PROCEEDINGS OF THE NINTH ESSLLI STUDENT SESSION

LAURA ALONSO I ALEMANY AND PAUL ÉGRÉ (EDITORS)

August 2004, Nancy, France
Preface

Since its creation in 1996, the ESSLLI Student Session has become a privileged platform for students working in the fields of logic, language and computation to present their work, either completed or in progress, and obtain fruitful feedback from an interdisciplinary audience, in many cases, for the first time.

The scientific quality and creativity of the contributions to this ninth edition of the Student Session was widely acknowledged among reviewers. A total of 56 papers were submitted, of which 18 will be presented orally and 8 as posters. The papers cover a wide range of topics, including computational linguistics, natural language syntax and semantics, formal pragmatics, intensional logic, modal logic, epistemic logic, logic programming, proof theory and cognitive science. This diversity, which reflects the interdisciplinary character of ESSLLI, is also what gives the Student Session its specificity and makes it particularly exciting to organize.

As chairs, we benefited a lot from the experience accumulated in the previous editions of the Student Session. Besides the wealth of documentation that they made available to us, the previous chairs have always been willing to lend a hand; a special mention should be made to Ivana Kruijff-Korbayová, Kristina Striegnitz, Malvina Nissim and Balder ten Cate. But also members of previous Programme Committees have been ready to help, and in general all the people involved in ESSLLI and FoLILI; among these, we would like to thank specially Carlos Areces, Patrick Blackburn, Willemijn Vermaat and Antal van den Bosch, who were in charge of the organization of ESSLLI this year and were most supporting during all the preparation of the Student Session.

It is not exaggerated to say that the whole of the Logic, Language and Computation community is willing to collaborate in the ESSLLI Student Session. We are proud to say that only very seldom does a researcher decline to do a review for the Student Session, and usually for non-academic reasons.

As follows, the Student Session is a unique event that provides an opportunity for students to become active part of their research community, getting feedback, tackling responsibilities, making decisions, learning to ask and get support, in sum: acquiring a more comprehensive understanding of what research means.

But, most of all, the Student Session is a very good way to enjoy your summer and get fresh ideas. We hope that you join us in it.

Laura Alonso i Alemany and Paul Égré

Barcelona and Paris, June 2004
Programme Committee

A lot of people are involved in the organization of the Student Session. Without them, the StuS would probably not be possible as it is now, and, in any case, the people involved in it would not learn as much as we do now.

In the first place, there are a lot of people who serve as reviewers for the StuS. They look at submitted papers carefully, and take their time to write lengthy and detailed reviews that provide high quality feedback for students to carry on with their work. We are happy to find that these reviews are at the same level, and many times much more insightful and helpful than those at bigger conferences.

As in previous years, we are happy to have had as reviewers researchers of recognized expertise in their areas. We are very grateful to all of them:

The ESSLLI Student Session is interdisciplinary in nature. Therefore, areas of expertise are divided so that papers can be dealt with an adequate level of expertise. This year, three areas have been distinguished: *Logic and Computation*, *Language and Logic* and *Language and Computation*.

For each of these areas, two or three student co-chairs have been in charge of studying the submitted papers that pertain to the area, looking for adequate reviewers for each of them, putting reviews together and making recommendations as to the adequacy of papers for presentation at the StuS. In all these tasks, they have been assisted by an area expert, a senior researcher who provides advice in difficult points and constitutes a reference point in the whole process.

This year, we have been very happy to work with the following people:

**Logic and Computation**

Carlos Areces (expert), LORIA, Nancy
Jaume Baixeries (co-chair), Universitat Politècnica de Catalunya, Barcelona
Willem Conradie (co-chair), Rand Afrikaans University, South Africa
Magdalena Ortiz de la Fuente (co-chair), Universidad de las Americas, Puebla

**Language and Logic**

Judit Gervain (co-chair), SISSA, Trieste
Bart Geurts (expert), University of Nijmegen, Nijmegen
Benjamin Spector (co-chair), Laboratoire de Linguistique Formelle, Université Paris 7, ENS

**Language and Computation**

Benoît Crabbé (co-chair), LORIA, Nancy
Ivana Kruijff-Korbayová (expert), Universität des Saarlandes, Saarbrücken
Marco Kuhlmann (co-chair), Universität des Saarlandes, Saarbrücken

We have been very lucky to work with all these people, who have made the task of putting everything together and making final decisions not only possible, but thoroughly enjoyable. We would like to thank them all for their dedication to the Student Session. We firmly believe that we have learnt a lot, and we can assure that we have had a very good time!

L.A.A. and P.É.
Contents

1 XML Query Evaluation via CTL Symbolic Model Checking  
   Loredana Afanasiev  
   1

2 Logic Program Transformation for Auxiliary Variable Elimination  
   Javier Alvarez  
   13

3 Defining Logical Constants, the Insight from Basic Logic  
   Denis Bonnay and Benjamin Simmenauer  
   27

4 Incremental Processing and the Generic Interpretation of Indefinites  
   Stefan Bott and Oriol Valentin  
   36

5 Why \( \lambda \text{HST}^* \) and \( \text{HST}^*_\alpha \) do not solve the Russell-Myhill paradox after all  
   Serge Bozon  
   48

6 A Simple Logic for Reasoning about Uncertainty  
   Nívea de Carvalho Ferreira  
   61

7 Norwegian Word Order in HPSG  
   Liv Ellingsen  
   72

8 Towards a Time Tagger for Romanian  
   Corina Forăscu and Daniela Solomon  
   82

9 Deduction of Numeral Grammars  
   Harald Hammarström  
   94

10 On some aspects of machine translation among related languages  
   Petr Homola  
   105

11 Implementing System BV of the Calculus of Structures in Maude  
   Ozan Kahramanoğulları  
   117

12 Word Sense Disambiguation of Estonian with syntactic dependency relations and WordNet  
   Kaarel Kaljurand  
   128
13 Towards Unsupervised Learning of Morphology Applied to Ukrainian
  Sophia Katrenko  
14 Multiple Long Distance Scrambling in Korean
  Jieun Kjaer  
15 Acquaintance resolution and belief *de re*
  Emir Maier  
16 Automatic Extraction of Subcategorization Frames for Bulgarian
  Svetoslav Marinov  
17 Why Not Use Query Logs As Corpora?
  Olena Medelyan  
18 On Epistemic Temporal Strategic Logic
  Sieuwert van Otterloo and Geert Jonker  
19 A Coreference Resolution Model on Excerpts from a Novel
  Oana Postolache and Corina Forăscu  
20 Bounded Model Property for Multi-Context Systems
  Floris Roelofsen  
21 Time and Focus: The Case of German *gerade*
  Gerhard Schaden  
22 Signaling games and non-literal meaning
  Merlijn Sevenster  
23 Expectation Reasoning: A Computationally Grounded Theory of Agency
  Binh Vu Tran, James Harland and Margaret Hamilton  
24 The indefinite; an extra-argument-slot analysis
  Hiroyuki Uchida  
25 Stable Categories as a Basis for Concepts
  Matthias Weisgerber  
26 Using Generalized Linear Regression to Train a Novel Sentence Detection
  System
  Yang Ye
# Programme

## First Week
9-13 August

### Monday 9th
**Gerhard Schaden**
**Nívea de Carvalho Ferreira**
*Chair: Judit Gervain*

- Time and Focus - The Case of German GERADE
- A Simple Logic for Reasoning about Uncertainty

### Tuesday 10th
**Oana Postolache**
**Corina Forăscu**
**Petr Homola**
*Chair: Marco Kuhlman*

- Coreference Resolution Model on Excerpts from a Novel
- On some aspects of machine translation among related languages

### Wednesday 11th
**Loredana Afanasiev**
**Sieuwert van Otterloo**
**Geert Jonker**
*Chair: Willem Conradie*

- XML Query Evaluation via CTL Symbolic Model Checking
- On Epistemic Temporal Strategic Logic

### Thursday 12th
**Jieun Kiaer**
**Liv Ellingsen**
*Chair: Benoît Crabbé*

- Multiple Long distance scrambling in Korean
- Norwegian Word Order in HPSG

### Friday 13th
**Ozan Kahramanoğlu**
**Dennis Bonnay**
**Benjamin Simenauer**
*Chair: Magdalena Ortiz*

- Implementing System BV of the Calculus of Structures in Maude
- Defining Logical Constants, the Insight from Basic Logic
# Programme

## Second Week

**16-20 August**

### Monday 16th

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harald Hammarström</td>
<td>Deduction of Numeral Grammar</td>
</tr>
<tr>
<td>Sophia Katrenko</td>
<td>Towards Unsupervised Learning of Morphology</td>
</tr>
<tr>
<td></td>
<td>Applied to Ukrainian</td>
</tr>
<tr>
<td><strong>Chair:</strong> Marco Kuhlman</td>
<td></td>
</tr>
</tbody>
</table>

### Tuesday 17th

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Olena Medelyan</td>
<td>Why Not Use Query Logs As Corpora?</td>
</tr>
<tr>
<td>Svetoslav Marinov</td>
<td>Automatic Extraction of Subcategorization</td>
</tr>
<tr>
<td></td>
<td>Frames for Bulgarian</td>
</tr>
<tr>
<td><strong>Chair:</strong> Benoît Crabbé</td>
<td></td>
</tr>
</tbody>
</table>

### Wednesday 18th

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serge Bozon</td>
<td>Why ( \lambda \text{HST}^* ) and ( \text{HST}_{\lambda}^* ) do not solve the Russell-Myhill paradox after all</td>
</tr>
<tr>
<td>Javier Álvez</td>
<td>Logic Program transformation&lt;br&gt;for auxiliary variable elimination</td>
</tr>
<tr>
<td><strong>Chair:</strong> Magdalena Ortiz / Willem Conradie</td>
<td></td>
</tr>
</tbody>
</table>

### Thursday 19th

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emar Maier</td>
<td>Acquaintance resolution and <em>de se</em> as <em>de re</em></td>
</tr>
<tr>
<td>Hiroyuki Uchida</td>
<td>The indefinite; an extra-argument-slot analysis</td>
</tr>
<tr>
<td><strong>Chair:</strong> Judit Gervain</td>
<td></td>
</tr>
</tbody>
</table>

### Friday 20th

**Closing Event**
## Programme

### Poster Presentations

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stefan Bott and Oriol Valentin</td>
<td>Incremental Processing and the Generic Interpretation of Indefinites</td>
</tr>
<tr>
<td>Corina Forăscu and Daniela Solomon</td>
<td>Towards a Time Tagger for Romanian</td>
</tr>
<tr>
<td>Kaarel Kaljurand</td>
<td>Word Sense Disambiguation of Estonian with syntactic dependency relations and WordNet</td>
</tr>
<tr>
<td>Floris Roelofsen</td>
<td>Bounded Model Property for Multi-Context Systems</td>
</tr>
<tr>
<td>Merlijn Sevenster</td>
<td>Signaling games and non-literal meaning</td>
</tr>
<tr>
<td>Bành Vũ Trân, James Harland and Margaret Hamilton</td>
<td>Expectation Reasoning: A Computationally Grounded Theory of Agency</td>
</tr>
<tr>
<td>Matthias Weisgerber</td>
<td>Stable Categories as a Basis for Concepts</td>
</tr>
<tr>
<td>Yang Ye</td>
<td>Using Generalized Linear Regression to Train a Novel Sentence Detection System</td>
</tr>
</tbody>
</table>
XML Query Evaluation via CTL Symbolic Model Checking

LOREDANA AFANASIEV
Informatics Institute, University of Amsterdam
lafanasi@science.uva.nl

Abstract.
The Extensible Markup Language (XML) was designed to describe the content of a document and its hierarchical structure. The XML Path language (XPath) is a language for selecting elements from XML documents. We propose and implement a linear embedding of the query evaluation problem for Core XPath, the navigational fragment of XPath, into the model checking problem for Computation Tree Logic (CTL). This allows us to evaluate Core XPath queries by exploiting a model checker for CTL. We report on experiments with the state-of-the-art model checker NuSMV, and compare our results with alternative academic XPath processors.

1 Introduction

Most data on the Web are unstructured, unorganized and diverse. Web information architects organize the Web data as semistructured documents with some underlying partial structure. The regular schemas used for conventional databases are rigid and simply inappropriate for manipulating with this type of data. The Extensible Markup Language (XML) (11) is a textual representation of the hierarchical structure of information that was designed by the World Wide Web Consortium to describe any type of textual information. It is able to represent missing or duplicated data as well as nested information, and, hence, provides a suitable data model for semistructured data (1).

XML, due to its simplicity, is becoming a universal format for data exchange. Whether it will be used also for storage of big amounts of data, as opposed to using conventional databases, depends on the ability of XML storage models to support efficient querying. Hence, the XML query evaluation problem, that is the problem of selecting elements from an XML document that meet particular requirements, is in the spotlight of recent research on semistructured data and has proved particularly appealing for computational logicians. In particular, query evaluation and equivalence problems have been recast in terms of reasoning problems in suitable logical languages.
XML path language (XPath) (12) is the most widely used XML querying language, proposed by the World Wide Web Consortium (W3C). XPath plays a crucial role in other XML technologies such as XSLT (14), XQuery (13). The XPath query evaluation problem is known to be solvable in polynomial time (polynomial in the combined size of the query and the document) (16). There exists several implementations of polynomial time algorithms for XPath query evaluation and of linear time algorithms for smaller fragments of XPath, in particular Core XPath (16; 17). Core XPath is the navigational fragment of XPath (16).

We propose a new, model checking-based, method for evaluating Core XPath queries. Recall that global model checking is the task of computing the states of the model (Kripke structure) that satisfy a logical formula (10). There is a close connection between the XPath query evaluation problem and the global model checking problem. An XML file can be represented as a node-labeled graph structure, and a Core XPath query can be interpreted as a temporal logic formula. Hence, query evaluation boils down to checking which nodes of the representation graph satisfy the formula representing the query. We present a linear embedding of Core XPath into Computation Tree Logic (CTL).

Model checking is a widely used technique for Software and Hardware verification (finite state concurrent systems verification). There are advanced industrial and academic software tools for this purpose. The embedding of query evaluation into model checking has the advantage of re-using existing ideas and even allowing the use of an actual model checker.

These motivations suggest the following central question:

*Can we use a CTL model checking approach in order to evaluate Core XPath queries?*

In this paper we answer this question by theoretical and experimental means. We devise a reduction from the query evaluation problem for Core XPath to the model checking problem for CTL. More precisely, we propose a document translation to convert an XML document into a model for CTL, and a query translation to map a Core XPath query into a CTL formula. We also consider a small fragment of Core XPath, Simple XPath, which admits a simpler translation to CTL. It contains queries frequently used in practical applications. We test the practical value of our approach by comparing its running times to two other very fast XPath query evaluators. Our implementation uses an existing symbolic model checker, NuSMV.

This paper aims to present our results to a general audience. It contains few technical details and it is based on examples. More technical details can be found in (2) and (3).

## 2 Related Work

The relation between model checking and query processing has been extensively explored in the setting of structured data. We refer to (18) as a good entry-point to the area. The relation between model checking and query processing for semistructured data goes back at
least to (4), where it was formulated in terms of suitable modal-like logics. Quintarelli (24) embeds a fragment of the graphical query language G-Log into CTL, and she sketches a mapping for subsets of other semistructured query languages, like Lorel, GraphLog and UnQL. Miklau and Suciu (22) and Gottlob et al. (15) sketch an embedding of the forward looking fragment of Core XPath into CTL and the full Core XPath into multi-modal CTL with past modalities and thus obtain linear time complexity result.

In (16) it is shown that the query processing problem for XPath admits a polynomial-time algorithm, and it can be solved in linear time in case of Core XPath. The authors propose an algorithm that translates a Core XPath query into an algebraic expression over sets of nodes of the tree representing an XML document, and that evaluates the algebraic expression in order to answer the query. The implementation of this approach is called XMLTaskForce engine and it will be used for comparison with the performance of XMChecker. A number of other Core XPath processors have been implemented afterward. Koch (19) translates a query into a tree automaton and uses the resulting tree automaton in order to process the query. A different approach to processing XPath on XML files is to embed XML documents into relational databases, to rewrite XPath queries as SQL ones, and to run an SQL engine to retrieve the answer set of the original XPath query (see, e.g., (28)). Finally, recently, Buneman et al. propose in (7) an efficient algorithm for Core XPath processing on compressed XML files. The core idea is to compress XML trees into directed acyclic graphs sharing common subtrees, and to evaluate the query directly on compressed XML documents. It borrows aspects from model checking techniques for model representation, namely from Ordered Binary Decision Diagrams (OBDDs). The resulting implementation carries the name MacMill, and is also used in our comparisons.

3 The XPath language and its fragments

An XML document can be represented as a tree with tag labels attached to the nodes (1). For example in Figure 1.1 we have the tree representation of a small XML document.

```xml
<bibliotheek>
  <boek>
    <auteur>Roux</auteur>
    <auteur>Combalusier</auteur>
    <titel>Database Systems</titel>
  </boek>
  <boek>
    <auteur>Smith</auteur>
    <titel>Database Systems</titel>
  </boek>
</bibliotheek>
```

![Figure 1.1: XML document and its tree representation.](image)

Such a tree can be represented as a tuple $T = (N, R^+, R^-, L : \Sigma \rightarrow \varphi(N))$, where $\Sigma$
is the set of labels corresponding to the XML element tags, \( N \) is the set of tree nodes, and \( L \) labels the nodes with elements from \( \Sigma \). The tree edge relation \( R_e \) represents the XML element inclusion. The sibling order defined by the document order is represented by the secondary relation \( R_s \). In this way, we represent only the hierarchical structure of the document. In this paper we will ignore the content of the XML elements and attribute information, focusing only on the hierarchical structure of the document and the tags of the elements. The XML tree representation and our method can be extended to the full case.

XML path language (XPath) \((12)\) is a language proposed by the World Wide Web Consortium (W3C) for selecting elements from XML documents. Core XPath \((16)\) is the clean logical core of XPath. It is often referred to as the navigational fragment of XPath, since it maintains the navigational power of XPath, but not its arithmetical and string operations.

Core XPath queries describe paths through an XML document. A Core XPath query is an arbitrarily long sequence of location steps defined inductively as follows:

\[
\begin{align*}
\text{xpath} & : = \text{locationstep} \mid /\text{locationstep} \mid \text{xpath}/\text{locationstep} \\
\text{locationstep} & : = \text{axis} :: l \mid \text{axis} :: l/\text{pred} \\
\text{pred} & : = \text{pred and pred} \mid \text{pred or pred} \mid \text{not pred} \mid \text{xpath} \\
\text{axis} & : = \text{child} \mid \text{parent} \mid \text{descendant} \mid \text{ancestor} \mid \\
& \quad \text{descendant_or_self} \mid \text{ancestor_or_self} \mid \text{following_sibling} \mid \\
& \quad \text{preceding_sibling} \mid \text{following} \mid \text{preceding}
\end{align*}
\]

where \( l \) is either an XML tag or \( * \), a wild-card label. Core XPath queries starting with the symbol / are called absolute, and the others are called relative.

The crucial notions of the Core XPath language are axes and location steps. Each axis corresponds to a binary relation on the set of nodes of the tree. For instance, in the example of Figure 1.1, the axis descendant relates the node 5 with the nodes 6 and 7. The location step \( \text{axis} :: l \) is interpreted as a binary relation on the set of nodes of the tree that relates \((n, m)\) if \( m \) is reachable from \( n \) through \( \text{axis} \) and the label of \( m \) is \( l \). For example, the pair of nodes \((5, 7)\) belongs to the relation identified by descendant :: title. The location step \( \text{axis} :: l/\text{pred} \) introduces additional constraints on node selection. The node with the label \( l \) must also satisfy the predicate \( \text{pred} \). For instance, the pair of nodes \((0, 1)\) belongs to child :: boek[child :: auteur[following_sibling :: auteur]]. Finally, a query is interpreted as the composition of the relations for each \( \text{locationstep} \) in the query. Absolute queries are evaluated from the root of the tree, while relative queries are evaluated at any node of the tree. The result of an absolute or relative Core XPath query is the set of nodes that can be reached through the corresponding relation starting at the tree root or at any node of the tree, respectively. For example, the query child :: boek[child :: auteur[following_sibling :: auteur]] asks for the books with at least two authors. The result of evaluating this query in the node 0 is the node 1. A formal definition of the syntax and semantics of Core XPath can be found in \((2)\).

Simple XPath is the fragment of Core XPath that contains only the queries of the form \( /\text{self} :: *[\text{pred}] \) or \( \text{self} :: *[\text{pred}] \), where \( \text{pred} \) contains only the downward axes: self, child,
descendant and descendant-or-self. Simple XPath is a very natural XPath fragment. Over
trees it can express exactly every bisimulation invariant (with respect to the child relation)
unary first order property (expressed in the signature with the descendant relation, and
unary predicates corresponding to tag names) (21). With respect to complexity, it is just
as difficult as Core XPath: query evaluation needs linear time, and checking whether a
Simple XPath expression is satisfiable (i.e., there exists a tree and a node such that the
node is in the answer set of the expression when evaluated on that tree) is complete for
exponential time (30).

4 Core XPath query evaluation via CTL model checking

Computation Tree Logic (CTL) is a temporal logic most commonly used for formal ver-
ification of computer systems. We assume the reader to be familiar with the syntax and
semantics of CTL (see, e.g., (10)). The truth set of a CTL formula with respect to a model
is the set of states of the model that satisfy the formula. Recall that the global model
checking is the problem of computing the truth set of a given formula with respect to a
given model.

Theorem 4.1 There exists a linear reduction from the Core XPath query evaluation problem
to the CTL global model checking problem.

The proof of this result can be found in (2). The reduction proceeds in two steps. First,
we reduce Core XPath query evaluation to global model checking for a multi-modal version
of CTL with four modalities. Each modality corresponds to a primitive direction in XPath,
namely down, up, left or right. Next, we show that global model checking for multi-modal
CTL can be reduced to global model checking for CTL. All in all, the reduction consists
of a linear translation of the XML tree into a CTL model and a linear translation of the
Core XPath query into a CTL formula.
We will now give an example of the translations. Consider again the XML document given in Figure 1.1. The intermediate translation of this document is shown in Figure 1.2. The down, up, left or right arrows label the corresponding multi-modal CTL relations. Figure 1.3 contains the uni-modal CTL model obtained from the multi-modal CTL model in Figure 1.2. Note that a, ↑, ↓, ←, → are now node labels additional to Σ.

Next, consider the following Core XPath query:

/child :: boek[child :: auteur[following_sibling :: auteur]]

Its translation in multi-modal CTL is as follows:

boek ∧ ¬EX↑EX↑⊤ ∧ EX↑(auteur ∧ EX→EF→auteur)

The uni-modal CTL translation of the example query is:

a ∧ boek ∧ EX(↑ ∧ EX(↑ ∧ EX(a ∧ ¬EX(↑ ∧ EX ↑)))) ∧ EX(↓ ∧ EX(↓ ∧ EX(a ∧ auteur ∧ EX(→ ∧ EXE(→, → ∧ EX(a ∧ auteur))))))

In general, the resulting CTL model has $5 \cdot |N|$ states, where $N$ is the set of nodes in the given XML tree, and has $\Sigma_{d \in \{↑, ↓, ←, →\}} |R_d| + 8 \cdot |N|$ edges, where $R_d$ are the XML tree relations$^1$ and $N$ is the set of nodes. The modal depth of the resulting CTL formula is at most 4 times bigger than the number of the location steps in the Core XPath query. Though theoretically uninteresting, these numbers play a big role in practical applications of the problem reduction.

The Simple XPath language, due to the restriction put on the axes appearing in the query, admits a more efficient reduction into CTL. For the CTL model we take simply the

$^1R↑, R↓$ are the inverse relations of $R↑$ and $R↓$, respectively.
XML tree only with the relation $R_1$. A Simple XPath query can be translated into a CTL formula with modal depth at most the number of location steps in the query.

As a corollary of Theorem 4.1, we obtain the following result.

**Corollary 4.2** Core XPath query evaluation can be done in time $O(|D| \cdot |q|)$, with $|D|$ and $|q|$ the size of the XML document and the Core XPath query, respectively.

Note that these results immediately transfer to Simple XPath. As a theoretical result, Corollary 4.2 is not new (16; 20). However, our reduction to CTL might be suitable for practical implementation. In the rest of the paper we test how feasible it is to use CTL model checking for evaluating Core XPath queries in practice.

## 5 Implementation and experiments

A major advantage of our approach to Core XPath query evaluation is that it is easily implementable; by using an existing CTL model checker, we reduce our task to implementing the document and query translations into CTL and the translation from the model checker output back to the world of XML. The resulting tool is called XMChecker (XML Model Checker). Our implementation uses NuSMV (9) as the CTL model checker. It is written in Perl and consists of two translation subroutines, one for translating the tree representation of the XML document to NuSMV input format, and another for translating Core XPath and Simple XPath queries to CTL formulas. Besides these translations, it contains a subroutine that runs NuSMV, and a subroutine that interprets the truth set of a CTL formula, given as output by NuSMV, as a set of elements of the original XML file.

We opted to use NuSMV because it is a state-of-the-art tool implementing many different optimization techniques, including symbolic model checking (10). Moreover, NuSMV is open source, modularly structured, and well documented (8). However, designed for the verification of computer hardware and software, NuSMV only allows an yes/no answer for model checking problem. The global model checking needed for our purposes was easily implemented by making a small modification to NuSMV’s source code.

The source code for XMChecker and for NuSMV, modified to perform global model checking, can be found at the XMChecker website (31).

**Experimental settings.** Our experiments were run on a Pentium IV, 1.60GHz, with 1.5GB RAM, running Redhat Linux version 2.4.21-ict1. We ran tests using a variety of XML documents, Core XPath and Simple XPath queries, using both artificially generated data and XML benchmark data. To illustrate our conclusions, we present the results obtained on the documents and queries generated by the XML benchmarking program XMark (27). We concentrate on a set of queries derived from XMark benchmark queries, containing relative and absolute Simple XPath and Core XPath queries. We measured the CPU times (in seconds) needed to execute the conversion of the CTL model into NuSMV’s internal, symbolic representation and building OBDD representation and model checking (query evaluation). We ignore the document and query translation times and the times needed to interpret the truth set of a CTL formula as a set of elements of
Figure 1.4: a. Processing CTL models into internal symbolic NuSMV format. b. Run times for Core XPath and Simple XPath query evaluation against documents with increasing size, using XMChecker, MacMill, and XMLTaskForce.

the original XML file, as they are insignificantly small. On the data just described, we compared XMChecker’s query evaluation time against the run-times of the following alternative XPath processors: XMLTaskForce Engine, the first polynomial-time XPath engine, proposed in (16), and MacMill (7), to the best of our knowledge the fastest implemented navigational XPath processor currently available.

Results. We first look at the model building times used by XMChecker. Figure 1.4.a. contains the time taken by NuSMV to process the input model into its internal symbolic format: to Boolean encode the model and to build OBDD representation. Observe that the numbers are extremely high. This suggests that run-time processing of the XML documents is not feasible in practice, thus this task must be done beforehand. Note also that the document translation for Core XPath produces bigger CTL models than the one for Simple XPath, leading to longer building times. We observed that the processing capacity of NuSMV is limited to input files of several megabytes. These are the main bottlenecks in applying NuSMV for our purposes.

Figure 1.4.b. contains the outcomes of the comparison between XMChecker, MacMill, and XMLTaskForce for evaluating the following query, on documents generated by XMark:

\[
Q = \text{/self::*[child::site/child::regions/child::africa/child::item/}
\text{child::description/child::parlist/child::listitem/child::text]}
\]

We picked this simple query to exemplify the performance of our two translations (for Core XPath and Simple XPath). Other experiments, described in details in (2) and (3), underline the same pattern and lead to similar conclusions.
The time that XMChecker takes for Simple XPath query evaluation when the model and the query are in main memory are of the same order of magnitude as the time that MacMill takes to process the queries under the same conditions. For Core XPath query evaluation, XMChecker takes up to three orders of magnitude more times than MacMill. For the Simple XPath experiments, we used the XML tree representation containing only the tags appearing in the query. This was done to obtain the smallest possible model relative to the given query. The same projection technique is implemented in MacMill (7). Other experiments were carried out, with the complete XML tree representation of the document. In this case, the models grow larger since they contain all the tag information carried by the XML documents, and the time needed by XMChecker to perform query evaluation is in the same order of magnitude as XMLTaskForce. However, as this processing is query independent, it can be done off-line.

In summary, our experiments show that the XMChecker Simple XPath query evaluation times (when the model is in main memory) can compete both with the MacMill query evaluation engine (in the case of document translation relative to the given queries) and with the XMLTaskForce query evaluation engine (in the case of full translation of the tree representation of XML documents). Ignoring the model processing times, these results are very encouraging. With respect to Core XPath query evaluation, many optimization to our symbolic model checking approach are needed for this approach to be applied in practice. In the next section we discuss possible improvements of the representation of the document trees.

6 Conclusions and future work

Let’s go back to the central question of the present investigation: Can we use a CTL model checking approach in order to evaluate Core XPath queries? We have shown that, from a theoretical point of view, this is indeed possible and that a linear time implementation can be provided. The first, naive, implementation is provided. It takes advantage of an existing symbolic model checker NuSMV and the comparison with the two state-of-the-art Core XPath engines has shown competitive potential for Simple XPath language. Though there are several bottlenecks to overcome, many other optimizations and improvements already suggest themselves.

Problems. NuSMV is a tool designed for software verification and it receives as input a program specification, which is in practice logarithmic in the size of its finite state system. NuSMV unfolds the program step by step, at the same time building and compressing the model representation by use of OBDDs. Using this algorithm, NuSMV can manage very large models, never dealing with the explicit model size. Our case, the XML document (i.e., the model) is given explicitly. Hence, in case of big XML documents, the load of the main memory before building the OBDD representation becomes too big for NuSMV to cope with. An immediate question appears: is it possible to turn an XML tree representation directly into OBDD format, in a more efficient manner than currently done by NuSMV? But more important is the question: can we use OBDDs for compressing
XML?

There are several questions on issues related to CTL, in particular to CTL model checking. By adding past operators to CTL language, that is, the backward looking analogues of EX and EF, it is possible to handle backward looking axes with the Simple XPath translation. Can we extend NuSMV with past operators?

**Improvements.** We see many directions for future work. Here we list some of them. Though in the worst case no reduction of the model is obtained we consider it worth studying the OBDDs representation for XML documents. The extent to which OBDDs can help compress the model crucially depends on the initial encoding of the model. Hence, to capture the redundancies in the XML structure we must employ clever binary encodings. For example, the binary encoding of a node must differ in as little bits as possible from the encoding of its children. In such way we guarantee a smaller OBDD representation for the transition relation of the model.

The next step that imposes itself is to combine the Buneman et al. (7) model compression technique with the OBDDs representation of the model. Note that these two techniques are essentially different. The main idea is to find first the smallest bisimilar model, which preserves the answer set of the query (7), and afterward to proceed with model checking techniques for efficient model representation and model checking. Experimental results in (7) show that the compressed model can be, in some cases, of size up to 10% and lower of the initial size, while the OBDD representation can achieve a logarithmic size of the model (10).

Finally, real XML documents (like XMark generated files) often are graphs (not trees) because of the use of ID/IDREF attributes. Query evaluation on graphs is not as easy to optimize as on trees, and has, in fact, hardly been addressed so far. Since model checkers are designed to work on cyclic structures, this seems an area in which a model checking-based approach has potential.

**Acknowledgements**

This research was (partially) funded by the Netherlands Organization for Scientific Research (NWO) under project number 612.000.207. Thanks to Balder ten Cate, Massimo Francheschetti, Maarten Marx, and Maarten de Rijke.
References


Logic Program Transformation for Auxiliary Variable Elimination

JAVIER ÁLVEZ
Dpto. de LSI de la UPV. Paseo Manuel de Lardizabal, 1. 20080 - San Sebastian, Spain
e-mail: jiba@isi.ceu.es

ABSTRACT.
In logic programming, it is very common the use of auxiliary variables for storing intermediate results that are passed from a literal to other in the body of a clause. This can make easier the programming task and also clearer the resulting programs. Nevertheless, these variables are especially problematic for dealing with negation. In particular, in approaches such as intensional and constructive negation, these variables raise universal quantified goals or constraints, depending on the LP or CLP approach. It is a well-known fact that the universal quantification in goals or constraints can jeopardize at least the efficiency, and at most the completeness, of the goal computation process. Hence, our purpose is to eliminate the auxiliary variables in normal logic programs. As a first step, in this paper, we present a syntactic method for transforming a subclass of definite logic programs into equivalent ones with no auxiliary variable. In future works, we hope to extend this transformation to: (a) the whole class of definite logic programs, and (b) the class of normal logic programs.

1 Introduction
Program transformation has been extensively studied mainly for improving the performance of functional and logic programs. For logic programs, transformations have been phrased in terms of three basic operations: unfold, fold and goal replacement. Unfold/fold transformations have been widely used for improving program efficiency, for reasoning about programs, for automated deduction and synthesis, for program specialization, etc. Unfold/fold transformation systems for definite logic programs were early introduced in the seminal paper (Tamaki and Sato 1984). Later, several extensions for different wider classes of programs have been proposed (e.g. (Aravindan and Dung 1995; Maher 1988;

1This work has been partially supported by the Spanish Project TIC 2001-2476-C03.
Maher 1993; Pettorossi and Proietti 1999; Roychoudhury, Kumar, Ramakrishnan, and Ramakrishnan 1999; Sato and Tamaki 1989; Seki 1993)) and their correctness w.r.t. various semantics has been proved.

In this paper, we present an unfold/fold-based transformation method for eliminating the auxiliary variables used to store intermediate results in logic programs. The underlying aim is to improve the performance of a practical implementation of constructive negation (cf. Álvez, Lucio, Orejas, Pasarella, and Pino 2004). It is a well-known fact that the auxiliary variables used to store intermediate results in logic programs cause several problems for solving negative goals, since they give rise to unavoidable universal quantification in the negation of a body clause. In the so-called intensional negation (cf. Barbuti, Mancarella, Pedreschi, and Turini 1990), such universal quantification prevented for achieving a complete goal computation mechanism. Afterwards, constructive negation was introduced in (Chan 1988) and extended in (Chan 1989; Drabent 1995; Stuckey 1995) to a complete and sound operational semantics for the whole class of normal logic programs. Although the existing experience in implementing negation (beyond negation as failure) is, to our knowledge, very limited, there are some works (e.g. Álvez, Lucio, Orejas, Pasarella, and Pino 2004; Bruscoli, Levi, Levi, and Meo 1994; Stuckey 1995) which give a hint about the efficiency problems of universal quantified goals. Hence, our purpose is to eliminate the auxiliary variables in normal logic programs. As a first step, in this paper, we present a syntactic method for transforming a subclass of definite logic programs into equivalent ones with no auxiliary variable. In future works, we hope to extend this transformation to:

(a) the whole class of definite logic programs, and (b) the class of normal logic programs.

We say that our method is unfold/fold-based in the sense that the correctness of every transformation step is resembled to a derivation sequence of programs where each step is an application of a basic operation from the unfold/fold system proposed in (Sato 1992). The results of (Sato 1992) allow us to prove equivalence with respect to three-valued logical consequence of program completion, which is the standard declarative semantics for normal programs.

Regarding to closely related works, we are aware of (Proietti and Pettorossi 1995) which focuses the problem of eliminating unnecessary variables in definite logic programs. The notion of unnecessary variable is less restricted than our notion of auxiliary variables. Thus, this method deals with more variables. However, the logic program subclass where this method works is more restricted than the subclass we deal with. In this way, our method is able to eliminate a more restricted class of variables in a wider class of logic programs. Unnecessary variables are also called internal variables in some other works (e.g. (Sato and Tamaki 1984)) where the need for their elimination is claimed. The CPS (Continuation Passing Style) Conversion proposed in (Sato and Tamaki 1989) is similar to our transformation. This work transforms program predicates by applying unfold/fold transformations and by adding extra-arguments that store the information needed for the following computation and that work like stacks. The unfold/fold transformation is done deterministically on the basis of a predicate partition that classifies arguments into input and output arguments. The most important difference is that we automatically obtain the predicate partition, that we call mode specification, from the program definition. Besides,
the transformation differs because our aim is to eliminate the auxiliary variables from the program definition, but not to gain efficiency.

Outline of the paper. Section 2 contains preliminary definitions and notation. Section 3 is devoted to an assistant concept – the mode specification – which helps for carrying out transformations. In section 4 we present the transformation method for any recursive definite logic program. Section 4 discusses future work.

2 Preliminaries

Every program \( P \) is built from symbols of a signature \( \Sigma \equiv \{FS_\Sigma, PS_\Sigma\} \) of function and predicate symbols, respectively, and variables from \( X \). Both function and predicate symbols have associated a number \( n \), called arity. A term is either a variable, a constant or a function symbol applied to \( n \) terms, where \( n \) is the arity of the function. We denote by \( \text{var}(t) \) the set of variables that occurs in \( t \). A bar is used to denote tuples, or finite sequences, of objects, like \( \overline{x} \) as abbreviation of the \( n \)-tuple of variables \( x_1, \ldots, x_n \). Concatenation of tuples is denoted by the infix \( \cdot \) operator, i.e. \( \overline{x} \cdot \overline{y} \) is the concatenation of \( \overline{x} \) and \( \overline{y} \). From now on, we use \( t, r, s \) to denote terms and \( x, y, z \) to denote variables, possibly with bar and sub/super-scripts.

A substitution \( \sigma \) is a mapping from a finite set of variables, called its domain, into a set of terms. It is assumed that \( \sigma \) behaves as the identity for the variables outside its domain. The most general unifier of a set of terms \( \{s_1, \ldots, s_n\} \), denoted \( \text{mgu}(\overline{s}) \), is an idempotent substitution \( \sigma \) such that \( \sigma(s_i) \equiv \sigma(s_j) \) for all \( i, j \in 1..n \) and for any other substitution \( \theta \) with the same property, \( \theta \equiv \sigma \cdot \sigma \) holds for some substitution \( \sigma' \).

An atom \( p(\overline{t}) \) is a \( n \)-ary predicate symbol \( p \) applied to a \( n \)-tuple of terms \( \overline{t} \). We denote by \( \text{var}(\overline{t}) \) the set of variables that occurs in \( \overline{t} \). A clause \( C \) is an expression of the form \( B : - B_1, \ldots, B_m \) where \( B \) (called its head) is an atom and \( B_1, \ldots, B_m \) (called its body) denotes a conjunction of atoms. When the body is empty (or equivalently true) the clause \( B \) is called a fact. A program \( P \) is defined by a collection of clauses. For a predicate \( p \), we denote by \( \text{Def}_P(p) \) the set of all clauses in \( P \) with head predicate \( p \). Without loss of generality, any conjunction of atoms \( p_1(\overline{s}_1, \overline{s}_0), \ldots, p_n(\overline{s}_1, \overline{s}_0) \) inside a clause body can be replaced with an atom \( p'(\overline{s}_1, \ldots, \overline{s}_i, \overline{s}_0, \ldots, \overline{s}_0) \), where \( p' \) is another new program predicate defined by the clause:

\[
p''(\overline{s}_1, \ldots, \overline{s}_i, \overline{s}_0, \ldots, \overline{s}_0) : - p_1(\overline{s}_1, \overline{s}_0), \ldots, p_n(\overline{s}_1, \overline{s}_0)
\]

We will implicitly use this equivalence in order to write clauses along the paper.

**Definition 1** Let \( C \) be a clause, a variable \( y \) is auxiliary in \( C \) iff \( y \) occurs in at least two different atoms of the \( C \)'s body but it not occurs in the \( C \)'s head. We denote by \( \text{auxvar}(C) \) the set of all auxiliary variables occurring in the clause \( C \).

The transformation method we are going to introduce is able to eliminate every auxiliary variable from a class of logic programs. However, it does not serve for eliminating the
variables that occur in just one atom of the body clause, but not in its head. These variables, which we call isolated, also raise unavoidable universal quantification. The notion of unnecessary variable (cf. (Proietti and Pettorossi 1995)), also called internal variables (cf. (Sato and Tamaki 1984)), covers exactly both notions of auxiliary and isolated variables. In our experience, isolated variables are rarely used in logic programs.

To define the semantics of a program $P$, Clark (Clark 1978) proposed to complete the definition of the predicates in $P$. The predicate completion formula of a predicate $p$ such that $Def_P(p) \equiv \{ p(\vec{t}) : \neg \overline{t}(\vec{s}) | i \in 1..m \}$ is the sentence:

$$\forall \overline{x}(p(\overline{x}) \leftrightarrow \bigvee_{i=1}^{m} \exists \overline{y}(\overline{x} = \vec{t}^i \land \overline{t}(\vec{s}^i)))$$

where each $\overline{t} = \text{var}(\vec{t}) \cup \text{var}(\vec{s})$. In particular, for $m = 0$ (or $Def_P(p) \equiv \emptyset$) the above disjunction becomes false. Hence, the formula is equivalent to $\forall \overline{x}(\neg p(\overline{x}))$. The Clark’s completion of a program $P$, namely $\text{Comp}(P)$, consists of the set $P^*$ of the predicate completion formulas for every $p \in PS_G$ together with the free equality theory $\text{FET}(\Sigma)$. Then, the standard declarative meaning of normal logic programs is $\text{Comp}(P)$ interpreted in three-valued logic (cf. (Kunen 1987)).

Unfold/fold transformation systems restrict the application of their basic operations (usually by syntactic conditions) in order to warrant the correctness w.r.t. the considered semantics for a class of logic programs. Since we are interested in negation, we base our results on the transformation system proposed in (Sato 1992) because it is equivalence-preserving w.r.t. three-valued logical consequences of program completion. However, due to the lack of space, we omit the proofs of the correctness theorems in this paper. They can be found in the technical report version of this paper.

3 Mode Specifications

Our transformation method relies in a preliminary partition of the predicate arguments according to its mode that can be input (in) or output (out). We need to assign a mode to (all or part of) the arguments of the predicate of an atom. The value null will be used to denote the absence of mode.

**Definition 2** The mode specification of the predicate $p \setminus n$ in an atom $a \equiv p(t_1, \ldots, t_n)$ is a $n$-tuple $(m_1, \ldots, m_n) \subseteq \{\text{in}, \text{out}, \text{null}\}^n$ and we denote it by $ms(a)$ or $ms(p(t_1, \ldots, t_n))$. We say that $ms$ is partial if $ms_j = \text{null}$ for some $1 \leq j \leq n$. Otherwise, it is total.

**Definition 3** Let $C \equiv B : - B_1, \ldots, B_m$ be a program clause, its mode specification $ms(C)$ is an $m + 1$-tuple $(ms^0(B), ms^1(B_1), \ldots, ms^m(B_m))$ such that each $ms^i(a)$ is a
mode specification of \( a \). If \( ms^t(a) \) is total for all \( a \in \{ B, B_1, \ldots, B_m \} \) we say that \( ms(C) \) is total. Otherwise, we say that \( ms(C) \) is partial.

This mode specification will be exclusively used at transformation time. That is, it will not serve to restrict predicates-uses in user-goals. By using the mode specification, we obtain a dataflow of the data inside the clauses. Sometimes, we suffice to have partial specification modes for transforming clauses. The intuition for assigning a mode specification relies on two common and well-known features of the logic program languages (mainly Prolog):

(1) Any logical variable can be instantiated only once

(2) The most common selection rule for goal literals is leftmost.

These two facts support the intuition behind Definition 4, although our method does not depend on the selection rule. On the basis of the Fact (1), we decide that each variable occurs in output arguments in at most one atom. The remaining arguments where the variable occurs in are assigned to be input arguments. Due to the Fact (2), we place the output arguments (if there are some) in the leftmost atom where the variable occurs in.

A partial mode specification can be extended to other (partial or total) mode specification in different forms:

**Definition 4** Let \( C \equiv p(t_1, \ldots, t_n) : \neg B_1, \ldots, B_m \) be a program clause such that \( B_j \equiv p_j(s_1, \ldots, s_{k_j}) \) for each \( 1 \leq j \leq m \). Let \( ms(C) \) be a partial mode specification for \( C \)

(a) if \( ms^t(B_j) \) is total for each \( 1 \leq j \leq m \), then head-extension of \( ms \) is made by changing each \( ms^0_h \) such that \( ms^0_h = \text{null} \) (\( 1 \leq h \leq n \)) to be out or in respectively depending on the following inclusion holds or not:

\[
\text{var}(t_h) \subseteq \bigcup_{j=1}^{m} \bigcup_{i=1}^{k_j} \{ v \mid v \in \text{var}(s_i) \text{ and } m^i_j = \text{out} \}
\]

(b) Otherwise \( ms^t(B_j) \) is partial for some \( 1 \leq j \leq m \), \( ms \) can be extended w.r.t some variable \( y \) that occurs in \( C \)'s body. To do that, let \( B_j \) be the leftmost atom where \( y \) occurs in. Then, the extension is defined in two steps:

(b1) For each \( i \) such that \( ms^j_i = \text{null} \), \( ms^j_i \) takes the value out whenever \( y \in \text{var}(s_i) \). Besides, if \( y \) is an auxiliary variable, each null \( ms^j_i \) such that \( y \notin \text{var}(s_i) \) takes the value in.

(b2) For each \( k \) such that \( k \neq j \) and for each \( i \) such that \( ms^k_i = \text{null} \), \( ms^k_i \) takes the value in whenever \( y \in \text{var}(s_i) \).

We call the variable-extension of \( ms \) in \( C \) to the result of successively extending w.r.t. each variable occurring in \( C \) provided that all the auxiliary variables are considered before the non-auxiliary ones.

\[17\]
Now, we explain the extension of the mode specification of an atom with predicate $p$ to a clause with head-predicate $p$ and, at the same time, to all the clauses defining (in a program) all the predicates that are mutually recursive with $p$.

**Definition 5** Let $P$ be a program, $C$ a clause $p(s_1,\ldots,s_n) : \neg B_1,\ldots,B_m$, the collection $q_1,\ldots,q_k$ of all the predicates that are mutually recursive with $p$ in $P$, and the collection of clauses $\mathcal{D} \equiv \bigcup_{i=1}^{k} \text{Def}_P(q_i)$. We extend any mode specification $\text{ms}(p(t_1,\ldots,t_n))$ to the set of clauses $C \cup \mathcal{D}$ as follows:

1. $\text{ms}(a)$ is assigned to be identical to $\text{ms}(p(t_1,\ldots,t_n))$ for every atom occurring in a clause in $C \cup \mathcal{D}$ with head predicate $p$. In particular, $C$’s head.

2. For each $B_j \equiv q_i(r_1,\ldots,r_k)$ such that $1 \leq i \leq k$, we firstly extend $\text{ms}$ to every clause $D \in \text{Def}_P(q_i)$ in two successive steps: (1) variable-extension in $D$ and (2) head-extension in $D$. Then, $\text{ms}^j(B_j)$ inherits the $\text{ms}(q_i(\overline{t}))$ from the clause heads in $\text{Def}_P(q_i)$. In this process, as soon as any predicate $q_i$ takes a mode specification, it is propagated to the rest of occurrences of $q_i$ in $\mathcal{D}$.

3. Finally, we non-deterministically extend $\text{ms}$ to a total mode specification for $C$.

**Example 1** Consider the following clause $E1.1$:

\[
\text{w_balanced}(\text{tree}(x_1,x_2,x_3),x_4) :~ \text{w_balanced}(x_2,y_1), \text{w_balanced}(x_3,y_2), \\
\quad \text{equal}(y_1,y_2), \text{sum}(y_1,y_2,y_3), \text{sum}(x_1,y_3,x_4)
\]

The first auxiliary variable is $y_1$ and $\text{w_balanced}(x_2,y_1)$ is the leftmost atom where $y_1$ occurs in. Thus, $\text{ms}(\text{w_balanced}(x_2,y_1)) = (\text{in}, \text{out})$. Now we are going to extend it to the clause as explained in Definition 5. We know that this clause belongs to a program where the set of mutually recursive predicates with $\text{w_balanced}$ is empty. First, it is extended to the remaining atoms with predicate $w_{\text{balanced}}$:

\[
\text{ms}(\text{w_balanced}(\text{tree}(x_1,x_2,x_3),x_4)) = \text{ms}(\text{w_balanced}(x_3,y_2)) = (\text{in}, \text{out})
\]

The remaining occurrences of the variable $y_1$ are in: $\text{ms}(\text{equal}(y_1,y_2)) = (\text{in}, \text{null})$ and $\text{ms}(\text{sum}(y_1,y_2,y_3)) = (\text{in}, \text{null}, \text{null})$.

For the next auxiliary variable $y_2$, the leftmost atom where the variable occurs in already has a non-null mode specification. For the remaining occurrences of $y_2$: $\text{ms}(\text{equal}(y_1,y_2)) = (\text{in}, \text{in})$ and $\text{ms}(\text{sum}(y_1,y_2,y_3)) = (\text{in}, \text{in}, \text{null})$.

The last auxiliary variables is $y_3$, so its leftmost atom $\text{ms}(\text{sum}(y_1,y_2,y_3)) = (\text{in}, \text{in}, \text{out})$ and, by extension, $\text{ms}(\text{sum}(x_1,y_3,x_4)) = (\text{null}, \text{in}, \text{null})$. Finally, $\text{ms}(\text{sum}(x_1,y_3,x_4))$ is non-deterministically completed.
Along the rest of the paper, we will write an atom with its internal terms split in two
tuples whenever the mode specification of the atom would be relevant. More precisely, we
will write $p(\overline{t}_i, \overline{t}_o)$ to denote that $\overline{t}_i$ and $\overline{t}_o$ respectively are the terms placed in input and
output arguments.

**Definition 6** A clause $C \equiv p(\overline{t}) : \neg p_1(\overline{s}^1), \ldots, p_m(\overline{s}^m)$ is called adjacent linked if each
auxiliary variable only occurs in $\overline{s}^j$ and $\overline{s}^{j+1}$ for some $j \in 1..m - 1$. A predicate $p$ is
adjacent linked in a program $P$ if all the clauses in Def$_P(p)$ are adjacent linked.

**Theorem 1** Any clause can be transformed into an equivalent set of adjacent linked
clauses.

**Proof.** Let $C \equiv p(\overline{t}) : \neg p_1(\overline{s}^1), \ldots, p_m(\overline{s}^m)$ be a not adjacent linked clause. Let $p_j(\overline{s}^j)$ be
the leftmost atom where some auxiliary $y$ appears and the condition fails. Notice that this
happens if $y$ does not appear in the next atom, but also if it appears in the next and also
in another one. In any case, we replace the atom $p_{j+1}(\overline{s}^{j+1})$ with a new atom $p_{j+1}'(\overline{s}^j)$ that
is obtained as follows:

1. $\overline{r} \equiv \overline{s}^{j+1} \cdot y \cdot y'$, where $y'$ is a fresh variable
2. $p_{j+1}'$ is a new predicate defined by the clause $p_{j+1}'(\overline{r} \cdot z \cdot z) : \neg p_{j+1}(\overline{r})$
3. Replace the variable $y$ with $y'$ in $p_k(\overline{s}^k)$ for every $j + 2 \leq k \leq m$

The above transformation is successively applied to the clause until an adjacent linked
clause is obtained. By unfolding, $C$ and the new set of clauses are equivalents.

On the basis of the Theorem 1, we will assume in the rest of the paper that every
program clause is adjacent linked.

**Example 2** The clause $E1.1$ in Example 1 is transformed into the adjacent linked clause:

$$E2.1 : \text{w\_balanced}(\text{tree}(x_1, x_2, x_3), x_4) : \neg \text{w\_balanced}(x_2, y_1), \text{w\_balanced'}(x_3, y_2, y_1, y_1'),
\text{equal''}(y_1', y_2, y_1, y_1', y_2),
\text{sum}(y_1', y_2', y_3),
\text{sum}(x_1, y_3, x_4)$$

**Definition 7** A predicate $p$ is called tail recursive w.r.t. a mode specification in a program
$P$ when Def$_P(p)$ consists of clauses of the form $p(\overline{t}_i, \overline{t}) : \neg q(\overline{s}), p(\overline{t}_i, \overline{s})$ where the atom $q$
does not recursively call the predicate $p$.  

\[19\]
Definition 8 A predicate $p$ is called linear recursive in a program $P$ when $\text{Def}_P(p)$ consists of clauses of the form $p(\overline{t}) : = p_1(\overline{s}^1), \ldots , p_m(\overline{s}^m)$ where at most one atom in $\{ p_k(\overline{s}^k) | 1 \leq k \leq m \}$ recursively call $p$ and all of them are also linear recursive.

Definition 9 A predicate $p$ is called non-linear recursive in a program $P$ when $\text{Def}_P(p)$ consists of clauses of the form $p(\overline{t}) : = p_1(\overline{s}^1), \ldots , p_m(\overline{s}^m)$ where more than one atom in $\{ p_k(\overline{s}^k) | 1 \leq k \leq m \}$ recursively call $p$.

4 Transformation Method for Auxiliary Variables Elimination

In this section, we will show a method for transforming a given clause, containing auxiliary variables, into a collection of auxiliary variables free clauses. The elimination of all auxiliary variables of a clause $C$ in a program $P$ is made in two steps:

1. The clause $C$ is transformed into an equivalent collection of clauses in the normalized form given in Fig. 2.1.

2. Any clause of the form of Fig. 2.1 is transformed into an equivalent finite set of auxiliary variables free clauses.

$$C \equiv p(\overline{t_i}, \overline{t_o}) : = q(\overline{t_i'}, \overline{t_o'}, p_k(\overline{t_i'}, \overline{t_o'}))$$

where $\text{auxvar}(C) \subseteq \text{var}(\overline{t_i'}) \cup \text{var}(\overline{t_o'})$ and $q$ is tail recursive w.r.t. $\text{ms}(q)$ in $P$ and $q$ does not recursively call $p$ (In part., $p \neq q$).

Figure 2.1: Normalized Form

Notice that $p_k$ can be the predicate $p$ itself or a different predicate. In particular, the definition clauses of a tail recursive predicate fulfill the form of the clause $C$ in Fig. 2.1.

The rest of this section is split in two parts. Subsect. 4.1 is devoted to the above second step and Subsect. 4.2 gives a method for carrying out the above first step.

4.1 Auxiliary Variable Elimination from a Normalized Clause

Given a program $P$ with a clause $C$ as in Fig. 2.1 where $q$ is defined (in $P$) by some clauses of one of the following two forms:

$$q(\overline{t_i}, \overline{t_o}) : = q_a(\overline{t_i'}, \overline{t_o'}) \quad (2.1)$$

$$q(\overline{s_i}, \overline{z}) : = q_b(\overline{s_i'}, \overline{s_o'}, q(\overline{s_i'}, \overline{z})) \quad (2.2)$$

Then, we define an equivalent set of auxiliary variables free clauses, $AVF(C)$, as follows:
1. \( AVF(C) := \{ p(t_i, t_o) \mid \neg p'(t_i, t_o) \} \) where \( p' \) is a new predicate. It is worthy to notice that the terms \( t_i^a \) and \( t_o^a \), and hence the auxiliary variables, have disappeared.

2. For each clause of the form (2.1) in \( Def_P(q) \) and \( \sigma = mgu(t_o, t_i^a) \):

\[
AVF(C) := AVF(C) \cup \{ p'(t_i, t_a) : q_a(t_i^a, t_o^a), p_k(t_i^a) \sigma, \sigma \}
\]

3. For each clause of the form (2.2):

\[
AVF(C) := AVF(C) \cup \{ p'(s_i, t_a) : q_b(s_i^b, s_o^b), p'(s_i, t_a) \}
\]

It suffices to eliminate, if there is some, the auxiliary variables in the definition of the new predicate \( p' \). This is made by iteration of the same method. Since the method never introduces new variables, this iterative process must be finite.

**Theorem 2** Let \( P \) be a program with a clause \( C \) as in Fig. 2.1, then \( P \) and \( P \{ C \} \cup AVF(C) \) are equivalents with respect to three-valued program completion.

**Example 3** Consider the following program \( P \):

\[
E3.1: \quad multiply(x_1, x_2, z) \leftarrow \text{mult}(0, x_1, x_2, z)
\]

\[
E3.2: \quad \text{mult}(x_0, 0, x_0)
\]

\[
E3.3: \quad \text{mult}(x_0, s(x_1), x_2, z) \leftarrow \text{sum}(x_0, x_2, y), \text{mult}(y, x_1, x_2, z)
\]

\[
E3.4: \quad \text{sum}(x_3, 0, x_3)
\]

\[
E3.5: \quad \text{sum}(x_4, s(x_5), z) \leftarrow \text{sum}(s(x_4), x_5, z)
\]

\[E3.3\] with \( ms(E3.1) \equiv ((\text{in}, \text{in}, \text{in}, \text{out}), (\text{in}, \text{in}, \text{out}), (\text{in}, \text{in}, \text{in}, \text{out})) \) is substituted by:

\[
E3.6: \quad \text{mult}(x_0, s(x_1), x_2, z) \leftarrow \text{sum}'(x_1, x_2, x_0, x_2, z)
\]

\[
E3.7: \quad \text{sum}'(x_1, x_2, x_3, 0, z) \leftarrow \text{mult}(x_3, x_1, x_2, z)
\]

\[
E3.8: \quad \text{sum}'(x_1, x_2, x_4, s(x_5), z) \leftarrow \text{sum}'(x_1, x_2, s(x_4), x_5, z)
\]

where \( y \) has been eliminated and \( E3.7 \) and \( E3.8 \) have been respect. obtained from \( E3.5 \) and \( E3.6 \).

**4.2 Transforming Clauses into its Normalized Form**

Suppose a given clause \( C \) that has some auxiliary variable and does not fulfill the requirements of Fig. 2.1. We can assume the following three facts about \( C \). First, it is adjacent linked. Second, it has exactly two body atoms, namely \( q_1(t_i), q_2(\overline s) \): at least two because it has auxiliary variables and exactly two because we can resume any conjunction of atoms to just one atom. Third, we can define a total specification mode \( ms(C) \): start in the body
w.r.t. the auxiliary variables, after extend it for the non-auxiliary ones, and finally do the head-extension. Hence, we can assume

\[ C \equiv p(\overline{t}_i, \overline{t}_o) : -q_1(\overline{t}_i, \overline{t}_o), q_2(\overline{s}_i, \overline{s}_o) \]

where \( \text{auxvar}(C) \subseteq \text{var}(\overline{t}_o) \cup \text{var}(\overline{s}_i) \). Therefore, if \( C \) does not fulfill the conditions of Fig. 2.1, then one of the two following facts must hold: (1) \( q_1 \) recursively calls \( p \) or (2) \( q_1 \) is not tail recursive w.r.t. \( ms(C) \). In the second case, \( q_1 \) is transformed into a tail recursive predicate w.r.t. \( ms(q_1(\overline{t})) \). After that the method of Subsect. 4.1 can be applied to \( C \). In the first case, \( p \) is a recursive predicate which is not tail recursive. Since tail recursion is a particular case of Fig. 2.1, we firstly transform \( p \) (hence, \( Def_{p}(p) \)) into a tail recursive predicate. After that, the clauses of the new \( Def_{p}(p) \) must be checked to find auxiliary variables.

The following two subsections give a method to transform a wide subclass of recursive definite programs into tail recursive ones. In the first, we transform any linear recursive predicate into a tail recursive one. In the second, we transform a subclass of general recursive predicates into linear recursive ones. Finally, we give an example using both transformation.

From Linear to Tail Recursive Predicates

Let us suppose that \( Def_{p}(p) \) consists of clauses the following two forms:

\[
\begin{align*}
    p(\overline{t}_i, \overline{t}_o) : & - q_a(\overline{t}_i, \overline{t}_o) \quad (2.3) \\
    p(\overline{s}_i, \overline{s}_o) : & - q_b(\overline{t}_i, \overline{t}_o, p_j(\overline{s}_i, \overline{s}_o), q_c(\overline{t}_i, \overline{t}_o)) \quad (2.4)
\end{align*}
\]

and \( p \) is linear recursive. Hence, each \( p_j \) is either \( p \) itself or a different predicate such that \( DEF_{p}(p_j) \) satisfies the same condition. We define an equivalent tail recursive set \( TR(p) \) of clauses as follows:

1. \( TR(p) := \{ p(\overline{x}, \overline{z}) : - p'(|| \cdot c_p \cdot \overline{x}, \overline{z}) \} \cup \{ p''(\overline{x}, \overline{z}) \} \) where \( p', p'' \) are new predicates and \( c_p \) is a constant associated to the predicate \( p \).

2. For each clause in \( DEF_{p}(p) \) of type (2.3):

\[
TR(p) := TR(p) \cup \{ p'(S \cdot c_p \cdot \overline{t}_i, \overline{z}) : - q_a(\overline{t}_i, \overline{t}_o), p''(\overline{t}_o[S], \overline{z}) \}
\]

3. For each clause \( C \in DEF_{p}(p) \) of type (2.4):

\[
TR(p) := TR(p) \cup \{ p'(S \cdot c_p \cdot \overline{s}_i, \overline{z}) : - q_b(\overline{t}_i, \overline{t}_o, p', (|| \cdot c_p[S] \cdot c_{p_j} \cdot \overline{s}_o, \overline{z})) \}
\]

\[
\cup \{ p''(\overline{s}_o[S], \overline{z}) : - q_c(\overline{s}_i, \overline{s}_o), p''(\overline{s}_o[S], \overline{z}) \}
\]

where \( c_{p_j} \) is a constant associated to the clause \( C \) and the predicate \( p \), \( c_{p_j} \) is a constant associated to the predicate \( p_j \), and \( \overline{z} = (\text{var}(\overline{s}_i \cdot \overline{s}_o) \cap \text{var}(\overline{t}_i \cdot \overline{t}_o)) \setminus \text{var}(\overline{s}_o) \).
4. We repeat the steps 2 and 3 for every clause $C \in Def_P(p_j)$ of each $p_j$.

**Theorem 3** Let $P$ be a program that defines a linear predicate $p$, then the programs $P$ and $P \setminus Def_P(p) \cup TR(p)$ are equivalents w.r.t. three-valued program completion.  

**From Non-linear to Linear Recursion**

In this subsection, we show how to transform a subclass of non-linear (or general) recursive predicate definitions into linear definitions. In spite of not being a general method, this transformation is useful in most cases. The non-linear recursive predicates we handle are defined by a set of clauses $Def_P(p)$ of the following form:

$$p(t_i, t_o) : = q_a(t_i^1, t_o^1), p_1(t_i^1, t_o^1), \ldots, p_n(t_i^n, t_o^n), q_b(t_i^n, t_o^n)$$

where each $p_j$ may be $p$ itself or another predicate $q$ that calls $p$ and

$$\text{var}(t_i^k) \cap \text{var}(t_o^k) = \emptyset \quad \text{and} \quad \text{var}(t_i^k) \cap \text{var}(t_o^k) = \emptyset \quad \text{for each} \quad k \neq h$$

$$\text{auxvar}(t_i) = \bigcup_{k=1}^n \text{auxvar}(t_i^k) \quad \text{and} \quad \text{auxvar}(t_o) = \bigcup_{k=1}^n \text{auxvar}(t_o^k)$$

We obtain an equivalent set of linear clauses, called $LR(p)$ as follows:

1. $LR(p) := \{ p(t_i, t_o) : = p'(([c_p, t_i]), [t_o]) \} \cup \{ p'([], []) \}$ where $p'$ is new and $c_p$ is a new constant associated to $p$

2. For each clause of the form (2.5) with $n = 0$:

$$LR(p) := LR(p) \cup \{ p'(([c_p, t_i]), [t_o]) : = q_a(t_i^1, t_o^1), q_b(t_i^n, t_o^n), p'(S_i, S_o) \}$$

3. For each clause of the form (2.5) with $n > 0$ (let $c_{p_j}$ be a constant associated to $p_j$):

$$LR(p) := LR(p) \cup \{ p'(([c_p, t_i]), [t_o]) : = q_a(t_i^1, t_o^1), q_b(t_i^n, t_o^n), p'(S_i, S_o) \}$$

4. Besides, for each non-linear $p_j(j \in 1..n)$, we add a new clause, exactly as in the above steps 2 and 3, except that $c_p$ is replaced with the corresponding constant $c_{p_j}$.

**Theorem 4** Let $P$ be a program that defines a linear predicate $p$, then the programs $P$ and $P \setminus Def_P(p) \cup LR(p)$ are equivalents w.r.t. three-valued program completion.  

**Example 4** Consider the following program that computes the preorder of a binary tree:

$E4.1 : \text{preorder}($null,$[])$

$E4.2 : \text{preorder}($tree($x_1, x_2, x_3), [x_1 | z_1]) : = \text{preorder}(x_2, y_1), \text{preorder}(x_3, y_2),$  

append($y_1, y_2, z_1$)

$E4.3 : \text{append}([$], x_4, x_4)$

$E4.4 : \text{append}([x_5 | x_6], x_7, [x_5 | z_2]) : = \text{append}(x_6, x_7, z_2)$
preorder is transformed into linear recursive w.r.t. \( \text{ms}(\text{preorder}) \equiv (\text{in, out}) \) as follows:

\[
\begin{align*}
E4.5 & : \text{preorder}(x_0, z_0) := \text{po}'([x_0], [z_0]) \\
E4.6 & : \text{po}'([], []) \\
E4.7 & : \text{po}'([\text{nil} | S_i], [[| S_o]) := \text{po}'(S_i, S_o) \\
E4.8 & : \text{po}'([\text{tree}(x_1, x_2, x_3)|S_i], [[x_1|z_1]|S_o]) := \text{po}'([x_2, x_3|S_i], [y_1, y_2|S_o]), \\
& \quad \text{append}(y_1, y_2, z_1)
\end{align*}
\]

The new predicate \( \text{po}' \) is also transformed w.r.t. \( \text{ms}(\text{po}') \equiv (\text{in, out}) \) as follows:

\[
\begin{align*}
E4.9 & : \text{po}'(x_0, z_0) := \text{po}''(x_0, [, ], z_0) \\
E4.10 & : \text{po}''([], S''_0, z_1) := \text{po}''_{\text{aux}}([|], S''_0, z_1) \\
E4.11 & : \text{po}''([\text{nil} | S''_i], S''_0, z_2) := \text{po}''(S''_i, [c^1_{po}] | S''_0, z_2) \\
E4.12 & : \text{po}''([\text{tree}(x_1, x_2, x_3)|S''_i], S''_0, z_3) := \text{po}''([x_2, x_3|S''_i], [x_1, c^2_{po}] | S''_0, z_3) \\
E4.13 & : \text{po}''_{\text{aux}}([x_4|[, |], x_4) \\
E4.14 & : \text{po}''_{\text{aux}}([x_5, c^3_{po}] | S''_0, z_4) := \text{po}''_{\text{aux}}([| | x_5] | S''_0, z_4) \\
E4.15 & : \text{po}''_{\text{aux}}([|[x_6, x_7|x_8], x_9, c^2_{po}] | S''_0, z_5) := \text{append}(x_6, x_7, y), \text{po}''_{\text{aux}}([|[x_0|y] | x_8] | S''_0, z_5)
\end{align*}
\]

The auxiliary \( y_1 \) and \( y_2 \) of E4.15 can be eliminated using the method of Subsect. 4.1.

5 Future work

We have designed a syntactic transformation method that eliminates the auxiliary variables from a wide class of definite program. We plan to generalize the presented transformation method for handling with: (1) the whole class of definite logic programs, and (2) the class of normal logic programs. We also plan to extend the transformation in order to eliminate the so-called isolated variables (see Ex. 1). Besides, we have the intention to integrate the transformational method into the implementation of constructive negation explained in (Álvez, Lucio, Orejas, Pasarella, and Pino 2004), as an improvement of this prototype.
References


438.
Stuckey, P. J. (1995). Negation and constraint logic programming. Information and Com-
putation 118(1), 12–33.
Tärnlund (Ed.), Proceedings of the Second International Logic Programming Confer-
Defining Logical Constants, the Insight from Basic Logic

DENIS BONNAY
IHPST / University Paris I
denis.bonnay@univ-paris1.fr

BENJAMIN SIMMENAUER
IHPST / University Paris I
benjamin.simmенauer@yahoo.fr

ABSTRACT.
We tackle the question: “what is a good definition of a logical constant?”. We work in the Basic Logic proof-theoretic setting, in which logical constants are introduced through definitional equations, the underlying idea being that they reflect structural links in the meta-language. Our main result is that general solution methods for definitional equations correspond to cut elimination. The cut-elimination proof is therefore built-in in the good definition of the connectives.

The topic of this paper is proof theory as applied to questions relative to the determination of the class of logical constants. This is both a logical and philosophical programme: the main contributions are (Gentzen 1934-35), (Prawitz 1965), (Hacking 1979), and, more recently, (Belnap 1982), (Dosen 1989), (Girard 1993). We wish to throw light upon three kind of problems:

1. How to define a logical constant? Relevant questions are: can we expect a completely explicit definition? In which language are we to state it?

2. How to unify logical constants across various logical systems?

3. How to exclude pathological cases like Prior’s famous Tonk?

We will use Basic logic, a new sequent calculus, developped in (Sambin et al. 2000). The main ideas are: a) a precise definition of logical constants through an analysis of the interaction between object- and meta-language and b) a very strict control of contexts and structural rules which can be liberalized step by step to gain a vast family of already known logics.
We will first introduce Basic logic and explain why it proposes an interesting approach to the first two problems. And then we will show how to use it to solve the third one. Our main result will thus be a correctness criterion for purported definitions of logical constants.

1 The Basic Logic Framework

1.1 A “consecutional” Meta-language

Basic Logic’s original aim is to restore the unity of logic beyond the diversity of substructural logics. In Basic Logic calculus, $B$, there are only identity axioms, exchange rules, rules for logical operators, and “composition rules” (that results from a splitting of cut rule). But there is a new constraint upon logical rules: they must be formulated according to a principle of visibility: in $B$-logical rules, contexts aside principal formulas have to be empty. $B$ is thus a kind of logical matrix: it provides once and for all a stock of logical constants, that remains unchanged, whatever action you perform to enrich this calculus. Two such kinds of actions are possible: adding structural rules, and liberalizing contexts. Different substructural logics can thus be seen as extensions of $B$: for instance, if you liberalize left and right contexts, you get linear logic.

In Basic Logic, logical connectives are selected according to a principle of reflection, which says that any logical connective must reflect a link in the meta-language. Basic logicians claim that two links only are required: “and” and “yields”. We can obtain such a language through a sequent calculus sometimes called “structural”,\(^1\) because of its deprival of logical connectives. It is better, as the further definitions should establish, to call this language a “consecutional” language.

First, we need a definition of structure:

**Definition 1 (structure)** Let $\mathcal{L}$ be any language with symbols for formulas: $A$, $B$, ... and sets of formulas : $\Gamma$, $\Delta$, .... In $B$ we will also need a binary punctuation sign (the comma). A structure is any string of symbols which is either a formula $A$ (or a set of formulas $\Gamma$) or a list of formulas (sets of formulas) punctuated by commas.

But that is not enough to obtain the meta-language used in Basic Logic. We wish to form consecutions, or sequents, in order to catch the link “yields”:

**Definition 2 (consecution)** Let $X$ and $Y$ be any structures as defined above. Then you can form the consecution $X \vdash Y$ with $X$ as antecedent and $Y$ as consequent.

According to the scope of the link considered, each of the two metalinguistic “and” and “yields” can be expressed through different constructions, according to their possible positions (inside or outside a sequent). For instance, the figure:

\(^1\)See (Dosen 1989), for example.
\[ \Gamma, A \vdash B \quad \Delta \vdash B \quad \frac{}{\Gamma, A, \Delta \vdash B} \]

should be read as follows:

\(((\Gamma \text{ and } A) \text{ yields } B) \text{ and } (\Delta \text{ yields } B)) \text{ yields } ((\Gamma \text{ and } A \text{ and } \Delta) \text{ yields } B)\)

So far for morphology. To complete the basic logic meta-language description, we must add two rules, immediately derived from the meaning attached to the link “yields”. These are the two composition rules (abbreviated as cut left and cut right):

\[\frac{\Gamma \vdash A \quad \Gamma' \vdash \Delta}{\Gamma'(\Gamma/A) \vdash \Delta} \quad \text{cut L} \quad \frac{\Gamma' \vdash \Delta' \quad A \vdash \Delta}{\Gamma' \vdash \Delta'(\Delta/A)} \quad \text{cut R}\]

The composition on the left rule allows one, provided that \(A \in \Gamma'\), to replace one occurrence of \(A\) by \(\Gamma\) in \(\Gamma'\). The composition on the right rule allows one, provided that \(A \in \Delta'\) to replace one occurrence of \(A\) by \(\Delta\) in \(\Delta'\). To sum up the justification of these half-cut rules, one can say that when \(A\) appears on the left of “yields”, it can be replaced by the set of hypotheses used to deduce \(A\) (the formulas that \textit{yield} \(A\)), and when it appears on the right, it can be replaced by the set of its conclusions (the formulas that are yielded by \(A\)).

### 1.2 Definitional equations

Reflection is carried out by so-called “definitional equations”, which take the form of equivalences between expressions of the structural meta-language and \textit{logical} sequents: the idea is that for each possible occurrence of a link in the meta-language, we can define a connective of the object-language (the logical language), in order to get a purely “consecutional” definition of the logical constants. A definitional equation for any constant \(\square\) has the following form:\(^2\)

\[ A \square B \vdash \Delta \text{ iff } A, B \vdash \Delta \quad (\Box) \]

To solve such an “equation” comes to giving two rules, corresponding to the two directions of the biconditional (it echoes, in Gentzen systems, left and right introduction rules). In left-to-right direction, the rule is straightforward: it tells us how to build a complex formula with \(\square\) as principal connective:

\[ \frac{A, B \vdash \Delta}{A \square B \vdash \Delta} \quad \square-\text{Formation} \]

This is the rule of \(\square\) formation (F for short). The other case is far more difficult. First, we should consider a symmetrical rule of \textit{implicit reflection}, that is, a rule which specifies the way \(\square\) reflects a meta-linguistic construction:

\[ \frac{A \square B \vdash \Delta \quad \text{ implicit } \square-\text{reflection}}{A, B \vdash \Delta} \]

\(^2\)Of course, in this precise example, \(\square\) reveals itself to be the linear multiplicative conjunction \(\otimes\).
One can construe this rule as an implicit characterization, not of the situation in which you can assert \( A \supset B \), but of the situation in which you can use \( A \supset B \) in the process of a proof. If we wish to account for the complete meaning of the logical connectives, we have to make this reflection explicit. In other words, we want a rule which is equivalent to implicit \( \Box \)-reflection (IR for short), but does not presuppose any knowledge of the complex formula, only a knowledge of the resources needed to assert this complex formula.\(^3\)

This task is accomplished in three steps:\(^4\)

1. Trivialization of the premiss of implicit reflection.

2. Application of implicit reflection to the result of 1. : the conclusion of this application is called “axiom of \( \Box \)-reflection”.

3. Composition of axiom of \( \Box \)-reflection with premises of the form \( \Gamma \vdash A \) and \( \Delta \vdash B \).

Finally the rule of explicit \( \Box \)-reflection (ER for short) is

\[
\Gamma \vdash A \quad \Delta \vdash B \quad \text{explicit} \quad \Box \text{-reflection}
\]

2 How to choose the right constants

2.1 Back to Tonk, the normative problem

The idea behind basic logic is somewhat similar to an idea Dosen’s exposed in his 1989 paper, “Logical constants as punctuation marks” (Dosen 1989). The idea was to construe logical constants as encoding certain structural features of deductions, hence the metaphor of the punctuation marks. This meant that logical constants should be introduced thanks to a double-line rule in the style of basic logic’s equations. For example, he suggested the following rule for the implication

\[
\Gamma, A \vdash \Delta, B \quad \text{\( \rightarrow \)}
\]

Thus, Dosen endorsed the program consisting in defining logical constants in a proof-theoretical setting. Well-known objections to this approach go back to Prior’s proposal to introduce a silly connective he named Tonk and which rendered inconsistent any language it would be added to (Prior 1960). And one could define through a double-line rule a connective as devastating as tonk (see the \( \odot \) of section 2.3 for an example).

So, how should one deal with that problem? Three different reactions to Prior’s provocation can be distinguished:

---

\(^3\)This point could be made clearer by a drawing a comparison with the use of the principle of inversion to obtain elimination rules from introduction rules in natural deduction.

\(^4\)See section 2.2 for how that looks.

30
Indifference This is the position Dosen praised. As far as what is aimed at is a
demarcation of logic, there is no reason why one should banish bad logics out of the
realm of logic. Dosen explains what form a satisfactory analysis of a logical constant
must have, but he does not bother to implement a normative element in his formal
categorization.

Global response Belnap, in his famous response to Prior (Belnap 1961), stated criteria
the definition of a logical constant should meet to be admissible. Namely, the ex-
tended system should be conservative over the old one. So Belnap does not seek to
explain the form a satisfactory analysis of a logical constant should have, but he gives
a normative global constraint any truly logical constant should satisfy.

Local response One would like to have it both ways: the normative element should be
built in into the form the analysis must have. In the setting of natural deduction,
Dummettian requisites of harmony play exactly this role see (Dummett 1991).

Taking for granted that from a philosophical point of view, the local response is the
most desirable one, our point is that from a logical point of view, basic logic is a good
setting to implement it.

2.2 Solvability, a built in solution to the normative problem

Here is the logical point. As soon as one does not go into higher-order languages, the
cut-elimination property ensures the subformula property, which in turn amounts to a
conservativity result. Therefore, Belnap’s conservativity constraint can be replaced at
little cost by the requisite that the extended deductive system should still enjoy the cut
elimination property. But the proof of cut elimination is modular enough; to check that cut
elimination is preserved, it is enough to check that one can perform the essential reduction
step for the new constant.5 Therefore, to implement the local response, we just have to
ensure the way logical constants are defined gives us that step.

Definition 3 (correct definition) A set of introduction and elimination rules define
correctly a logical constant if and only if when they are added to a basic deductive sys-
tem, the new system still enjoys cut elimination.

Definition 4 (truly solvable equation) An equation is truly solvable iff

• one can find a proof of ER from IR which 1) starts with one application of implicit
  reflection to an axiom and 2) uses only elementary means (axioms, structural rules
  and cutL or cutR on subformulas of the formula whose main connective is being
  defined).

5See (Sambin et al. 2000) for details about cut elimination in basic logic; visibility, which prevents
contexts from occurring aside a principal formula simplifies greatly the usual proof.

31
• one can find a proof of IR from ER  

Claim 1: If a definitional equation is truly solvable, then the corresponding formation and explicit reflection rules define correctly a logical constant.

The claim is rather simple to prove, even if one has to reach a sufficient degree of generality. We shall consider an arbitrary constant \( \odot \) whose definitional equation (E) has been truly solved. Without loss of generality, we assume that \( \odot \) is a binary connective and appears in (E) on the left of the sequent. (E) is of the form

\[
\frac{Seq_1(A, B/\Delta, \Delta')}{A \odot B \vdash \Delta, \Delta'} (E)
\]

where \( Seq_1(A, B/\Delta, \Delta') \) stands for one or two sequents whose premisses are among \( A \) and \( B \) and whose conclusions are structures among \( \Delta \) and \( \Delta' \).

By hypothesis, (E) is solved by one proof schema, let’s call it (Sol):

\[
\frac{A \odot B \vdash A \odot B}{Seq_\Gamma(A, B/A \odot B)} \odot IR
\]

\[
\vdots
\]

\[
\frac{Seq_2(\Gamma, \Gamma'/A, B)}{\Gamma, \Gamma' \vdash A \odot B} (Sol)
\]

where \( Seq_\Gamma(A, B/A \odot B) \) is an instantiation of \( Seq_1(A, B/\Delta, \Delta') \) that is the (or one of) the sequent(s) \( Seq_1(A, B/\Delta, \Delta') \) with \( A \odot B \) standing for \( \Delta \) (one of \( \Delta \) and \( \Delta' \)); and where \( Seq_2(\Gamma, \Gamma'/A, B) \) represent in the same manner the premiss(e(s) of the explicit reflection rule(s).  

Now, suppose we have to deal with a principal cut on a formula \( A \odot B \), whose form will be

\[
\Pi_1 \quad \Pi_2
\]

\[
\vdots
\]

\[
\frac{Seq_1(A, B/\Delta, \Delta')}{A \odot B \vdash \Delta, \Delta'} \odot F \quad \frac{Seq_2(\Gamma, \Gamma'/A, B)}{\Gamma, \Gamma' \vdash A \odot B} \odot ER
\]

\[
\Gamma, \Gamma' \vdash \Delta, \Delta'
\]

Explicit reflection has been obtained through solving (E). Thanks to (Sol), we can obtain a proof of the same sequent using implicit reflection instead of explicit reflection.

---

6 Adding this second condition corresponds to the shift from mere harmony to stability in a natural deduction setting.

7 Therefore Sol is also schematic in that it may have to give two explicit reflection rules, as in the case of the additive disjunction. This will happen because one demands not only that implicit reflection yields explicit reflection, but that implicit reflection be equivalent to explicit reflection.
\[
\begin{align*}
\Pi_1 & \quad \frac{A \otimes B \vdash A \otimes B}{\text{Seq}_v(A, B/A \otimes B)} \otimes IR \quad \Pi_2 \\
\vdots & \quad \vdots \\
\text{Seq}_1(A, B/\Delta, \Delta') \otimes F & \quad \vdots \\
A \otimes B \vdash \Delta, \Delta' & \quad \text{Seq}_2(\Gamma, \Gamma'/A, B) \quad (Sol)
\end{align*}
\]

Then we can glue the leftmost part of the proof on top of the axiom in order to obtain a proof of the final sequent without the last cut: we just have to substitute \(\Delta, \Delta'\) for \(A \otimes B\) in the consequence of the axiom and in linked occurrences downward.\(^8\)

\[
\begin{align*}
\Pi_1 & \quad \vdots \\
\text{Seq}_1(A, B/\Delta, \Delta') \otimes F & \quad \vdots \\
A \otimes B \vdash \Delta, \Delta' & \quad \text{Seq}_2(\Gamma, \Gamma'/A, B) \quad (Sol)
\end{align*}
\]

What we have is not a real proof, because \(\otimes IR\) is not deductive rule. But happily, the implicit reflection rule just follows a formation rule, so we can eliminate this roundabout step.

\[
\begin{align*}
\Pi_1 & \quad \vdots \\
\text{Seq}_1(A, B/\Delta, \Delta') \otimes F & \quad \vdots \\
A \otimes B \vdash \Delta, \Delta' & \quad \text{Seq}_2(\Gamma, \Gamma'/A, B) \quad (Sol)
\end{align*}
\]

By hypothesis, S contains only axioms and cuts on formulas of lower complexity than \(A \otimes B\). Therefore, we have completed the desired reduction step. QED.

### 2.3 Further questions

We would like to know how the set of equations of basic logic relates to our notion of truly solvable equation.

First, note that our criterion is not trivial. We could imagine defining a new constant \(\otimes\) through the following equation

\(^8\)This kind of substitution on proof-trunks preserves the property of being a proof-trunk, as shown by the substitution lemma 4.1 in (Sambin et al. 2000).
$$\Gamma \vdash A \quad \Gamma' \vdash B \quad \therefore \quad \Gamma, \Gamma' \vdash A \odot B$$

But one can easily check that \((\odot)\) is not solvable. In fact, adding \((\odot)\) would license an incoherent way of separating contexts.\(^9\) In contrast, defining equations for conjunctions and disjunctions (multiplicative and additive) are truly solvable as expected.

Moreover, we would like to know whether the set of equations of basic logic is complete with respect to true solvability. If we restrict our attention to binary connectives defined along \((E)\), it is possible to look at all possible equation (in a purely combinatorial way). One can check that one gets three possible equations on each side, two of them give rise to genuine basic connectives, and the last one is not truly solvable, like \((\odot)\). But the restriction to connectives of arity no greater than 2 is not harmless in a substructural setting like the one of basic logic; one could introduce a ternary \(\otimes\) which would not be definable from the binary one.

Another problem is that one cannot find a satisfactory definitional equation for implication within the framework of \(B\), that is satisfying visibility. Actually, we would like to have a definitional equation like:

$$\Gamma \vdash A \rightarrow B \text{ iff } \Gamma, A \vdash B \quad (\rightarrow)$$

But in \((\rightarrow)\), visibility is betrayed: it means that, to solve the equation for the conditional, we are compelled to liberalize left contexts (that is, to solve the equation not in \(B\), but in \(BL\)). The alternative proposal is to admit a richer meta-language, in which one can use “nested” occurrences of \(\vdash\):

$$\Gamma \vdash A \rightarrow' B \text{ iff } \Gamma' \vdash (A \vdash B) \quad (\rightarrow')$$

But this is scarcely acceptable, because one still ignores how to use such a weird meta-language. We conclude with another remark: the case of implication is more entangled because we would like \(\rightarrow\) to reflect directly the meta-linguistic link \(\vdash\) (ie, we would like a deduction theorem). But: first, it seems we cannot have such a theorem in \(B\), and second, if we admit that what varies among distinct substructural logics is the nature and meaning of \(\vdash\) (the consequence relation), how could we account for a unique and definable-once-for-all-in-\(B\) connective \(\rightarrow\)?

**Acknowledgements**

We thank the anonymous referees for their useful remarks, Jacques Dubucs and Mikaël Cozic for stimulating comments.

---

\(^9\)Applied to an axiom, the upward rule gives \(\vdash B\). Therefore, conservativity will fail, so the equation is not truly solvable, as one could guess by trying to solve it.
References

Incremental Processing and the Generic Interpretation of Indefinites

Stefan Bott and Oriol Valentín
GLICOM
Universitat Pompeu Fabra
{stefan.bott, oriol.valentin}@trad.upf.es

Abstract.
In this paper we propose a dynamic model of semantics which extends Heim’s theory of File Change Semantics in two aspects. First it is fully compatible with an incremental model of discourse interpretation based on Lambek Calculus. Secondly, we distinguish between two different kinds of references, one for the discourse referent itself and one for the class of which the referent is an instance of. Each specific NP is associated with a referent of both types, while generic NPs only refer to their class. We show that this distinction is needed in order to explain a series of defocusing phenomena which are sensitive to class information. This analysis can be carried over to the treatment of the distinction between stage-level and individual-level predicates within a strictly incremental language processing model. We present a formalised sketch of the incremental processing of indefinite NPs which assigns generic and specific interpretations depending on the nature of the predicates by which they are selected.

1 Generic and specific reading of indefinites and bare plurals
Bare plural and singular indefinite NPs may either get a specific or a generic reading. Among other factors the type of reading seems to depend largely on the type of predicate which selects the NP. Carlson (1977) introduced a distinction between stage-level and individual-level predicates (S-level and I-level predicates respectively). They differ in that the first type makes an assertion about an individual as such, while the second makes an assertion about an individual within a stage, a spatiotemporal slice of an individual.

(1) Pirates are dangerous.
(2) Pirates are approaching.
(3) A pirate is vicious.
(4) A pirate is furious.

The subjects in (1) and (3) get a generic interpretation while the subjects of (2) and (4) are most naturally interpreted as specific referents. Accordingly ‘dangerous’ and ‘vicious’ are typical I-level predicates, while ‘furious’ and ‘approaching’ are S-level predicates. An important observation is that the subjects selected by I-level predicates can only get a generic interpretation while the subjects of S-level predicates are generally ambiguous between a specific and a generic reading, where the generic reading is usually less preferred. In addition to its specific reading (4) may have a reading in which ‘being furious’ is a general property of pirates. Such a reading can often be enforced by adverbs as ‘always’, ‘usually’ or ‘generally’, such as in ‘a pirate is always angry’. On the other hand indefinite subjects of I-level predicates can not have a specific reading. (1) cannot get interpreted in a way where there is a specific set of pirates which at a given point in time and space are dangerous.

The generic reading of S-level predicate subjects may be blocked when the event has a spatiotemporal anchor:

(5) a. A pirate sings pirate songs.
   b. A pirate sang pirate songs.
   c. A pirate is angry today
   d. *A pirate is vicious today

Since it carries present tense the verb ‘sings’ may have a time-independent reading in (5a), which might be interpreted as ‘it is a general property of pirates that they sing pirate songs’. (5b) is tensed and a generic interpretation is clearly impossible. In (5c) the adverb today block the generic reading. Finally (5d) shows that temporal adverbs are incompatible with generic readings. We will treat this as a feature mismatch below, where the I-level is incompatible with the requirement of a temporal adverb to refer to a spatiotemporal stage.

Different authors have given different explanations to this phenomenon. Carlson (1977) attributed the different readings to a difference in the selectional properties of the predicate in combination with a generic operator. Within the range of a generic operator S-level and I-level predicates select both for an individual reference, but he subdivides individual references into object references and kind references. S-level predicates (under the influence of the generic operator) select for a object reference (a specific reading) while I-level predicates select for a kind reference (which yields the generic reading). Diesing (1992) and Kratzer (1995) explain the different interpretations within a framework where the different readings of indefinites follow from different base and surface positions. They interpret subjects of S-level predicates as generated VP internally, while I-level predicates - according to them - generate their subjects as genuine external arguments which are generated outside the VP domain. They assume that existential closure (Heim 1982) applies to NPs within the VP while free NPs outside the VP must be bound by an operator, in this case the one
which is associated with genericity. From this it follows that the VP-internal subjects of I-level predicates move for independent reasons outside the VP but may be reconstructed to their original position at LF. This explains the two possible readings for I-level predicates. The subjects of S-level predicates cannot be reconstructed inside the VP since they were generated externally from the beginning.

An interesting proposal which is not based on invisible syntax was made in Erteschik-Shir (1997). Erteschik-Shir argues that the difference can be reduced to a difference in focus structure, and can hence be interpreted on a semantic-pragmatic level. She uses the file card system similar to Heim’s (1982) File Change Semantics to represent discourse referents. This point of view has the advantage that no non-overt syntactic operators and LF reconstruction rules must be stipulated (as in Diesing’s proposal). An additional advantage is that it also explains the constraints on focus-background partitioning (or Focus Structure in her terms). She adopts Heim’s novelty-familiarity condition which predicts indefinites to typically introduce a new discourse referent, while the definite article signals that the discourse referent must be discourse old. She argues that generic indefinites are exceptional, however, in that they may serve as topics. Her framework explains the difference between generic and specific indefinites under two assumptions: 1) All sentences must have topics. If no overt topic is available the stage (the spatiotemporal anchor) may serve as an implicit topic. Foci enter a predication relation with topics. 2) Only S-level predicates select for a stage topic, while I-level predicates are incompatible with stage topics, because the events they introduce cannot be anchored in time and space. It follows from the two assumptions that the subjects of I-level predicates must be overt topics, hence discourse-given, while the subjects of S-level predicates may be focused and discourse new. In the latter case the stage serves as an implicit topic, allowing for the rest of the sentence to be focused.

The appeal of Erteschik-Shir’s approach lies clearly in that it derives the two different readings from purely discourse-semantic factors and does away with the need for covered syntactic devices. In this paper we want to follow a similar approach, but we will argue that the distinction between generic and specific readings follow only from their representation in a formal discourse model. What we ultimately derive is a processing model which only reads from the surface string of words and interprets the relevant semantic properties of individual words (or more technically: increments) as instructions for (one or more) file changes. In order to acheive this we employ Lambek Calculus which proceeds incrementally. In this way the syntactic-semantic process does not have to come to the end of the sentence in order to derive a discourse level representation. We will explain the details of this proposal in section 5.

Incremental natural language models have received increasing attention within the last years because they explain some psycholinguistic findings and offer an effective strategy to reduce ambiguity locally without processing a whole sentence (Steedman 2000). They can handle garden pathening phenomena and explain non-constituent coordination, a phenomenon which is hard to account for in traditional phrase structure grammars. In addition, Steedman points out that only incremental language processing models do not have to apply different grammars for competence and performance. We shall see in section
4 that a dynamic model of semantics such as File Change Semantics fits perfectly in a incremental language processing model such as Lambek Calculus.

None of the mentioned attempts to explain the generic/specific distinction is compatible with incremental language processing. In Steedman’s model discourse information plays an important role in local disambiguation within the syntactic/semantic process, an idea which we will exploit in our proposal. Incremental processing implies that the processor has no look-ahead access to the information about the type of predicate at the moment when the indefinite subject ‘pirates’ in (1) and (2) is processed, but this information will finally determine the nature of the nominal reference. The representation of the indefinite NP must therefore be ambiguous or underspecified with respect to the specific/generic distinction until the point where the predicate is processed.

2 Information structure and discourse referents

A good deal of the justification of the model presented here comes from facts about information structure and appropriateness conditions on focus-background partitioning in a given context. We agree with Ertesich-Shir (1997) that both information structure and the different readings of indefinites must be resolved on the level of discourse semantics.

Although the exact nature of the constraints which discourse semantics impose on information structure are subject to an ongoing debate, there is a general agreement that the occurrence of a discourse referent influences the ability of this discourse referent to serve as a focused element in subsequent sentences. In languages which mark foci phonologically (e.g. English) we can observe a deaccentuation of the element in question, that means the shift of an accent from a position where it would fall in neutral contexts to a position to the left. The conditions on defocusing are in some way similar to the conditions of pronominalisation. Capital letters mark the main stress of the sentence. In (6a) ‘John’ is deaccented and the stress moves to the next word on the left which is in focus, in this case ‘TRUST’.

(6)  
\[ \begin{align*}
\text{a. } & \text{ John is a nice lad. Nevertheless I don’t TRUST John/him.} \\
\text{b. } & \text{ I left a cookie on the table. Now it’s not there any more. Somebody must have EATEN the cookie/it.}
\end{align*} \]

However, it is not always the discourse referent itself that causes defocusing. In many cases the mention of the class which the discourse referent belongs to may cause the defocusing of a second discourse referent which belongs to the same class. We will call a new discourse referent instance of his class. Although the conditions for defocusing are similar to the ones for pronominalisation they are not identical. Note that pronominalisation is not possible, because the referent is evidently not the same. A pronoun requires an explicit discourse referent as an antecedent while defocusing may happen on the basis of information about the class of a referent.

(7)  
\[ \begin{align*}
\text{a. } & \text{ Susan ate a COOKIE. Also BILL ate a cookie/*it.}
\end{align*} \]
b. John ate an APPLE. Mary PAINTED an apple/*it.

Defocusing may also apply between a generic NP and a specific instance of this class. Again the conditions for pronominalisation are different from the conditions on defocusing.

(8) a. Firemen are altruistic. Mary FANCIES a fireman/*him.
    b. Pirates are hideous. Nevertheless Sue MARRIED a pirate/*him.

A simple explanation of these cases in terms of discourse referents is not plausible, since the extension of the defocused element in the examples above has not the same extension as the antecedent by virtue of which they are defocused and deaccented. In this respect defocusing is clearly distinct from pronominalisation. What these examples have in common is that defocusing happens as a result of the mention of the class, either directly or via the mention of an instance of this class.

3 The double reference hypothesis

In order to explain both the defocusing phenomena and specificity phenomena in incremental processing we propose that each (specific) discourse referent has two references, one for its class and one for the instance which it represents. The class referent is obligatory, since every common noun belongs to a class. The instance reference is optional, since generic NP do not point to an instance of their class, but to the class itself.

Consider (7a) above. After the first sentence has been processed the file will contain the following cards:

\{x; Mary(x), x ate y\}, \{y; y belongs to class z\}, \{z; class: cookie\}.

The extension of file change semantics we propose here is the introduction of a file card of the type exemplified by z. Such cards merely represent the class to which some referents belong to. In this example y belongs to class z, an individual cookie belonging to the class of cookies. Once y has been mentioned in the discourse it will become activated, which means that it has a salient status a the given point in the discourse. At the same time y gets activated, z will be activated as well, because it contains important information about y. The activation of z will in turn licence the defocusing of any new token belonging to the class z. This explains why cookie in Also BILL ate a cookie is deaccented.

The class reference we assumes also captures Carlson (1977)'s idea of kinds. According to Carlson scorpions in scorpions are poisonous are kind-referring. In our model the property of being poisonous will be annotated on the class card and no specific card (i.e. a representation of a specific set of scorpions) will be created. As we have seen above the correct interpretation of a bare plural subject depends on the nature of the predicate. The main problem we encounter when we try to process such sentences incrementally, is the following: at the time when a preverbal indefinite subject is processed, the processor has no information about the semantic nature of its predicate. For this reason the ambiguity between a specific and a generic reading cannot be resolved at this point. If we interpret semantic processing as a continuous update of a file, we cannot say anything about the
nature of the new file card which must be created for an indefinite according to Heim’s novelty- familiarity condition.

In our model a class reference will be created/activated for any new indefinite NP. In addition a temporal specific reference will be created to capture the potential of any indefinite NP to be specific. If the incremental process goes on and finds an S-level predicate, this will preserve the class and the instance reference. The temporal specific reading will be consolidated in this case. When an I-level predicate follows, the instance referent will be destroyed since it is not compatible with an I-level predicate. The creation of a temporal instance referent is necessary in order to preserve incrementality and avoid look-ahead devices. In cases like (5b) a modification or a spatiotemporal anchor may, in turn, force the specific reading and convert the temporally created instance reference into a permanent one, which may not be erased by an I-level predicate.

Metaphorically speaking the instance referent is created ‘on-top’ of the class referent. The effect of this is that, if it is present, only the instance referent can serve as an antecedent for a pronoun, the class referent will be inaccessible. If no instance referent is available the class referent will be accessible for pronouns and a co-referring pronoun can only refer to the class. Since a pronoun referring to the class must be plural, singular indefinites are not accessible for a pronoun (as in (9a)), while bare plurals are (cf. (9b)). (9c) shows that specific singular indefinites are accessible for pronouns and the pronoun will get a specific interpretation as well.

(9)  a. A pirate is dangerous. *He is/ they are not altruistic.
    b. Pirates are dangerous. They are also dirty.
    c. A pirate came in. He was in a bad mood.

Krifka, Pelletier, Carlson, ter Meulen, Link, and Chierchia (1995) point out that singular indefinites are not genuinely kind referring. One test they present to detect kind reference is the following: some predicates, such as become extinct or be invented, only select for kind references. For this reason (10a) is grammatical and (10b) is not.

(10)  a. Dinosaurs became extinct
    b. *A dinosaur became extinct.

The inaccessibility of a pirate for pronominalisation in (9a) could be explained along the same lines. A pirate would then be inaccessible both by a singular pronoun because it is not existential and by a plural pronoun by virtue of not being kind-referring. Our explanation here is simpler: we assume that the plural pronoun is ungrammatical here because it has no linguistically realized plural antecedent. This implies that singular indefinite can indeed be kind-referring and that a kind-referring lecture of indefinites like the one in (10a) must be blocked by other factors. We will have to leave this question open here, but we are confident that it can be solved in further work.

The difference of the constraints on defocusing and pronominalisation follows directly from the present model. Defocusing is sensitive to all activated file cards, including the ones which are introduced or activated in a non-overt way, e.g. a class reference which is

41
activated by an activation of a specific instance of its class. Pronouns must refer to an
activated file card and in addition they must have an overt linguistic (or deictic) antecedent
which matches its grammatical features, such as number and gender. In addition a pronoun
can only refer to file cards which are topmost in the class-instance hierarchy. If only a class
reference is available, a pronoun may refer to it if there is no grammatical mismatch. If
both a class and a instance reference is available the instance reference will win out because
it is higher up in the hierarchy and in this sense ‘topmost’.

Note also that we have not represented genericity as an operator at LF. LF-raising is
not a possible option in a monostratic framework such Categorial Grammar (which is here
represented by Lambek Calculus). On the contrary we have conserved the effects on the
semantic representation in treating genericity as an operation which is carried out as a
manipulation of the file which represents the discourse semantics. In this respect we have
moved away from File Change Semantics as proposed in Heim (1977). This follows mainly
from the fact that we use a model of syntax and semantics which knows no LF because of
its pronounced monostratic nature. In fact, a framework such as Lambek Calculus has no
need for an LF which cannot be read from surface string because the parse is at the same
time a syntactic and a semantic object. The manipulation of file cards is done in parallel
as the incremental process proceeds.

4 Incremental processing and incremental semantic processing

Until this point our discussion has been rather informal. Now we would like to give a
sketch of how this idea can be properly formalised at the syntax-semantic level for which
a framework like Lambek Calculus (LC) may be wellsuited. Assuming a syntax structure
model, such as LC, allows to incorporate both competence and performance grammar in the
processing system, a fact that phrase structure grammars hardly exhibit (Steedman 2000).
Moreover, as it has been noted by Morrill (2000), LC allows an elegant analysis of incremen-
tal processing of sentences or fragments with the help of proofnets (Roorda 1991), which at
the processing level roughly consists of a system of unresolved dependencies which are to be
completed trough incremental parsing. Again, phrase structure grammar has troubles to
account for this aspect. Furthermore, proofnets can be seen as a syntax-semantic object, for
they represent the proofs whithout spurious ambiguity. They also represent the different
semantic readings via the Curry-Howard isomorphism. Finally, at each step of incremen-
tal processing, a semantic form can be synchronously computed (Johnson 1998), through
interpolation. In the rest of the paper we will use Pregroup grammars as a simplication of
the syntactic model which exhibit almost the same properties as LC grammars, including
a proofnet processing model. In addition to the syntactic/semantic computation discussed
by Morrill and Johnson, we assume that discourse information will be created and used for
disambiguation at each step of the computation, i.e. for each increment. This corresponds
roughly to what Steedman (2000) calls the Oracle, a process that at each step eliminates
unnecessary local ambiguity in order to avoid an explosion of global readings. The oracle operates on the basis of context information. We assume that the context might be modelled as a filing cabinet, as proposed by Heim (1982), with the addition of the Double Reference Hypothesis described above.

Let \( A_t \) be a finite set of atomic syntactic types, for instance np, n, s, pp, .... Consider the infinite set of syntactic types (LCtypes): \( T := A_t \setminus \{ T \mid T \in T \cdot T \} \)

In order to save space, we consider a simplified and very useful approximation to LC-grammars: the P(regroup)-grammars for which a very simple notion of proof net can be defined (Lambek 2001):

Definitions: Consider the infinite set of P-types: \( T := A_t \setminus \{ T^i \mid T \in T^i \cdot T^r \} \)

The set of inequations is the least set composed of expressions \( X_1 + \ldots + X_n \rightarrow X \), where \( X_i \) and \( X \) are P-types.

Rule governing (Free) Pregroup grammars are: \( \rightarrow \) is a partial order, \( X^i \cdot X \rightarrow 1 \) \( X \cdot X^r \rightarrow 1 \), if \( \rightarrow Y \) and \( V \rightarrow W \) then \( XV \rightarrow YW \), and finally \( 1 \rightarrow X^r \cdot X \) and \( 1 \rightarrow XX^i \), although, according to Lambek, the last two inequations are without linguistic interest. The superindices \( l \) and \( r \) are closely related to for the directionality expressed by the slash and backslash in LC types. It can be verified that in general, neither \( nll - n \) nor \( nrr - n \) are derivable. It is readily seen also that \( nlr - n \) and \( nlr - n \) are derivable inequations. Superscripts like \( ll \), \( lr \) etc. result from multiple application of the translation rules between LC types and P-types (see below). A P-grammar and its language generated is similarly defined as for LC-grammars. A P-proofnet for a P-inequation \( X_1 + \ldots + X_n \rightarrow X \) is a graph, such that every contraction (left or right) is signalled with an edge connecting them (See figures in section 5). Any edge connecting two atoms must span between pairs of atoms ‘a ar’ or ‘al a’ (in that order). These connecting edges are called axiom-linkings. Moreover two global constraints in the graph must hold: every atom must be connected to another one and only one atom, and no edge can cross another edge (planarity).\(^1\)

Theorem: A P-inequation is derivable iff there is a P-proofnet for it. The corresponding proof is easy to obtain and left to the interested reader.

Let us set up a correlation between Lambek Calculus and Pregroups: Consider the following translation between LC-types and P-types. We suppose the set of atomic P-types is equal to the one of LC-types:

\[ \tau : \text{Types}(\text{LC}) \rightarrow \text{Types}(\text{P}). \quad \tau(A) = A \text{ if A atomic type, } \tau(A \setminus B) = \tau(A)^r \tau(B), \tau(B / A) = \tau(B) \tau(A)^l, \tau(A.B) = \tau(A) \tau(B) \]

A derivable LC-sequent \( \Delta \vdash A \) translates to a derivable P-inequation \( \tau(\Delta) \rightarrow A \rightarrow \tau(A) \). The reader must be warned that given \( \Delta \rightarrow A \), if \( \tau(\Delta) \rightarrow A \) is P-derivable this doesn’t imply that \( \Delta \rightarrow A \) is LC-derivable! Counterexample: \((A \cdot B) / C \) is A \cdot (B / C) translate both to \( ABC^l \) and \((A \cdot B) / C \rightarrow A \cdot (B / C) \) is not LC-derivable. Given \( \tau(\Delta) \rightarrow \tau(A) \) P-derivable, we can rebuild the corresponding proofnet and check in linear time whether \( \Delta \vdash A \) is LC-derivable, as well as obtain the semantic reading (The algorithm, “semantic trip” (Retore and Groote 1996) ). In less technical terms, we have shown that we

\(^1\)Observe how the complexity correctness criterium is reduced compared to the correspondent ones in Lambek Calculus, for instance planarity acidity for every switching
can safely use P-Grammars instead of LC-Grammars without loosing much of their power. P-Grammar should be seen as an approximation to LC grammars, which are in practice much more convenient to handle.

5 Incremental processing and processing in context

In this section, we present a concrete model of incremental processing. We will adapt Morrill's (2000) indeterministic double shift reduce parser to P-grammars. This parser can be seen as an indeterministic double stack automaton with the following valid moves: R(EDUCE), P(POSTPOSE), M(OVE), C(ONSUM). The tape contains a sequence of types $S, A_1, ..., A_n$ corresponding to the target category and every Ai to a word (prosodic form) ai. The automaton works with a Global Stack (GS) and a Local Stack (LS). Informally the automaton works like this:

Initial state: push $t(S)$ to the LS (Local stack) EITHER, if there are two atoms reducible (that is, a and ar, or al and a), make a reduction (R) OR postpose, i.e, no reduce in a potential reduction (P), OR pop the LS and push the element popped to the global stack GS (M) OR spreads $\tau(A_i)$ taken at the automaton tape over the LS (C)

Acceptance state: GS and LS empty, and tape completely read.

Crash states: Other configurations where no valid move can be done more.

Let us now turn to the lexical entries for generic/specific readings: Let's assume we have at our disposal the feature variable F subsuming both features gen and spec. Then, $F \cap gen = gen$ and $F \cap spec = spec$ Unification in the P-proofnet takes place at the axioms-links. The reader should not forget that semantic readings are obtained from the corresponding LC-proofnet. In order to account for scope alternations (Which is known to hold for specific indefinites as well as non-specific), following Moortgat (1997), we assign indefinites either specific or non-specific, a quantifier-like syntactic type, e.g $s/(np s)/n$ for a subject indefinite. Feature percolation occurs at the level of lexical assignment (See toy-lexicon below), and feature instantiation at the level of axiom-linkings.

Let's see how this works:

Let us first define a toy lexicon (with LC types):

- $a - (s_F \setminus s_{F})/n_F pirate - n_F$
- $is - (np \setminus s_F)/(n_F/n_F) angry - n_F/n_F vicious - n_{gen}/n_{gen}$ today - $s_{spec} \setminus s_{spec}$

Let us now see how a copula sentence with the adjective angry is resolved in a P-proofnet. Angry is the head of a S-level predicate and leaves its subject ambiguous between a generic and a specific reading. In order to reduce space we will annotate the feature gen-spec, gen and spec as the subindices g-s, g and s respectively:

\begin{equation}
\text{(11) A pirate is angry.}
\end{equation}

---

2 gen could be $[spec-]$ and and spec $[spec+]$

3 To be translated via $\tau$ into P-proofnets. See P-proofnets
Figure 4.1: Ambiguous between a generic and a specific reading

Here we can get two possible readings, one generic and the other specific. The intuitions behind this is that the underspecified feature gen-spec can survive within the whole proofnet because none of the features it subsumes is blocked, e.g. by a I-level predicate which would block the spec reading.

It is important to note that in parallel to the synchronous syntactic/semantic process the according File cards in the pragmatic discourse model are manipulated. Remember that the instance file card is introduced as a temporal data-structure which might be erased when the processor encounters an I-level predicate. The shift from gen-spec in the proofnet to gen will cause such a destructive action in the discourse model.

Now let us consider the same P-proofnet with the increment of a temporal sentential adverb today. This adverb will introduce an spatiotemporal anchor which will block the generic reading. On the level of the discourse model this means that the temporal instance file card will be converted into a permanent one which cannot be erased later on. The axiom linking is only possible if the features unify. If one of the features is underspecified this underspecification will be resolved in favour of the unifying feature.

(12) A pirate is angry today.

Figure 4.2: Today forces the specific reading

Note that a+pirate+is+vicious, cannot be modified with the previous adverb today, for there is a conflictive unification (An invalid axiom-link, or invalid reduction). The spatiotemporal anchor introduced by ‘today’ is not compatible with a generic reading of the subject either: (14) *a+pirate+is+vicious+today

6 Conclusion

This paper leaves several open questions. We want to stress two of them to conclude the discussion.
In this paper we have tried to recast Carlson’s distinction between *kinds* and *objects* in a file change model. An interesting question which this proposal raises is if Carlson’s entire nominal hierarchy could be expressed as different types of file cards. This would only mean the introduction of a file card type for *stages*. Such a move seems plausible, but at the moment we are not sure if it involves further complications.

So far we have not said anything about object nouns. The F-feature (which subsumes *gen* and *spec*) introduced here affects the readings of subject nouns. In part this is a consequence of the sketchy nature of the feature representation of each lexical element. If we compare the present proposal to full-size feature structure based approaches such as HPSG (Pollard and Sag 1994), we can think of complex feature structures as a possible solution for the increasing complexity of the analysis. In a theory with wider scope we could use complex restrictions on all predicate arguments imposed by the predicate head. The integration of complex feature structures is, however, not a trivial task and must be left for further work.

**Acknowledgements**

We would like to thank Enric Vallduví, Louise McNally, Glyn Morrill, Gemma Boleda, Martí Quixal as well as three anonymous reviewers for comments and discussion about the form and content of this paper.


References
Why $\lambda$HST* and HST$_\lambda^*$ do not solve the Russell-Myhill paradox after all

SERGE BOZON
IH(resultSet - University Paris 1
bozonserge@hotmail.com

ABSTRACT.
Cocchiarella (2000) claims that Russell’s paradox of propositions (Principles, § 500) is resolved in his systems $\lambda$HST* and HST$_\lambda^*$. We prove that these systems can not solve a property-theoretical variant of this paradox, due to Philippe de Rouilhan.

We briefly recall the paradox involved and its underestimated historical importance, then prove that $\lambda$HST* and HST$_\lambda^*$ can not solve it after all, and conclude by showing that any hyperintensional logic has to accept the general hyperintensional laws needed in the derivations of the paradox. What does ‘hyperintensional’ mean? The intensional versus hyperintensional distinction is classic since Cresswell (1975). To put it briefly and roughly, we will call an intensional logic ‘hyperintensional’ if the granularity (of the identity criteria) of its intensional entities is finer than necessary equivalence. More specifically, we will be interested in logical frameworks, like the Logic of Sense and Denotation (Alternative (0)) of Church-Anderson, Bealer’s (1982) $T_2$, the substitutional type-free theories of Russell (as reconstructed by Landini 1998), the simple theory of types of Russell (as reconstructed by Church 1984), Orilia’s (2000) $P^*$, Menzel’s (1986) LPRP, etc., which are faithful to the ‘principle of maximum distinction’ (Myhill 1958, p. 81), that is frameworks in which these identity criteria are (to speak vaguely) as fine-grained as possible. Remember that Myhill (1958) rediscovered Russell’s propositional paradox in a Fregean disguise, whence the name of this paper, and that such stringent criteria of identity may be suitable for dealing with the propositional attitudes, the paradox of analysis and other related issues that are of crucial importance for any intensional logic.1

1Bealer (1982) shows that such very fine-grained criteria are also implied by the idea that every complex intensional entity has a unique complete logical decomposition. Lastly, in reply to a remark of an anonymous referee, we will merely point out that the principle of maximum distinction does not restrict the possible interpretations (of the systems involved) to syntactical ones, as one can see by reading the quoted papers. On the contrary, the authors involved have devised powerful arguments against nominalistic proposals to do away with intensional entities altogether in favour of syntactical entities (cf., for instance, Church 1950),

Proceedings of the Ninth ESSLLI Student Session
Laura Alonso i Alemany and Paul Égré (editors)
Chapter 5, Copyright © 2004, Serge Bozon
1 Introduction

1.1 What is the Russell-Myhill paradox?

Russell writes in Appendix B of his Principles (p. 527):

If $m$ be a class of propositions, the proposition “every $m$ is true” may or may not be itself an $m$. But there is a one-one relation of this proposition to $m$: if $n$ be different from $m$, “every $n$ is true” is not the same proposition as “every $m$ is true”. Consider now the whole class of propositions of the form “every $m$ is true”, and having the property of not being members of their respective $m$’s.

Let this class be $w$, and let $p$ be the proposition “every $w$ is true”.

The contradiction appears if one asks whether $p$ is a $w$. One should note that the hyperintensional principle (that we will call ‘(*)’) used by Russell in the second sentence amounts to assume an injection from the power-set$^2$ of the set of propositions to the set of propositions, which Cantor’s power-set theorem forbids. We write “hyperintensional” because one could show that (*) is not valid if the granularity of the propositions is not finer than necessary equivalence.$^3$ Let us use (like Russell) ‘$m$’ as a variable of class of propositions and ‘$F$’ as a variable of property of properties. By replacing classes of propositions by properties of properties, the proposition that every $m$ is true, which is somehow the “conjunction” of $m$, becomes then the property which is somehow the “conjunction” of $F$, namely the property of having all the properties which are $F$; the class $w$ becomes the property $W$ of being such an $F$-conjunction which is not $F$; and finally $p$ becomes the $W$-conjunction property $\land W$, the contradiction appearing again by asking whether $\land W$ is $W$, using this time the following hyperintensional principle, which amounts to assume an injection from the set of properties of properties to the set of properties: (*) if the property of having all the properties $F$ is (identical to) the property of having all the properties $G$, then $F$ is (identical to) $G$. Keeping Cantor’s theorem in mind, one must recall that, in a hyperintensional framework, the cardinality of the set of properties of properties is strictly bigger than the cardinality of the power-set of the set of properties, since the properties are individuated more finely than by coextensionality.$^4$

This property-theoretic variant of Russell’s hyperintensional paradox is due to Routley (1996, p. 230).

1.2 The historical importance of the Russell-Myhill paradox

What is the usual way to tell the story of the development of Russell’s theories of types? Something like this: after a tumultuous decade, Russell has devised a type theory which

$^2$The opposition set versus (proper) class is irrelevant here.

$^3$As a rough sketch of this missing argument, take for $m$ the class of propositions expressed by tautological sentences and for $n$ the empty class of propositions. Then the associated propositions (that every $m$ is true, that every $n$ is true) are logico-mathematical truths, thus necessarily equivalent, but $m$ is distinct from $n$. One must keep in mind that this notion of truth is not semantical, truth being here a property of propositions and not of sentences.

$^4$For a discussion of Cantor’s theorem in the light of Cocchiarella’s systems, see Cocchiarella (1992).
is his "great success", much relative success after all since a lack of conceptual clearness prevented him to see the fundamental distinction distinguishing the logical paradoxes from the so-called "semantical paradoxes", constraining him twice to give up coherent theories (the simple substitutional theory and the simple theory of types) to resolve semantical paradoxes by a procedure of ramification which seems now ad hoc in the light of the (so clear) Tarskian distinction of level of languages. But authors like Church (1984), Rouilhan (1996), Landini (1998) have shown that this story is at least misleading: Russell's simple theory of types and his simple substitutional theories of 1904-1907 are incoherent. The paradox involved is none other than Russell's propositional paradox, thus this incoherence could be ignored only because one had unfaithfully forgotten the hyperintensionality of Russell's ontology.5 This paradox being not semantical (cf. Rouilhan 1996, p. 290-291), the traditional classification of the paradoxes is ineffective, ramification is not ad hoc and its legitimation does not presuppose in any way giving up realism and identifying abstracts objects with mental constructs. To conclude this (much too) elliptical summary, the usual account of this tangled and fundamental period of modern logic is incorrect.

1.3 What are $\lambda$HST* and HST*$_\lambda$?

The systems $\lambda$HST* and HST*$_\lambda$ of Cocchiarella are two non-standard (or type-free) second-order logics which have been applied to a wide range of philosophical issues, including metaphysics (theory of universals), modality, logicism, semantics of natural language, fiction, intensional contexts, etc. Now one knows that type-freedom opens the door to Russell's paradox of predication. To achieve consistency, $\lambda$HST* imposes a grammatical restriction on the formation on $\lambda$-abstracts, only the homogeneously stratified ones being taken to be well-formed. HST*$_\lambda$ has no morphological restriction but allows for $\lambda$-conversion only if, roughly speaking, all the terms occurring as arguments of a given $\lambda$-abstract are denoting. Thus HST*$_\lambda$ relies on a free logic and some singular $\lambda$-terms are provably nondenoting in HST*$_\lambda$. But HST*$_\lambda$ grants that at least those $\lambda$-abstracts that contain no nondenoting terms, are homogeneously stratified and are bound to individuals are denoting. These systems are consistent relative to weak Zermelo set theory (cf. Cocchiarella 1986, p. 231).

In terms of identity conditions, $\lambda$HST* and HST*$_\lambda$ are neutral (neither extensional nor intensional), even if Cocchiarella claims that the denotata, if any, of nominalized predicates are "intensional objects". Now one can "use" these systems without "going outside" their purely logical framework, but one can also apply them to extra-logical notions, using $\lambda$HST* and HST*$_\lambda$ as, for example, the underlying logics of some natural language semantics, like in Cherchia (1984), or of some "philosophical logic", like in Orilia (1994). In the first case, Cocchiarella has shown that platonic realism, for example, would already induce in the pure framework of predication theory the choice of intensionality (cf. Cocchiarella

5Church (1984) shows that this "extensionalization" loses some of the characteristically Russellian contributions to logic: for instance, 'the motivation for Russell’s contextual definitions of class abstracts and of descriptions largely disappears' (p. 21).

6To keep this paper self-contained, all the technical informations (concerning Cocchiarella’s systems) needed here are given in the Appendix.
1986, p. 232-234). In the second case, it is clear that the applications involved impose also the choice of intensionality. So it seems that intensionality is vital to the use of these systems, and the well-known difficulties concerning propositional attitudes contexts would even classically militate (in the second case) in favour of hyperintensionality (cf. Anderson 1984, Bealer 1982, etc.). The crucial question is thus: “how far can we go in this (intensional) way with λHST* and HSTλ∗”? Cocchiarella (2000) could let one think that the answer is “as far as one wish”, since the propositional Russelian paradox is resolved in λHST*, the (complex) predicate expressing the property of propositions (corresponding to) w being not homogeneously stratified, and also in HSTλ∗, (the nominalization of) the sentence expressing the proposition p being provably nondenoting. But our paper will show that the correct answer is in fact: “not far enough”.

Orilia (1996) has proved that λHST* and HSTλ∗ can not solve an extra-logical paradox. This paradox is extra-logical because it involves adding (to the purely logical vocabulary of λHST* and HSTλ∗) extra-logical notions, namely psychological notions, and extra-logical assumptions governing these psychological notions.⁷ Thus Orilia’s paradox only shows an eventual problem of application (of λHST* and HSTλ∗). Eventual, because one can always choose not to accept these psychological assumptions. That is why Orilia calls his paradox ‘contingent’. On the other hand, the two property-theoretic variants of Russell’s hyperintensional paradox presented here are pure. They do not involve any extra-logical notion or assumption, but are on the contrary expressible in the purely logical language of these systems, thus they are not contingent but unavoidable (if one does not wish, of course, to give up hyperintensionality).

To sum up, λHST* and HSTλ∗, which are always presented by Cocchiarella as theories of predication and concepts (as opposed to theories of membership and sets), can simply not be (purely logically extended to get) hyperintensional logics, and thus are (perhaps) not adequate for one of their main intended application.⁸

2 Proofs

Contrary to Cocchiarella (2000), who uses directly (*), we won’t do the same for (*′), since we believe that a derivation of the paradox from more general and natural principles is more illuminating.⁹ So the hyperintensional logical laws needed here are the universal

---

⁷One must notice that these extra-logical assumptions are not general normative principles (as the classical laws of knowledge involved in the famous knower’s paradox), but only particular descriptions of contingent psychological facts (like ‘there is only one man who is... and who thinks of — at time t’). Therefore, this paradox is contingent in a strong sense (namely stronger than the sense induced by any extra-logical paradox): the knower’s paradox is extra-logical (since ‘knowing’ is not a logical notion) but it is not contingent in this sense.

⁸A straightforward formalization of ΛW (cf. section 1.1) in λHST* is given in the proof of Theorem 2.2. In HSTλ∗, a much more convoluted definition is needed because of the complications induced by (∃/HSCEP∗).

⁹Even if we do not have the space to derive here (*′) from our three principles, a simple inspection of the proof of Theorem 2.1 is enough to understand how it could easily be done (in HSTλ∗ and λHST*). Of
closure of (the instances of) the following two schemata:

\[(Hyp. Q) \quad [\lambda z^m \forall y \varphi(z^n, z, y)] = [\lambda z^m \forall \psi(z^n, z, y)] \rightarrow \forall y ([\lambda z \varphi(z^n, z, y)] = [\lambda z \psi(z^n, z, y)])\]

where \(Q \in \{\forall, \exists\}, n, m \geq 0, y\) is an individual or predicate variable

\[(Hyp. \bullet) \quad [\lambda z^m (\varphi(z, y) \bullet \psi(z, y))] = [\lambda z (\varphi(z, y) \bullet \psi(z, y))] \rightarrow

([\lambda z \varphi(z, y)] = [\lambda z \psi(z, y)]) \land ([\lambda z \psi(z, y)] = [\lambda z \psi(z, y)])\]

where \(\bullet \in \{\rightarrow, \leftrightarrow, \land, \lor\}, m, r \geq 0, y_i, 1 \leq i \leq r, \) is an individual or predicate variable

and the axiom:

\[(Hyp. At.) \quad \forall F \forall X (\forall y^n ([\lambda z^m F] = [\lambda z^m X]) \rightarrow [\lambda x^n F] = [\lambda x^n X])\]

where \(m \geq 0, n \geq 1, y_i, 1 \leq i \leq n, \) is an individual or predicate variable, \(F\) and \(X\) are predicate variables.

**THEOREM 2.1** \(HST^*_\lambda + (Hyp. Q), (Hyp. \bullet), (Hyp. At.) \vdash \perp\)

**Definition** \(W =_d [\lambda y \forall X (\exists w (w = X) \rightarrow

((y = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz))] \rightarrow Xy)))]\)

**Definition** \(\wedge W =_d [\lambda t \forall F (\exists x (x = F) \rightarrow (WF \rightarrow Ft))]\)

In primitive notation, \(\wedge W =_d [\lambda t \forall F (\exists x (x = F) \rightarrow ((y = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz))] \rightarrow Xy)))]\)

\(\wedge W\) is homogeneously stratified by any function \(f\) such that \(f(z) = f(t) = 0, f(y) = f(x) = f(Y) = f(F) = 1, f(X) = f(w) = 2,\) thus \(W\) is also homogeneously stratified. \(\wedge W\) is bound to individuals, thus \(W\) is also bound to individuals.

**Proof.** By \((\exists/HSCP^*_\lambda),\) since \(W\) is purely logical, closed, homogeneously stratified and bound to individuals,

\[(1) \quad \exists s (s = W)\]

thus, since \(\wedge W\) is purely logical, closed, homogeneously stratified and bound to individuals, by \((\exists/HSCP^*_\lambda),\)

\[\forall G \forall X (\forall t F (\exists x (x = F) \rightarrow (GF \rightarrow Ft)) = [\lambda t \forall F (\exists x (x = F) \rightarrow (XF \rightarrow Ft))] \rightarrow G = X)\]

An expression of the form ‘\(\varphi^{n}\)’ is short for ‘\(s_1, \ldots, s_m\)’. The letters ‘\(s\), ‘\(t\),‘\(z\),‘\(y\),‘\(w\),‘\(z\),’ will be used to refer to individual variables, the superscript of a predicate variable will indicate its arity, and the superscript of a variable will be written only for the first occurrence of the variable. Otherwise, we follow standard metalinguistic conventions (so ‘\(\varphi^{n}\)’ means that the free variables of \(\varphi\) form a subset of \(\{x_1, \ldots, x_m\}\)). As usual, we will use in the following semi-formal proof the simplified version of the existential instantiation rule and we will drop some parentheses and superscripts in obvious ways. To keep the presentation as general as possible, we won’t take into account the fact that several logical constants (\(\exists, \land, \lor, \rightarrow\)) are not primitive but (classically) defined in \(HST^*_\lambda\) and \(HST^*_\lambda\). Finally, we won’t give all the hyperintensional laws implied by the principle of maximum distinction, but only the laws needed in the following antinomy. For example, it is obvious that the schema \((Hyp. \bullet)\) should be completed by:

\[(Hyp. \neg) \quad [\lambda z \neg \varphi(z, y)] = [\lambda z \neg \psi(z, y)] \rightarrow [\lambda z \varphi(z, y)] = [\lambda z \psi(z, y)]\]
(2) \( \exists s (s = \land W) \).

By def. of \( W \), (\( \exists / \lambda - \text{conv} \)) and standard laws,

(3) \( \exists s (s = \land W) \rightarrow (W (\land W) \leftrightarrow \forall X (\exists w (w = X) \rightarrow ((\land W = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz)]) \rightarrow X(\land W))))) \)

so, by (2), (3) and modus ponens,

(4) \( W(\land W) \leftrightarrow \forall X (\exists w (w = X) \rightarrow ((\land W = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz)]) \rightarrow X(\land W))))) \)

Assume

(5) \( W(\land W) \)

then, by (4) and equivalence-elimination,

(6) \( \neg \forall X (\exists w (w = X) \rightarrow ((\land W = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz)]) \rightarrow X(\land W))) \)

so, by elementary transformations,

(7) \( \exists X (\exists w (w = X) \land (\land W = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz)]) \land \neg X(\land W)) \)

thus, by existential instantiation and conjunction elimination,

(8) \( (\land W = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz)]) \land \neg X(\land W) \)

thus, by conjunction elimination and def. of \( \land W \),

(9) \( [\lambda t \forall F (\exists x (x = F) \rightarrow (WF \rightarrow Ft))] = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz)]) \)

whence, by (A6)\textsuperscript{\textbullet} (twice) and (Hyp. Q),

(10) \( \forall F ([\lambda t (\exists x (x = F) \rightarrow (WF \rightarrow Ft))] = [\lambda t (\exists x (x = F) \rightarrow (XF \rightarrow Ft)]) \)

so, by (Hyp. \textbullet) and standard laws,

(11) \( \forall F ([\lambda t (WF] = [\lambda t (XF)] \)

whence, by (Hyp. At.),

(12) \( [\lambda x Wx] = [\lambda x Xx] \)

thus, by (ID\textsubscript{x*}),

(13) \( W = X \)

whence, by (8), conjunction elimination and (Id* / Pred*),

(14) \( \neg W(\land W) \).

Assume

(15) \( \neg W(\land W) \)

then, by (4) and elementary transformations,

(16) \( \forall X (\exists w (w = X) \rightarrow ((\land W = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (XY \rightarrow Yz)]) \rightarrow X(\land W))) \)

so, by second-order universal instantiation (cf. Appendix, remark b)), instantiating \( X \) to \( W \),

(17) \( \exists w (w = W) \rightarrow ((\land W = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (WY \rightarrow Yz)]) \rightarrow W(\land W)) \)

so, by (1), the law of rewrite of quantified variables and modus ponens,

(18) \( (\land W = [\lambda z \forall Y (\exists x (x = Y) \rightarrow (WY \rightarrow Yz)]) \rightarrow W(\land W) \)

whence, by def. of \( \land W \), (A6)\textsuperscript{\textbullet} (twice) and modus ponens,

(19) \( W(\land W), \)

thus, by (double) implication introduction (on (5)-(15) and (16)-(20))

(20) \( W(\land W) \leftrightarrow \neg W(\land W). \)
**Theorem 2.2** \( \lambda \text{HST}^* + (Hyp. Q), (Hyp. \bullet), (Hyp. At.) \vdash \bot \)

**Definition** \( W =_d \{ y \exists X ((y = [\lambda z \forall Y (XY \to Yz)] \land \neg Xy) \} \)

**Definition** \( \forall W =_d \{ \lambda w \forall F (WF \to Fw) \} \).

\( \forall W \) and \( W \) are homogeneously stratified by any function \( f \) such that \( f(z) = f(w) = 0, \)

\( f(y) = f(Y) = f(F) = 1, f(X) = 2. \) The proof parallels the preceding one (using this time full \( (\lambda - \text{Conv}^*)), but is however simpler and is thus left to the reader.\(^{11}\)

### 3 Justification of the three hyperintensional laws

Let us begin by the first two laws. They correspond directly to the idea that identical complex (hyper)intensions must have the same constituents. To use Bealer’s (1982) framework, where this metaphorical notion of ‘constituent’ can be formally (and algebraically) defined,\(^{12}\) if these intensions are analysable in terms of operations of composition (ontological analogues of the syntactical operations on the expressions which express these intensions), then these operations must (among other things) be injective. To take just one simple example: if the “negation” of (the proposition) \( p \) is (identical to) the “negation” of (the proposition) \( q \), then \( p \) is (identical to) \( q \), and this injectivity generalizes to any such \( n \)-ary (ontological) operation and any intensional entity (properties, relations, etc.).\(^{13}\)

One can find these two laws (or notational variants of them) in all the hyperintensional logics faithful to the principle of maximum distinction: they are easy theorems of Bealer \( T2 \), by his principles \( A10 \) and \( A11 \); in the Logic of Sense and Denotation (Alternative 0), they are analogous to the axioms \( A(0) \theta^a \theta \) and \( A(0) \eta^a \beta \);\(^{14}\) in the type theory of Russell (as reconstructed by Church 1984), our second law is a generalization of his axioms 9 and 10, the first is a property-theoretical variant of the converse of his axiom 11\(^{15} \), etc.

---

\(^{11}\)In the (forthcoming) paper “The Basic Problem of the Logic of Meaning”, Rouslan uses his property-theoretical variant of Russell’s hyperintensional paradox to prove the inconsistency of a simple (i.e. heterogeneously) stratified (first-order) property theory of his own (§ 8.2), where the notion of stratification is inspired by NF. Considering the fact that \( \lambda \text{HST}^* \) is equiconsistent with NFU and relies only (contrary to \( \text{HST}^*_1 \)) on (homogeneous) stratification to avoid paradoxes, this second theorem is therefore not really surprising.

\(^{12}\)For lack of space, this definition will be omitted here.

\(^{13}\)To put it in misleading (cf. Appendix, remark c)) semantic terms, if \( \neg \varphi \) is synonymous with \( \neg \psi \), then \( \varphi \) is synonymous with \( \psi \). In \( (Hyp. Q) \), the operations involved are the “(type-free) existential or universal quantification” of an \( m \)-ary relation. In \( (Hyp. \bullet) \), the operations involved are the (type-free) ontological analogues of the usual binary connectives.

\(^{14}\)Cf. Anderson 1984, p. 378. This analogy would need to be explained, because the well-known peculiarities of a Fregean framework (like the purely conceptual nature of senses) complicate any direct comparison. Of course, we use here ‘conceptual’ in Church’s sense.

\(^{15}\)Church (1984) didn’t intend to give all the hyperintensional laws implied by the principle of maximum distinction in a typed Russellian framework, but only those needed in the formal reconstruction of Russell’s propositional paradox.
To simplify the interpretation of the third law, let us restrict ourselves (without loss of
generality) to the monadic case (i.e. when \( n = m = 1 \)). Then this axiom says (leaving
implicit the initial second-order universal quantifiers): if, for every \( x \), the property of being
an \( y \) such that \( x \) is \( F \) is identical to the property of being an \( y \) such that \( x \) is \( G \), then
the properties \( F \) and \( G \) are identical. One can justify this axiom in two steps. Firstly,
the principle of maximum distinction obviously implies that, for any \( x \): if the proposition
that \( x \) is \( F \) is distinct from the proposition that \( x \) is \( G \), then the associated “constant”
propositional properties (namely being an \( y \) such that \( x \) is \( F \) and being an \( y \) such that \( x \)
is \( G \)), which contain these propositions in Bealer’s precise sense, are distinct. Whence, by
contraposition, for every \( x \): if the property of being an \( y \) such that \( x \) is \( F \) is identical to
the property of being an \( y \) such that \( x \) is \( G \), then the associated propositions are identical.
Thus, by elementary predicate logic, if, for every \( x \), the property of being an \( y \) such that
\( x \) is \( F \) is identical to the property of being an \( y \) such that \( x \) is \( G \), then, for every \( x \), the
proposition that \( x \) is \( F \) is identical to the proposition that \( x \) is \( G \). Secondly, if, for every \( x \),
the proposition that \( x \) is \( F \) is identical to the proposition that \( x \) is \( G \), then the properties \( F \)
and \( G \) are identical. This is trivial if the properties are construed as propositional functions
(and these functions in turn defined in the usual set-theoretical way), as in Church or
Russell.\(^{16}\) In the opposite case, as in Bealer, it is anyway (by contraposition) an immediate
consequence of the principle of maximum distinction, with or without nominalization (the
nature of \( x \), concrete or abstract object, possible reference of an ordinary name or of an
intensional abstract, does not matter).\(^{17}\) Whence, by conjoining the two steps, this last
law is justified.

4 Conclusion

To sum up, this paper should simply be taken as one more example of a general lesson,
often stressed by Philippe de Rouilhan (and ironically by Cocchiarella 1985 himself)\(^ {18}\): one
should not think that the classical methods used to resolve the non-intensional paradoxes
will permit to resolve the (hyper)intensional ones. To quote a mysterious sentence of
Gödel (as reported by Myhill 1984): ‘There never were set-theoretic paradoxes, but the
property-theoretic paradoxes are still unresolved’.

\(^{16}\) In a Fregean framework, a propositional function sends individual concepts, and not individuals, into
the associated propositions (which are not singular propositions). Moreover, in a Russellian framework,
the extensionality principle for functions (implied by the set-theoretical definition involved) raises some
problems for the intended application (in finite domains) of the associated hyperintensional logic (cf.

\(^{17}\) Formally, it is easy to see, using his axioms A10 and A11, that (the first-order schema implied by) this
last law is a (schematic) theorem of Bealer’s T2.

\(^{18}\) I am indebted to an anonymous referee for this reference to Cocchiarella’s review of Bealer (1982).
References

Appendix

We recall here the technical material (concerning Cocchiarella’s systems) that we use in the proofs. We shall rely primarily on Cocchiarella (1986) and thus any reference below will be with respect to this work, unless otherwise indicated.

Morphology of $HST^*$ and $\lambda HST^*$

Definition The set of well-formed expressions of $HST^*$ and $\lambda HST^*$

The inductive definition of $T_n$, the set of well-formed expressions of type $n$, is the following, where $T_0$ (resp. $T_1$, $T_{n+1}$, $n > 0$) is the type of singular terms (resp. of formulas, of $n$-ary predicate expressions):

1. every individual variable (or constant) belongs to $T_0$, every $n$-ary predicate variable (or constant) belongs to $T_{n+1}$ and $T_0$
2. if $t, s \in T_0$, then $(t = s) \in T_1$
3. if $\pi \in T_{n+1}$ and $a_1, ..., a_n \in T_0$, then $\pi(a_1, ..., a_n) \in T_1$
4. if $\varphi \in T_1$ and $x_1, ..., x_n$ are pairwise distinct individual variables, then $[\lambda x_1...x_n \varphi] \in T_{n+1}$
5. if $\varphi \in T_1$, then $\neg \varphi \in T_1$
6. if $\varphi, \psi \in T_1$, then $(\varphi \rightarrow \psi) \in T_1$
7. if $\varphi \in T_1$ and $y$ is an individual or predicate variable, then $(\forall y)\varphi \in T_1$
8. if $\varphi \in T_1$, then $[\lambda \varphi] \in T_0$
9. if $n > 1$, then $T_n \subseteq T_0$.

By (1), (8), (9), one sees in what sense this syntax allows nominalization (and is thus non-standard): $[\lambda \varphi]$ is the nominalization of $\varphi$, the second occurrence of $F$ in ‘$F(F)$’ is nominalized, etc. In the framework of a hyperintensional logic, it is natural to identify $[\lambda \varphi]$ (resp. $[\lambda x \varphi(x)]$) with the proposition that $\varphi$ (resp. the property of being $\varphi(x)$), etc. Strictly speaking, only the nominalized occurrences of ‘$[\lambda x \varphi(x)]$’ can be so construed. For the other occurrences, it depends on the intended intepretation of juxtaposition (of a predicate and a singular term).

Let us also remind that, in $\lambda HST^*$, only the homogeneously stratified $\lambda$-abstracts are taken to be well-formed.

Definition Homogeneous stratification

A formula or an abstract $\varphi$ is homogeneously stratified iff there is an assignment $f$ of natural numbers to the terms and predicates occurring in $\varphi$ (including $\varphi$ itself if $\varphi$ is an abstract) such that:
(1) if $t_1 = t_2$ occurs in $\varphi$, then $f(t_1) = f(t_2)$
(2) for all $n \geq 1$, if $r = [\lambda x_1...x_n \, \psi]$ occurs in $\varphi$, then $f(x_i) = f(x_j), 1 \leq i, j \leq n$, and $f(r) = f(x_1) + 1$
(3) for all formulas $\psi$, if $r = [\lambda \psi]$ occurs in $\varphi$ and $l_1,...,l_n$ are all the terms or predicates occurring in $\psi$, then $f(r) \geq \max(f(l_1),...,f(l_n))$
(4) for all $n \geq 1$ and all $n$-ary predicates $\pi$, if $\pi(t_1,...,t_n)$ occurs in $\varphi$, then $f(t_i) = f(t_j), 1 \leq i, j \leq n$, and $f(\pi) = f(t_1) + 1$.

That definition is taken from Cocchiarella (2000), §1, because the clause (3) was overlooked in earlier formulations.\(^\text{19}\)

Definition Bound to individuals

A well-formed expression $\xi$ is bound to individuals iff, for all $n \geq 0$, all $n$-ary predicate variables $F$ and all formulas $\varphi$, if $(\forall F) \varphi$ occurs in $\xi$, then, for some individual variable $x$ and some formula $\psi$, $\varphi$ is $(\exists x \, (x = F) \rightarrow \psi)$.

Principles of HST\(_\lambda^*\) and $\lambda$HST\(_\lambda^*\) used in section 2

- HST\(_\lambda^*\)

### (3)/conv*

$[\lambda x_1...x_n \, \psi](a_1,...,a_n) \leftrightarrow \exists x_1...\exists x_n (a_1 = x_1 \wedge ... \wedge a_n = x_n \wedge \psi)$, where no $x_i$ is free in any $a_j, 1 \leq i, j \leq n$ (p. 221).

(CP\(_\lambda\))

$\exists F^n ([\lambda x_1...x_n \, \psi] = F^n)$, where $F^n$ does not occur free in $\psi$ (p. 224).

### (3)/HSCP\(_\lambda\)

$(\exists y (a_1 = y) \wedge ... \wedge \exists y (a_n = y) \rightarrow \exists y (y = [\lambda x_1...x_n \, \psi]))$, where $[\lambda x_1...x_n \, \psi]$ is homogeneously stratified, $\psi$ is bound to individuals, $y$ is an individual variable not occurring in $\psi$ and $a_1,...,a_n$ are all the variables (or nonlogical constants) occurring free in $[\lambda x_1...x_n \, \psi]$ (p. 229).

\(^{19}\text{By analogy with (3), one could block the paradoxes of section 2 by replacing the clauses (2) and (3) by a single (2-3'): for all } n \geq 0, \text{ if } r = [\lambda x_1...x_n \, \psi] \text{ occurs in } \varphi \text{ and } l_1,...,l_m \text{ are all the terms or predicates occurring in } \psi, \text{ then } f(x_i) = f(x_j), 1 \leq i, j \leq n, \text{ and } f(r) \geq \max(f(l_1),...,f(l_m)); \text{ but such a "ramified" stratification would deprive HST\(_\lambda^*\) and } \lambda\text{HST}\(_\lambda^*\) \text{ of the expressive power that gives precisely to type-free theories their interest, and the aforementioned applications of HST\(_\lambda^*\) and } \lambda\text{HST}\(_\lambda^*\) \text{ would thus be undermined: for example the (innocuous) proposition that Bob is intrigued by the property of being taller than Bob would be no more available in } \lambda\text{HST}\(_\lambda^*\) \text{ and HST}\(_\lambda^*\) \text{ (the associated propositional abstract being, by (2-3') and (4), not even well-formed in } \lambda\text{HST}^*\text{). One must notice that the (innocuous) proposition expressed by 'Bob thinks that he is pretty' is already ruled out in } \lambda\text{HST}^* \text{ and HST}\(_\lambda^*\) \text{ by (3) and (4), so Cocchiarella's (2000) new definition is already much too drastic! Lastly, relaxing (4) to avoid these shortcomings would block the relative consistency proofs for } \lambda\text{HST}^* \text{ and HST}\(_\lambda^*\) \text{ (p. 193-205, 230-231). Anyway, this possible replacement of (2)-(3) by (2-3') shows once more (cf. section 1.2) how hard it is to escape (some kind of) ramification in hyperintensional logic, even if one must notice that the situation here is already much worse than in Russell's ramified theory of types, where the preceding propositions do exist.}
• \(\lambda HST^*\)

\((\lambda - Conv^*)\) \([\lambda x_1...x_n \psi](a_1, ..., a_n) \leftrightarrow \psi(a_1/x_1, ..., a_n/x_n)\),
where each \(x_i\) is free for \(a_i\) in \(\psi\), 1 \(\leq\) \(i\) \(\leq\) \(n\) (p. 222).

\((HSCP_{\lambda}^*)\) \(\exists F^n ([\lambda x_1...x_n \psi] = F^n)\),
where \([\lambda x_1...x_n \psi]\) is homogeneously stratified and \(F^n\) does not occur free
in \(\psi\) (p. 224).

• **Principles common to both systems**

\((A6)\) \([\lambda x_1...x_n \psi] = [\lambda y_1...y_n \psi(y_1/x_1, ..., y_n/x_n)]\),
where no \(y_i\), 1 \(\leq\) \(i\) \(\leq\) \(n\), occurs
in \(\psi\) (p. 221).

\((ID_{\lambda}^*)\) \([\lambda x_1...x_n P(x_1, ..., x_n)] = P\),
where \(P\) is an \(n\)-ary predicate variable or
constant (p. 221).

\((Id */Pred^*)\) \(\pi = \sigma \rightarrow (\pi(a_1, ..., a_n) \leftrightarrow \sigma(a_1, ..., a_n))\),
where \(\pi\) and \(\sigma\) are \(n\)-ary predicate
expressions (p. 221).

**Remarks**

a) Some of the principles listed above are in fact theorems of the associated systems (for
example \((Id */Pred^*)\) is a theorem of both systems).

b) In \(HST_{\lambda}^*\) (resp. \(\lambda HST^*\), \((CP_{\lambda}^*)\) (resp. \((HSCP_{\lambda}^*)\)) guarantees that all closed \(\lambda\)-abstracts
(resp. all closed and homogeneously stratified \(\lambda\)-abstracts) stand for values of variables
occurring in predicate position, whence our use in section 2 of the corresponding second-
order universal instantiation rule.

c) To put it in semantic terms, which are simpler to state but misleading (since a hyper-
intensional logic is not a theory of the semantical relations between expressions and their
meanings, but a theory of the ontological relations between entities suitable to be mean-
ings), \((A6)\) can be interpreted in our framework as guaranteeing that alphabetic change of
variables bound by the abstraction operator preserves meaning. Obviously, one also wants
that alphabetic change of quantified variables preserves meaning, thus the following more
general principle is needed, where the notion of ‘alphabetic variant’ is understood as usual:

\((A6)\) \([\lambda x_1...x_n \varphi] = [\lambda y_1...y_n \psi]\),
where these two \(\lambda\)-terms are alphabetic variants
of each other.

Note that (i) we could have used only \((A6)\) in section 2, but the definitions would then
have been less intuitive because of the presence of quantifiers \(Q\) (binding some variable \(v\))
whose scope is a formula containing a proper subformula in which \(v\) is already bound by
another quantifier, (ii) we could even have done without \((A6)\) at all, at the extra-expense
of the same unease with \(\lambda\)-bound variables.
Acknowledgements

I would like to thank the anonymous referees for their very valuable comments and Paul Égré and Denis Bonnay for their patience with my “Latex incompetence”. A preliminary version of the results contained in this paper was announced in a “Journée des doctorants” at the Institut d’Histoire et Philosophie des Sciences et Techniques, Paris, June 2, 2003.
A Simple Logic for Reasoning about Uncertainty

NíVEA de CarvalHO FerREIRA
University of Liverpool - Department of Computer Science
niveacf@csc.liv.ac.uk

ABSTRACT.
The introduction of powerful logical formalisms has had significant impact upon the development of knowledge representation over a number of years. In many realistic scenarios, agents that have to decide what action to perform next do not have sufficient information to infer that they know or believe what the current situation is. Although they may be biased to certain properties that hold, they are uncertain about the logical facts. In this paper we present a logic that, on the one hand, builds upon the natural framework of Kripke models, while allowing reasoning about uncertainty. Contrary to many logical approaches to probabilistic reasoning, the logic is compact and conceptually simple. Thus it represents a good candidate for representing and reasoning about uncertainty within computational agents. The logic is proved complete, moreover a decision procedure is provided.

1 Introduction

In both reasoning about agents and in reasoning within agents, it is vital to choose tools that allow the representation of knowledge at an appropriate level of abstraction, yet being simple enough to be mechanised. It is clear, however, that in realistic scenarios, such descriptions need to incorporate some elements of uncertainty. While there have been some steps in developing logics of uncertainty or logics of probability (see Section 6) many of these have either been complex to understand or complex to reason about.

In this paper we present a logic that builds upon the natural framework of Kripke models (the basis of modal logics), while allowing reasoning about uncertainty. The $P_{F}KD45$ Logic extends in some aspects the system $P_{F}D$ given in (van der Hoek 1997),

---

1This work is part of the author’s PhD project (still in progress) that is being developed under the supervision of Professors Michael Fisher and Wiebe van der Hoek.
which in turn was inspired by (Fattorosi-Barnaba and Amati 1987). The basic modal operator $P^\succ$ allow us to write formulas as $P^\succ_{0.5}\varphi$, meaning that the “agent believes $\varphi$ with probability strictly greater than 0.5”. The operators (which have self-explanatory meaning) $P^\geq$, $P^\leq$, $P^\preceq$ and $P^\succeq$ can be defined in terms of the basic one. Since probabilities range from 0 to 1, $P^\preceq_1$ is identified with the classical modal operator $B$ (Chellas 1980).

The focus of $P_F D$ was not on a doxastic interpretation of modalities, leading us to include two properties in $P_F K D 45$, expressed in axioms A8 and A9, in order to incorporate that characteristic. For instance, this allow us to have a probability distribution independent of world, and have nesting belief formulas equivalent to formulas without nesting. Those issues are considered in more details in Subsection 2.1.

A peculiar property of the logic semantics is that it only allows probability measures (for each world) that are in a base set $F$. Although this restricts probability assignments to a finite range, it is still possible to express and reason about arbitrary probabilities. Axiom A7 guarantees that arbitrary values collapse to values in the set $F$. The main motivation for using $F$ is the restoration of compactness for the logic.

Logics that allow us to express that $Prob(\varphi) \sim r$ are, in general, not compact\(^2\), witness the set of premises $\Gamma$

$$\{Prob(q) > \alpha \mid \alpha \in Q \cap [0, 1]\} \quad (6.1)$$

Then, obviously, we have $\Gamma \models Prob(q) = 1$, but there is no finite subset of $\Gamma$ that proves this conclusion. This has a computational counterpart: a mechanical device verifying whether a set of premises $\{Prob(\varphi) \sim r\}$ is satisfiable in $Q \cap [0, 1]$, in principle has to check an infinite number of assignments of probabilities to formulas $\varphi$. For these reasons, we assume that the range of allowed probabilities in $P_F K D 45$ is a finite base set $F \subseteq [0, 1]$.

Although the use of the set base $F$ causes logical restriction, it is possible to highlight some interesting aspects (cf.(van der Hoek 1997)). For instance, it we take $F = \{0, 1\}$, we have the classical modal logical. Having Driankov’s linguistic estimates (as in (Driankov 1987)) impossible, extremely unlikely, very low chance, small chance, it may, meaningful chance, most likely, extremely likely, certain would be modeled by a 9-element $F$. In other words, the granularity of $F$ can be chosen according to the intended agent’s application.

Once one of our main interests is to use the $P_F K D 45$ logic for describing and implementing uncertain agents, then having a mechanism for calculating the set $F$ is a desirable property. For this purpose, the basic idea is to have $F$ determined by a set of arbitrary probability values, extracted from the agent specification provided. Considerations about this topic is briefly addressed in Section 5.

Consequently, contrary to many logical approaches to probabilistic reasoning, our logic is compact and conceptually simple. Thus it represents a strong candidate for representing and reasoning about uncertainty within computational agents.

The paper is organised as follows. In Section 2 we present a syntax description of the language, and we provide its semantics and establish its properties. A decision procedure for the logic has been developed and implemented, and this result presented in Section 3. Due to space restrictions, only a small example showing the versatility of the approach is

\(^2\)Compact in the sense that inference in terms of infinite sets coincide with inference from finite sets.
provided in Section 4. In 5 we discuss briefly how to calculate the base set \( F \). Finally, related work and final remarks are presented in Section 6.

## 2 Language Description

The language \( L \) of \( P_FKD45 \) consists of a countable set of propositional symbols, the logical connectives \( \neg \) and \( \lor \) (with standard definitions for \( \bot, \top, \land, \rightarrow, \leftrightarrow \)), and parentheses. The basic modal operator, \( P^>_x \), is also defined, where \( x \) is a rational number within the interval \([0,1]\).

**Definition 1** A set \( F \) is a base for a logic \( P_FKD45 \) if it satisfies:

1. \( F \) is finite;
2. \( \{0,1\} \subseteq F \subseteq [0,1] \);
3. \( x, y \in F \) and \( (x + y \leq 1) \Rightarrow (x + y) \in F \);
4. \( x \in F \Rightarrow (1 - x) \in F \).

The logic is defined relative to a fixed base set \( F = \{x_0, x_1, ..., x_n\} \subseteq [0,1] \). It is assumed that \( x_i < x_{i+1} \), if \( i < n \) (implying \( 0 = x_0 \) and \( x_n = 1 \)). The basic operator is \( P^>_x \), with intended meaning of \( P^>_x \varphi \) being: “\( \varphi \) is believed to have a probability strictly greater than \( x \).”

The following abbreviations are used (in the subsequent schemes, \( x \) and \( y \) represent arbitrary rational values over \([0,1]\), and \( x_i, x_{i+1} \) are elements of the base set \( F \)):

- **D1.** \( P^>=x \varphi \equiv \neg P^<=1-x \neg \varphi \)
- **D2.** \( P^<=x \varphi \equiv P^>=1-x \neg \varphi \)
- **D3.** \( P^=x \varphi \equiv \neg P^<=x \varphi \)
- **D4.** \( P^=x \varphi \equiv \neg P^>=x \varphi \land \neg P^<=x \varphi \)

The inference rules (\( R1 \) and \( R2 \)) and axioms (A1–A9) of \( P_FKD45 \) are defined as follows.

- **R1** From \( \varphi \) and \( \varphi \Rightarrow \psi \) infer \( \psi \) (modus ponens)
- **R2** From \( \varphi \) infer \( P^>=x \varphi \) (necessitation rule)

- **A1.** All propositional tautologies
- **A2.** \( P^>=1 \varphi \Rightarrow [(P^>=x \varphi \to P^>=x \psi) \land (P^>=x \varphi \to P^>=x \psi) \land (P^>=x \varphi \to P^>=x \psi)] \)
- **A3.** \( P^>=1 \varphi \Rightarrow (P^>=x \varphi \to P^>=y \psi) \) (where \( y < x \))
A4. $P^\ge_0 \varphi$

A5. $P^\ge_{x+y} (\varphi \lor \psi) \rightarrow (P^\ge_x \varphi \lor P^\ge_y \psi)$ (where $x + y \in [0, 1]$)

A6. $P^\ge_{1-x} (\varphi \land \psi) \rightarrow ((P^\ge_x \varphi \land P^\ge_y \psi) \rightarrow P^\ge_{x+y} (\varphi \land \psi))$ (where $x + y \in [0, 1]$)

A7. $P^\ge_{x+1} \varphi \rightarrow P^\ge_x \varphi$

A8. $(P^\ge_0 P^\ge_x \varphi \rightarrow P^\ge_x \varphi) \land (P^\ge_0 P^\le_x \varphi \rightarrow P^\le_x \varphi)$

A9. $(P^\ge_x \varphi \rightarrow P^\ge_{x+1} \varphi) \land (P^\le_x \varphi \rightarrow P^\le_{x+1} \varphi)$

The axioms A1–A6 all reflect basic properties of probabilities. Axiom A7 reflects the peculiarity of having a base set $F$: it says that, if a probability is bigger than a certain value in $F$, it must be at least the next value.

Axioms A8 and A9 are included to emphasize the relationship with the modal logic $KD45$ and they make our agents doxastically introspective. The intuition behind these additional axioms is as follows. Axiom A8 denotes that, if the agent assigns a positive probability to some probabilistic judgment, then it incorporates this judgment. Axiom A9 states that the agent is absolutely sure about its own probabilistic beliefs. Those two axioms are introspective doxastic properties of $P_FKD45^3$.

The following lemma shows the benefit of having a finite base $F$: it guarantees that we can express, in the language, that every formula has a probability.

**Lemma 1** For all $\varphi \in L$, the following is a $P_FKD45$-theorem: $\neg(P^=\varphi \land P^=\psi), x \neq y; \text{ and } P^=_{x_0} \varphi \lor P^=_{x_1} \varphi \lor \ldots \lor P^=_{x_n} \varphi$ (recall: $F = \{0 = x_0, x_1, ..., x_n = 1\}$)

### 2.1 Semantics and Properties

The classical Kripke Model semantics refers to a collection of possible worlds — including the actual one — in which sentences of the language are set to true or false. Different worlds may have different interpretations for the sentences. In a way, the worlds can be identified by the formulas they verify.

In our Probabilistic Kripke Model over $F$, we have the same picture of possible worlds, adding a concept of probability distribution. In other words, there is an assignment of probability values (that are in $F$) to the set of possible worlds. Consequently, once those worlds model sentences of the language, we have assignment of probabilities to the formulas itself. That is, $P^\ge_x \varphi$ is true at a world $w$ if, and only if, the probability values assigned to the possible worlds (from $w$) that verify $\varphi$ sum up to a value greater than, or equal to, $x$.

The formal definition of the $P_FKD45$ models is as follows.

---

3Due to space limitations, some lemmas, theorems and remarks that follow directly from axioms could not be shown here. Besides, full proofs are generally omitted, but can found in the associated technical report (de C. Ferreira, Fisher, and van der Hoek 2004).
Definition 2 For each base set, $F$, $\mathcal{P}_F K\mathcal{D}45$ is the class of all models $M = \langle W, P_F, \pi \rangle$ for which:

- $W$ is a non-empty set (of worlds);
- $P_F$ is a function $P_F : W \to F$, satisfying$^4$:
  \[\sum_{w \in W} P_F(w) = 1\]
- $\pi$ is a valuation: $W \times L \to \{\text{true}, \text{false}\}$
- The truth definition for formulas is defined intuitively as:
  - $(M, w) \models p$ iff $\pi(w)(p) = \text{true}$, for atomic sentences $p$
  - $(M, w) \models \neg \varphi$ iff $M \not\models \varphi$
  - $(M, w) \models \varphi \land \psi$ iff $M \models \varphi$ and $M \models \psi$
  - $(M, w) \models P_{\exists x}^\varphi$ iff for each $w'$ s.t. $(M, w') \models \varphi$, $P_F(w') \geq x$.

Note that the probability distribution is independent of the world.

Lemma 2 $P_F K\mathcal{D}45$ is sound with respect to $\mathcal{P}_F K\mathcal{D}45$.

Completeness

Let $\varphi$ be a consistent formula of $P_F K\mathcal{D}45$. Below we show how to construct a model that satisfies $\varphi$.

Let $\Psi$ be the set of sub-formulas of $\varphi$ closed under single negation and satisfying, for any $\sim$ within $\{\langle, \rangle, \preceq, \succeq, =\}$, $(P_{\pi}^\varphi \in \Psi \Rightarrow \{P_{\pi}^x | x \in F \} \subseteq \Psi)$. With $\Psi$ being finite, say $|\Psi| = k$, we can define the $\Psi$-maximal consistent sets as $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$, $n \leq 2^k$. Let $\gamma_i$ be the conjunction of formulas in $\Gamma_i, i \leq n$. Then, we have:

1. $\Gamma \vdash \neg(\gamma_i \land \gamma_j)$, where $i \neq j$;
2. $\Gamma \vdash (\gamma_1 \lor \ldots \lor \gamma_n)$
3. $\Gamma \vdash \psi \leftrightarrow \gamma_{\psi_1} \lor \ldots \lor \gamma_{\psi_r}$, where $\gamma_{\psi_1} \lor \ldots \lor \gamma_{\psi_r}$ are exactly those $\gamma$’s which contain $\psi$ as a conjunct, for each $\psi \in \Psi$.

Since $\varphi$ is consistent and, by construction of the $\Gamma$’s, there is at least one $\Gamma_\varphi$ such that $\varphi \in \Gamma_\varphi$. Given this $\Gamma_\varphi$, we construct a set $\Phi \supseteq \Gamma_\varphi$ as follows. From Theorem $T8^5$, we know that for every consistent set $\Gamma$ and formula $\psi$, at least one set of the sequence

$$
\Gamma \cup \{P_0^- \psi\}, \Gamma \cup \{P_{x_1}^- \psi\}, \ldots, \Gamma \cup \{P_{x_{n-1}}^- \psi\}, \Gamma \cup \{P_1^- \psi\}
$$

$^4$In (van der Hoek 1997), the definition was $P_F : W \times P(W) \to F$ allowing different sets of worlds to have different probability distributions.

$^5$T8: $P_x^\varphi \rightarrow P_{\pi}^\varphi \backslash P_{\pi}^x \varphi \lor \ldots \lor P_{\pi}^x \varphi$, with $x_i = x^{\uparrow}$, where $x \uparrow = \min\{s \in F | s > x\}$, having $x \in (0, 1]$. 65
is also consistent. 
Now, we obtain $\Phi$ from $\Gamma_\varphi$ as follows:

1. let $\Phi_0 = \Gamma_\varphi$ (this set is consistent);
2. for $i = 1$ to $n$, we know that there is some $x \in F$ such that $\Phi_{i-1} \cup \{P^x_\varphi \gamma_i\}$ will be consistent, and we make the corresponding choice for $\Phi_i$.

Let $\Phi$ be $\Phi_n$; this is a consistent extension of $\Gamma_\varphi$, which contains a probability in $F$ for every ‘world’ $\Gamma_i$ ($i \leq n$). We are now ready to define a canonical model $M^c = \langle W^c, P^c, \pi^c \rangle$ as follows:

1. $W^c = \{\Gamma_\varphi\} \cup \{\Gamma_i \mid \exists x > 0 P^x_\varphi \gamma_i \in \Phi\}$;
2. $P^c(\Gamma_i) = x \iff P^x_\varphi \gamma_i \in \Phi$;
3. $\pi(\Gamma_i)(p) = \text{true if } p \in \Gamma_i$, and $\text{false}$ otherwise.

**Lemma 3 (Coincidence Lemma)** For all $\psi \in \Psi$ and $\Gamma \in W^c$

$$M^c, \Gamma \models \psi \iff \psi \in \Gamma$$

The following is immediate from Lemma 2 and Coincidence Lemma:

**Theorem 1 (Soundness and Completeness, Finite Models)** For any formula $\varphi$, we have $\mathcal{P}_F KD45 \models \varphi \iff P_F KD45 \vdash \varphi$. Moreover, every consistent formula has a finite model.

**Nested Beliefs**

Considering $P_F KD45$ as a language for representing properties within individual agents, below we show that for every nested belief formulas there is an equivalent to some formula without nesting.

**Lemma 4 (Independence of Probability Distribution)** Let $M = \langle W, P_F, \pi \rangle$ be a $\mathcal{P}_F KD45$ model. Then: $\exists w \in W(M, w) \models P^\geq_\varphi \beta \iff \forall u \in W(M, u) \models P^\geq_\varphi \beta$.

We are now going to show that nested beliefs are superfluous, in $P_F KD45$. This result is a generalisation of (Meyer and van der Hoek 1995, Theorem 1.7.6.4), where it is proved for $S5$, which means that their result still goes through when weakening the logic to $KD45^6$, and even when having probabilistic operators.

**Definition 3** We say that a formula $\psi$ is in normal form if it is a disjunction of conjunctions of the form $\delta = \omega \land \beta^1 \land \beta^2 \land \cdots \land \beta^m \land \alpha^1 \land \alpha^2 \land \cdots \land \alpha^k$, where $\omega, \beta^i, \alpha^j, (i \leq n, j \leq k)$ are all purely propositional formulas. The formula $\delta$ is called the canonical conjunction and the sub-formulas $P^\geq_\varphi \beta^i$ and $P^\geq_\varphi \alpha^j$ are called prenex formulas.

---

*The correspondence between $P_F KD45$ and $KD45$ logic is shown in (de C, Ferreira, Fisher, and van der Hoek 2004).*
Lemma 5 If $\psi$ is in normal form and contains a prenex formula $\sigma$, then $\psi$ is equivalent to a formula in the form $\pi \lor (\lambda \land \sigma)$ where $\pi, \lambda$ and $\sigma$ are in normal form.

This lemma guarantees that prenex formulas can always be moved to the outermost level.

Lemma 6 (Removal of Nested Beliefs) We have the following two equivalences in \(\mathcal{P}_{KD45}\):

\[
P^\geq_\alpha (\pi \lor (\lambda \land P^\geq _\gamma \beta)) \iff (P^\geq_\alpha (\pi \lor \lambda) \land P^\geq_\gamma \beta) \lor (P^\geq_\alpha \pi \land \neg P^\geq_\gamma \beta) \tag{6.3}
\]

\[
P^\geq_\alpha (\pi \lor (\lambda \land P^\geq _\gamma \beta)) \iff (P^\geq_\alpha (\pi \lor \lambda) \land P^\geq_\gamma \beta) \lor (P^\geq_\alpha \pi \land \neg P^\geq_\gamma \beta) \tag{6.4}
\]

In this way, we can bring all the probabilistic operators to the outermost level, giving us:

Theorem 2 Every formula $\varphi$ is equivalent to a formula $\psi$ in normal form, i.e., a formula without nesting of probabilistic operators.

3 Decision Procedure

As previously explained, the semantic definition for $\mathcal{P}_{KD45}$-formulas is based on Probabilistic Kripke Models. For each world $w$ there is a set of worlds that $w$ considers possible and each one of these possible worlds is specified according to the formulas it satisfies. So, by evaluating formulas, we identify the worlds where those formulas are satisfied\(^7\). As a result, we can obtain the values that, once assigned to the set of possible worlds, can satisfy the modal formula present in the agent’s specification.

The idea is to convert the set of formulas into constraint (in)equations. The inequation components represent all the possible truth valuations for the propositional symbols.

For instance, consider that the agent specification is expressed by the set of formulas: \(\{P^\geq_\alpha p, P^\geq_\alpha q\}\) and $F = \{0, 0.1, 0.2, ..., 0.9, 1\}$\(^8\).

In this case, the set of constraints generated is:

\[
p0q0 + p0q1 + p1q0 + p1q1 = 1 \tag{6.5}
\]

\[
p1q0 + p1q1 \geq 0.8 \tag{6.6}
\]

\[
p0q1 + p1q1 \geq 0.7 \tag{6.7}
\]

The first equation expresses the fact that probability values have to sum up to 1. The two inequations represent constraints on the worlds in which $p$ holds and worlds where $q$ holds, respectively.

Solving the constraint (in)equations determines which are the values in set $F$ that obey the constraints imposed by the formulas and can be, consequently, applied to the set of worlds. Therefore, the decision procedure turns out to be a mechanism of finding

---

\(^7\)Note that possible worlds refers exactly to a combination of truth values of propositional symbols.

\(^8\)The four possible sets of worlds (characterised by the truth-assignments) are: \(p1q1\) (where both $p$ and $q$ holds), \(p1q0\) (in which $p$ holds and the negation of $q$ holds), \(p0q1\) (in which the $q$ and negation of $p$ hold) and \(p0q0\) (where both negations hold).
all the possible probability assignments for the set of possible worlds that would satisfy the specified formulas, as long as this set of formulas is consistent. Otherwise, no possible assignment exists.

**Theorem 3 (Decision Procedure)** A formula \( \varphi \) in \( P_{F}KD45 \) is satisfiable if, and only if, there is a solution for the set of (in)equations generated from \( \varphi \) within the domain \( F \).

**Remark 1** The overall idea is that \( P_{F}KD45 \) system is used to verify if a set of formulas is consistent. Being a complete system, the idea is to try to establish a model for the set of formulas to show that this set is satisfiable (like in the Kripke model semantics), and, consequently, have the proof that the set is consistent.

## 4 Example

We present a simple example to show what an agent specification would look like in the \( P_{F}KD45 \) language. This is a variety of the common “travel agent” scenario whereby once the travel agent believes you might be interested in a holiday, it sends you information. Consider \( F = \{0, 0.1, ..., 0.9, 1\} \). The basic formula are given as follows\(^9\).

A. \( \text{ask}(you, x) \Rightarrow P_{0.8}^{\geq} \text{go}(you, x) \), i. e., “if you ask for information about the destination \( x \), I believe that you wish to go to \( x \) with probability greater than, or equal to, 0.8”

B. \( P_{1}^{=} [ \text{go}(you, x) \Rightarrow \text{buy}(you, holiday, x) ] \), i. e., “I believe that, if you wish to go to \( x \), then you will buy a holiday in \( x \)”

C. \( P_{0.5}^{=} \text{buy}(you, holiday, x) \Rightarrow \text{sendinfo}(you, x) \), i. e., “if I believe that you will buy a holiday for \( x \) with probability greater than, or equal to, 0.5, I will send information about holidays at \( x \)”

D. \( \text{ask}(you, x) \), i. e., “you ask for information on destination \( x \)”

>From D and A and R2 we have: \( P_{0.8}^{\geq} \text{go}(you, x) \) \hspace{1cm} (Res1)
>From Res1, A3 and item B: \( P_{0.7}^{=} \text{buy}(you, holiday, x) \) \hspace{1cm} (Res2)
>From Res2 and T6\(^{10}\): \( P_{0.5}^{=} \text{buy}(you, holiday, x) \) \hspace{1cm} (Res3)
>From Res3 and item C: \( \text{sendinfo}(you, x) \)

Referring to the decision procedure execution, there are three formulas to be evaluated (the ones that express degrees of beliefs):

1. \( P_{0.8}^{=} \text{go}(you, x) \) \hspace{1cm} (from A)
2. \( P_{1}^{=} [ \text{go}(you, x) \Rightarrow \text{buy}(you, holiday, x) ] \) \hspace{1cm} (from B)

---

\(^9\)Note that, for simplicity, we use predicates, rather than propositions, to represent this example, but the finiteness of the domain ensures that this can be reduced to a propositional problem if necessary.

\(^{10}\)T6: \( P_{x}^{=} \varphi \Rightarrow P_{y}^{=} \varphi \), where \( y \leq x \)
3. \( P_{0.5}^{\text{buy}}(\text{you}, \text{holiday}, x) \)  

(from C)

We obtain 6 solutions when solving those rules. Which means that, whatever solution is chosen as a possible value assignment, the antecedent of the third rule is true. Or, independently of the assignment \( \text{sendinfo}(\text{you}, x) \) is a logical consequence of the knowledge theory, and six assignments can be considered as options when building a model for the agent specification\(^{11}\).

5 Limiting \( F \)

Further results concerning the base set \( F \) include an appropriate base set for a formula to be calculated. In general we have that satisfiability is preserved when considering bigger sets: if \( F \subseteq F' \), then \( P_{F \cdot K \cdot D 45} \text{-satisfiability implies } P_{F' \cdot K \cdot D 45} \text{-satisfiability. As a consequence, we have that a formula } \varphi \text{ is } P_{F \cdot K \cdot D 45} \text{-satisfiable for some } F \text{ if, and only if, it is } P_{F \cdot K \cdot D 45} \text{-satisfiable for some generated } F'. \) Given a formula \( \varphi \), can we generate a \( F \) which is sufficient for satisfiability of \( \varphi \)? Succeeding in this automatic generation of \( F \), leaves to the system the task of determining this set, i.e., the user does not need to bother about the specific \( F \).

Let us sketch a way to construct an \( F \) from the formula \( \varphi \). Consider the formula \( \varphi \). Rewrite all the occurrences of \( P_x^\infty \psi \) in \( \varphi \) in such a way, that they all have a common denominator \( d \): every \( P_x^\infty \psi \) gets rewritten as a \( P_x^\infty \psi \). So, \( \varphi \) has denominator \( d \).

Proposition 1 Let \( \varphi \) be a formula in the language, with denominator \( d \). Then, \( \varphi \) is satisfiable for some \( F \) iff \( \varphi \) is satisfiable for \( F_{\varphi} = \frac{r}{2d \cdot 2^r} \), where \( k \) is the number of atoms occurring in \( \varphi \).

6 Related Work and Conclusion

Several methods have been developed to deal with uncertain information, often being split between numerical (or quantitative) or symbolic (or qualitative) ones (Parsons and Hunter 1998). \( P_{F \cdot K \cdot D 45} \) is a system that combines logic and probability. In this sense, it is related to other work that showed how this combination would be possible in different ways (Howson 1997; Fernando 1998). One of those possible approaches is the interpretation of the modal belief operator according to the concept of ‘likelihood’ (as in (Halpern and Robin 1987)). In this logic, instead of using numbers to express uncertainty one would have expressions like “\( p \) is likely to be a consistent hypothesis” (as a state is taken as a set of hypotheses “true for now”). That is, a qualitative notion of likelihood rather than explicit probabilities.

\( P_{F \cdot K \cdot D 45} \) was designed for reasoning with (exact) probabilities. Its Probabilistic Kripke Model semantics is similar to the one presented in (Fagin, Halpern, and Megiddo 1990; \(^{11}\)In this case, the six assignments for \([B0G0,B0G1,B1G0,B1G1] \) are: \([0,0,0,1],[0,0,1,9],[0,0,2,8],[1,0,0,9],[1,0,1,8] \) and \([2,0,0,8] \) (where “B” represents “buy(...)” and “G” “go(...)”).}
Fagin and Halpern 1994). In their formalism, a formula is typically a boolean combination of expressions of the form \( a_1 w(\varphi_1) + \ldots + a_k w(\varphi_k) \geq c \), where \( a_1, \ldots, a_k, c \) are integers, and each \( \varphi_i \) is propositional. One important remark is that the restriction of having \( \varphi \)'s as purely propositional does not apply to \( P_{T,KD45} \). Besides, the system in (Fagin, Halpern, and Megiddo 1990; Fagin and Halpern 1994) includes, as axioms, all the formulas of linear inequalities; consequently, their proofs of completeness rely on results in the area of linear programming. \( P_{T,KD45} \) logic is conceptually simpler. Finally, \( P_{T,KD45} \) differs mainly from other systems for representing beliefs and probability by allowing only a finite range of probability values, an assumption that at the same time imposes restrictions about the values that can be assigned to the possible worlds and permits the restoration of compactness for the logic.

In this paper, it was presented the \( P_{T,KD45} \) system, a system that combines modal logic and probability. Despite the inclusion of new axioms, theorems and slight changes in the logical semantics, it was shown how the logic preserves important results about soundness, completeness, finite model and decidability of the previous system \( P_{T,D} \) (van der Hoek 1997). In addition, new results about nested beliefs were presented, a decision procedure for the logic has been developed, and results of calculation of base set \( F \) shown. A a brief example was used to show how the language can serve and an appropriate agent specification language. In summary, we proposed not only a complete axiomatization for the logic, but also a decision procedure that permits us to verify satisfiability of \( P_{T,KD45} \)-formulas.

Maybe the work closest to ours is that of (Ognjanovic and Rakovic 2000). It considers languages for first-order probabilities, and the compactness of \( P_{T,KD45} \) easily follows from (Ognjanovic and Rakovic 2000)|Theorem 11]. They also consider the case in which all the worlds are assigned the same probability function, but for a language that forbids iteration.

The use of a finite range \( F \) of probability values is a peculiar property of our logic. Although the use of the set base \( F \) causes logical restriction, it is possible to chose its granularity according to the intended agent’s application. Besides, as discussed earlier in Section 1, the compactness that it brings has benefits. Furthermore, having a finite range of probability values reduces the computational effort when building a model for the agent logical description. Finally, this work on \( P_{T,KD45} \) represents one step towards our main goal: an agent programming language capable of specifying and implementing agents that deal with uncertain information, proposing new mechanisms for handling this uncertainty in executable specifications. Future work will concentrate on the developing an executable framework that combines the probabilistic approach of \( P_{T,KD45} \) with the dynamic approach of Temporal Logics.

**Acknowledgements**

The author gratefully acknowledges support by the Brazilian Government under CAPES-scholarship. The author also thanks to the anonymous referees for their relevant suggestions.
References

about uncertainty. Technical report, Department of Computer Science - University
of Liverpool. Available online: http://www.csc.liv.ac.uk/~niveacf/.
belief. In IDA annual research report, pp. 113–120. Linköping University.
Fagin, R., J. Y. Halpern, and N. Megiddo (1990). A logic for reasoning about probabil-
Intelligence 32(3), 379–405.
ence 48, 517–531.
Science. Cambridge University Press.
Computer Science 247, 191–212.
A. Hunter and S. Parsons (Eds.), Applications of Uncertainty Formalisms. Berlin:
Springer-Verlag.
Norwegian Word Order in HPSG

Liv Ellingsen
Universitetet i Oslo
liv.ellingsen@ifl.uio.no

Abstract.
This paper presents an HPSG analysis for main word order phenomena in Norwegian. The analysis has been built on the grammar 'starter-kit' the LinGO Grammar Matrix (Bender, Flickinger, and Oopen 2002), and it has been implemented in the LKB system (Copestake 2001). The main phenomena treated in the analysis are the verb-second (V2) constraint in Norwegian main clauses, the distribution of certain types of sentence adverbials and the contrast between main and subordinate clause structures.

1 Introduction

The goal of the work described in this paper has been to provide an HPSG analysis for certain word order phenomena in Norwegian. The analysis should preferably be simple enough to accomodate a running implementation with the tools I had at hand, i.e. the LKB grammar engineering platform\(^1\). Part of the project was also to see to which extent such a grammar would benefit from being built on a language-independent grammar 'starter-kit' the LinGO Grammar Matrix.

The data accounted for by the analysis are described in section 2, while parts of the actual analysis and implementation is described in section 3.

2 Data

Norwegian is mainly viewed as a V2 language, i.e. the finite verb always occupies the second position in a Norwegian declarative main clause, as shown in (1).\(^2\) Combined with

\(^1\)The LKB system: http://www-csli.stanford.edu/~aac/lkb.html.

\(^2\)A discussion of cases where the finite verb may additionally be preceded by a modifier is not within the scope of this paper, but it should be possible to include this phenomenon in the analysis and implementation presented here.
the possibility of topicalization, this implies that the subject has more than one possible position in this type of clause. In a clause where another element has been topicalized, the canonical subject position is after the finite verb, before any of the verb’s complements, as seen in (1-b).

(1) a. Gyrd ga boka til Inge.
    Gyrd gave the-book to Inge
    Gyrd gave the book to Inge.

b. Til Inge ga Gyrd boka.
    to Inge gave Gyrd the-book

The subordinate clause structure differs from the main clause structure. Topicalization is not allowed in a subordinate clause structure, and so it has the strict SVO order shown in (2).

(2) a. [da] Gyrd ga boka til Inge.
    [when] Gyrd gave the-book to Inge

b. *[da] til Inge ga Gyrd boka.
    [when] to Inge Gyrd gave the-book

Another interesting structural difference is in the distributional possibilities of adverbials in the two clause types, and here especially those of so-called sentence adverbials, which will be described further in this paper. In main clauses, an adverbial like sikkert (surely) can be placed directly after the finite verb as seen in (3), or directly after an inverted subject. In subordinate clauses, the sentence adverbial can be placed directly before the finite verb, as the V2 constraint does not apply to this type of structure.

(3) a. Gyrd ga sikkert boka til Inge.
    Gyrd gave surely the-book to Inge
    Gyrd surely gave the book to Inge.

b. *Gyrd sikkert ga boka til Inge.
    Gyrd surely gave the-book to Inge

    [when] Gyrd surely gave the-book to Inge

    [when] Gyrd gave surely the-book to Inge

Sentence adverbials (and other adverbials) can mainly be placed in three different positions in a Norwegian clause: (i) in the sentence-initial position (topicalized) (ii) in the middle field after the finite verb but before any of its complements in main clauses, or directly before the finite verb in subordinate clauses and (iii) at the sentence end. By

---

3I use the term ‘main/subordinate clause structure’ instead of ‘main/subordinate clause’, as the common complementizer at (that) can head subordinate clauses with both main and subordinate clause structure.

4Because of their distributional possibilities, names like nexus adverbial (Diderichsen 1962) and central adverbial have also been used to describe sentence adverbials.
comparing the three sentence adverbials *sikkert, heldigvis (luckily)* and *alikevel (still)*, the sentences in (4) show how sentence adverbials differ with regard to their distributional possibilities.

(4)  

a. Til Inge ga Gyrd heldigvis / sikkert / alikevel boka.  
    to Inge gave Gyrd luckily / surely / still the-book  
    Luckily/Surely/Still, Gyrd gave the book to Inge  

b. Gyrd ga boka til Inge *heldigvis / *sikkert / alikevel.  
    Gyrd gave the-book to Inge *luckily / *surely / still  

 c. Heldigvis / *Sikkert / Allikevel ga Gyrd boka til Inge.  
    luckily / *surely / still gave Gyrd the-book to Inge

A very small group of sentence adverbials (like *alikevel*) can be placed in all the three possible positions in a clause. The large remaining group of sentence adverbials can not be placed at the end of the sentence after any of the verbs complements\(^5\). This group is then divided into ‘heavy’ sentence adverbials that can be stressed, and hence topicalized (like *heldigvis*), and ‘light’ sentence adverbials that can only stand in the middle field of the sentence.

Sentence adverbials that can stand at the sentence end in a main clause can also stand in this position in a subordinate clause structure. Most sentence adverbials can stand directly to the left of the finite verb in a subordinate clause, while attempts at placing them to the left of the subject usually result in ungrammatical or highly doubtful constructions for most adverbials\(^6\), as shown in (5-b).

(5)  

    [when] Gyrd luckily / surely / still gave the-book to Inge  

b. [da] *heldigvis / *sikkert / *alikevel Gyrd har gitt boka til Inge.  
    [when] *luckily / *surely / *still Gyrd gave the-book to Inge

3 Analysis and implementation

The goal of the work described in this paper has been to provide an analysis for these phenomena of the Norwegian language that also could serve for an actual implementation of them in the framework of HPSG.

The analysis and implementation presented follow mainstream theory for Head-driven Phrase Structure Grammar (HPSG), as represented by Pollard and Sag (1994) and Sag and Wasow (1999). The HPSG framework is constraint-based and lexicalized, and it uses typed feature structures to model (all) linguistic objects. The types are organized in a type

---

\(^5\)If none of the verb’s complements or adjuncts are placed at the end of the sentence, sentence adverbials in the middle field might physically end up at the end of the sentence, though. Additionally, some adverbs can be extraposed after the sentence, using comma-intonation, but this will not be taken into account here.

\(^6\)Specific exceptions to this pattern exist, including adverbials like *ikke (not)*. In some cases the context or properties of the subject might also make this pattern the preferred one.
hierarchy where multiple inheritance is used to express generalizations across linguistic objects on different levels.

The analysis and implementation of the phenomena are built on the LinGO Grammar Matrix (Bender, Flickinger, and Oepen 2002), a language-independent ‘starter kit’ for grammar developers. This core grammar provides basic types for fundamental classes of constructions, such as head-subject, head-complement and head-adjunct phrases. Obviously, the Matrix aims to avoid any constraints on surface order of daughter constituents in these phrases.

The LinGO Grammar Matrix is mainly based on the English LinGO ERG (Flickinger 2000). This makes it interesting to see whether the basic types can be directly transferred to a Norwegian grammar, or if they need to be adjusted in any way before they are put to use.

3.1 The subject

Concerning the possibility of subject inversion in Norwegian, this construction is also found in English (for instance with inverted auxiliaries in polar questions, e.g. Did Kim sleep?) and it should therefore be possible to analyse and implement it based on the Matrix’ construction types. One possible way to handle inversion in English is to use a lexical rule that moves the subject to the front of the verb’s complement list (Sag and Wasow 1999). As the order of the inverted subject and the complements are fixed in Norwegian, this would be a possibility. When the sentence adverbials are introduced, this solution becomes problematic, though: Because the sentence adverbials can be placed on each side of an inverted subject, but usually not between two complements, it is necessary to explicitly keep track of the subject.

Another possible solution for English is to use flat rules that bind the subject and the complements at the same time, all as sisters of the verb (Ginzburg and Sag 2001). This is also a possibility for Norwegian, but again the sentence adverbials complicate the picture. Because the adjuncts would have to be taken into account in a flat rule, a large number of rules would be needed to account for all possible combinations of subject, adjuncts and complements.7

Clause schemata

A traditional way to describe Scandinavian clause structure originates in Diderichsen’s clause schemata (Diderichsen 1962). The schemata are ordered sequences of fields with specific constraints on the content of each field. Figure 7.1 shows Diderichsen’s clause schemata for main and subordinate clause structures, slightly adapted for Norwegian (Faarlund, Lie, and Vannebo 1997).

---

7 This approach will grow even more complicated when the analysis is extended to cover the phenomenon of object shift as well, where light, unstressed subjects and complements like pronouns precede sentence adverbials in the middle field (Sells 1998).
Main clause schema

<table>
<thead>
<tr>
<th>Fund. field</th>
<th>Nexus field</th>
<th>Content field</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>v a1 n a2</td>
<td>V N A</td>
</tr>
<tr>
<td>Til Inge</td>
<td>ga - Gyrd</td>
<td>sikkert - boka</td>
</tr>
<tr>
<td>Til Inge</td>
<td>har - Gyrd</td>
<td>sikkert gitt boka</td>
</tr>
<tr>
<td>(to Inge)</td>
<td>has Gyrd</td>
<td>surely given the-book</td>
</tr>
</tbody>
</table>

Subordinate clause schema

<table>
<thead>
<tr>
<th>Comp. field</th>
<th>Nexus field</th>
<th>Content field</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>a1 n a2 v</td>
<td>V N A</td>
</tr>
<tr>
<td>At</td>
<td>- Gyrd sikkert ga</td>
<td>boka til Inge</td>
</tr>
</tbody>
</table>

Figure 7.1: Diderichsen's clause schemata, adapted for Norwegian

It is possible to use these schemata as a basis for an alternative HPSG analysis of the phenomena they are describing. This view is also presented for Danish in Jensen and Skadegauge (2001). Their analysis is inspired by Kathol (1995), where topological fields and a topological hierarchy are used to handle linearization in German. In his work, Kathol modifies standard HPSG theory, introducing discontinuous constituents and a 'domain union' relation (Reape 1994).

However, since the goal of this work has been to provide an efficient working implementation of the analysis presented, analyses that need heavy machinery to enable an implementation, like those involving separate linearization domains, was dispreferred. The hope was that since Norwegian, like English, is a language with a relatively fixed word order, it could be described without any use of the heavy machinery developed for languages with far more free word order, like German.

A rule-based account

The simplest way to account for the word order phenomena (from an implementational point of view) seems to be a constructionist analysis. However, such a rule-based analysis must keep its number of rules down to still qualify as 'simple', and at the same time cover all the selected phenomena in a linguistically adequate manner. For subject inversion in Norwegian main clause structures, two such rule-based accounts have already been shown to be inadequate for the purposes of this grammar (see page 75).

To keep track of the subject in an inverted structure while still benefiting from using binary branching structures, it is necessary to define the subject position in the nexus field (after the finite verb) in a main clause structure as a canonical subject position for this type of clause. In a subordinate clause structures, the only possible subject position is the position before the finite verb in the nexus field.

It is thus necessary at least to define two head-subject rules, one head-final, combining
the finite verb with its subject to the left, and one head-initial, combining the finite verb with its inverted subject to the right and marking the structure as a main clause structure. This marking can for instance be done via the feature MC (main clause), which takes a boolean as its value.

It is also possible to follow Diderichsen’s schemata even more closely in this respect and say that a subject in the sentence-initial position is extracted from its canonical inverted position and topicalized in the same way as complements and adjuncts. In that case, the head-final version of the head-subject rule marks the structure as a subordinate clause structure ($MC -$). This analysis has been implemented in the grammar on which this paper is based and it results in the two structures shown in figure 7.2 for the (correctly syntactically ambiguous) sentence Inge beundrer Gyrd (Inge admires Gyrd).

![Figure 7.2: Main clause structure with subject and object extraction](image)

To accomodate the head-initial head-subject rule, a few changes had to be made to the basic types in the Matrix, for instance when it came to the treatment of complements. The Matrix did not expect the subject to be combined with its verb before all complements were gone from the COMPS (complement) list and something equivalent to a VP had been built. When the verb is combined with its subject before any of the complements, violating traditional VP structures, the complements had to be passed up from the head daughter to the mother in a head-subject phrase, so that a head-complement phrase could apply afterwards. Originally, the mother’s COMPS list was set to be the empty list in the Matrix.

Giving up on a traditional VP in this manner can be debated. However, if a surface-oriented approach to word order is adopted, as is the case here, it might (as already mentioned) be a necessary step to account for the Norwegian data. It is of course still necessary to be able to build a traditional VP for other constructions included in the grammar.

### 3.2 Sentence adverbials

This approach lets us keep track of the subject in a clause, but it is not quite enough to ensure correct placement of sentence adverbials in the nexus field. The nexus field of a main clause structure spans from the finite verb to the content field, which starts with
the first complement of the finite verb. But as the Matrix uses binary head-complement phrases, cancelling the complements of the \texttt{comps} list one by one\textsuperscript{8}, it is not possible to see whether a structure still has a complete \texttt{comps} list, or whether one or more complements have already been cancelled of the list.

A solution to this problem is to define a feature \texttt{nucl}\textsuperscript{9} (nucleus) in the verb’s feature structure, whose boolean value tells us whether we still are in the nexus field or not. The \texttt{nucl} feature turns out to supply all the information that need to be transferred from the clause schemata to the grammar for the purpose of covering the phenomena presented in this paper.

All finite verbs start out with the \texttt{nucl} value \texttt{+}. The inverted head-subject rule leaves the \texttt{nucl} value of the phrase unchanged, while any version of a head-complement rule changes the value to \texttt{−}. Modification by a sentence adverbial in the nexus field also leaves the \texttt{nucl} value unchanged.

Head-modifier rules handling adverbials in the nexus field mark the structure as a main or subordinate clause structure in the same way as the different types of head-subject rules, as the head-final/head-initial distinction plays the same role in these cases.

The different types of adverbials themselves can be defined as shown in the type hierarchy in figure 7.3. The hierarchy defines the types of modifiers according to their distributional possibilities only, so cross-classification with types that contribute information from other dimensions (for instance part-of-speech) is necessary.

\begin{center}
\begin{tikzpicture}

\node (mod) {mod};
\node (final-or-strictly-final-mod) [below left of=mod] {final-or-strictly-final-mod}
\node (nex-or-final-mod) [below right of=mod] {nex-or-final-mod};
\node (nex-mod) [below right of=final-or-strictly-final-mod] {nex-mod};
\node (strictly-final-mod) [below left of=nex-mod] {strictly-final-mod};
\node (final-mod) [below right of=nex-mod] {final-mod};
\node (emph-mod) [below right of=strictly-final-mod] {emph-mod};
\node (non-emph-mod) [below right of=final-mod] {non-emph-mod};

\draw (mod) -- (final-or-strictly-final-mod);
\draw (mod) -- (nex-or-final-mod);
\draw (final-or-strictly-final-mod) -- (nex-mod);
\draw (final-or-strictly-final-mod) -- (strictly-final-mod);
\draw (final-or-strictly-final-mod) -- (final-mod);
\draw (nex-or-final-mod) -- (emph-mod);
\draw (nex-or-final-mod) -- (non-emph-mod);
\end{tikzpicture}
\end{center}

\textbf{Figure 7.3: Type hierarchy below \textit{mod}}

The type \texttt{nex-or-final-mod} has no constraints on its placement, and sentence adverbials that can be placed in all possible positions of a clause are defined as instances of this type. They are then forced down to the appropriate subtype by a rule application (i.e. the rule specifies its daughter to be of a specific subtype of \texttt{nex-or-final-mod}), to receive constraints necessary for obtaining correct parses.

The type \texttt{nex-mod}, sentence adverbials that can not stand at the sentence end, is constrained to modify elements with \texttt{nucl} value \texttt{+}. Its two subtypes, \texttt{emph-mod} and \texttt{non-emph-mod}, have no constraints on their own, but the extraction rules make use of this

\textsuperscript{8}This is the preferred method, as using flat head-complement rules would complicate matters once one wants to include object shift in the grammar.

\textsuperscript{9}The feature name \texttt{NEX} has already been put to use in the LinGO ERG.
distinction to make sure that only heavy sentence adverbials are topicalized. Finally, the type *final-or-strictly-final-mod* are constrained to modify elements with an empty *comps* list.\(^\text{10}\)

Figure 7.4 shows two parse trees where the type of clause is decided on by the placement of the sentence adverbials. The parse tree for the left sentence unambiguously shows a main clause structure because of the *nex-mod* sentence adverbial placed to the right of the finite verb. The structure to the right can only be a subordinate clause structure due to the left-modification of the verb.

If we compare the left main clause structure in figure 7.4 with the two possible structures for the simple sentence in figure 7.2, we see that the placement of the *nex-mod* adverbial *neppe* also disambiguates the structure, deciding that *Gyrd* can only be the inverted subject of the clause.

![Figure 7.4: Main and subordinate clause structure, type of structure decided by adverbial placement](image)

4 Conclusion

The analysis presented in this paper covers the main phenomena of Norwegian word order, and although this is not discussed presently, it supports an extension to cover more special, but important phenomena like object-shift. The relative simplicity of the analysis has made

\(^{10}\)The distinction between *strictly-final-mod* and *final-mod* is made for purely technical reasons, to avoid spurious parses in cases where an adverbial can be extracted from more than one position in the sentence.
it well-suited for implementation, and a running version of the grammar has been built and tested using the LKB system.\footnote{Downloadable grammar: http://folk.uio.no/livel/nor_grammar/}

Using the LinGO Grammar Matrix as a grammar ‘starter-kit’ turned out to be an advantage for this project. Although some changes had to be made in order to facilitate the Norwegian grammar, most of these changes can likely be accommodated by further underspecification in future versions of the Matrix.

**Acknowledgements**

I would like to thank Jan Tore Lønning and Lars Hellan for advise and support throughout and after the completion of my Master’s thesis. I would also like to thank four anonymous reviewers for helpful comments on this paper, and especially for an insightful and detailed review of the data.
References


Towards a Time Tagger for Romanian

Corina Forăscu  
Faculty of Computer Science, University of Iași, Romania  
corina@infoiasi.ro

Daniela Solomon  
Faculty of Computer Science, University of Iași, Romania  
solomon@infoiasi.ro

Abstract.  
The paper describes the temporal annotating process on a Romanian version of a parallel corpus from the literature register - using the TimeML mark-up language that we discuss comparatively with TIMEX2 standard\(^1\), some solutions to the problems we had to deal with, methods for extracting time expressions and their links, preliminary results in automatic tagging\(^1\) and some possible applications for temporal annotated information.

1 Introduction

Recent work in document analysis starts focusing on the temporal information in documents, mainly for their use in many practical NLP applications such as question answering (questions like "when", "how often" or "how long"), summarization (temporally ordered information, biographic summaries), information extraction or machine translation (translated and normalized temporal references).

We took a deeper look into the temporal information in a Romanian corpus from the literature register because we wanted to see how the TimeML annotation scheme could be applied to our language and to another register than the news - for which TimeML was developed. Our guess was that some slight modifications are needed when applied to Romanian, due to the language particularities.

The paper starts with an overview of the work done in the temporal annotation field. In section 3, 4 we briefly describe the corpus used for annotation and TimeML - the mark-up language that we applied and that we discussed comparatively to TIMEX2 standard. Some

\(^1\)Many thanks to the anonymous reviewers for suggesting these improvements to us.
preliminary observations and results are also reported in section 4. In section 5 we present the process of merging between annotation standards. In the following section we discuss the algorithms we used towards creating an automatic time tagger and their evaluation. Intended work and applications are presented in section 7 and finally, in section 8, the conclusions.

2 Passing through TIME

Regarding the temporal annotation schemes, the 7th Message Understanding Conference (MUC 1998) defines the Named Entity (NE) task intended to recognize and classify time expressions. They had to be determined only for events of interest: the task requires the SGML marking up for the fully-specified time expressions and their classification using the DATE and TIME attribute type. Type DATE/TIME refers to complete and partial calendar dates/day times but not to those that are context-dependent.

As revealed in a study on a print and broadcast news corpus (Mani and Wilson 2000), more than two-thirds of time expressions are context-dependent. Hence in (Wilson et al. 2001) a different annotation scheme was introduced. It augments the former with a wider range of flagged time expressions, a richer representation of time values. It represents the time expressions by TIMEX tag of type TIME or DATE, using the ISO-8601 standard with several extensions for some commonly occurring temporal units. There was no tag assignment for the time periods; generic, indefinite and ambiguous time expressions were tagged without a value.

During the ACL-2001 Workshop on Temporal and Spatial Information Processing some other approaches were reported. In (Schilder and Habel 2001) the "event" and the temporal relations between events and time were located in German newswire articles, by using a semantic tagging system for temporal expressions. Another approach (Filatova and Hovy 2001) assigns a time stamp to every event - only clauses but no nominalizations were taken into account, ignoring the relations between events. In (Katz and Arosio 2001), through an intrasentential annotation, the temporal relations between events (verbs) were detected in a multilingual corpus.

Following researches and a pilot study, in (Setzer 2001) a fine-grained annotation scheme capturing all events, times, signals and temporal relations is described. The scheme introduces the EVENT tag with ten attributes, some of which were intended to deal with the temporal relations, the simple TIMEX tag (with three attributes) and the complex one (with three additional attributes). Using a graphical annotation tool specially developed, the annotation scheme has been evaluated through the construction of a trial news corpus. Some improvements in the annotation tool, the importance of annotating temporal relations and the representation of extracted temporal information as a time-event graph are discussed in (Setzer and Gaizauskas 2002).

The TIDES programme (Translingual Information Detection, Extraction and Summarization) developed a guideline for annotating time expressions, a method for associating

\[^2\text{http://www.itl.nist.gov/iaui/894.02/related\_projects/muc/main.html}\]
a canonical representation to times and a method for extracting such time expressions from multiple languages (Wilson et al. 2001). Working in two steps (flagging a temporal expression by using lexical trigger words; identifying the time value that the expression designates), the annotation process requires assigning values for 7 attributes in the TIMEX2 tag: VAL, MOD, SET, PERIODICITY, GRANULARITY, NON_SPECIFIC, COMMENT. Using lexical criteria, the tag extent can comprise a noun, a noun phrase (NP), an adjective, an adverb or an adjectival/adverbial phrase, but not a prepositional phrase or a clause of any type. The VAL attribute permits to represent calendar/clock time and durations in a normalized form (based on ISO 8601, with extensions) of the contextually determined interpretation of the time designated by the temporal expression. When applying temporal tags, there are two basic annotation principles (Ferro et al. 2001): "If a human can determine a value for a temporal expression, it should be tagged" and "VAL must be based on evidence internal to the document that is being annotated". The MOD attribute captures the basic semantics of quantifier modifiers and lexicalised aspect markers; it clarifies the interpretation of the value of VAL. The SET attribute represents sets of times, i.e. times that recur regularly or irregularly. The unit of time denoted by the members of the set is represented using the GRANULARITY attribute. The frequency of regular recurrence is expressed using the PERIODICITY attribute. The NON_SPECIFIC attribute permits to represent the generically used NPs, the singular indefinite NPs that are used referentially and other non-specific NPs.

The TIMEX2 annotation scheme was applied not only to news, but also to meeting scheduling dialogs (Mani et al. 2001). The complete annotation guideline is given in (Ferro et al. 2001). The TIMEX2 scheme is applied to the Automatic Content Extraction (ACE\textsuperscript{3}) evaluation, Relation Detection and Characterization (RDC) task, whose goal is to detect and characterize Entity-Entity and Event-Entity relations.

During the Temporal and Event Recognition for Question Answering Systems\textsuperscript{4} (TERQAS) workshop in 2002, the major deliverables of the ARDA funded contract was the creation of TimeML and TIMEBANK (Pustejovsky et al. 2002). TimeML is a metadata standard for marking events, their temporal anchoring and links in news articles. Among other accomplishments, there have been presented algorithms for recognizing temporal and event expressions, anchoring of events in time, ordering the events on a time axis, a text segmented closure algorithm, guidelines for temporal annotation, a scoring and inter-annotator evaluation tool. By means of seven tags, which we will briefly describe in section 4, the TimeML v1.0 language is probably the most used in temporal annotations. Continuing the activity of the workshop, in (Pustejovsky et al. 2003) a graphical annotation toolkit for TimeML was developed.

Taking into consideration both TIMEX2 and TimeML schemes, we decided to use the TimeML v1.0 because it is more complex and it treats unitarily the temporal aspects of texts. Even if TIMEX2 deals only with the temporal expressions (and somehow it does it better than TIMEX3, its TimeML descendent), we must consider it as a component technology in ACE, conceived to fill the temporal attributes for extracted relations and

\textsuperscript{3}http://www.nist.gov/speech/tests/ace/index.htm
\textsuperscript{4}http://www.cs.brandeis.edu/~jamesp/arda/time/terqas/
events. In section 4 we will discuss the pros and cons for TIMEX2 compared to TIMEX3, taking also into consideration the last versions of TIMEX2 (Ferro et al. 2004) and TimeML (Sauri et al. 2004).

Time taggers have been developed for: English (TempEx - G. Wilson at MITRE), Korean (KTX - SeokBae Jang at Georgetown), any language (TDL - Jennifer Baldwin at Georgetown; tested on English, French, Spanish, Korean). Another time tagger was developed in Spanish (Saquete et al. 2002), using a specific annotation scheme.

Regarding the Semantic Web, some significant efforts have taken place in order to develop ontologies of time, for expressing the temporal content of web pages and temporal properties of web services (ONTOLINGUA\textsuperscript{5}, SUMO\textsuperscript{6}, CYC\textsuperscript{7} between others). One of the latest collaborative efforts is the DAML-Time ontology\textsuperscript{8} (Hobbs 2002), which covers the basic topological temporal relations on instants and intervals, measures of duration, clock and calendar units, months and years, time and duration stamps. A mapping between TimeML and DAML-Time is presented in (Hobbs and Pustejovsky 2003), in order to facilitate reasoning over events, their orderings and their relative granularity in texts.

Both TimeML and DAML-Time are very well disseminated. In 2003 there was an advanced ESSLLI course\textsuperscript{9} on TimeML\textsuperscript{10}, conducted by J. Pustejovsky, R. Gaizauskas and G. Katz and a course on ontologies for the Semantic Web, including DAML-Time\textsuperscript{11}, presented by Jerry Hobbs at the EUROLAN 2003 summer school\textsuperscript{12} - to mention just some examples.

3 Corpus

The corpus we have chosen to work with is George Orwell’s ”1984”. It is a parallel corpus consisting of the original English version of the novel and six integral translations in Romanian, Bulgarian, Hungarian, Czech, Slovene and Estonian. Developed during the MULTTEXT-EAST Copernicus project\textsuperscript{13}, the corpus was consistently SGML encoded, sentence aligned and morpho-syntactically annotated (Tuﬁş et al. 1997). During the EC funded project BalkaNet\textsuperscript{14}, the corpus was enriched with the Greek, Serbian and Turkish translations of the novel, which have the same annotation layers.

The advantages of a parallel corpus in language engineering are notable. Cross-lingual information extraction, automatic sense tagging, extraction of translation equivalents and hence of a multilingual dictionary are only some examples of the applications of a parallel corpus. Besides, for Romanian-English, there are many layers of annotation that had been

\textsuperscript{5}http://www.ksl.stanford.edu/software/ontolingua/
\textsuperscript{6}http://ontology.teknowledge.com
\textsuperscript{7}http://www.cyc.com/cycdoc/vocab/time-vocab.html
\textsuperscript{8}http://www.cs.rochester.edu/~ferguson/daml/daml-time-nov2002.txt
\textsuperscript{9}http://www.logic.at/esslli03/giveabs.php?13
\textsuperscript{10}http://www.cs.brandeis.edu/~jamesp/arda/time/esslli.html
\textsuperscript{11}http://www.racal.ro/EUROLAN-2003/html/presentations/JerryHobbs/romania-aug03-2.ppt
\textsuperscript{12}http://www.racal.ro/EUROLAN-2003/html/
\textsuperscript{13}http://inljs.si/ME
\textsuperscript{14}http://is.dblab.upatras.gr/public_collection/ Balkanet

85
added to the original version of the corpus, making it a heavily annotated and hence a very useful one for different levels of natural language processing.

For the beginning we used a fragment from the first chapter of the novel (Romanian version), summing approx. 2700 words, 95 clauses. Each word in the text was annotated to part of speech (POS-tagged), context-disambiguated lemma and other word-level morpho-syntactic information.

The Romanian (and English) text was also NP-tagged and marked for reference expressions. We did not use these extra XML-annotation levels when we have annotated the text, but, as we will see, they can be useful in an automatic temporal tagger. A sense tagger for the parallel corpora (Tufiş and Ion 2003) developed at the Romanian Academy Institute for Artificial Intelligence is under evaluation using the same corpus.

4 Hand-Annotation for temporal events and relations

For annotation we used the TimeML language - a standard for marking events, their temporal anchoring and links in news articles - and the PALinkA annotator\(^\text{15}\) - a fairly advanced interface which allows a wide range of annotations. We will briefly present the seven tags together with our results and some remarks.

The EVENT tag is used to annotate situations that happen or occur, states or circumstances in which something obtains or holds true. Syntactically, EVENTS are typically verbs, although nominalizations, adjectives, predicative clauses or prepositional phrases are also annotated as events. Among the four attributes, ID, class, tense, aspect, the class attribute specifies if the event is of type REPORTING, PERCEPTION, ASPECTUAL, I_ACTION, I_STATE, STATE or OCCURRENCE.

We tagged all the events, including those that are not temporally anchored or related with other events, even if in the news register the guidelines recommend not to tag the generic events. Especially in the fiction register, having all the events annotated could be of use in question answering or in discourse processing, as we will see in the section 7. We had problems with the "aspect" attribute, which has to be filled for verbs, because the Romanian Grammar, issued by the Romanian Academy (Graur 1966), does not take into consideration this grammatical category for verbs. A clear distinction made between perfective and imperfective verbs (Irimia 1997) allowed us to tag with a specific aspect only some of the verbs.

The distribution that we obtained for the events, according to their class, is given in the table 8.1:

The TIMEX3 tag is used to mark up explicit temporal expressions, such as times of a day, dates - calendar dates or ranges, durations. Dates and times are represented using the W3C-defined profile of ISO 8601, 1997, with some extensions (Ferro et al. 2001) that allow representing NL temporal expressions with different levels of precision.

\(^{15}\)Developed by Constantin Orășan at the University of Wolverhampton, http://clg.wlv.ac.uk/projects/PALinkA/
<table>
<thead>
<tr>
<th>OCC</th>
<th>STATE</th>
<th>REPORT</th>
<th>I_ACT</th>
<th>ASP</th>
<th>I_STATE</th>
<th>PERCEPT</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instances</td>
<td>219</td>
<td>48</td>
<td>17</td>
<td>14</td>
<td>12</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>%</td>
<td>65.7</td>
<td>14.4</td>
<td>5.1</td>
<td>4.2</td>
<td>3.6</td>
<td>3.6</td>
<td>3.4</td>
</tr>
</tbody>
</table>

Table 8.1: The distribution of EVENT classes

Our guess that TIMEX3 has a sparse distribution in the fiction register has shown to be true, as it is illustrated in the table 8.2:

<table>
<thead>
<tr>
<th>Register</th>
<th>Fiction</th>
<th>News1</th>
<th>News5</th>
<th>News2</th>
<th>News6</th>
<th>News3</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>1</td>
<td>2.45</td>
<td>1.89</td>
<td>1.50</td>
<td>1.47</td>
<td>1.11</td>
</tr>
</tbody>
</table>

Table 8.2: The distribution of TIMEX3 on 100 words in fiction and news register

Also, the timex are more frequently underspecified, especially if we compare our results with those obtained in the news register (Setzer and Gaizauskas 2002). In TIMEX3 the underspecified temporal values, including the non-specific ones, are made explicit by setting the attribute temporalFunction with the value true and so a postprocessor will try to exactly determine them. The generic temporal expressions (I love December) cannot be assigned with a full-specified value. In the absence of an equivalent TIMEX3 attribute as the NON_SPECIFIC attribute from TIMEX2 standard, the postprocessor will try in vain to handle these generic expressions.

A useful TIMEX2 attribute, SET - used to mark expressions of sets of times (every week) - is already added to TIMEX3 in its last version (Sauri et al. 2004). The GRANULARITY and PERIODICITY TIMEX2 attributes were also imported to the present TimeML version in the TIMEX3 tag: the quant is a literal that quantifies over the expression of type SET and freq contains an integer and a time granularity that represents the frequency at which the temporal expression regularly reoccurs.

None of the proposed values for the functionInDocument attribute (which indicates the function of the TIMEX3 tag in providing a temporal anchor for other temporal expressions in the document) can be applied to a story from the literature register; so it would be necessary another predefined value to indicate, for example, the beginning of the story.

The SIGNAL tag is used to annotate sections of text, typically function words that indicate how temporal objects are to be interrelated. Syntactically, the signals are temporal prepositions, conjunctions and/or modifiers, negative expressions, modal verbs, prepositions signaling modality and special characters denoting ranges in temporal expressions.

The MAKEINSTANCE tag is based on the event annotation, indicating different instances or realizations of a given event. Every EVENT introduces at least one corresponding MAKEINSTANCE.

For those events that had more than one realization, we chose to represent the cardinality by a specific number; another option is to create as many MAKEINSTANCE tags as the number of realizations. Following the event annotation, we have automatically generated a MAKEINSTANCE tag for each event. If an event had more than one realization, we manually introduced the corresponding cardinality.

87
The proportion of tags, without MAKEINSTANCE and links is as in table 8.3.

<table>
<thead>
<tr>
<th></th>
<th>EVENT</th>
<th>TIMEX3</th>
<th>SIGNAL</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instances</td>
<td>333</td>
<td>27</td>
<td>171</td>
<td>531</td>
</tr>
<tr>
<td>%</td>
<td>62.7</td>
<td>5.1</td>
<td>32.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.3: The distribution of EVENT, TIMEX3 and SIGNAL tags

Link tags encode the various relations that exist between the temporal elements of a document. There are three types of link tags.

The TLINK represents the temporal relation between two temporal elements (event-event, event-timex); it establishes a link between the involved entities, stating what type of relation holds.

The SLINK is a subordination link for contexts that introduce event-event or event-signal relations involving negation, modality, evidentials and factives.

The ALINK is an aspectual link that indicates an aspectual connection between two events: the aspectual event and its argument event.

Annotating the set of all possible temporal relations is a difficult task mainly due to their high density. As it was proved in (Pustejovsky et al. 2002), the number of possible temporal relations is quadratic to the number of events and timex. We had to deal with this time-consuming task and we managed to annotate only a part of our corpus. The fragment contains 271 words and 14 sentences (counted from period to period), with 39 EVENT and 6 TIMEX3 tags. The distribution of manually annotated links is given in table 8.4.

<table>
<thead>
<tr>
<th></th>
<th>TLINK</th>
<th>ALINK</th>
<th>SLINK</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instances</td>
<td>36</td>
<td>3</td>
<td>17</td>
<td>56</td>
</tr>
<tr>
<td>%</td>
<td>64.3</td>
<td>5.4</td>
<td>30.3</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.4: The distribution of LINK types

The following is an example of a full temporal annotated sentence:

*Acum îşi dădea seama că tocmai din cauza acestui incident se hotărâse el brusc să vină acasă şi să-şii înceapă jurnalul tău astăzi.* (in Romanian) – It was, he now realized, because of this other incident that he had suddenly decided to come home and begin the diary today.

<TIMEX3 temporalFunction="true" tid="t152" type="TIME" value="PRESENT_REF"> Acum</TIMEX3> îşi <EVENT aspect="PROGRESSIVE" class="OCCURRENCE" eid="e153" tense="PAST"> dădea</EVENT> <MAKEINSTANCE eiid="e159" eid="e153" cardinality="1"/> seama <SIGNAL sid="e154" că</SIGNAL> tocmai din cauza acestui <EVENT aspect="NONE" class="OCCURRENCE" eid="e156" tense="NONE">
5 Merge between XML annotation standards

The annotation completed, we found out that some temporal expressions are denoted by coreferential expressions (yesterday – that day). It would not have been possible to tag them automatically on a document missing the NP-annotation. An anaphora resolution engine could detect the referential NPs. Knowing the time value for one NP, the coreferential one would be tagged as a timex with the same value.

In order to combine different levels of annotation we performed a "merge" operation between two documents conforming to different standards and which have the same hub document. Following (Cristea and Butnariu 2004) the annotation standards are organized as a DAG hierarchy. Each standard (node), with a unique symbol, inherits all features of all its parents, adding other features defined by any number of <tag> and <ref> labels. A <tag> label records a new XML element tag that has a name and a list of attributes. A semantic relation (dependency) between two annotation standards is recorded by a <ref> label. A standard A is superior to a standard B if and only if there is a path from B to the root of the hierarchy that passes through A. Informally a node A subsumes a node B in the hierarchy if B is a descendent of A that is more informative than A and/or defines more semantic constrains. The merged XML document contains the union of the annotation
tags of the two original documents.

6 Towards a temporal tagger

A visual inspection of the annotated material containing the word, NP, coreference and temporal level helped to distinguish a relevant set of tags and attributes which should be involved in an automatic tagging process. Following this investigation we proposed an algorithm for event tagging which is sensible to the Romanian particularities and local context. So far it handles only the verb-denoted events. Without taking into consideration any of the EVENT attributes, the algorithm is evaluated as in the table 8.5:

<table>
<thead>
<tr>
<th>Selected</th>
<th>True Positive</th>
<th>Target</th>
<th>precision</th>
<th>recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>232</td>
<td>215</td>
<td>226</td>
<td>92.6</td>
<td>95.1</td>
</tr>
</tbody>
</table>

Table 8.5: The evaluation for verb-denoted events

Using a list of verbs classified as type REPORTING, PERCEPTION, ASPECTUAL, I_ACTION, I_STATE or OCCURRENCE and morpho-sintactic information, we managed to automatically assign values for the class and tense attributes, as illustrated in table 8.6:

<table>
<thead>
<tr>
<th>Selected</th>
<th>True Positive</th>
<th>Target</th>
<th>precision</th>
<th>recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>232</td>
<td>79</td>
<td>178</td>
<td>34.9</td>
<td>44.4</td>
</tr>
</tbody>
</table>

Table 8.6: The evaluation for class and tense attributes of EVENTS

The information that we extract from the word-level annotation allowed us to automatically tag some of the signals. In order to obtain a higher performance we should look to the local context and define some specific Romanian collocations ("in timp ce" (RO) - while). The performance of our algorithm is given in the table 8.7:

<table>
<thead>
<tr>
<th>Selected</th>
<th>True Positive</th>
<th>Golden</th>
<th>precision</th>
<th>recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>111</td>
<td>171</td>
<td>100.0</td>
<td>64.91</td>
</tr>
</tbody>
</table>

Table 8.7: The evaluation for SIGNALS

Given a negative signal, a very simple algorithm allows to automatically tag with SLINK of type "NEGATIVE".

The above algorithms were implemented in Java.

7 Future work and possible applications

One of the first applications we thought of is a study of the grammatical category of the aspect of verbs in Romanian. Using the translation equivalents derived from the parallel
corpus (Tufiş and Barbu 2001) and the English version with the verbs annotated as events with tense and aspect, we will tag the corresponding Romanian verbs with the same aspect and tense. It could be a step toward defining the category of aspect in Romanian and a statistical evidence of the time mapping of the verbs between English and Romanian.

We intend to annotate also the English version of the corpus and using the TREQ-AL word alignment system (Tufiş and Barbu 2002) to automatically tag the Romanian version. The obtained results should show the weak points of the temporal aspects in Romanian and hence we would pinpoint some extensions of the community standards, needed in Romanian.\footnote{We are grateful to our reviewers for these suggestions.}

When dealing with temporal links, an interesting interrelation between them shows that temporally linked events indicate a continuous sequence of events in the narration. An interruption reveals another sequence that can be time-independent of the former one or nested in it; in discourse processing a break in temporal links can indicate a specific discourse phenomenon like the start/end of a flash-back. The events without a temporal anchor give background information or they can indicate a specific type of communication act (direct or indirect speech). In this way we intend to use all the temporal annotated data together with time ontologies in order to represent the temporal structure of the discourse and its possible relations with other discourse structures, such as, for example, Rhetorical Structure (Mann and Thompson 1987).

We obtained the present temporal annotation version by agreement, always discussing together the best solution to be adopted. We should double-annotate this version and compute the kappa coefficient of agreement (Carletta 1996) as a measure of the applicability of TimeML to other language\footnote{We are grateful to our reviewers for these suggestions.} and register than the standard was created for.

The initial aim of our work - developing a temporal tagger for Romanian - will be reached and further developed by testing and studying the tagger for different types of text registers, using time ontologies to obtain inferences about time and events.

8 Conclusions

The paper presents the hand-annotation process of the temporal information in a parallel corpus from the fiction register. We took the first steps towards creating a Romanian temporal tagger. As we got more into the problem, we found problems that we have solved or intend to tackle with in the future. Some interesting applications and further developments are also discussed.

Acknowledgements

The authors would like to thank professor dr. Dan Cristea and professor dr. Dan Tufiş for their helpful discussions, comments and continuous encouragements when the TIME didn’t work up for us.
References


Deduction of Numeral Grammars

Harald Hammarström
Chalmers University
harald2@cs.chalmers.se

Abstract.
We describe an unsupervised method to learn how numeral expressions (e.g. fifty-three, ninety-three etc.) are formed. This problem is seen as finding a minimal-size set of atomic forms and concatenation rules, and the method for abstracting powerful concatenation rules is through k-means clustering of the strings on the edit-distance. We evaluate the method on the number words for 1–101 in 90 diverse languages and discuss success rates and limitations.

1 Introduction

In this paper we shall describe experiments on automatic abstraction of natural language numeral systems. To be more precise, for a given list of the names for the numbers 1 to n in a language we get a grammar and a set of atoms. By grammar we simply mean a set of concatenation rules and atoms are those items which show no possibilities of regular decomposition. For the problem to be interesting, we also want the grammar to be as tight and complete as possible. For example, in English the numerals ≤ 100 are as follows:

one eleven twenty-one thirty-one ... seventy-one eighty-one ninety-one
two twelve twenty-two thirty-two ... seventy-two eighty-two ninety-two
three thirteen twenty-three thirty-three ... seventy-three eighty-three ninety-three
...

The atoms would be 1–12 + variants and the grammar would consist of rules like:

{thir|four|fif|six|seven|eigh|nine}teen
{twen|thir|four|fif|six|seven|eigh|nine}{ty-one|ty-two|...|ty-nine}

...
It’s obvious that such grammars are subsumed by context-free grammars where the atoms correspond to terminals, and since the sets of strings discussed in this paper are all finite, regular expressions can account for them too.

The grammar can be non-empty whenever there are regularities like in the English numeral system. The methods used aren’t specialized for numerals but work on any set of strings that has similar regular properties. However, numeral systems typically have strong and abundant rule-like features, yet at the same time irregularities and idiosyncrasies so characteristic of natural language. This makes them ideal for testing one’s methods and moreover, numeral systems are much more comparable across languages than almost any other subset of grammar.

We have done experiments for numbers 1–101 on about 90 languages from a numerals project centered around GF (Ranta 2004) at Chalmers University. The methods and results are discussed below.

2 Method

The problem can be formulated as picking out the optimal grammar from the giant set of combinatorially possible grammars. Instead of brute-force going through the whole search space we can heuristically deduce what at least a near-optimal grammar should be. This (tractable) computation is carried out along the following lines:

- Iterate until no more rules can be abstracted:
  - Generate a $k$-clustering for $k = 1, 2, 3, \ldots, i$ (for some suitable $i$)
  - Select the optimal clustering on basis of average rule size
  - Rule abstraction from clusters

Strings that have been abstracted into a rule in an earlier iteration can still be subject to further abstraction.

2.1 Clustering

The clustering phase uses a straightforward $k$-means classifier (Russell and Norvig 2003). The similarity measure is the edit-distance (Wagner and Fischer 1974) with settings that penalize different length strings more than same-length different strings. Some quick checks indicated that the most useful edit costs were 0.25 for substitution versus 1 for deletion.

All the strings are in our internal 8-bit ascii representation which is that of the language itself or a phonematic approximation of the orthography of the source. It is truer to language to use some kind of phonemic distance (rather than orthographic distance) as in e.g (Kondak 2002) but that would be overkill for this experiment.

Now, the crucial question is how many clusters one should divide into because, as shall be explained, the rule abstraction is done on basis of the cluster divisions. Table 9.1 has some examples of clustering outcomes for some $k$’s over English.

The classic clustering dilemma is to find “the right” $k$. Ideally one will want that the clusters are as tight and as disjoint as possible, but there’s no simple way to optimize for maximum tightness, disjointness or both for different $k$’s that does not trivially favour the case where each
<table>
<thead>
<tr>
<th>$k$</th>
<th>$x$</th>
<th>$n$</th>
<th>$w$</th>
<th>$k$</th>
<th>$x$</th>
<th>$n$</th>
<th>$w$</th>
<th>$k$</th>
<th>$x$</th>
<th>$n$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>26</td>
<td>73</td>
<td>11</td>
<td>58.4</td>
<td>51</td>
<td>74</td>
<td>14</td>
<td>44.3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>27</td>
<td>63</td>
<td>8</td>
<td>69.4</td>
<td>52</td>
<td>71</td>
<td>15</td>
<td>42.6</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>28</td>
<td>56</td>
<td>9</td>
<td>50.9</td>
<td>53</td>
<td>83</td>
<td>16</td>
<td>43.8</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>1</td>
<td>35.0</td>
<td>29</td>
<td>59</td>
<td>9</td>
<td>57.2</td>
<td>54</td>
<td>80</td>
<td>17</td>
<td>42.9</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>1</td>
<td>35.0</td>
<td>30</td>
<td>81</td>
<td>11</td>
<td>66.1</td>
<td>55</td>
<td>75</td>
<td>12</td>
<td>51.9</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>31</td>
<td>80</td>
<td>13</td>
<td>51.8</td>
<td>56</td>
<td>81</td>
<td>16</td>
<td>48.2</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>32</td>
<td>71</td>
<td>13</td>
<td>45.9</td>
<td>57</td>
<td>72</td>
<td>12</td>
<td>54.8</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>1</td>
<td>35.0</td>
<td>33</td>
<td>84</td>
<td>14</td>
<td>51.4</td>
<td>58</td>
<td>83</td>
<td>17</td>
<td>44.2</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>1</td>
<td>35.0</td>
<td>34</td>
<td>58</td>
<td>10</td>
<td>56.5</td>
<td>59</td>
<td>84</td>
<td>18</td>
<td>42.7</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>1</td>
<td>35.0</td>
<td>35</td>
<td>87</td>
<td>13</td>
<td>60.7</td>
<td>60</td>
<td>74</td>
<td>20</td>
<td>34.2</td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>3</td>
<td>67.7</td>
<td>36</td>
<td>81</td>
<td>13</td>
<td>55.8</td>
<td>61</td>
<td>80</td>
<td>13</td>
<td>57.8</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>2</td>
<td>61.5</td>
<td>37</td>
<td>79</td>
<td>14</td>
<td>51.5</td>
<td>62</td>
<td>81</td>
<td>16</td>
<td>45.8</td>
</tr>
<tr>
<td>13</td>
<td>25</td>
<td>4</td>
<td>57.3</td>
<td>38</td>
<td>80</td>
<td>12</td>
<td>60.5</td>
<td>63</td>
<td>73</td>
<td>16</td>
<td>40.3</td>
</tr>
<tr>
<td>14</td>
<td>27</td>
<td>4</td>
<td>60.8</td>
<td>39</td>
<td>78</td>
<td>13</td>
<td>50.6</td>
<td>64</td>
<td>72</td>
<td>15</td>
<td>45.8</td>
</tr>
<tr>
<td>15</td>
<td>23</td>
<td>3</td>
<td>46.3</td>
<td>40</td>
<td>87</td>
<td>13</td>
<td>60.2</td>
<td>65</td>
<td>65</td>
<td>15</td>
<td>39.9</td>
</tr>
<tr>
<td>16</td>
<td>39</td>
<td>5</td>
<td>74.2</td>
<td>41</td>
<td>83</td>
<td>16</td>
<td>47.1</td>
<td>66</td>
<td>65</td>
<td>15</td>
<td>38.9</td>
</tr>
<tr>
<td>17</td>
<td>55</td>
<td>7</td>
<td>75.9</td>
<td>42</td>
<td>82</td>
<td>15</td>
<td>49.5</td>
<td>67</td>
<td>65</td>
<td>15</td>
<td>41.3</td>
</tr>
<tr>
<td>18</td>
<td>47</td>
<td>6</td>
<td>65.8</td>
<td>43</td>
<td>81</td>
<td>16</td>
<td>45.9</td>
<td>68</td>
<td>70</td>
<td>15</td>
<td>44.1</td>
</tr>
<tr>
<td>19</td>
<td>63</td>
<td>8</td>
<td>64.4</td>
<td>44</td>
<td>87</td>
<td>16</td>
<td>49.9</td>
<td>69</td>
<td>78</td>
<td>18</td>
<td>40.9</td>
</tr>
<tr>
<td>20</td>
<td>63</td>
<td>8</td>
<td>70.4</td>
<td>45</td>
<td>83</td>
<td>16</td>
<td>46.9</td>
<td>70</td>
<td>65</td>
<td>15</td>
<td>39.9</td>
</tr>
<tr>
<td>21</td>
<td>55</td>
<td>8</td>
<td>66.6</td>
<td>46</td>
<td>75</td>
<td>15</td>
<td>45.2</td>
<td>71</td>
<td>50</td>
<td>12</td>
<td>34.6</td>
</tr>
<tr>
<td>22</td>
<td>63</td>
<td>8</td>
<td>69.4</td>
<td>47</td>
<td>87</td>
<td>15</td>
<td>52.9</td>
<td>72</td>
<td>65</td>
<td>14</td>
<td>44.9</td>
</tr>
<tr>
<td>23</td>
<td>63</td>
<td>9</td>
<td>60.1</td>
<td>48</td>
<td>75</td>
<td>18</td>
<td>38.4</td>
<td>73</td>
<td>43</td>
<td>9</td>
<td>40.3</td>
</tr>
<tr>
<td>24</td>
<td>66</td>
<td>11</td>
<td>53.5</td>
<td>49</td>
<td>84</td>
<td>18</td>
<td>42.8</td>
<td>74</td>
<td>25</td>
<td>7</td>
<td>38.4</td>
</tr>
<tr>
<td>25</td>
<td>66</td>
<td>11</td>
<td>54.6</td>
<td>50</td>
<td>87</td>
<td>17</td>
<td>46.9</td>
<td>75</td>
<td>79</td>
<td>13</td>
<td>54.9</td>
</tr>
</tbody>
</table>

Table 9.1: $x$ is the number of strings affected in rules induced by the clustering. $n$ is the number of induced rules and $w$ their average size. There is a random factor in initializing but the result vary only meagerly from run to run.
object is a single cluster. Although there are some interesting general approaches to this problem around (Ghosh 2003), they require commitment to supervision or models that we might not want to require on our sets of strings.

So instead we use the average induced rule size as the optimization goal. By rule size we mean how many cases a rule applies to plus the length of the generalization. So e.g a rule \{thir\|four\|fif\|six\|seven\|eigh\|nine\}teen gets size 7 * 4 = 28. This is unsupervised and entertains the intended intuition that we want strong rules in the end.

From the average rule sizes in table 9.1 we see that the best cluster sizes for English seem to be around 10 or 20, reflecting the fact that English is base-10 and we get general rules if we cluster into teens, twenties, thirties, ..., nineties plus one cluster of units or ten different clusters as in one for each unit (for they have nothing in common except not belonging to the other classes).

2.2 Rule Abstraction

For each cluster, the best generalization is calculated by matching all pairs of cluster members and greedily choosing the longest most applicable one. Most applicable means that the highest number of members of the cluster satisfy it. A generalization of a set of strings is a common prefix, suffix or circumfix (other generalizations, such as infixes (Yu 2003), are possible in natural language but did not seem to occur in our sample). For example, if a cluster contains \{twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety, ninety-four\}, the most applicable generalization is a suffix -ty which is common to all members except one (= 8 which is more than e.g 2 for -rty, f-ty, ninety-) and it is longer than -y, which would also apply to the same eight members.

To filter out a large class of unwanted rules we also postulate that a rule must:

- be built on more than one example
- generalize more than one character

The last criterion is because a lot of one char suffixes and prefixes occur by chance.

A few more comments are in order:

- Once a rule has been extracted from a cluster, any other cases outside the cluster that match the rule are incorporated.
- All rules are disjoint in the sense that the same string is never part of two parallel rules.
- Maximally only one rule is abstracted from a cluster at each step, since rules are never revoked and thus it’s very important that the rules are sound and maximal.

2.3 Related Methods

The method described is similar to induction algorithms by de Marcken (de Marcken 1996) and (Baroni 2000) in that it sees rules as compression – a good rule is one which makes description shorter. And like Goldsmith’s Linguistica (Goldsmith 2001) it relies on the existence of affixes that are frequent, as food for generalizing. The difference in our case is the use of clustering, where potentially variants like three and thir have a better chance of being recognized. It is a turnoff,
then, to note that in the experiments aimed at in this paper we have not set out to exploit this property. It is also beyond the scope of this paper to look empirically at how clustering predicts affix divisions compared to Goldsmith’s heuristic or Johnson and Martin’s (Johnson and Martin 2003) hub detection.

3 Experiments

3.1 Evaluation Metric

Each language has been annotated with a gold standard of the components of its numeral system. The components are the atoms and their allomorphs, other morphemes and orthographic links (e.g and, -, spaces etc), “irregular extensions”, and formation rules. Irregular extensions are expressions that are not monomorphemic but occur in only one or two cases – too few to merit a rule. To give the reader an idea, the raw annotation for English is given below:

```plaintext
#English
b = 10
a = {}
a[1] = ‘one’
a[2] = ‘two’
a[3] = ‘three’
a[4] = ‘four’
a[5] = ‘five’
a[6] = ‘six’
a[7] = ‘seven’
a[8] = ‘eight’
a[9] = ‘nine’
a[10] = ‘ten’
a[100] = ‘hundred’
a[1000] = ‘thousand’
irr_ext = {}
irr_ext[12] = (2, ‘twelve’, ‘leaves’)

v = {}
v[2] = [Variant(‘twen’, {PART_ROLE: ‘small’})]
v[3] = [Variant(‘thir’, {PART_ROLE: ‘small’})]
v[5] = [Variant(‘fif’, {PART_ROLE: ‘small’})]
v[8] = [Variant(‘eigh’, {PART_ROLE: ‘small’})]
v[10] = [Variant(‘ty’, {VARIANT: ‘ten’}),
            Variant(‘teen’, {VARIANT: ‘teen’})]
```

98
AND = Element('' and ′′, AND)

cr = {}
cr[13] = Rule([[MOD, REM, Compound(ADD, [SMALL, BIG], ′′teen′′)])
cr[20] = Rule([[MOD, Q, Compound(MUL, [SMALL, BIG], ′′ten′′)],
                             (RES, REM, Compound(ADD, [BIG, SMALL]))])
cr[100] = Rule([[MOD, Q, Compound(MUL, [SMALL, SPACE, BIG])]),
                             (RES, REM, Compound(ADD, [BIG, SPACE, AND, SPACE, SMALL]]))],
                     None,
                     OVERRIDE)
cr[1000] = Rule([[MOD, Q, Compound(MUL, [SMALL, SPACE, BIG])]),
                             (RES, REM, Compound(ADD, [BIG, SPACE, SMALL]]))],
                     None,
                     OVERRIDE)

The reader need not worry about the syntax and semantics of the annotation scheme, but
should assume it allows retrieval of the details of each numeral system in the sample in a uniform
way. Since the database was originally designed for typological comparison, it happens annotated
in too high detail than what is needed, and we shall now proceed to the actual benchmark.

The irregular extensions, atoms and allomorphs of the gold standard are treated as atoms,
just as orthographic links and non-numeric morphemes are incorporated into the formation rules.
Now, think of a rule as a set of strings, namely those strings it generates when applied to its
domain. The numeral system as a whole, a set of atoms and a set of rules, is now a set of strings
(of atoms) and set of sets of strings. There’s no reason to handle the atoms separately, so a
numeral system can be treated as a set of sets of strings.

So, for clarity, denote an induced numeral system \( I = \{r_1, \ldots, r_m\} \) vs. the gold standard
\( G = \{R_1, \ldots, R_n\} \). Of course \( \bigcup I = \bigcup G = 1-101 \) in some language.

A rule \( r \) is a set of strings, \( s_1, \ldots, s_k \), and theoretically these may belong to as many as
\( k \) different “real” rules according to the gold standard, but (hopefully), they only belong to \( i \)
different real rules. In these terms, the precision of a rule is \( \frac{k-i}{k} \). So e.g. if they all belong in
one rule in the gold standard too, then the precision is a perfect 1 and vice versa. Formally:

\[
\text{precision}(r, G) = |\{R_x \in G | \exists s \in r, R_x\}|
\]  \hspace{1cm} (9.1)

Now, define the accuracy as:

\[
\text{Accuracy} = \frac{1}{|I|} \sum_{r \in I} \text{precision}(r, G) + \frac{1}{|G|} \sum_{r \in G} \text{precision}(r, I) \]  \hspace{1cm} (9.2)

Note that this accuracy-metric only measures which strings belong to which rules, and we
leave out details as to whether the actual formation/split in the rule is correct.

Again, we take a fictitious example of a fraction of English:

3.2 Results

We give a summary in table 9.2 of the algorithm’s performance on the 90 languages in the test
set. Kwaza (van der Voort 2000) and Kayardild (Evans 1995) have been excluded since they do
not really have numerals over five.

99
<table>
<thead>
<tr>
<th>I</th>
<th>G</th>
<th>precision($r_1, G$)</th>
<th>precision($r_2, I$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1 = {\text{one, seventeen}}$</td>
<td>$R_2 = {\text{one, two, \ldots , ten}}$</td>
<td>$\frac{2}{2}$</td>
<td>$\frac{7}{10}$</td>
</tr>
<tr>
<td>$r_2 = {\text{nineteen}}$</td>
<td>$R_3 = {\text{thirteen, \ldots , nineteen}}$</td>
<td>$1$</td>
<td>$\frac{7}{7}$</td>
</tr>
<tr>
<td>$r_3 = {\text{two, \ldots , ten}}$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$r_4 = \ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
</tr>
</tbody>
</table>

We will give examples to illustrate the ratings. For German we get:

\[
\{\text{drei} \mid \text{zw\text{"{u}l}}\} \text{ neun | acht | f\text{"{u}n}f | sechs | ein | vier | sieben} \text{ und } \{\text{neun | zw\text{"{u}l}}\} \text{ f\text{"{u}n}f | vier | sechs | acht | zig}\}
\]

\[
\{\text{drei} \mid \text{zw\text{"{u}l}}\} \text{ neun | acht | f\text{"{u}n}f | neun | sechs | sieb\text{"{e}n | vier}\} \text{ und } \text{ dreissig}
\]

\[
\{\text{drei} \mid \text{zw\text{"{u}l}}\} \text{ neun | acht | f\text{"{u}n}f | neun | sechs | sieb\text{"{e}n | vier}\} \text{ zw\text{"{u}l}}
\]

hundert{\{\text{eins}\}}

And the atoms eins, \ldots , neun, dreissig, elf, zw\text{"{u}l} which is pretty much optimal. One should think about whether dressig should be a separate case or part of a larger -\text{ig} rule, but the algorithm favoured -\text{ig} abstraction ($7^*3$ vs. $8^*2$). The situation is not uncommon, e.g many Romance languages have 30 with a suffix slightly different from 40, \ldots , 90 as in treinta, quaranta, cinquanta, seixanta, setanta, vuitanta, noranta (in Catalan (Maria Mas and Vergés 2000)).

The German case illustrates another prominent feature, namely grouping of words like zwei and drei, that are short and have a common sequence of length 2. This is responsible for unfortunate abstractions in e.g Guahibo (Queixalos 1986), Kabardian (COLARUSSO 1992), Malay (DODDS 1977). Classical Arabic (HAYWOOD and NAHMAD 1962) and Pashto (PENZ 1955) are even more vulnerable as they are implemented in romanizations of a short-vowel-less Arabic script, and so generally have shorter string and chance resemblances become harder to distinguish. These unwanted generalizations could easily be made illegal by imposing restrictions also on length 2 abstractions. But we don’t want to do this on the general level since length 2 abstractions are also often wanted and unproblematic (e.g Biblical Hebrew (NYBERG 1972), also short-vowel-less, abstracts fine).

The other insuperable problem is tone-variant Tibetan (HERBERT BRUCE 1978) or highly inflecting languages like Polish and Old Church Slavonic (NANDRIQ 1959). Similarly, the closely related sandhi-intensive Dravidian languages Irula (PERIALWAR 1978), Kodagu (BALAKRISHNAN 1977) and Tamil yield many classes of abstractions with three or four members. We see no simple remedy for treating these cases in general.

There are many languages that use base-20 in addition to a base 5 or 10, especially for numbers $\leq 100$. E.g Nootka (FOLAN 1986), Breton (TRÉPOS 1980), Maybrat (PHILOMENA HEDWIG 1999), Danish (BROUNDUM-NIELSEN 1974), Totonac (MACKAY 1999), Burushaski (GRUNE 1998). These have the desired regularities but can be harder for the clustering to single out. The base-12 language Nungu (MATHES 1917) abstracts so well that one suspects that the source is idealized.

An interesting observation is that the algorithm frequently proceeds by finding “horizontal” abstractions, that is, it often groups twenty-one, thirty-one, \ldots , ninety-one rather than twenty-one, twenty-two, \ldots , twenty-nine. If this behaviour should be considered unwanted it’s easy to penalize for for a particular language, but there’s really nothing wrong with
<table>
<thead>
<tr>
<th>Language</th>
<th>Accuracy</th>
<th>Language</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>af tunni</td>
<td>0.62</td>
<td>albanian</td>
<td>0.90</td>
</tr>
<tr>
<td>amharic</td>
<td>0.95</td>
<td>basque</td>
<td>0.82</td>
</tr>
<tr>
<td>bearlake slave</td>
<td>0.51</td>
<td>biblical hebrew</td>
<td>0.86</td>
</tr>
<tr>
<td>breton</td>
<td>0.85</td>
<td>bulgarian</td>
<td>0.70</td>
</tr>
<tr>
<td>catalan</td>
<td>0.91</td>
<td>classical arabic</td>
<td>0.22</td>
</tr>
<tr>
<td>classical greek</td>
<td>0.92</td>
<td>croatian</td>
<td>0.77</td>
</tr>
<tr>
<td>czech</td>
<td>0.40</td>
<td>dagur</td>
<td>0.97</td>
</tr>
<tr>
<td>danish</td>
<td>0.92</td>
<td>dutch</td>
<td>0.87</td>
</tr>
<tr>
<td>english</td>
<td>0.92</td>
<td>finnish</td>
<td>0.55</td>
</tr>
<tr>
<td>french</td>
<td>0.47</td>
<td>fulfulde</td>
<td>0.42</td>
</tr>
<tr>
<td>ge'ez</td>
<td>0.95</td>
<td>german</td>
<td>0.88</td>
</tr>
<tr>
<td>guahibo</td>
<td>0.60</td>
<td>guarani</td>
<td>0.60</td>
</tr>
<tr>
<td>hindi</td>
<td>0.49</td>
<td>hungarian</td>
<td>0.46</td>
</tr>
<tr>
<td>icelandic</td>
<td>0.88</td>
<td>irish</td>
<td>0.31</td>
</tr>
<tr>
<td>irula</td>
<td>0.35</td>
<td>italian</td>
<td>0.86</td>
</tr>
<tr>
<td>japanese</td>
<td>0.97</td>
<td>kabardian</td>
<td>0.26</td>
</tr>
<tr>
<td>kambera</td>
<td>0.66</td>
<td>kawaiisu</td>
<td>0.48</td>
</tr>
<tr>
<td>khmer</td>
<td>0.97</td>
<td>khowar</td>
<td>0.67</td>
</tr>
<tr>
<td>kodagu</td>
<td>0.39</td>
<td>kolyma yukaghir</td>
<td>0.36</td>
</tr>
<tr>
<td>korean</td>
<td>0.72</td>
<td>kulung</td>
<td>0.51</td>
</tr>
<tr>
<td>kwami</td>
<td>0.88</td>
<td>lalo</td>
<td>0.97</td>
</tr>
<tr>
<td>lamani</td>
<td>0.97</td>
<td>latvian</td>
<td>0.92</td>
</tr>
<tr>
<td>lithuanian</td>
<td>0.92</td>
<td>lotuxo</td>
<td>0.97</td>
</tr>
<tr>
<td>maale</td>
<td>0.97</td>
<td>malay</td>
<td>0.52</td>
</tr>
<tr>
<td>maltese</td>
<td>0.52</td>
<td>mapuche</td>
<td>0.97</td>
</tr>
<tr>
<td>margi</td>
<td>0.66</td>
<td>maybrat</td>
<td>0.45</td>
</tr>
<tr>
<td>miya</td>
<td>0.88</td>
<td>modern greek</td>
<td>0.88</td>
</tr>
<tr>
<td>mongolian</td>
<td>0.77</td>
<td>nenets</td>
<td>0.97</td>
</tr>
<tr>
<td>nungu</td>
<td>1.00</td>
<td>nootka</td>
<td>0.66</td>
</tr>
<tr>
<td>old church slavonic</td>
<td>0.43</td>
<td>oromo</td>
<td>0.97</td>
</tr>
<tr>
<td>pashto</td>
<td>0.39</td>
<td>polish</td>
<td>0.35</td>
</tr>
<tr>
<td>portuguese</td>
<td>0.92</td>
<td>quechua</td>
<td>0.97</td>
</tr>
<tr>
<td>romanian</td>
<td>0.92</td>
<td>russian</td>
<td>0.51</td>
</tr>
<tr>
<td>samoan</td>
<td>0.97</td>
<td>sango</td>
<td>0.97</td>
</tr>
<tr>
<td>sanskrit</td>
<td>0.77</td>
<td>slovak</td>
<td>0.51</td>
</tr>
<tr>
<td>sorani</td>
<td>0.77</td>
<td>spanish</td>
<td>0.92</td>
</tr>
<tr>
<td>stieng</td>
<td>0.97</td>
<td>supyire</td>
<td>0.67</td>
</tr>
<tr>
<td>swahili</td>
<td>0.97</td>
<td>swedish</td>
<td>0.92</td>
</tr>
<tr>
<td>swiss french</td>
<td>0.88</td>
<td>tamil</td>
<td>0.39</td>
</tr>
<tr>
<td>tibetan</td>
<td>0.35</td>
<td>totonac</td>
<td>0.43</td>
</tr>
<tr>
<td>tuda-daza</td>
<td>0.97</td>
<td>tukang besi</td>
<td>0.97</td>
</tr>
<tr>
<td>turkish</td>
<td>0.97</td>
<td>votic</td>
<td>0.67</td>
</tr>
<tr>
<td>welsh</td>
<td>0.51</td>
<td>yasin burushaski</td>
<td>0.67</td>
</tr>
<tr>
<td>yucatec</td>
<td>0.57</td>
<td>zaiwa</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Table 9.2: Language and grammar deduction accuracy
4 Discussion

We have shown that a fairly simple hybrid unsupervised/natural algorithm can abstract much of the syntax/morphology of number words \(\leq 100\) across languages. Much higher accuracy is of course achievable if one drops generality, and we have data to add supervised learning stages, but the generality is what makes the method interesting.

The method works well also on sets of strings that are not numeral systems and on incompletely specified numeral systems – for an input of 50 randomly chosen numbers in 1..100 in English the output is similar.

However, the main issue for wider use of the method is scalability. Suppose we want to cluster a lot more than 101 strings. The bottleneck part of the algorithm is to compute the distance between any two strings, which is \(O(n^2 m^2)\) for \(n\) strings of length \(m\). The fact that there no such thing as the mean for a set of strings (as there is for vectors) also adds to the time consumption in practice in the \(k\)-means computation. Intelligent choosing for which \(k\)’s to try clustering is not a time-complexity issue. We presume that the best option for an asymptotic speed-up, still keeping the general architecture, is to replace the edit-distance with a heuristic linear-time distance measure. However, we also expect accuracy to decrease with larger input sets if the input data is “incomplete”.

With only a few further assumptions, there is a clear sense in which the algorithm can say which numeral systems are more or less complicated, that is which that have most irregularities and varied formation rules. This would be hard to achieve without computers and unsound if abstraction is supervised.

Acknowledgements

We wish thank the consulted libraries of Stockholm, Uppsala and Gothenburg and their xerox-machines. A huge debt of gratitude is to Dr. Aarne Ranta and the VINNOVA project for financial support.
References

Studies 1, 83–93.
The Athlone Press, University of London.
Number 2 in Program in Oriental Languages Publications Series B. American Council of
Learned Societies, Washington, D.C.
65–80. Etymology hints are in the introduction (pp. 5-20) of the same volume.
Tokyo.
the Association for Computing Machinery 21(1), 168–173.
California, Berkeley.
On some aspects of machine translation among related languages

PETR HOMOLA
Institute of Formal and Applied Linguistic, Charles Universtity, Prague, Czech Republic
homola@ufal.mff.cuni.cz

ABSTRACT.
As previous MT projects at our department have shown that we are not able to develop a good syntactic parser for Czech at the moment, an experimental ‘parser-less’ MT system Česíko from Czech to closely related languages has been implemented. We have crossed the boundary of the Slavonic language family and extended this system to the pair Czech-Lithuanian (a Baltic language). Main components of the extended system are shallow syntactic parser and partial transfer. The paper summarizes our findings based on the experiments with shallow MT and describes some aspects of machine translation among related languages, especially what types of similarity can be exploited to simplify the parser and transfer. We explain by means of examples what problems may occur in the transfer phase and how we solve them.

1 Introduction
Machine translation (MT) has a long tradition at our department. Two MT systems have been developed in the eighties: The system APAC3.2 (Kirschner 1987) translates technical manuals from English to Czech, whereas RUSLAN (Oliva 1989) translates software manuals from Czech to Russian. In both systems, a rule-based syntactic parser has been used. Problems related with the parser for Czech initiated a couple of questions; the most important one was whether it is necessary to parse whole sentences, in other words, whether we need one complex syntactic structure for every sentence. This idea was at the beginning of the system Česíko (Hajič et al. 2000), a ‘parser-less’ MT system for closely related languages. A closer look at morphological and syntactic differences between Czech and Polish (Dębowski et al. 2002) shows that the translation quality could be improved by a partial transfer that would focus on problematic phenomena (cf. (Homola 2002) for Russian). Thus we have tried to extend the system to an even less related target language and adapted Česíko to Lithuanian, a Baltic language (Homola and Rimkutė 2003). A simple shallow syntactic parser has been developed and a few transfer rules used; the translation quality increased significantly. This fact was a strong impulse to analyze the most problematic phenomena in more detail in order to improve this approach.
In section 2, we list the levels of language similarity and explain how they may affect the quality of MT. In section 3, we focus on types of language similarity. In section 4, we describe the architecture and components of Česíko. In section 5, we explain what dependencies are processed by the shallow parser. In section 6, we go into detail about the structure of bilingual glossaries and their entries. Syntactic transfer rules are presented and explained by means of examples in section 7. Finally, we describe how we evaluated the results in section 8.

2 Levels of language similarity

Natural languages are grouped in language families. Usually, languages from the same language family are more similar than languages from different language families. Although the division of languages into families is not a perfect criterion for classification, it provides a raw hierarchy. We distinguish the following levels of proximity of related languages:

Very closely related languages (e.g., Czech and Slovak or Upper Sorbian) are very similar in morphology, syntax and vocabulary. Semantic ambiguities are rare. We suppose that no syntactic parser is needed in MT systems for these languages.

Closely related languages (e.g., Czech and Polish or Russian) are similar in morphology and widely in vocabulary, although some semantic ambiguities occur. Syntactic constructions are not fully compatible, e.g., counterparts of analytic constructions are synthetic and vice versa or different lexems are used. Transfer (at least partial) is needed to perform MT among these languages.

Related languages (e.g., Czech and Lithuanian or Latvian) are not as similar as closely related languages, although there are still many similarities (because of common origin and/or influencing each other strongly). The morphological system is similar and there are many one-to-one correspondences in the vocabulary. Although the syntax is widely the same, there are differences (as in the previous case) and, moreover, some constructions that do not have direct counterparts in the other language.

3 Types of language similarity

We distinguish four types of similarity: typological, lexical, morphological and syntactic. We believe that the first type is the most important one. For example, both Czech and Lithuanian are languages with rich inflection and an extremely free word order, thus we can suppose that it will not be necessary to change word order at the level of actants. The shallow approach would definitely be unsuccessful for translation between languages of different typology.¹

The lexical similarity does not mean that the vocabulary has to have the same origin, i.e., that words have to be derived from the same (proto-)stem. What is important for shallow MT (and for MT in general), is the semantic correspondence (preferably one-to-one relation).

¹It would be probably impossible to translate, e.g., the Hungarian one-word-sentence Énekelhetetlek “I may make you sing” (Alberti and Kleiber 2004) to a language without agglutination by means of shallow NLP.

106
Similar morphological systems simplify the transfer. For example, Slavonic languages (except of Bulgarian and Macedonian) have 6–7 cases. The case system of East Baltic languages is very similar, although it has been reduced formally in Latvian (Dini 2000). (Ambrazas 1996) gives seven cases for Lithuanian, but there are in fact at least eight cases (or ten cases but only eight of them are productive (Erika Rimkutė, personal communication)). Nevertheless the case systems of Slavonic and East Baltic languages are very similar which makes the shallow MT approach possible. Significant differences occur only in the verbal system, East Baltic languages have a huge amount of participles, half-participles and other transgressive forms that have no direct counterpart in Czech.²

Syntactic constructions have usually the same structure. There are often two equal possibilities in the other language. For example, a condition can be expressed by subjunctive or imperative in Russian, as in the example from (Panevová 1980):

(1)  Pridi na čas rašče, ...
    come-IMP,SG he-NOM on hour-ACC sooner
    “If he would have come one hour sooner, ...”

In Czech, only one form is possible, namely subjunctive. This particular case of syntactic ambiguity is no problem for MT from Czech, as the syntactically similar construction can be used.³

A well known problem is the inherent semantic ambiguity of the transgressive. We do not analyze this vague half-sentential construction although the translation may be incorrect in some cases due to wrong congruence, e.g., Czech Odešli zakončitše práci “They-MASC left after having finished the work” → Lithuanian Išėjo pabaigė darbą, but Odešly zakončitše práci “They-FEM left after having finished the work” → Išėjo pabaiðgusio darbą; Polish Rozmawiano pijac herbate “It has been talked while drinking tea” → Czech pij¼ ãaj (transgressive cannot be used with general subject in Czech (Jarmila Panevová, personal communication), but the sentence would be understandable).

4 Shallow MT system Česlíko

The MT system Česlíko is an experimental system for automatic translation from Czech to Balto-Slavonic languages.⁴ Since both the source and target languages of the system Česlíko are (closely) related, we suppose that it is possible to simplify the translation process by omitting the

²A special problem are various suffixes in Polish. Although the agglutination itself can be solved by the tagger, there are cases of ambiguity that are hard or even impossible to resolve, e.g., in miadem (“crushed material”-INS or “I had”), both the stem (miad) and the suffix (-em) are ambiguous (Dębowski 2004); or, the morph -ś is inherently ambiguous, e.g., coś powiedział means “he said something” or “what did you say” etc.

³In many cases, a literally translated syntactic construction is stylistically unusual but still grammatical or at least understandable, e.g., Polish Pamiętam go pijanego/pijanym “I remember him drunken” (accusative or instrumentative can be used) vs. Lithuanian Atsil puosimenu girta (the adjective in a nominal predicate is usually congruent with the subject in case, but the instrumental form would be understood as well).

⁴At the moment, we have resources for the following target languages: Slovak, Polish and Lithuanian.
full syntactico-semantic analysis and transfer and replace these components by a partial (shallow) syntactic analysis and a lemma-to-lemma or term-to-term transfer.

The basic version of the system uses the following components:

**Morphological analyzer for Czech** Czech is a language with rich inflection, a word has usually many different endings that express case, number, person etc. It is necessary to assign a lemma and morphological tags (a tagset) to every word form.

**Morphological disambiguation** Morphological analysis assigns often more lemmas or tagsets to word forms, therefore it would be impossible to use a shallow MT method. The morphologically analyzed text has to be disambiguated first.

**Domain-related bilingual glossaries** These glossaries are used first and provide not only a lemma-to-lemma, but preferably a term-to-term translation. If there are more glossaries, the most specific ones are applied first.

**General dictionaries** These dictionaries are used to translate lemmas that were not found in domain-related glossaries.

**Morphological synthesis of the target language** This final phase generates word forms in the target language.

The extended version of the systems includes a shallow parser and transfer component. Because shallow processing takes into account only dependencies that occur at the lowest syntactic level (unlike full-fledged parsing), the morphologically processed input of the syntactic component of the system must be disambiguated, i.e., only one lemma and one tagset can be delivered for every word form. We satisfy this requirement by a statistical tagger, which is available for Czech (Hajič 2001). This solution is a bit problematic because the output of the tagger contains errors, but there is no possibility to get better results at the moment.

The shallow parser for Czech is based on LFG (Bresnan 2001). As the syntactic analysis is partial, we do not use the whole LFG framework. For example, we leave out the completeness and coherence conditions and anaphoric binding. f-structures are processed by the transfer component whose main task is to convert all lexical entries. The conversion involves the translation of lemmas (basic word forms) and modification of morphological tags, if necessary. It is obvious that the change of morphological properties can break the agreement within a constituent, if the head of the constituent governs attributes that have to agree with the governor, as in ((2)).

\[(2) \quad \text{velká-FEM} \quad \text{banka-FEM}\]
\[\quad \ast \text{didelé-FEM} \quad \text{bankas-MASC}\]
\[\quad \text{“big bank”}\]
\[\quad \text{(didelis bankas)}\]

---

5 (Žákčová 2002) has shown that it is not possible to disambiguate Czech texts morphologically merely by means of shallow parsing.

6 The tagger is the only statistical component in the system, since we have no resources to train statistical methods on.

7 Correct translation is given in brackets.
Thus another task of the transfer component is to modify morphological categories of dependents of the translated item to preserve agreement. The same concerns the agreement between prepositions and their objects, as in ((3)).

(3)  
do-GEN Prahy-GEN  
*i-ACC Prahoš-GEN  
“to Prague”  
(i Praha)

Converted f-structures are linearized. In general, the word order of the source sentence is preserved, but it is necessary to change the order of constituents in some cases (mostly in noun phrases), as in ((4)).

(4)  
bratr1 otce2  
těv02 broliš1  
“father’s brother”

Finally, the linearized sentences are synthesized morphologically.\(^8\)

Let us have a look at example ((5)) that shows partial feature structures.\(^9\)

(5)  
Před domem stoji zelený autobus.  
in-front-of house-ins stays-3sg green-NOM bus-NOM  
“There is a green bus in front of the house.”

\[
\begin{align*}
[ & \text{PRED ‘před(↑ OBJ)’} ] \\
[ & \text{OBJ [ PRED ‘dům’ ] } ] \\
[ & \text{PRED ‘stát’ } ] \\
[ & \text{PRED ‘autobus’} \\
[ & \text{ADJ { [ PRED ‘zelený’ ] } } ]
\end{align*}
\]

5 Syntactic underspecification

In shallow syntactic analysis, only some dependencies in the sentence are analyzed, mostly those in smaller constituents, such as NP or PP. A closer look at our language pairs shows that most differences occur at the surface level.

Let us have a look at example ((6)). Dependencies analyzed by the shallow parser are expressed by the solid line, not recognized ‘deeper’ dependencies by the dotted line.

(6)  
Iš tolo matomas namas miško pakraštyje.  
from far visible-MASC,SG,NOM house-NOM forest-GEN border-LOC  
“a/the from far visible house at the border of the wood”

\(^8\)For Lithuanian, we use a modified version of the software described in (Zinkevičius 2001).

\(^9\)We do not give all attributes in the feature structures for the sake of simplicity.
In the Czech source sentence, the word order of constituents is very similar as in (6). The only difference is in the translation of *iš tolo* (in Czech *zdáleka*) and the word order in the NP *miško pakraštyje* (genitive attributes follow the governing noun in Czech).

Omitted dependencies (dotted lines in (6)) can be told to be syntactically underspecified. The syntactic structure of a sentence built by a shallow parser is incomplete and could be optionally extended by another module.\(^\text{10}\)

A serious problem for NLP of languages with rich inflection represents the non-projectivity. In these languages, non-projective sentences are still understandable because the word order (at the level of actants) has almost no grammatical meaning. For example, more than one third of sentences are non-projective in Czech.\(^\text{11}\) In our approach, we do not consider non-projectivity, because both Czech and Lithuanian use the same type of non-projective dependencies and because the simple parser is based on LFG.\(^\text{12}\)

For example, the syntactic structure of (8), a non-projective Lithuanian sentence, is the same as of its Czech translation.

\begin{equation}
\text{(8) Ša knygą pradėsiu skaityti rytoj.}
\end{equation}

\begin{flushright}
*this-FEM,ACC book-ACC I-will-start read-INF tomorrow* \\
"I will start to read this book tomorrow."
\end{flushright}

\(^{10}\) (Zeman 2001) describes experiments with a regular grammar and statistical parsers.

\(^{11}\) Various types of non-projectivity in Czech are described and formalized in (Kuboň 2001).

\(^{12}\) Non-projective sentences would be a problem, if the parser would be full-fledged, as there are examples of (Czech) sentences that cannot be parsed by LFG because of the insufficient formal power of this formalism (Valia Kordoni, personal communication). Nevertheless LFG is sufficient for our shallow approach.
6 Bilingual glossaries

The glossaries are a sub-component of the transfer component. Their task is to provide lexical translation of constituents analyzed by the shallow parser. We will describe this component by means of examples from the MT pair Czech-Lithuanian.

There are two major problems concerning the glossaries. The first one is the semantic ambiguity of lexems in both languages, the other one is the incompatibility of the morphological components for both the source and target languages.

The semantic ambiguity is no problem for the pair Czech-Slovak, since these languages are very similar at all linguistic levels. Some problems may arise for other Slavonic languages, although they will be quite rare. A different problem occurs in Lithuanian. Although the lexical ambiguity is not higher than for Slavonic languages, the verbal aspect causes problems because of its specific use in East Baltic languages.

In the following, we list the most frequent changes of morphological categories that are necessary to achieve correct translation.

**Gender and number** Many corresponding words have different gender in Czech and Lithuanian. In some cases, the number can differ as well. Dict. entries look as follows: *důvod-MASC1 “reason”* → *priežastis-FEM*

**Case of preposition** Since the valence (case) of corresponding prepositions is different usually, it has to be modified. This information is used later in the transfer phase to change the case of the congruent object of the preposition. A special case are prepositions that express location, as they are omitted and the case of the object is changed to locative. Dict. entries look as follow: *do-GEN “into”* → *i-ACC*

**Participles** A rather technical problem are participles. These verb forms are annotated as adjectives in Czech. Unfortunately, the tool for morphological synthesis of Lithuanian

---

13 There are some statistical methods that could help (cf. (Pecina and Holub 2002)). We solve this problem manually in glossaries at the moment, although this solution is suitable for narrow domains only.

14 Some Balticists assert that there is no verbal aspect in Baltic languages and argue for a classification of verbs by Aktionsart, similarly as in German (Račienė 1999; Forssman 2001). Nevertheless we are able to determine this category in the MT system by quite simple correspondence rules coded in an additional glossary information.

---

111
requires them to be marked as verbs. Thus we have separate entries for these forms and define what categories have to be replaced (POS) or added (voice, tense etc.). Other categories (gender, case, number etc.) preserve their values. Dict. entries look as follows: koupený-ADJ "bought" → pirkš-VERB,PART,PASS,PAST

Adjectives of possibility Adjectives that express possibility are derived from verbs. The same meaning is expressed by Lithuanian passive participles (e.g., vyslovičiščný “pronounceable” → išlariamo), thus the dict. entries are similar as in the previous case.

7 Shallow rule-based transfer

The transfer is partial, i.e., only smaller constituents are taken into account. We do not consider dependencies at the level of actants. This leads to problems with valence but this approach may be sufficient for related languages in most cases.

The transfer consists of simple rules that adapt category values, such as gender or case, to preserve agreement between a governor and its dependents. Other rules modify word order, if the original one would make the phrase ill-formed. Word order is changed only locally, constituents at the highest level remain at their positions.

Morphological categories are modified mainly in noun phrases, word order changes concern the order of genitive attributes and some adpositions. The most frequent rules for the pair Czech-Lithuanian are presented in the following.

Congruence in noun phrases When gender of a noun is changed, all congruent dependents have to change this category as well, e.g., vážný důvod-MASC “serious reason” → svarbi priežastis-FEM (*svarbus-MASC).

Congruence of prepositions When a Lithuanian preposition requires an other case than its Czech counterpart, the case of its object has to be changed. This modification includes also the adaptation of all congruent dependents (as in the previous case), e.g., do Prag-GEN “to Prague” → į Pragą-ACC (*Pragos-GEN).

Order of genitive attributes Genitive attributes are placed behind the governing noun, e.g., kniha bratra “brother’s book” → knišio knyga (the governing noun is in bold).

Order of adpositions Some counterparts of Czech prepositions are placed behind their object, e.g., k lesu “to the forest” → miško link (the noun is in bold).

8 Evaluation

As has been explained in the previous sections, our method provides only a partial machine translation. This means that the translation is not expected to be perfect and post-editing of the result may be necessary. Nevertheless, the evaluation shows that the quality of the Czech-to-Lithuanian translation is better than the results achieved for Polish by (Dębowski et al. 2002), although Polish is more related to Czech than Lithuanian.

We translated two fragments of a technical manual from Czech to Lithuanian automatically by the MT system and manually (correcting the automatic translation) and computed the accuracy
of the translation using the Trados Translator’s Workbench. The weighted average of the fuzzy match is 87.6%. (Debowski et al. 2002) give 71.4% as the result for Polish being target language.

Table 1 summarizes the MT results from Czech to three different target languages using Česáko. Results of our work are emphasized, the baseline for English (a commercial MT system has been used) is given in italic due to (Hajič et al. 2003).

<table>
<thead>
<tr>
<th>target language</th>
<th>weighted avg</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>30%</td>
<td></td>
</tr>
<tr>
<td>Slovak</td>
<td>90%</td>
<td></td>
</tr>
<tr>
<td>Polish</td>
<td>71.4%</td>
<td></td>
</tr>
<tr>
<td>Lithuanian</td>
<td>87.6%</td>
<td>91.8%</td>
</tr>
</tbody>
</table>

Table 10.1: Accuracy of the translation

In the following, we describe the most common translation errors.

**Incorrect lemma.** In some cases, lemmas are translated incorrectly. The cause is often semantic. For example, the noun phrase jméno otce (‘father’s name’ or ‘patronymic’) can be translated as tévo vandos or tévavardis. Both translations have different meaning, which is rather semantic. Thus, we have no criterion to decide at the syntactic level, how to translate this phrase, and the translation may be incorrect.

**Incorrect verb forms** are used sometimes. This problem can occur especially for participles (and similar verb forms, such as half-participles and other transgressive forms), the subjunctive and periphrastic verb constructions.

**Incorrect inflection** concerning the declination of substantives, adjectives etc. occurs quite often. The negation of the verbs or their different valence can cause the case to be translated incorrectly. This is the most frequent problem which occurs mainly in more complicated syntactic constructions that have not been fully analyzed by the partial grammar. For example, a relative pronoun at the beginning of an embedded sentence is not marked as a dependent of its head noun, i.e., its gender is preserved even if the gender of the noun has been changed during the transfer.

## 9 Conclusions and further work

We have described the experimental MT system Česáko and analyzed some aspects concerning shallow parsing and MT. It is obvious that shallow NLP methods cannot achieve as good results as the full-fledged approach, but due to the fact that there is no syntactic parser for Czech with sufficient quality, shallow methods may fill in the gap until a better, full-fledged parser will be developed.

Although Česáko does not generate perfect translation, the results are quite encouraging. The translated text has to be post-processed by the user, of course, nevertheless the result is useful and can be used, for example, in combination with translation memory tools (Homola and Tolvaj 2004).
Our further work will include improvements of the shallow parser and transfer rules as well as extending the MT system to other language pairs. Furthermore, we would like to integrate the system SProUT (Becker et al. 2002) and its linguistic resources for Central and Eastern European languages (Drożdżyński et al. 2003) with Česíko to simplify the development of grammars and improve the translation quality by integrating named entity recognition in the system (Babych and Hartley 2004; Piskorski et al. 2004).

Acknowledgements

The work described in this paper has been supported by the grant of the MŠMT ČR ME642, No. LN00A063 and partially supported by the grant of the GAČR No. 405/03/0914.
References


Implementing System BV of the Calculus of Structures in Maude

Ozan Kahramanoğullari
University of Leipzig and Dresden University of Technology, ICCL
ozan@informatik.uni-leipzig.de

Abstract.
System BV is an extension of multiplicative linear logic with a non-commutative self-dual operator. We first map derivations of system BV of the calculus of structures to rewritings in a term rewriting system modulo equality, and then express this rewriting system as a Maude system module. This results in an automated proof search implementation for this system, and provides a recipe for implementing existing calculus of structures systems for other logics. Our result is interesting from the view of applications, specially, where sequentiality is essential, e.g., planning and natural language processing. In particular, we argue that we can express plans as logical formulae by using the sequential operator of BV and reason on them in a purely logical way.

1 Introduction

The calculus of structures is a proof theoretical formalism, like natural deduction, the sequent calculus and proof nets, for specifying logical systems syntactically. It was conceived in (Guglelmi 2002) to introduce the logical system BV, which extends multiplicative linear logic by a non-commutative self-dual logical operator. Then it turned out to yield systems with interesting and exciting properties for existing logics such as classical logic (Brünnler 2003), linear logic (Straßerburger 2003) and modal logics (Stewart and Stouppa 2003), and new insights to their proof theory. In (Tiu 2001), Tiu showed that BV is not definable in any sequent calculus system. Bruscoli showed in (Bruscoli 2002) that the non-commutative operator of BV captures precisely the sequentiality notion of process algebra, in particular CCS (Milner 1989).

In contrast to sequent calculus, the calculus of structures does not rely on the notion of main connective and, like in term rewriting, it permits the application of the inference rules deep inside a formula (structure). In this paper, exploiting this resemblance, we present a general procedure turning derivations in logical systems of the calculus of structures into rewritings in term rewriting systems modulo equality. We illustrate our procedure on system BV of the calculus of structures. Then, we encode the resulting term rewriting system in Maude (Clavel, Durán, Eker, Lincoln,...)
Martí-Oliet, Meseguer, and Talcott 2004; Clavel, Durán, Eker, Lincoln, Martí-Oliet, Meseguer, and Talcott 2003) which results in an implementation of an automated proof search tool for system BV. We also argue that we can employ system BV on applications where sequentiality is essential. In particular, we refer to our encoding of the conjunctive planning problems in the language of BV which allows to express plans as logical formulae. Space restrictions do not permit to present this encoding in detail, we refer to (Kahramanoğulları 2004a).

2 System BV

In this section, we will shortly present the system BV of the calculus of structures, following (Guglielmi 2002). Systems in the calculus of structures for other logics (Brünnler 2003; Straßburger 2003; Stewart and Stouppa 2003) are designed by respecting the scheme in this section.

There are countably many atoms, denoted by \(a, b, c, \ldots\) Structures of the language BV are denoted by \(R, S, T, \ldots\) and are generated by

\[
S ::= \circ | a | \langle S_1 \ldots ; S \rangle | [S, \ldots, S] | (S, \ldots, S) | \overline{S}
\]

(11.1)

where \(\circ\) stands for any atom and \(\circ\), the unit, is not an atom. \(\langle S_1 \ldots ; S \rangle\) is called a seq structure, \([S, \ldots, S]\) is called a par structure, and \((S, \ldots, S)\) is called a copar structure, \(\overline{S}\) is the negation of the structure \(S\). Structures are considered equivalent modulo the relation \(\approx\), which is the smallest congruence relation induced by the equations shown in Figure 11.1.\(^1\) There \(\overline{R}, \overline{T}\) and \(\overline{U}\) stand for finite, non-empty sequences of structures. A structure context, denoted as in \(S\{\ \}\), is a structure with a hole that does not appear in the scope of negation. The structure \(R\) is a substructure of \(S\{\overline{R}\}\) and \(S\{\ \}\) is its context. Context braces are omitted if no ambiguity is possible: for instance \(S\{\overline{R}, T\}\) stands for \(S\{\overline{[R, T]}\}\). A structure, or a structure context, is said to be in negation normal form when the only negated structures appearing in it are atoms, no unit \(\circ\) appears in it and no parentheses can be equivalently eliminated.

In the calculus of structures, a typical (deep) inference rule is a scheme of the kind

\[
\frac{S\{T\}}{S\{R\}}^\rho
\]

where \(\rho\) is the name of the rule, \(T\) is its premise and \(R\) is its conclusion. Such a rule specifies the implication \(T \Rightarrow R\) inside a generic context \(S\{\ \}\), which is the implication being modeled in the system\(^2\). An inference rule is called an axiom if its premise is empty. Rules with empty contexts correspond to the case of the sequent calculus.

A (formal) system \(S\) is a set of inference rules. A derivation \(\Delta\) in a certain formal system is a finite chain of instances of inference rules in the system. A derivation can consist of just one structure. The topmost structure in a derivation, if present, is called the premise of the

\(^1\)In (Guglielmi 2002) axioms for context closure are added. However, because each equational system includes the axioms of equality, context closure follows from the substitutivity axioms.

\(^2\)Due to duality between \(T \Rightarrow R\) and \(\overline{R} \Rightarrow \overline{T}\), rules come in pairs of dual rules: a down-version and an up-version. For instance, the dual of the \(\text{ai} \downarrow\) rule is the cut rule. In this paper, we only consider the down rules, which provide a sound and complete system.
Associativity
\[ \langle \bar{R}; \langle \bar{T}; \bar{U} \rangle \rangle \approx \langle \bar{R}; T; \bar{U} \rangle \]
\[ \bar{R}, [\bar{T}] \approx [\bar{R}, \bar{T}] \]
\[ (\bar{R}, \langle \bar{T} \rangle) \approx (\bar{R}, \bar{T}) \]

Commutativity
\[ [\bar{R}, \bar{T}] \approx [\bar{T}, \bar{R}] \]
\[ (\bar{R}, \bar{T}) \approx (\bar{T}, \bar{R}) \]

Negation
\[ \bar{\sigma} \approx \circ \]
\[ \langle \bar{R}; \bar{T} \rangle \approx \langle \bar{T}; \bar{R} \rangle \]
\[ [\bar{R}, \bar{T}] \approx [\bar{T}, \bar{R}] \]
\[ \langle R, T \rangle \approx [\bar{R}, \bar{T}] \]

Unit
\[ \langle \circ; \bar{R} \rangle \approx \langle \bar{R}; \circ \rangle \approx \langle \bar{R} \rangle \]
\[ [\circ, \bar{R}] \approx [\bar{R}] \]
\[ (\circ, \bar{R}) \approx (\bar{R}) \]

Singleton
\[ \langle \bar{R} \rangle \approx [\bar{R}] \approx (\bar{R}) \approx \bar{R} \]

Figure 11.1: The equational system underlying BV.

deviation, and the bottommost structure is called its conclusion. The length of a derivation is the number of instances of inference rules appearing in it.

The system \{\circ\downarrow, \text{ai}\downarrow, \text{s}, \text{seq}\downarrow\}, shown in Figure 11.2, is denoted BV and called basic system V, where V stands for one non-commutative operator\(^3\). The rules of the system are called unit (\text{circ}\downarrow), atomic interaction (\text{ai}\downarrow), switch (\text{s}) and seq (\text{seq}\downarrow). We consider ai\downarrow to be a schema for all positive atoms a.

There is a straightforward correspondence between structures not involving seq and formulae of multiplicative linear logic. For example \([\langle a, b \rangle, \text{c}]\) corresponds to \((a \otimes b) \otimes c \otimes d\), and vice versa. Units 1 and \perp are mapped into \circ, since 1 \equiv \perp, when the rules mix and mix0 are added to MELL. For a detailed discussion on the proof theory of BV and the precise relation between BV and multiplicative linear logic the reader is referred to (Guglielmi 2002).

3 From Derivations to Rewritings

In this paper, we assume that the reader is familiar with the notions of term rewriting such as terms, positions, replacements, substitutions, equations and rewrite rules as can be found in e.g. (Baader and Nipkow 1998; Plaisted 1993). However, we will recapitulate the definition of the rewrite relation \(R/E\) that will be used extensively. This section is partly a summary of the technical report (Höllocher and Kahramanoğulları 2004).

Given terms \(s, t\), a term rewriting system \(R\) and an equational system \(E\), \(s\) rewriting to \(t\) wrt \(R\) and \(E\), denoted by \(s \rightarrow_{R/E}(\rho, \pi, \sigma)\) \(t\) if there are terms \(s', t'\), a rewrite rule \(\rho = l \rightarrow r\), a position \(\pi \in \text{pos}(s')\) and a substitution \(\sigma\) such that \(s \approx_{E} s', s'|_{\pi} = \sigma(l)\), \(t' = s'|_{\sigma(r)}|_{\pi}\) and \(t' \approx_{E} t\). In other words, \(s \rightarrow_{R/E}(\rho, \pi, \sigma)\) \(t\) if \((\exists s', t')\) \(s \approx_{E} s' \rightarrow_{R}(\rho, \pi, \sigma)\) \(t' \approx_{E} t\).

\(^{3}\)This name is due to the intuition that \(W\) stands for two non-commutative operators.
$$\frac{Q}{Q'} \approx \frac{T'}{T} \rho \begin{array}{|c|c|} \hline \Delta & n_{22}(S) = \\
R & \begin{cases} 
\circ & \text{if } S = \circ, \\
S & \text{if } S \text{ is an atom,} \\
{\frac{n_{22}(R)}{n_{22}(T)}} & \text{if } S = \overline{R}, \\
\langle n_{22}(R); n_{22}(T) \rangle & \text{if } S = \langle R; \overline{T} \rangle, \\
(n_{22}(R), n_{22}(T)) & \text{if } S = (R, \overline{T}), \\
|n_{22}(R), n_{22}(T)| & \text{if } S = [R, \overline{T}]. 
\end{cases} \\
\end{array}$$

Figure 11.3: **Left:** A derivation from $R$ to $Q$ **Right:** Transformation $n_{22}$

### 3.1 Replacing Equivalence Classes by Equality Steps

For this purpose, we separate the notion of a structure from the equivalence class defined by the equations shown in Figure 11.1. From this point on, a structure is an expression of the form delivered in (11.1) and no longer an equivalence class of these expressions.

A structure $R$ is a *derivation from $R$ to $R$. If $\Delta$ is a derivation from structure $R$ to structure $T$, $T \approx T'$, there is an instance of an inference rule $\rho$ with conclusion $T'$ and premise $Q'$, and $Q' \approx Q$ then the derivation on the left-hand-side of Figure 11.3 is a *derivation from $R$ to $Q$. For notational convenience we combine two subsequent equality steps occurring in a derivation into a single equality step. The notion of a proof can be analogously redefined, that is, $\Delta$ is a proof of $R$ iff $\Delta$ is a derivation from $R$ to $T$ and $T \approx \circ$.

Because $\approx$ is the finest congruence relation generated by the equational system shown in Figure 11.1, each derivation and each proof as defined in Section 2 can be transformed into a derivation and a proof as defined in this subsection, respectively. We have thus clarified the role of the equational theory underlying derivations in BV. The same kind of changes to BV have already been considered in (Brünnler 2003).

### 3.2 Replacing n-ary Operators by binary Ones

We will now restrict ourselves to binary operators. The recursive transformation on the right-hand-side of Figure 11.3 turns each structure into a structure, where only the binary operators $\langle \cdot, \cdot \rangle$, $\langle \cdot \cdot \rangle$ and $\langle \cdot \circ \rangle$ are used.\textsuperscript{4} As a consequence, we will also simplify the equations defining the syntactic equivalence leading to a refined set of equations as shown in Figure 11.5, where the equations for singleton become superfluous. Because the inference rules for BV (see Figure 11.2) use only binary seq-, par- and copar-operators, there is no need to change them.

Because $n_{22}(S) \approx S$, derivations wrt n-ary seq-, par- and copar-operators can be equivalently turned into derivations with only binary seq, par- and copar-operators and vice versa. This may lead to less intelligible structures, but the n-ary operators may be reintroduced as abbreviations (e.g. (Fitting 1996; Hölldobler 2001)).

\textsuperscript{4}While applying this transformation, due to associativity of the structures, it is important to observe the equivalence $[R, T] = [R, T_1, \ldots, T_n] = [R, [T_1, \ldots, T_n]]$ of structures where $n \geq 1$.  

120
\[
R_{\text{Neg}} = \begin{cases} 
\delta & \rightarrow \circ \\
\langle R; T \rangle & \rightarrow \langle \tilde{R}; \tilde{T} \rangle \\
[R; T] & \rightarrow \langle \tilde{R}; \tilde{T} \rangle \\
(R; T) & \rightarrow [\tilde{R}; \tilde{T}]
\end{cases}
\]

\[
eq - \circ = \circ \\
\text{eq} - \langle R; T \rangle = \langle - R; - T \rangle \\
\text{eq} - [ R; T ] = \{ - R; - T \} \\
\text{eq} - \{ R; T \} = \{ - R; - T \} \\
\text{eq} - - R = R.
\]

Figure 11.4: Left: T. R. System \( R_{\text{Neg}} \) Right: Corresponding Maude code

3.3 Replacing Structures by Terms

We replace the structures by terms, and consider terms over variables, thus formalizing the concept of structures with variable occurrences. Let

\[\Sigma_{\text{BV}} = \{ \circ, \circ, [\circ, \circ], (\circ, \circ), (\circ, \circ, \circ) \cup \{ a \mid a \text{ is a positive atom} \}\].

Then, structures as defined in Section 2 are simply \( \Sigma_{\text{BV}} \)-terms over the empty set of variables, i.e., ground \( \Sigma_{\text{BV}} \)-terms. On the other hand, by considering a non-empty set \( \mathcal{V} \) of variables, we obtain \( \Sigma_{\text{BV}} \)-terms over \( \mathcal{V} \), which correspond to structures with variables.

This way, we can use the notions structure and \( \Sigma_{\text{BV}} \)-term synonymously and replace the notion of context in derivations within \( \text{BV} \) by the notion of a position, thus being precise about which substructure or subterm is replaced in a derivation step: the notion of positions, subterms and the replacement of a subterm by another one at a particular position take over the role of a context in \( \text{BV} \).

3.4 Orienting the Equalities for Negation

The inference rules of \( \text{BV} \) can be applied only to the structures which are not under the scope of negation sign. Since these rules do not introduce any new negation signs, neither when they are applied bottom-up nor top-down, we can orient the equalities for negation as rewrite rules from left to right to get the negation normal form at the beginning of a derivation:

**Lemma 1** Term rewriting system \( R_{\text{Neg}} \) on the left-hand-side of Figure 11.4 is (i.) terminating, (ii) confluent. (iii.) Let \( s \) be a \( \Sigma_{\text{BV}} \)-term. The normal form of \( s \) with respect to \( R_{\text{Neg}} \) is in negation normal form.

**Sketch of Proof** (i) It suffices to take the lexicographic path order \( - \succ_{\text{lpo}} [\circ, \circ] \succ_{\text{lpo}} (\circ, \circ) \succ_{\text{lpo}} \circ \) as stated in (Baader and Nipkow 1998). (ii) Since \( R_{\text{Neg}} \) is terminating, the result follows from the analysis of the critical pairs. (iii) \( s \) being in negation normal form and applicability of a rewrite rule of \( R_{\text{Neg}} \) are contradictory.

\[\square\]

3.5 Replacing Inference Rules by Rewrite Rules

In the final step, we define the term rewriting system \( \text{RBV} \) and the equational theory \( \text{EBV} \) corresponding to \( \text{BV} \) such that derivations in \( \text{BV} \) correspond to rewritings \( \rightarrow_{\text{RBV/EBV}} \). The context
occurring in inference rules is eliminated and inference rules are turned into rewrite rules. Each inference rule occurring in BV as shown in Figure 11.2 except $\phi \downarrow$ is turned into a rewrite rule as shown in Figure 11.6 by dropping the context $S$. As before, $a \downarrow \phi$ is a schema for all positive atoms $a$.

**Proposition 2** Let $s$ and $t$ be two $\Sigma_{BV}$-terms or structures, where $t$ is in negation normal form. (i) There is a derivation in BV from $s$ to $t$ having length $n$ if and only if there exists a rewriting $s \xrightarrow{R_{Neg}} s' \xrightarrow{n_{RBV/EBV}} t$. (ii) There is a proof of $s$ in BV having length $n$ if and only if there exists a rewriting $s \xrightarrow{R_{Neg}} s' \xrightarrow{n_{RBV/EBV}} \phi$.

**Sketch of Proof** The proof of (i) follows immediately from the discussion in this and the previous subsections and Lemma 1, by induction on the length of the derivation in BV and on the number of rewrite steps in RBV/EBV, respectively, for the if part and the only if part, respectively. (ii) follows immediately from (i).

## 4 Implementation in Maude

The language Maude (Clavel, Durán, Eker, Lincoln, Martí-Oliet, Meseguer, and Talcott 2003) allows implementing term rewriting systems modulo equational theories due to the built-in very fast matching algorithm that supports different combinations of associative, commutative equational theories, also with the presence of units. Another important feature of Maude that makes it a plausible platform for implementing systems of the calculus of structures is the availability of the search function since the 2.0 release of Maude. This function implements breadth-first search which is vital for complete search for derivations and proofs.

The Maude system module in Figure 11.7 implements the system RBV modulo EBV where the equalities for associativity, commutativity and unit become operator attributes “assoc”, “comm” and “id : o”. The module presumes that the $\Sigma_{BV}$-terms are in negation normal form. To get
the negation normal form of a $\Sigma_{BV}$-term, we can employ a functional module with the *Maude equations* on the right-hand-side of Figure 11.4.

mod BV is
sorts Atom Unit Structure .
subsort Atom < Structure .
subsort Unit < Structure .

op o : -> Unit .
op -` : Atom -> Atom [ prec 50 ] .
ops a b c d e : -> Atom .

var R T U V : Structure . var A : Atom .

rl [s] : [ { R , T } , U ] => [ [ R , U ] , T ] .
rl [q-down] : [ < R ; T > , < U ; V > ] => < [R,U] ; [T,V] > .
endm

Figure 11.7: The system module that implements BV.

We can then use the Maude 2 search command for searching for proofs or derivations:

```
search [ - c , [< a ; {c,- b} >] , < - a ; b > ]] =>+ o .
```

Maude> search [ - c , [< a ; {c,- b} >] , < - a ; b > ] =>+ o .

search in BV : [ - c , [< a ; {c,- b} >] , < - a ; b > ] =>+ o .

Solution 1 (state 2229)
states: 2230 rewrites: 196866 in 930ms cpu (950ms real) (211683
rewrites/second)
empty substitution
No more solutions.
states: 2438 rewrites: 306179 in 1460ms cpu (1490ms real) (209711
rewrites/second)

After a successful search, we can display the proof steps by using the command
"show path <state_number_displayed> ".

Maude> show path 2229 .
state 0, Structure: [- c , [< a ; {c,- b} >] , < - a ; b > ] ]

```
att [ rl [ [ R , T ] , < U ; V > ] => [ R , U ] ; [ V , T ] ] [label q-down] ] =>
state 20, Structure: [- c , [< a ; - a ] ; [b , {c,- b} ] ]

```
```
att [ rl [ A , - A ] => o [label ai-down] ] =>
state 178, Structure: [b , [ - c , {c,- b} ]] 
```
```
att [ rl [ U , [ R , T ] ] => [ T , [ R , U ] ] [label s] ] =>
state 634, Structure: [b , [ - b , {c,- c} ]] 
```
```
att [ rl [ A , - A ] => o [label ai-down] ] =>
```

123
state 1492, Structure: [b,- b]
===[ r1 [A,- A] => o [label ai-down] . ]==> 
state 2229, Unit: o

It is also possible to display all the one step rewrites of a $\Sigma_{BV}$-term by using the Maude command “search <term> =>1 R .”.

5 Planning within BV

In (Kahramanoğulları 2004a), we present an encoding of the conjunctive (multiset rewriting) planning problems (see e.g. (Große, Hölldobler, and Schneeberger 1996)) in the language of BV, where plans are not extracted from the proof of a planning problem, but are explicit premises of derivations, which result from bottom-up search. However, in such an encoding, being restricted to BV, while going up in a derivation, the actions in the problem structure at the conclusion of the derivation must be used precisely once. In order to overcome this, there, we employ system NEL (Guglielmi and Straßburger 2002), the extension of BV with the exponentials of linear logic, to express the availability of actions arbitrarily many times.

In (Bruscoli 2002), Bruscoli showed that there is a correspondence between system BV and a fragment of CCS (Milner 1989): the sequential composition corresponds to the non-commutative operator $\text{seq}$. Parallel composition is naturally mapped to the commutative linear logic operator $\text{par}$. However, as it is the case in CCS, there only the actions (labels) are included in the language, but not the resources that are consumed and produced by the actions.

Similar to (Bruscoli 2002), by exploiting the non-commutative operator of system BV, and the commutative logical operator $\text{par}$, we are able to observe concurrent plans, where the parallelism between plans is respected. Since our encoding is propositional, no unification mechanism is needed. This allows system NEL to give the complete operational semantics of our method, and establish the first step of a uniform formalism that connects concurrency and planning. This way, it becomes possible to transfer methods from concurrency to planning.

6 Discussions

In this paper, we showed that system BV of the calculus of structures can be expressed as a term rewriting system which can be implemented in Maude for automated proof search and automated application of inference rules. This way, we have also provided a tool for implementing a fragment of CCS which was shown to be equivalent to BV in (Bruscoli 2002).

We observed that orienting the equalities for $\text{unit}$ by modifying the inference rules to preserve completeness causes a gain in efficiency in proof search. In (Kahramanoğulları 2004b) we present equivalent systems to system BV where equalities for unit become redundant. Furthermore, due to the non-deterministic application of inference (rewrite) rules, often there are several rewritings of a structure, but in general, only a few of them lead to a proof. A similar problem was solved in (Verdejo and Martí-Oliet 2002) by employing the conditional rules of Maude and by means of a strategy at the meta-level (Clavel, Durán, Eker, Meseguer, and Stehr 1999).

The methods presented in this paper can be analogously applied to the existing systems in the calculus of structures for classical logic (Brünnler 2003) and linear logic (Straßburger 2003),

124
since these systems can also be expressed as term rewriting systems (Hölldobler and Kahramanoğulları 2004). However, termination of proof search in our implementation is a consequence of BV being a multiplicative logic. For the logics with an additive behavior, e.g., classical logic, some strategy must be introduced. Different Maude modules for the systems in the calculus of structures, including BV, classical logic and linear logic, are available for download at http://www.informatik.uni-leipzig.de/ozan/maude_cos.html.

Carrying our results to full linear logic is of particular interest, since the sequent calculus presentation of linear logic was previously encoded into rewriting logic within Maude modules (see, e.g., (Clavel 2000)). However those modules are not directly executable, in particular due to the promotion rule: in contrast to the calculus of structures, in the sequent calculus, promotion rule requires a global view of the formulae, which makes it difficult to express as an implementable rewriting rule.

Acknowledgements

This work has been supported by the DFG Graduierten-kolleg 446. I am grateful to Alessio Guglielmi, Steffen Hölldobler, Steven Eker, Lutz Straßburger and the members of the proof theory group at the International Center for Computational Logic at the TU Dresden. I would like to thank anonymous referees for valuable remarks and improvements.
References


Word Sense Disambiguation of Estonian with syntactic dependency relations and WordNet

KAAREL KALJURAND
Tartu University
Kaarel.Kaljurand@ut.ee

ABSTRACT.
This paper describes a method for noun sense disambiguation in Estonian texts using Constraint Grammar syntactic categories and WordNet as a knowledge source. First we convert the syntactic categories into syntactic dependency relations, focusing currently on subject, object and adjective modification relations. This information lets us build a database of syntactically similarly behaving nouns. We assume that those nouns are also semantically similar and use them to disambiguate the noun senses in texts. WordNet hyponymy/hypernymy hierarchy is used to calculate the semantic closeness of the noun senses.

1 Introduction
The task of word sense disambiguation (WSD) is to automatically detect the meaning of a word in a given context. All the possible meanings of the word are usually described in a manually compiled dictionary. The WSD system has to match the word and its context with a meaning in the dictionary.

The current paper describes a method for noun sense disambiguation in Estonian texts which relies on the existing language resources, namely Estonian Constraint Grammar parser and Estonian WordNet. The first is used to detect syntactic categories in order to describe the context of the nouns under analysis and the second as a dictionary of noun meanings.

WordNet (Fellbaum 1998) is different from traditional dictionaries as it groups words into sets of synonyms (synsets). These sets correspond to word meanings. If a word has several meanings then it belongs to more than one synset. All synsets are connected by semantic relations. There is a wide variety of different relation types, the most important being a hypernymy/hyponymy relation which connects sets that are related by subconcept relationship. In our approach, the WordNet dictionary functions merely as a dictionary that provides sense distinction. We ignore most of the descriptive information in WordNet, using only the information on the grouping of the words into synsets and their hypernymy/hyponymy relations.
We focus more on the contribution of syntax to WSD. We use a large collection of syntactically annotated texts to compile a database of syntactically similar behaving nouns. The annotation is based on the Estonian Constraint Grammar formalism which attempts to specify a syntactic category for each word in the sentence. As such, the sentences are not structured by the annotation. Therefore, we first convert the syntactic categories into syntactic dependency relations, specifying explicitly the participants in the relations. This information lets us to build the fore-mentioned database.

Following the Distributional Hypothesis, we assume that nouns which appear in the similar syntactic contexts are also semantically similar. When disambiguating a noun, we use its syntactic context to retrieve a list of similarly behaving nouns from the database. Comparing their senses with the senses of the analyzed noun in the WordNet hypernymy/hyponymy hierarchy lets us to choose the most probable sense for the analyzed noun.

One of the aims of this work is to find out what type of syntactic context has the main influence on the senses of the nouns in Estonian. We focus currently on subject, object and adjective modification relations. The main idea behind this method can be easily generalized to disambiguate also verbs and adjectives. The method is applicable also to other languages, given that they have a Constraint Grammar parser and a WordNet dictionary available.

Our aim is to provide high precision disambiguation. We tolerate low coverage for now, having the coverage still high enough to make the system useful and give some credibility to the evaluation results. The coverage will grow as the external language resources evolve.

In this paper, section 2 describes existing work on WSD for English and Estonian, section 3 describes linguistic resources available for Estonian that can be used for our purposes, section 4 gives an overview of the possible use of Constraint Grammar in a WSD application, section 5 specifies the WSD algorithm, section 6 gives the results obtained by a preliminary disambiguation model and finally section 7 talks about our future plans in Estonian WSD.

## 2 Existing work

Most work done in the field of WSD is for English. (Ide and Véronis 1998) give an overview of different WSD methods, mentioning also syntax-based methods as promising. (Yarowsky 1993) claims that syntactic functions relevant for WSD include objects for verbs, noun heads for adjectives and adjective or noun modifiers for nouns. (Lin 1997) uses a dependency parser to compile a database of syntactic relations and uses it to disambiguate word senses. The results strongly indicate that local syntactic context has influence on the word meaning. (Martínez et al. 2002) describe a WSD method in which a broad range of syntactic features are extracted with a dependency parser and used in machine learning algorithms. High precision WSD can be achieved, given that low coverage is tolerated. (Lee and Ng 2002) experiment with different combinations of context features and machine learning algorithms. They conclude that no clear winner exists neither among the learning algorithms nor the context features. In order to model the syntactic nature of nouns they extract the head of the noun, its part-of-speech, voice ("active", "passive" or "0" if the head is not a verb), and relative position ("left" or "right" from the analyzed noun). Such description of the context is simple and is likely to work well for English where the voice of the head verb and its relative position can model subject and object relations with high precision, relying on the fixed word order.

For Estonian WSD, so far a very simple bag-of-words model has been used for the context of
the analyzed nouns and verbs: words of the same part-of-speech which are located next to each other in a window of predefined size are considered semantically similar (Vider and Kaljurand 2001). The disambiguation system retrieves their corresponding senses from the Estonian WordNet. The group of senses located closest to each other in the hypernym/hyponym hierarchy is output as the proposed word sense reading for the window of words. This method correctly assumes that the context provides evidence for fixing the meaning of the words. However, its use of the context is very primitive, as it doesn’t see it as syntactically structured. Even more, when disambiguating nouns this method makes no use of their surrounding adjectives and verbs. Therefore, it is not surprising that the results obtained are low, almost close to the random baseline.

3 Linguistic resources for Estonian

The Estonian language belongs to the Finno-Ugric language family, being relatively close to Finnish. It is characterized by a free word order, an elaborate declinational and conjugational system — nouns, adjectives, pronouns, and numerals have 14 cases, and the ending of the verb depends on the tense, person and number. The stems of the words tend to change in declination and conjugation. Estonian has no prepositions, but there are several postpositions. Estonian uses no articles and no gender. Influenced by German, the Estonian syntax makes a lot of use of phrasal verbs.

Previous work on Estonian language technology has created several resources useful for our purposes. The integral part of the described WSD system is formed by Estonian Constraint Grammar (ESTCG) parser (Müürisepp et al. 2003) and the Estonian WordNet (EstWN).

Constraint Grammar (CG) (see e.g. (Karlsson et al. 1995) and (Tapanainen 1996)) is a dependency grammar which does not attempt to specify full syntactic relations occurring in the sentence, but only shallow syntactic categories for words. CG parsers have shown highly reliable results and constraint grammars have been developed for many languages. The only disincentive of using constraint grammar in language processing applications is that it doesn’t provide a full dependency structure as the analysis of the sentence. Still, it has been shown that CG can serve as a preliminary form of annotation in an attempt to describe a real dependency structure, e.g. (Järvinen and Tapanainen 1997) describe a Functional Dependency Grammar parser for English which uses CG as an underlying formalism.

In ESTCG, the main syntactic categories are subject (SUBJ), object (OBJ), main finite verb (+FMV), adjective modifier (AN) or <AN>, participle modifier (VN) or <VN>, adverbial (ADV), and predicative (PRD). Altogether, ~30 categories are used. Except for the modifiers of nouns, the direction of the dependency is not specified. The sentence “Suur koer hammustas meest” (A big dog bit a man) is analyzed in ESTCG as: Suur/AN> koer/SUBJ hammustas/+FMV meest/OBJ.

The ESTCG parser is a rule-based system, it has been evaluated on fiction corpora and has show a precision of 78% and recall of 96%. The parser uses an external morphological analysis component. Naturally, the analysis result depends on the efficiency of morphological analysis. If this is done by hand then the results are much less ambiguous (precision 88%, recall 99%).

EstWN focuses on hypernym/hyponym relation and noun sense distinction. It currently contains ~10,000 synsets, describing ~10,000 nouns, ~4000 verbs and ~500 adjectives.

In addition, we can make use of two corpora. Corpus where words are annotated with EstWN sense numbers can be used to train and evaluate automatic WSD programs. It is mostly based
on fiction, plus a small amount of news texts, containing ~90,000 words. Corpus manually annotated with ESTCG tags\(^1\) can be used as a reliable knowledge source of the syntactic context of the words. It currently contains ~200,000 words. The texts currently originate only from fiction. The two corpora share a large number of the same source texts.

4 Syntactic dependency relations

4.1 Syntactic dependency relations in WSD

Given that we can extract syntactic relations in the form relation(head, dependent) from the sentences, we have to specify which relations are relevant for our purposes, and what kind of further processing is needed in order to use the pure syntactic relations in semantic analysis.

Basically, every relation which describes nouns should be used in the disambiguation model. However, adverbial relations have proved to be difficult to detect because the information on whether the adverbial attaches to a noun (meest metsast ... (man from the foods ...)) or a verb (... vaadati teleskoobiga (... was seen with a telescope)) is not always present locally. Therefore, we do not currently focus on adverbials, as well as some other infrequently occurring relations.

CG is a theory of syntax. WSD, on the other hand, requires a more semantic basis and therefore the CG tags shouldn't be directly used. Figure 12.1 shows a possible grouping of the relations that accounts for the fact that (1) certain adjectives (participles) are semantically similar to verbs and (2) the use of copula is sometimes similar to modifier use. This classification is made on the basis of morphological tags of the words (substantive, adjective, active participle, etc.). The WSD system could consider the relations in the same group as equivalent, e.g. applying the same reliability weights to them.

4.2 Syntactic relations’ database

Compiling the syntactic relations' database can be split into two steps: (1) generate a corpus annotated with ESTCG tags, (2) extract syntactic relations and store them in a database.

There are several ways to generate ESTCG-annotated documents. Automatic annotation with the ESTCG parser gives us more freedom in choosing the texts and lets us generate a resource of arbitrary size. As a downside, the output of the parser can be very ambiguous and sometimes incorrect. On the other hand, using an existing manually annotated ESTCG corpus provides us with more reliability at the expense of obtaining only a small amount of data. Naturally, compromises exist between the two extremes. E.g. the external morphological analysis component of the parser can be switched off and the parser can be applied to a manually morphologically annotated corpus which is much larger that the ESTCG corpus. The analysis results will be less ambiguous. Also, it would be possible to extract sentences from a large corpus (or from the web) that are likely to have a simple structure (e.g. sentences containing only 1 noun and 1 verb). The ESTCG parser could be applied to such “simple” sentences, again obtaining a reliable output.

An extremely simple dependency relation extraction algorithm was designed, which still performs well. We only try to extract a small number of relations, namely subject, object, and adjective modification (see figure 12.2). The method doesn’t handle coordination, that results in

---

\(^1\)http://math.ut.ee/~heliu/syntkorpus.html
<table>
<thead>
<tr>
<th>Relation</th>
<th>Context</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>subj(0+FMV,0SUBJ)</td>
<td>head verb</td>
<td>koer haugub <em>(dog barks)</em></td>
</tr>
<tr>
<td>attr(S,0VN)</td>
<td>active participle modifier</td>
<td>haukuv koer <em>(barking dog)</em></td>
</tr>
<tr>
<td>pred(0SUBJ,0PRD A)</td>
<td>active participle predicative</td>
<td>koer on haukuv <em>(dog is barking)</em></td>
</tr>
<tr>
<td>Patient relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>subj(0+FMV,0OBJ)</td>
<td>head verb</td>
<td><em>(ta) toidab koera (he feeds the dog)</em></td>
</tr>
<tr>
<td>attr(S,0VN)</td>
<td>passive participle modifier</td>
<td>toidetav koer <em>(fed dog)</em></td>
</tr>
<tr>
<td>pred(0SUBJ,0PRD A)</td>
<td>passive participle predicative</td>
<td>koer on toidetav <em>(dog is fed)</em></td>
</tr>
<tr>
<td>Property relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>attr(S,0AN)</td>
<td>pure adjective modifier</td>
<td>suur koer <em>(big dog)</em></td>
</tr>
<tr>
<td>pred(0SUBJ,0PRD A)</td>
<td>pure adjective predicative</td>
<td>koer on suur <em>(dog is big)</em></td>
</tr>
<tr>
<td>Other relations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>attr(S,0NN)</td>
<td>noun modifier</td>
<td>mehe koer <em>(man's dog)</em></td>
</tr>
<tr>
<td>attr(S,0NN)</td>
<td>noun head</td>
<td>koera kont <em>(dog's bone)</em></td>
</tr>
<tr>
<td>pred(0SUBJ,0PRD S)</td>
<td>predicative noun</td>
<td>koer on loom <em>(dog is animal)</em></td>
</tr>
<tr>
<td>pred(0SUBJ,0PRD S)</td>
<td>copula subject</td>
<td>Muri on koer <em>(Fido is a dog)</em></td>
</tr>
</tbody>
</table>

Figure 12.1: Semantic grouping of syntactic relations. The examples use the noun ‘koer’ *(dog)* for which the syntactic context is provided. ° denotes the ESTCG syntactic category, S and A stand for morphological categories noun and adjective, respectively.

low coverage. The precision can decrease for several reasons: (1) we don’t check agreement when extracting adjective modification relations; (2) we don’t detect phrasal verbs (i.e. the head of the subject or object can come out to be semantically impossible); (3) we assume that the head of the object is always the main verb *(FMV)*, but often it is tagged as an adverbial *(ADV)* or object *(OBJ)*, following the official Estonian grammar *(Erelt et al. 1993)*.

Even though the extraction method is simple, the preliminary analysis of the results shows reasonably good precision. The problem with low coverage can be overcome by relying on data redundancy, i.e. increasing the size of the analyzed corpus. The problem with object extraction precision can be dealt with, considering only sentences with simple structure as extraction input. (The extraction algorithm and reliability statistics are not given in this paper due to the size limits.)

5 Disambiguation algorithm

Given that we have a database of syntactic relations which are known to exist in Estonian texts, we now need to describe the WSD algorithm that makes use of that knowledge source.
Figure 12.2: Syntactic dependency structure of the sentence “Suur koer hammustas meest”. For each word, its syntactic dependency relation, ESTCG category and lemma is provided.

For each analyzed noun in the input text, the algorithm has to find the dependencies which involve the noun, then consult the database of syntactic relations, and retrieve a list of other nouns which are known to occur in the same syntactic context (so called “similar nouns”). These, in turn, provide information to decide the sense of the analyzed noun. The basic structure of the algorithm is shown on figure 12.3.

1. Foreach sentence $S$ in the input text
   
   (a) Parse $S$ with ESTCG parser
   
   (b) Extract syntactic relations in $S$ (if the sentence contains polyseymous nouns)
   
   (c) Foreach polyseymous noun $W$ in $S$

   i. Foreach syntactic relation $R$ involving $W$
   
      A. Retrieve the “similar” nouns from the database that correspond to $R$ and the head (or dependent) of $W$

      B. Retrieve the senses of the “similar” nouns from EstWN

   ii. If no syntactic relations where found for the noun $W$ in sentence $S$ or the found relations were not described in the database then use a fall-back strategy (or leave the noun $W$ ambiguous)

   iii. Otherwise compare the nouns and their senses and disambiguate the noun $W$

Figure 12.3: The disambiguation algorithm.

As we do not focus on coverage for now, the fall back strategy (step 1(c)ii) can be described in a future specification. The most important part of the algorithm is in any case step 1(c)iii which processes the set of similarly behaving nouns, using them to disambiguate the word under analysis. The used model should consider the following (among other things):

- different syntactic relations can have different reliability/informativity for our purposes, independent of the words they bind

- different syntactic relations can be semantically similar (as discussed in section 4.1)

- the words are the more similar the more syntactic relations they share (and not share)

133
• the polysemy of the contextual words: there are (general) adjectives (‘new’, ‘good’, ‘big’) which can modify semantically different words (‘dog’, ‘law’, ‘table’, ‘result’). Such low restrictions on the context occur also in case of certain verbs (‘go’, ‘do’ etc). We should discard them as uninformative.

Having found reliable similar words, selecting the right sense for the analyzed word \( W \) is done by using the WordNet-distance measure: for all the pairs \( (w, x) \), where \( w \) is a sense of \( W \) and \( x \) is a sense of the "similar" word, minimize the distance of \( (w, x) \) in the hypernymy/hyponymy hierarchy. The winning pair will decide the sense of \( W \). This definition of WordNet-distance assumes that semantically similar words are located close to each other in the hypernymy/hyponymy hierarchy (this, of course, might not always hold).

6 Experiment

In order to look for a good model that could be applied by the algorithm, a few experiments were made. First, a syntactic relations' database was automatically generated using the manually annotated ESTCG corpus as input. Only subject, object, adjective premodification and participle premodification relations were stored in the database. As a result, the database described \( \sim 12,400 \) different syntactic relations, involving \( \sim 4200 \) different nouns.

A WSD system was implemented in Perl, and was applied to a small set a texts containing \( \sim 14,000 \) words. Those texts belong to both the ESTCG and the word sense corpora, making it possible to (1) try the WSD system on reliably ESTCG-annotated texts, and (2) evaluate the results automatically. The WSD system used two slightly different models (see the following sections 6.1 and 6.2) in the analysis.

6.1 Model 1

Similar nouns with frequency higher than 1 are ranked by (1) in how many different contexts they have evidence of being found (2) the WordNet-distance of their proposed sense. Voting among 3 best ranking similar nouns determines the selected sense.

6.2 Model 2

Same as Model 1, but contextual verbs and adjectives that are among the top 10 frequent of their part-of-speech in Estonian, are not allowed to be used to retrieve similar nouns. Reason: frequency is a good indication of polysemy, which in turn means less restrictions on the context.

6.3 Results

The results (see table 12.1) clearly indicate the usefulness of syntactic information in WSD. As this information is not always present (or not reliable), the coverage can turn out to be quite low. Still, the high precision shows that such a WSD system can at least be used in a preprocessing phase of disambiguation, making the text a bit less ambiguous. We also believe that it is possible to come up with better performing models.
<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>44%</td>
<td>100%</td>
</tr>
<tr>
<td>Commonest</td>
<td>82%</td>
<td>100%</td>
</tr>
<tr>
<td>Model 1</td>
<td>82%</td>
<td>62%</td>
</tr>
<tr>
<td>Model 2</td>
<td>89%</td>
<td>54%</td>
</tr>
</tbody>
</table>

Table 12.1: Results of automatic disambiguation of ~1700 nouns, which had a description in EstWN (i.e. both mono- and polysemous nouns). Random baseline is a trivial method which selects a sense at random. Commonest baseline selects the most frequent sense in language.

It is also clear that WordNet has quite a high granularity when it comes to differentiating between word senses. Even if the system performs badly by WordNet standards, it can still be useful in a practical application that works with coarse-grained senses. Better evaluation measures (see e.g. (Resnik and Yarowsky 1997)) should therefore be used to detect the exact problematic parts of our approach.

7 Conclusions and future work

7.1 Finding the right model (precision problem)

There is a large set of parameters that influence the outcome of the disambiguation. Somehow the best configuration has to be chosen. Interesting is to evaluate the reliability of different types of syntactic information, e.g. whether the adjective modification really provides the most information for the disambiguation of nouns, as claimed for English.

Generating the “right” knowledge-base of similarly behaving nouns has to be thought through more carefully. The knowledge-base should be of the same domain as the analyzed texts, e.g. technical texts that use figurative language less frequently shouldn’t be analyzed relying on a database generated from fiction texts.

In general, we intend to apply machine learning methods to handle the wide variety of parameters that influence the results and help us to deduce the most effective model of disambiguation.

7.2 Data sparseness (coverage problem)

Given the Zipfian properties of word frequency distributions, the obvious problem with the described lexicalized approach is the data sparseness. To fight this, instead of using a concrete verb (or adjective) as the context word, we could use a more general semantic class (based on e.g. FrameNet), e.g. verbs-of-utterance, verbs-of-movement, etc. Of course, mapping a word to a semantic class, succeeds only if all of its senses belong to the same class.

Also, we could try to recover implicit syntactic relations in the sentences, such as “hidden” subjects in control structures. E.g. the sentence “Koer plaanib haukuda.” (*The dog intends to bark.*) implies a subject relation not only between “koer” (*dog*) and “plaanib” (*intends*), but also
“koer” and “haukuma” (to bark). Such relations are currently difficult to extract because the underlying syntax formalism of ESTCG is too uninformative.

Of course, we can include other syntactic relations, such as adverbials, that are currently ignored, to the knowledge source, in case the compromise between their informativity and extraction reliability influences the final results positively.

Independent of the size of the relations’ database, the disambiguation is still likely to confront words for which no information is available. This information can then be generated at run-time, by using the web (e.g. through the Google API\(^2\)). Context words which trigger no similar nouns can be searched with a search engine, sentences containing them can be syntactically analyzed for similar nouns.

### 7.3 Other methods for WSD

A small number of polysemous words in Estonian can be disambiguated on morphological grounds. E.g. ‘kurk’ (THROAT or CUCUMBER) has different word-forms in all the cases except for the nominative, depending on the sense. Disambiguation of such words requires no syntactic processing.

Some syntactic relations apply to semantically similar words, e.g. coordination. The sentence “I like cats and laws” seems weird, “I like cats and animals” also, but “I like cats and dogs” does not. The coordinating word (‘and’, ‘or’) puts no restrictions on the meaning of its dependents, it only requires that they were similar (but then again: not connected via hypernym relations, as in “cats and animals”). The current approach doesn’t deal with such relations at all.

The analysis could also be made more WordNet-specific, by using other relations than hypernymy/hyponymy, e.g. meronymy, cause, role etc.

Finally, how to solve problems arising from figurative speech? E.g. ‘keel’ (LANGUAGE, or TONGUE/TISSUE, or TONGUE/FOOD, etc.) in the phrase “Mürgise keelgä mees” (literally: Man with a poisonous tongue, really: Man who likes to insult) is currently likely to be disambiguated as TONGUE/FOOD.

### Acknowledgements

This research was partly funded by the Estonian Information Technology Foundation as part of the Tiigrilikool program and by the grant sponsored by the ESSLI 2004 local organization. I’m also grateful to the two anonymous reviewers for useful comments on this work.

---

\(^2\)http://www.google.com/api
References


Towards Unsupervised Learning of Morphology Applied to Ukrainian

Sophia Katrenko
Lviv Polytechnic National University
sophijka@yahoo.com

Abstract.
This paper presents an approach to learn morphology in an unsupervised manner. It primarily focuses on inflectional morphology but it can also be extended to derivational one. The approach makes use of contextual information in order to build inflectional paradigms. Although it has been tested on Ukrainian, it is applicable to other highly inflectional languages.

1 Introduction

Lately, several approaches have been presented to learn morphology in an unsupervised manner, i.e., without usage of any linguistic knowledge. The main argument for applying unsupervised learning to variety of tasks is the low cost of acquiring knowledge.

According to (Goldsmith, 2000), the previously done approaches can be divided into several groups. Some of them aimed at morpheme boundaries identification (Harris, 1967), while others focused, e.g., on derivational rules (Gaussier, 1999). Unsupervised learning of morphology has also been used by (Sharma et al., 2002) to build a lexicon. Not all of the researchers have treated morphology in a classical way, making use of the theory of Whole Word Morphology (Neuvel et al., 2000).

On the other side, as it has been argued by (Goldsmith, 2000), such a learning could be a step towards constructing a linguistic theory. In his research, Goldsmith has not differentiated between derivational and inflectional morphology allowing the derivational and inflectional affixes to represent the same patterns (signatures) which is not appropriate to our approach.

The approach presented in this paper concerns itself with the identification of inflectional affixes. It differs from the previously proposed approaches. Firstly, it makes use of the context of a token and, secondly, it uses clustering methods in order to induce morphosyntactic information. It has been argued by Goldsmith that syntactic and semantic information would not change the learning of morphology drastically. However, as (Schone et al., 2000) have demonstrated, the results can be improved using latent semantic analysis. In this paper, the aim is to explore if syntactic information can be of interest for inducing inflectional paradigms. The intuition
underlying the approach is that if we cluster words w. r. t. parts of speech, we may expect to form clusters containing more granular information. E.g., wordforms belonging to the same part of speech category will be clustered together and, remembering that inflection does not change the part of speech category, one can use clustering to build inflectional morphological patterns. It is to be mentioned that some approaches based on unsupervised learning of morphology have been using labeled corpora. In order to make this approach as much unsupervised as it might be, the morphosyntactic information is to be induced from a corpus. The question arises what can be learnt from raw data having only limited/no knowledge about a given language. Another reason why clustering techniques are used is that for many languages (including Ukrainian in this case) neither annotated corpora nor taggers are available.

Some approaches (Hughes et al., 1994) revealed interesting facts about clustering words using n-grams or given syntactic patterns. They have shown that clustering implies grouping of tokens as to syntactic and semantic similarity. Although our approach relies on the accuracy of clusters, a threshold must be set in order to use the information which is the most reliable. Furthermore, if clusters provide an information about parts of speech tokens belong to and capture different inflectional morphemes, it is possible to induce inflectional morphemes for a given base using the information about inflectional classes of other bases it shares the same clusters with.

There is a range of applications making use of morphological properties of words (such as stemming, lemmatization for text categorization tasks or hyphenation). Besides this many tree-banks include lemmatization into their annotation schemes.

The paper is organized as follows. We first outline the approach, providing then more information on clustering methods and the way of producing signatures. The method presented here is tested on Ukrainian corpora as it is described in Section 4.

2 A Proposal to Learn Morphology

This approach includes the following steps:

- Find K closed-class words based on the token frequency
- Find the N most frequent open-class words
- Form the vectors containing tokens chosen on the previous two steps such, that each vector can be described as one consisting of number of closed-class words occurrences w.r.t. a given open-class word. Let a size of context window be X, then each vector is (X-1)*K long. As vectors are constituted for each of the closed-class words, the number of vectors is N.
- Apply clustering methods
- Group members of each cluster in such a way, that each group is formed by the alignment of wordforms.
- Form signatures

Finding the closed-class words is facilitated by using token frequency lists. (Smith et al., 1995) have explored several methods of acquiring closed-class words from corpora. The one based on a frequency list has been called a zero-order test. It is a well-known fact that for varieties
of languages the most frequent words are prepositions or articles, i.e., closed-class words. The problem remains how to split a list in a way that two sublists with closed- and open-class words are produced. Unfortunately, it can not be done fully automatically requiring, thus, a human judgement.

2.1 Clustering

In cluster analysis objects are grouped together if they are closely related. The clustering methods are distinguished according to the way clusters are built and fall into non-hierarchical and hierarchical methods. Degree of similarity is measured by distance functions, such as Euclidian or Manhattan metrics.

(Hughes et al., 1994) have experimented with a clustering of words in order to induce parts of speech information. They have explored several clustering techniques such as single linkage, complete linkage, median, etc. The best results have been obtained using Ward’s clustering method and Manhattan metric.

Definition 1 Given two m-dimensional objects \( x = (x_1, x_2, \ldots, x_m) \) and \( y = (y_1, y_2, \ldots, y_m) \) the Manhattan distance is measured as follows:

\[
d(x, y) = \sum_{i=1}^{m} |x_i - y_i|
\] (13.1)

Definition 2 Ward’s hierarchical clustering method uses the notion of information loss and looks for partitions such, that the information loss associated with each cluster is minimized. The clusters are thus grouped together if their fusion leads to minimal information loss which is defined in terms of smallest sum-of-squares criterion.

The literature on clustering indicates that it is far more important to choose an appropriate metric than a clustering method. For our purposes hierarchical clustering methods are of the most interest. On the one hand, one can represent results as a dendrogram - graphical representation of clusters hierarchy - and cut it to produce a desired number of clusters. On the other hand, such methods as K-means depend on the choice of the number of clusters and would not be a good choice for clustering wordforms. The hierarchical methods of clustering fall into agglomerative and divisive. If there is a strong clustering tendency, such methods provide similar results.

The evaluation of clustering results has previously been performed by means of a tagset, i.e., a cluster is said to be a ‘noun cluster’ if the most members of it are nouns. (Roberts, 2002) has tested clustering methods on English and Spanish corpora but despite a rich morphology of Spanish, only a small tagset has been used. We believe, morphosynactic features can be captured by means of clustering as well. Thus we have also done it in order to prove this hypothesis. In general, however, the approach of acquiring inflectional paradigms does not require this kind of evaluation, since it implies that wordforms will be clustered w.r.t. parts of speech but it is not intended to do tagging.

2.2 Producing signatures

A definition of a signature is close to one given by (Goldsmith, 2000). However, as it has been mentioned earlier, the latter includes all kind of affixes while our looks only for inflectional ones.
Thus, a signature is to be defined as follows:

**Definition 3** A signature $S = < B, I >$ is a morphological pattern consisting of a set of bases $B$ and a set of inflections $I$, such that $B \neq \emptyset$

**Algorithm of acquiring signatures**

1. Set a threshold $T_{set} = \alpha, \alpha \in [0, 1]$. The set of signatures is empty, $S = \emptyset$
2. Finding signatures
   For each cluster $C_i$, $i \in I; \forall (j \in I \land k \in I \land j \neq k) : C_j \cap C_k = \emptyset$
   $\bigcup_{i=0}^{n} C_i = C, k_c = \text{card}(C_i)$

   When aligning wordforms from the end, find a signature $S_k = < B_k, P_k >$
   where $B_k$ is a set of possible bases,
   and $P_k$ is a set of possible inflectional affixes, each member of which is the
   longest substring obtained by aligning.
   Let $k = \text{card}(B_k)$
   Find a ratio of $k$ to a size of cluster $C_i$, $T = k/k_c$
   If $T \geq T_{set}$, add a signature $S_k$ to set of signatures $S$,
   $S = S \cup \{S_k\}$

   If there are several signatures which may be produced for a single cluster, then repeat
   the procedure.

   Signatures, produced at this step will consist of a set of bases and a one-element set of
   inflectional affixes each,
   $S = \{S_1, S_2, \ldots\}, B = \{B_1, B_2, \ldots, B_k, \ldots\}, P = \{P_1, P_2, \ldots, P_k, \ldots\},$
   $\forall (P_i \in P): \text{card}(P_i) = 1$
3. Forming extended signatures
   **Form a new set of extended signatures $S^N = \{S_1^N, \ldots, S_d^N\}$,**
   whose element is $S_i^N = < B_i^N, P_i^N >$
   built by means of comparison $B_g$ and $B_d$ as
   $P_i^N = P_g \cup P_d$
   $B_i^N = B_g \cap B_d$, $B_i^N \neq \emptyset$

**Signatures prediction algorithm**

1. Perform Step 1 from the algorithm of producing signatures
2. Perform Step 2 from the algorithm of producing signatures
3. look for a signature $S_j = < B_j, P_j >$, such that $B_k \cap B_j \neq \emptyset$, $B_j \subset B_k$
   and form a new signature $S_R = < B_k \setminus B_j, P_j >$

   There are several points one has to account for. Obviously, wordforms grouped into one
   cluster and having no endings they share do not necessarily belong to different parts of speech.
   They are likely to be wordforms with zero-infections. On the other hand, it would be useful to
   induce more knowledge from the built signatures. One way of doing this is to look for signatures
   whose bases are subsets of other signatures. That is what we call "extended signatures". Given
   signatures received on the step 2, we can try to form extended signatures looking for all possible
inflections for given bases. It is more difficult to evaluate overlapping signatures, in this case the overlapping part of them must be found and, using the information about the cardinal number of sets, the decision is made if the signatures are merged together. At the moment, this is left to future work.

3 Inflectional paradigms in Ukrainian

The method of acquiring inflectional affixes can be used for different inflective languages. In this paper one of them is explored, Ukrainian, which is a highly inflectional Slavic language. The approach is restricted to the learning of morphology since it would be rather difficult to learn morphophonology due to its allomorphy phenomena. Historical phonology of Ukrainian is extensively described in (Shevelov, 1979). There are four groups of declining nouns in Ukrainian. They are defined using the information about inflections in nominative case and a consonant a base ends with. Verbs can be conjugated w. r. t. tense, aspect and agreement features (person and number). Table 13.1 provides some examples on declining nouns.

<table>
<thead>
<tr>
<th>case</th>
<th>Type I</th>
<th>Type II</th>
<th>Type III</th>
<th>Type IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>nom.</td>
<td>a hope</td>
<td>a surname</td>
<td>a quality</td>
<td>a name</td>
</tr>
<tr>
<td>gen.</td>
<td>наді-я</td>
<td>прізвиць-е</td>
<td>якість</td>
<td>ім'я</td>
</tr>
<tr>
<td>dat.</td>
<td>наді-и</td>
<td>прізвиць-а</td>
<td>якост-і</td>
<td>імен-і</td>
</tr>
<tr>
<td>acc.</td>
<td>наді-ю</td>
<td>прізвиць-у</td>
<td>якість</td>
<td>ім'я</td>
</tr>
<tr>
<td>instr.</td>
<td>наді-єю</td>
<td>прізвиць-єм</td>
<td>якост-ю</td>
<td>імен-ем</td>
</tr>
<tr>
<td>prep.</td>
<td>наді-и</td>
<td>прізвиць-і</td>
<td>якост-і</td>
<td>імен-і</td>
</tr>
<tr>
<td>voc.</td>
<td>наді-є</td>
<td>прізвиць-е</td>
<td>якост-е</td>
<td>ім'я</td>
</tr>
</tbody>
</table>

Figure 13.1: Groups of noun declension

4 Experiments

4.1 Data

The corpus used to test this approach on consists of 55,984 distinct tokens and 368,572 orthographic words. It includes texts from the newspaper "Dzerkalo tyznya" ("Mirror-Weekly") varying in size and topics.

Considering table 13.1 the task of learning inflectional morphology can be seen as construction of subsets (if possible, the whole sets) of morphological paradigms. As it follows from algorithms presented here, the signatures should cover different declination groups.

4.2 Evaluation

The settings being used are the following:
• the 12 most frequent closed-class words,
• the 200 most frequent open-class words to be clustered,
• the size of the context window - bigrams right after and before given open-class word.

When clustering, several methods and metrics have been used.

![Dendrogram](image)

**Figure 13.2: A part of a dendrogram**

Manhattan metric and Ward's method proved to be the best choice, although such techniques and metrics as nearest-neighbour (single linkage), furthest-neighbour (complete linkage), squared Euclidian distance have been tested as well.

As expected, the words have been grouped into clusters according to the parts of speech they belong to. It is also worth noting that clustering captures not only regular forms but also suppletion-like. E.g., the form *je* (3rd Person, sing. of *to be*) is associated with other forms of *to be*, such as *buio, buly* (past tense, 3rd person sing and past tense, 3rd person pl.), etc. A part of a dendrogram is presented on a figure 13.2.

As it has already been mentioned, some evaluation has been carried out. Clustering is one of the unsupervised learning methods, i.e. there are no predefined classes. In our case, the evaluation is nevertheless done using expert knowledge about a given language. Precision of each cluster is to be calculated as a ratio of the number of words reflecting the most common tag for a cluster, to the size of this cluster. Then, using precision of each cluster, an overall precision is calculated as an average of them. Depending on the size of a tagset, the accuracy of clustering varied from 92.93% for a small tagset (representing mainly parts-of-speech) to 70.79% for a large one (containing tags with more information about parts-of-speech).

Clusters are used as an input to algorithms of building signatures. Some of the signatures received are given below:
The method proposed here deals with built signatures to infer more information producing new signatures. Even though a word does not occur in a given corpus (figure 13.4), it may be inferred building, e.g., a signature on the base of given ones \(^1\) (figure (1), signatures 1 and 2).

\[\text{Signature 4 (using Signatures prediction algorithm):} \]
\[
\begin{align*}
\text{вклад} & \quad \text{президент} \\
\text{работ} & \quad \text{прав} \\
\text{країн} & \quad \text{країн} \\
\text{людин} & \quad \text{людин}
\end{align*}
\]

\[\begin{array}{c}
\{a\} \\
\{i\}
\end{array}\]

Figure 13.4: Sample from the output of the signatures prediction algorithm.

For the tests the threshold \(T_{set}\) was set to 0.4 and only clusters with more than 2 members were considered. Table 13.1 illustrates some of the results obtained by means of Ward’s and average linkage clustering methods in combination with such distances as Euclidian, squared Euclidian and the one based on cosine of vectors. \(N\) and \(K\) correspond to the numbers of open-class words and closed-class words, respectively. In this case, precision was calculated using reduced tagset containing parts of speech categories as described above. Precision given in parentheses was computed using weights, i.e., information about how many members each cluster consists of. This measure is especially important in case there is a high number of one-member clusters. Simple averaging will result in high scores and will not reflect the actual accuracy. E.g., centroid clustering has resulted in 30 clusters one of which included 149 words from the 200. Employing weights in this case yields lower precision than simple average (60.5% against 88.1%).

Considering recall to be a percentage of the correct elements of a signature which are found, recall for the signatures received by making use of clusters from the test ‘Ward’s method+squared Euclidian distance’ equals 56%. This value can be explained by the fact that the relevant elements of a signature have been assigned to the different clusters. Since a signature is built on the base

\(^1\)Assuming they are correct
of single cluster, it does not include its possible members from other clusters (in the best case two different signatures will be constituted).

Signatures received using different clustering methods and distances do not differ very much from each other. Those provided on figure 13.5 present patterns for nouns, adjectives and pronouns. It is usually a case that pronouns and adjectives are clustered together and, as a result, they can belong to the same pattern. Though the words are of different parts of speech, they are correctly split into base and inflectional affixes. Moreover, as they follow the same declination rules, more information about possible inflectional affixes in other cases than given can be obtained.

The more precise evaluation of results is needed. There are several issues to be taken into account. First, the precision and recall of the signatures relies on clustering accuracy. The threshold was proposed in order to relax this. The first type of errors is therefore treating words as being of the same part of speech when, in fact, they are not. Another type of errors occurs when an inflection set contains inflection+suffix (see figure 13.5). Both types are rare and can be avoided by careful choice of clusters and the threshold. This is also the reason why one prefers clusters with relatively high number of members. But it should be mentioned that very small number of clusters results in lower precision.

Second, the algorithms do not consider words with zero-inflections so a criterion must be undertaken indicating if all of them form one signature. Finally, the evaluation needs to be carried out separately on simple signatures and extended ones.

### 4.3 Using Linguistica

The same corpus was used as input data to Linguistica software implemented by (Goldsmith, 2000). However, the direct comparison is not possible. This software uses all words in a corpus when looking for signatures (note, a definition of signature given by Goldsmith differs

<table>
<thead>
<tr>
<th>Method</th>
<th>Metric</th>
<th>Precision</th>
<th>N</th>
<th>K</th>
<th># clusters</th>
<th># signatures</th>
<th># predicted</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ward’s</td>
<td>Sq. Euclidian</td>
<td>85.93%</td>
<td>200</td>
<td>12</td>
<td>30</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(79%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>Cosine</td>
<td>85.47%</td>
<td>200</td>
<td>12</td>
<td>30</td>
<td>7</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>linkage</td>
<td>of vectors</td>
<td>(78%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Ward’s</td>
<td>Euclidian</td>
<td>92.93%</td>
<td>200</td>
<td>12</td>
<td>40</td>
<td>7</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Ward’s</td>
<td>Sq. Euclidian</td>
<td>62.7%</td>
<td>679</td>
<td>19</td>
<td>60</td>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 13.1: Results
from ours). This results in increasing of the number of signatures while the method proposed here
uses the N most frequent words in a corpus (but one may also expect to extend it to consider other
words in case signatures are accurate). The output of Linguistica included 556 suffixes and 5,041
signatures. The top-ranked signatures usually covered adjectives. As noted by (Goldsmith,2000),
two suffixes can collapse into one and it is often the problem with noun signatures.

Linguistica turned out to be useful when finding probable affixes but as already mentioned,
the signatures (in terms of Goldsmith) include both derivational and inflectional affixes making
further processing more complicated. Consider, e.g., a signature produced by Linguistica:

(1) \{doziln\{ist'.iszym.a.e.ym.yh.osti\}

It includes stem doziln and a set of affixes. The words this signature is built on are of different
parts of speech, namely adjective, adverb and noun. It seems it would be difficult to infer
more knowledge or to extend already given signatures. In contrast, the signatures prediction
algorithm makes use of the information found in two signatures obtained by algorithm of acquiring
signatures. It is able to produce signatures with words which do not occur in a corpus.

5 Conclusions and future work

The approach outlined in this paper proved to be successful when learning inflections. As it has
been shown, the inflectional paradigms can be constituted by means of such signatures. However,
in some cases there are difficulties with building signatures. Ukrainian is characterized by vowel
allophony (see examples in Section 3). It can be seen on the example of a word rik (a year) -
when declining it changes to riku, rozi. One can thus rely on inflections received but the bases
will not be added to the extended signatures. One of the possible ways of handling it is to use
Levenshtein distance. This means to incorporate knowledge about vowels and consonants tending
to change when declining. (As noted by (Yarowsky et al., 2000), vowels but not consonants usually
change but it is not necessary true).

This method does not consider parts-of-speech ambiguity since the hard clustering has been
used. It would be interesting to try soft clustering methods which allow multiple membership,
so wordforms could be assigned to several clusters. It may be particularly interesting to combine
this method with already proposed statistical ones. Another direction of further research may be
learning of derivational affixes.

Acknowledgements

The work presented in this paper was supported by EACL. I would also like to thank anonymous
reviewers for their valuable suggestions and remarks.
Figure 13.5: Signatures
References


Zellig Harris (1967). *Morpheme boundaries within words: Report on a computer test*. Transformations and Discourse Analysis Papers 73, Department of Linguistics, University of Pennsylvania.


Multiple Long Distance Scrambling in Korean

JIEUN KIAER
King's College London
kiaer@dcs.kcl.ac.uk

ABSTRACT.
This paper aims to explain the incremental nature of structure building from left to right with special reference to multiple long distance scrambling construction in Korean. In this paper, I adopt parsing-based framework, Dynamic Syntax (Kempson, Meyer-Viol, and Gabbay 2001), and show characteristics of multiple long distance scrambling in Korean as consequences of monotonic-tree growth processes. In this paper, I shall show that on-line interaction between prosody and case specification play a crucial role in disambiguating possible parse sequences and optimise structure-building with relatively less effort. The challenge of this paper is to find the best way to pair syntax and phonology to most closely reflect the nature of the human parser.

1 Introduction
1.1 Subject Scrambling in Korean
Korean is a verb-final language with very free NP-ordering: it allows both clause-internal scrambling as in (1-a) to (1-b) or long-distance scrambling as in (1-c) to (1-d).

(1) a. Jina-ka sakwa-rul mek-ess-ta
   Jina_{NOM} apple_{ACC} eat_{PAST,DECL}
   Jina ate an apple
b. Sakwa-rul Jina-ka mek-ess-ta
   apple_{ACC} Jina_{NOM} eat_{PAST,DECL}
   Jina ate an apple.
c. Mina-ka Jina-ka sakwa-rul mek-ess-ta-ko malhay-ss-ta
   Mina_{NOM} Jina_{NOM} apple_{ACC} eat_{PAST,DECL,COMP} say_{PAST,DECL}
   Mina said that Jina ate an apple
   Jina said that Mina ate an apple. [Subject Scrambling]
d. Sakwa-rul Mina-ka Jina-ka mek-ess-ta-ko malhay-ss-ta
apple\text{ACC} Mina\text{ NOM} Jina\text{ NOM} \text{eat}_{\text{PAST}, \text{DECL}, \text{COMP}} \text{say}_{\text{PAST}, \text{DECL}}

An apple, Mina said that Jina ate [only object sakwa scrambled]
Jina said that Mina ate an apple [object-subject pair scrambled]

In Korean even subjects can scramble across clause boundary as in (2):\textsuperscript{1}

(2) Mina-ka Jina-ka pemin ira-ko malhay-ss-ta
Mina\text{ NOM} Jina\text{ NOM} criminal BE_{\text{COMPL}} \text{say}_{\text{PAST,DECL}}
(a) Mina said that Jina is a criminal.
(b) Jina said that Mina is a criminal. [Subject Scrambling]

In the on-line parsing process, the role of phonological information such as prominence, pitch contour or intonational break is crucial. For example, for (2) to have reading (a), there should be no intonational break between Mina-ka and Jina-ka and Jina-ka becomes more prominent than Mina-ka. On the other hand, for (2) to have reading (b), there must be an intonational break between Mina-ka and Jina-ka and Jina-ka becomes more prominent\textsuperscript{2} than Mina-ka. By combining the effects of processing case and intonation specifications, I want to show that structure building in Korean is incremental, rather than remaining unresolved until the end of parsing, as in LFG or HPSG approaches.\textsuperscript{3}

Based on Combinatory Categorial Grammar, Hoffmann (1995) explains local scrambling in Turkish by assigning a set of permutable categories to each verb rather than doubling lexicon by multiplying categories. Baldridge (2002) adds to this account, showing that long-distance scrambling requires in addition type-raising and forward cross composition. His Multi-Modal Combinatory Categorial Grammar, an extension of Combinatory Categorial Grammar (Steedman 2000), assumes \textit{Domain Union} Operation (Reape 1994) and thus there is no strict order preserving like CCG. Also, each combinatorial rule is applied according to which mode is chosen (whether right, left, or order-neutral slash for instance). In short, by loosening order-perserving nature in CCG and setting up a set of permutable categories, Baldridge is able to capture basic local and long-distance scrambling phenomenon in Turkish. Such explanation works perfectly well in Korean when there exists only one level of subordination (e.g., local scrambling). However, a problem occurs when there is more than one level of subordination as in (2).

Baldridge (2002) escapes the problem of yielding multiple choice in long-distance scrambling by assigning a different grammatical function to Fatma and Esra’nin as in (3).

(3) Kitabi Fatma Esra’nin okudugunu biliyor
book_{\text{ACC}} Fatma_{\text{ NOM}} Esra_{\text{ GEN}} read_{\text{GER}} \text{ know}_{\text{PROG}}

As for the book, Fatma knows that Esra read it.

Accordingly, the categories of predicates \textit{okudugunu}‘read’ and \textit{biliyor} become different as in (4).

\textsuperscript{2}A focused item has a longer duration, higher amplitude and a larger pitch range than a neutral item. See Jun and Lee (1998). Sun-Ah Jun (p.c.) points out that prominent/focused words are followed by intonational break occasionally.
\textsuperscript{3}Thanks to Mary Dalrymple and Ruth Kempson for many useful discussions.
(4) \( \text{okudugunu} := S_{\text{ACC}} \{ \text{NP}_{\text{GEN}}, \text{NP}_{\text{ACC}} \} \)
\( \text{biliyor} := S \{ \text{NP}_{\text{NOM}}, \text{NP}_{\text{ACC}} \} \)

So, it becomes clear which verb takes which NP as its subject and there is no confusion. But re-consider (2) in this connection:

(2) Mina-ka Jina-ka pemini ira-ko malhay-ss-ta
\( \text{Mina}_{\text{NOM}} \text{Jina}_{\text{NOM}} \text{criminal BE}_{\text{COMP}} \text{say}_{\text{PAST,DECL}} \)
(a) Mina said that Jina is a criminal.
(b) Jina said that Mina is a criminal. [Subject Scrambling]

Following Baldridge (2002), pemini-‘be a criminal’ and malhay-‘say’ will have a category as in (5) and now the problem occurs as there is no rule to match which verb with which subject unless we set up some language-specific mode for Korean.\(^4\)

(5) pemini:= \( S \{ \text{NP}_{\text{NOM}} \} \)
malhay:= \( S \{ \text{NP}_{\text{NOM}}, \text{NP}_{\text{ACC}} \} \)

Moreover, Multi-Modal CCG cannot explain why (6) and (7) can’t have (b) reading but only (a) reading, whereas (8) can have both (a) and (b) reading:

(6) Yuna-ka sakwa-rul cwu-ess-ta-ko Jina-ka Mina-ekey pokohay-ss-ta
\( \text{Yuna}_{\text{NOM}} \text{apple}_{\text{ACC}} \text{give}_{\text{PAST,DEC,COMP}} \text{Jina}_{\text{NOM}} \text{Mina}_{\text{DAT}} \text{report}_{\text{PAST,DEC}} \)
(a) Jina reported to Mina that Yuna gave an apple
*(b) Jina reported that Yuna gave an apple to Mina.

(7) Yuna-ka sakwa-rul cwu-ess-ta-ko Jina-ka pokohay-ss-ta Mina-ekey
\( \text{Yuna}_{\text{NOM}} \text{apple}_{\text{ACC}} \text{give}_{\text{PAST,DEC,COMP}} \text{Jina}_{\text{NOM}} \text{report}_{\text{PAST,DEC}} \text{Mina}_{\text{DAT}} \)
(a) Jina reported to Mina that Yuna gave an apple
*(b) Jina reported that Yuna gave an apple to Mina.

(8) Mina-ekey Yuna-ka sakwa-rul cwu-ess-ta-ko Jina-ka pokohay-ss-ta
\( \text{Mina}_{\text{DAT}} \text{Yuna}_{\text{NOM}} \text{apple}_{\text{ACC}} \text{give}_{\text{PAST,DEC,COMP}} \text{Jina}_{\text{NOM}} \text{report}_{\text{PAST,DEC}} \)
(a) Jina reported to Mina that Yuna gave an apple.
(b) Jina reported that Yuna gave an apple to Mina.

\( \text{Mina-ekey} \) in both (6) and (7) should be interpreted within the structure projected by the matrix verb \( \text{pokohay} \)-‘report’ because of the Right Roof Constraint, a well-known but little understood constraint (Ross 1967). However, Baldridge (2002) cannot explain this Right Roof Constraint. According to him, \( \text{Mina-ekey} \) in (6) and (7) should equally well be interpreted as argument to the matrix verb \( \text{pokohay} \)-‘report’ or to the subordinate verb \( \text{cwu} \)-‘give’ as in (8). Therefore, the asymmetry found between (7) and (8) is hard to express in Baldridge’s terms. With respect to rather different case-stacking phenomena, based on the Lexical Functional Grammar (LFG) framework, Nordlinger (1998) argued that in radically non-configurational languages like Wambaya, case morphology can provide exactly the same types of information about grammatical

\(^4\)Yet, it’s still problematic unless we allow two different modes to get two readings.
function that can be provided by phrase structure in more configurational languages, hence in principle able to disambiguate possible relations between the NP arguments and the verb. Furthermore, she showed that case morphology can also construct information about phrases other than the one to which the suffix is attached. In particular, in Walpiri, the outermost case-suffix provides information about other NPs to be found in the string and their structural role. With such help of an additional outermost case marker, structure-building in Walpiri can be done with relatively less effort and less syntactic ambiguity. In Korean, however, no such morphological help is available to distinguish available interpretations, and LFG would have no obvious mechanism to distinguish the various interpretations of a string such as (2), which is ambiguous when prosodic factors are ignored. In Section 3, based on Dynamic Syntax framework, I will show how case and prosody interact and builds up optimised structure together with relatively less effort. In particular, I argue that prominence and intonational break\(^5\) helps the parser to make a local decision at each step of parsing.

1.2 Pair-wise reading in (Multiple) Long Distance Scrambling

A particular challenge posed by scrambling is to explain multiple long-distance scrambling as in:

(9) Sakwa-rul Mina-ka Jina-ka mek-ess-ta-ko malhay-ss-ta
apple\textsubscript{ACC} Mina\textsubscript{NOM} Jina\textsubscript{NOM} cat\textsubscript{PAST,DECL,COMP sayPAST,DECL}

An apple, Mina said that Jina ate [only object sakwa scrambled]
Jina said that Mina ate an apple [object-subject pair scrambled]

This can involve either just the object or the pair of subject and object being construed as part of the subordinate structure. The latter is particularly problematic for standard accounts, since it appears to involve multiple long-distance dependency. Here again each reading can be distinguished in the on-line parsing process, when proper phonological information is accessible. For example, when there is an intonational break between sakwa-rul and Mina-ka, two lexical elements cannot be interpreted as one constituent or one pair. Yet, when there is a break between the first subject Min-ka and the second subject Jina-ka, the object sakwa-rul and Mina-ka must yield a pair-wise reading, even though there is no obvious way in which they form a constituent. Therefore, (10) sounds odd when the intonational break is between sakwa-rul and halmeni-ka, not between halmeni-ka and Jina-ka:\(^5\)

(10) Sakwa-rul halmeni-ka Jina-ka tu-sie-ss-ta-ko malhay-ss-e
apple\textsubscript{ACC} the grandmother\textsubscript{NOM} Jina\textsubscript{NOM} cat\textsubscript{HON,PAST,DECL,COMP sayPAST,DECL}

Jina said that the grandmother ate an apple.

Previous analyses of such surprising constituents attempted to motivate a distinct movement, (e.g. the remnant movement analysis of Koizumi (2000) and the oblique movement analysis of Takano (2002) to create a constituent. Yet, the actual motivation of such movement is not clear. On the contrary, CCG generalises surface constituency to provide substrings like Marcel proved and even a policeman a flower the full status of constituents. Prevost and Steedman (1994)

---

\(^5\)According to Pierrehumbert and Beckman (1988), prosodic phrases are solely specified in terms of pitch accent and the boundary.

\(^6\)HON means honorific.
argue that the two tunes $L + H^*$ LH $%$ and $H^*L$ are respectively associated with the “theme” and “rheme” of the sentence. Based on this observation, Steedman (2003) claimed that combinatorial rules require categories with proper intonational pattern. This might well be sufficient as long as the supposed concepts of theme and rheme are computed for a simple predicate-argument structure, yet the problem occurs in explaining multiple long-distance dependency across an embedded structure. With apparently unrelated data, Nordlinger (1998) shows how stacked-cases in Wambaya unfold structure corresponding to such surprising constituents, and is able to build up constituents step by step using a so-called Inside Out Function. However, her analysis depends on the presence of a morphological trigger, and for Korean scrambling where there is no morphological distinction between the multiple- and simple- long-distance scrambled interpretations, there is no basis for explaining the facts. In this paper, I show how such a pair-forming operation in long distance scrambling can be seen to be derived from general tree-growth processes in Dynamic Syntax.

2 Dynamic Syntax

2.1 Preliminaries

In terms of its internal architecture, Dynamic Syntax (DS) is like minimalism (Chomsky 1995) in that the level of logical form, a tree structure representation of functor-argument relations, is the only syntactic level, and the sole explanation of natural language syntax resides in the progressive projection of such structure. Further, DS trees are transparent representations of semantic content, which are incrementally built up, following the left-to-right dynamics of natural language processing in context. The rootnode of a tree is decorated with a logical term ($Fo(a)$), a type specification, ($Ty(t)$), and a treenode identifier ($Tn(0)$), and all other nodes are decorated with subterms of that formula with appropriate types and treenode identifier.$^{89}$

2.2 Structural Underspecification and Update: Using three kinds of Adjunction

The central concept of Dynamic Syntax is the building up of tree-structure representations of content, articulating different forms of structural underspecification and update within the construction process. These actions may be computational or lexical. For example, verbs in pro-drop languages project a template for an entire tree of type $t$. Though such actions construct fixed tree-relations, an important property of the tree-growth system is that nodes in the partial tree may be introduced without having particular relation initially fixed; and their syntactic position

---

7 Thanks to Mary Dalrymple for useful discussion on this issue - regarding the combined use of Inside Out and Outside In function.

8 Only a small number of types is used, $Ty(t)$, $Ty(t)$, $Ty(e \rightarrow t)$, for example, and for all such types, $Formula$ values may include metavariables, $(U, V)$, in particular for pronouns, which get updated during construction.

Nodes are described by modal operators, $\langle \downarrow \rangle$, $\langle \uparrow \rangle$, for daughter and mother respectively, and $\langle L \rangle \langle L^{-1} \rangle$, defining a LINK relation between trees, and others defined in terms of these, eg $\langle \uparrow \rangle$, $\langle \downarrow \rangle$ and $D_{r}, U$ which are the reflexive transitive closure of LINK and $\downarrow$ and of Inverse-LINK and $\uparrow$ respectively (functor daughter relations and argument daughter relations are distinguished as $\downarrow_1$ and $\downarrow_0$ respectively).
may be found/resolved subsequently. This is the basis of long-distance dependency, where the concept of movement is replaced by one of positional underspecifications. There are three types of Adjunction process depending on whether such underspecification has to be (i) locally determined (i.e., within an individual predicate-argument array) - Local* Adjunction,

\[
\{\ldots \{Tn(a), \ldots, \diamond \}\ldots \} \\
\{\{Tn(a), \ldots\} \ldots \{\langle \uparrow_0 \rangle Tn(a), ?\langle \uparrow_1 \rangle Tn(a), \ldots, \diamond \}\} 
\]

(ii) determined within a single tree (i.e., within an individual proposition) - *Adjunction

\[
\{\{Tn(a), \ldots ?Ty(t), \diamond \}\} \\
\{\{Tn(a), \ldots, ?Ty(t)\} \ldots \{\langle \uparrow_0 \rangle Tn(a), \ldots, ?\exists x Tn(x), ?Ty(e), \diamond \}\} 
\]

or (iii) determined merely within the overall construction process (i.e., possibly across a sequence of trees) - Generalised Adjunction.

\[
\{\ldots \{Tn(a), \ldots, \diamond \}\ldots \} \\
\{\{Tn(a), \ldots\} \ldots \{\langle U \rangle Tn(a), \ldots, \diamond \}\} 
\]

These are the direct analogue of the locality restrictions associated with anaphora resolution.\(^{10}\) The availability of these three processes means that there will be more than one alternative for any initial step of parsing long-distance scrambling construction in Korean, and as we shall see the role of intonation is to eliminate the disjunctions which these alternatives set up. Within a simple clause, either process of *Adjunction will allow noun phrases to be successively processed, and the role of case is to indicate relative positioning in the emergent semantic tree structure:

(11) sakwa-rul Jina-ka mek-ess-ta
     apple\text{ACC} Jina\text{NOM} eat_{\text{PAST,DECL}}
     Jina ate an apple.

Nominative, the suffix -\text{ka}, indicates a node to be immediately dominated by the top node projected from the clause, ?\langle \uparrow_0 \rangle Ty(t); accusative, the suffix -\text{rul}, indicates a node to be immediately dominated by the predicate node, ?\langle \uparrow_1 \rangle Ty(e \rightarrow t), and these may be used to induce an abduction step determining that the node they decorate may be taken immediately as fixed (see Kempson (2003)).\(^{11}\)

**Local Update of the nodes decorated by Case-marked NPs via Lexical Action of Case markers: Parsing (11)**

(i) Parsing Sakwa-rul:

\(^{10}\)In the following trees, both variants of *Adjunction are indicated by a dashed line. Generalised Adjunction is indicated by a dotted line.

\(^{11}\)Lexical actions are projected by the words of the language and map one tree description into the next one, adding information in the process. Lexical actions are defined as conditional actions (see Kempson, Meyer-Viol, and Gabbay (2001)).
3. DS Analysis on (multiple) long distance scrambling

The primary indicator of scrambling for both single and multiple long-distance scrambling is the use of interruption.

According to what the system licenses, both Local Adjunction and *Adjunction could apply to the parsing of the initial sahara-ru in (12). If the updating process is to be in any way delayed by the building of some unified node for later update, there will be no indication of this, and in such a case, interruption serves to preclude the possibility of parsing sahara-ru morphology, structural, or contextual input. I add to this by incorporating phonological information into the input information which the parsing process manipulates (see also Cam.

(i) Parsing Jina-ka

Case specification helps the parser to updates the nodes decorated by case-marked NPs to the immediately following local structure. Yet, there are some cases where such update procedure is somehow delayed, and the nodes decorated by case-marked NPs are not updated immediately to its local structure. These are the long-distance scrambling data to which now turn.

(ii) Parsing Jina-ka

* updates (\( \ast \) \( Tn(0) \)) into (\( Tn(0) \)).
IF $Fo(\psi), Ty(x)$,
THEN IF $\langle \uparrow o \rangle \langle \uparrow t \rangle Ty(t)$
BREAKEND
ELSE Abort
ELSE Abort

The result of parsing a noun phrase with the requisite intonation will accordingly be to reduce the alternative parse sequences to only that involving *Adjunction.\textsuperscript{12} We can see the steps up to the point at which the subordinate verb mek-‘eat’ is parsed in (12):

\begin{center}
\begin{tikzpicture}
  \node (Tn) {$\langle \uparrow o \rangle \langle \uparrow t \rangle Ty(t)$};
  \node (Fo) [above left] {$\langle \uparrow o \rangle Ty(e \rightarrow t)$};
  \node (M) [above right] {$\langle \uparrow o \rangle Tn(o)$};
  \node (J) [below right] {$\langle \uparrow o \rangle \langle \uparrow U \rangle Tn(o)$};
  \node (W) [below left] {$\langle \uparrow o \rangle \langle \uparrow U \rangle Tn(o)$};
  \node (Mek) [below right] {$\langle \uparrow o \rangle \langle \uparrow U \rangle Tn(o)$};
  \node (Mina) [left] {$Fo(x, Mina)$};
  \node (Sakwa) [left] {$Fo(Sakwa(x))$};
  \node (Mek) [right] {$Fo(Mek')$};
  \node (W) [below left] {$Fo(W)$};

  \draw[->] (Tn) -- (Fo);
  \draw[->] (Tn) -- (M);
  \draw[->] (Tn) -- (J);
  \draw[->] (Tn) -- (W);
  \draw[->] (M) -- (Mina);
  \draw[->] (M) -- (Sakwa);
  \draw[->] (W) -- (Mek);
  \draw[->] (W) -- (Mina);
  \draw[->] (W) -- (Sakwa);
\end{tikzpicture}
\end{center}

The first step involves parsing sakwa-rul with the intonational break, which results in the node decorated by sakwa-rul being introduced as unfixed. In the next step, Mina-ka is introduced at a node locally fixed as subject of its local, here main, structure. In order to parse the second subject, there has to be an intervening step of Generalised Adjunction, with Jina-ka then being processed like Mina-ka as the subject of the predicate mek, which induces another local structure of $Ty(t)$ with a meta-variable $W$ decorating the object-denoting node. In order to create a structure to which the step of Merge can apply to unify this node decorated by $Fo(W)$ with the sentence-initial unfixed node, a step of abduction\textsuperscript{13} has to take place to enrich what was initially only a weak embedding relation to a fixed relation of immediate subordination.\textsuperscript{14}

\begin{center}
\begin{tikzpicture}
  \node (Tn) {$\langle \uparrow o \rangle \langle \uparrow t \rangle Ty(t)$};
  \node (Fo) [above left] {$\langle \uparrow o \rangle Ty(e \rightarrow t)$};
  \node (M) [above right] {$\langle \uparrow o \rangle Tn(o)$};
  \node (J) [below right] {$\langle \uparrow o \rangle \langle \uparrow U \rangle Tn(o)$};
  \node (W) [below left] {$\langle \uparrow o \rangle \langle \uparrow U \rangle Tn(o)$};
  \node (Mek) [below right] {$\langle \uparrow o \rangle \langle \uparrow U \rangle Tn(o)$};
  \node (Mina) [left] {$Fo(x, Mina)$};
  \node (Sakwa) [left] {$Fo(Sakwa(x))$};
  \node (Mek) [right] {$Fo(Mek')$};
  \node (W) [below left] {$Fo(W)$};

  \draw[->] (Tn) -- (Fo);
  \draw[->] (Tn) -- (M);
  \draw[->] (Tn) -- (J);
  \draw[->] (Tn) -- (W);
  \draw[->] (M) -- (Mina);
  \draw[->] (M) -- (Sakwa);
  \draw[->] (W) -- (Mek);
  \draw[->] (W) -- (Mina);
  \draw[->] (W) -- (Sakwa);
\end{tikzpicture}
\end{center}

\textsuperscript{12}The steps subsequent to this follow the sequence of steps proposed in Kempson (2003) for Japanese scrambling.

\textsuperscript{13}Such abduction process is driven by the complementizer ko- which indicates the successful completion of the subordinate structure.

\textsuperscript{14}See Kempson (2003), where this is motivated in detail.
Once this step of abduction has taken place, the only structurally unresolved node is the node decorated by the left-dislocated *sakwa-rul*. This node is then merged with the node decorated by $Fo(W)$, updating both the formula value of the metavariable and the tree node relation.

### 3.1 Pair-wise reading in Long-Distance Scrambling

(13) *Sakwa-rul Mina-ka Jina-ka mek-ess-ta-ko malhay-ss-ta*

apple$_{ACC}$ Mina$_{NOM}$ Jina$_{NOM}$ e$^{\#}$PAST$^{\#}$DECL$^{\#}$COMP$^{\#}$ say$^{\#}$PAST$^{\#}$DECL

Jina said that Mina ate an apple

In (13), when there is an intonational break between the first subject *Mina-ka* and the second subject *Jina-ka*, the challenge is how to analyse *sakwa-rul Mina-ka* as forming a constituent and so yielding a pair-wise reading. Here the phonological action of BREAK is activated after parsing *Mina-ka*, the second NP. I analyse a pair-wise reading of multiple dislocated elements as a combination of * Adjunction and local * Adjunction. The first step in the parsing of *sakwa-rul Mina-ka* involves a step of * Adjunction, to introduce a new node decorated with requirement $?Ty(t)$\textsuperscript{15}. In that structure, first the node for *sakwa-rul* to decorate, and then the node for *Mina-ka* to decorate, are constructed by Local *Adjunction and locally fixed through abduction, using case specification exactly as in local scrambling:

\[
\begin{align*}
&Tn(0), ?Ty(t), \\
&\quad F_o(Jina) \\
&\quad \langle \uparrow_o \rangle Tn(o) \\
&\quad F_o(Mina) \\
&\quad ?(\uparrow_o) Ty(t) \\
&\quad F_o(E, x, \text{Sakwa}(x)) \\
&\quad ?(\uparrow_o) Ty(e \leftarrow t) \\
&\quad F_o(U) \\
&\quad \langle \uparrow_o \rangle \langle \uparrow_1 \rangle Tn(o) \\
&\quad ?Ty(t), \uparrow_o \uparrow_1 Tn(o, \gg) \\
&\quad F_o(W) \\
&\quad F_o(\text{Mek}')
\end{align*}
\]

The lack of any intonational break between *sakwa-rul* and *Mina-ka* is accordingly expected. Yet there is an intonational break between *Mina-ka* and *Jina-ka*, indicated during the parsing of *Jina-ka* (given the paralinguistic status of intonation distributed across one or more expressions); and this determines that all other alternatives are eliminated. The parser then processes *Jina-ka* and fixes it to its local structure, using case information as before. Here local structure matches with main structure. The verb *mek‘eat’ then provides a full template of the local structure and provides meta-variables $U$ and $W$ to the object and subject-denoting node, respectively.

At this juncture, through abduction process, which is driven by the lexical action of -ko, its propositional template of structure is fixed immediately subordinate to the main structure (see Kempson (2003)). The unfixed, partial structure dominating the nodes decorated by $F_o(Mina)$ and $F_o(sakwa)$ is then merged with the top node of this substructure, with the consequence that the subject node of this structure gets updated into $F_o(Mina)$ and the object node gets updated.

\textsuperscript{15}Here, * Adjunction feeds local * Adjunction. The original definition on * Adjunction was to be applied to $Ty(e)$ (See Kempson, Meyer-Viol, and Gabbay (2001)). I think it needs to be revised as either $Ty(e)$ or $Ty(t)$.
into Fo(sakwa).

3.2 Only One Unfixed Node?

There is a particular challenge posed for Dynamic Syntax, when faced with multiple scrambling languages, as to whether these can be characterised with only one unfixed node at a time, as required by the framework. In this connection, consider (14) and (15):

(14) Yuna-ekey Mina-ka sakwa-rul cwu-ess-ta-ko malhay-ss-ta
    Yuna\textsubscript{DAT} Mina\textsubscript{NOM} apple\textsubscript{ACC} \textit{give} PAST, DECL, COMP \textit{say} PAST, DECL
Mina said that she \textit{gave} an apple to Yuna.

(15) *Yuna-ekey sakwa-rul Mina-ka Jina-ka mek-ess-ta-ko malhay-ss-ta-ko
    Yuna\textsubscript{DAT} apple\textsubscript{ACC} Mina\textsubscript{NOM} Jina\textsubscript{NOM} \textit{eat} PAST, DECL, COMP \textit{say} PAST, DECL, COMP
    \textit{say}nggakhay-ss-ta
    think\textsubscript{PAST, DECL}
    *Jina thought that she \textit{said} to Yuna that Mina ate an apple.

In (14), dative NP \textit{Yuna-ekey} is apparently freely interpreted within either main or subordinate structure. Despite this, (15) is unacceptable. It is because the sequence of NPs \textit{sakwa-rul} Mina-\textit{ka} following \textit{Yuna-ekey}, has to be interpreted within the same structure. But the trouble is that \textit{mek'-eat'} can't take \textit{Yuna-ekey} as its argument. As we've seen in Section 3.1, the pair \textit{sakwa-rul} Mina-\textit{ka} will have to be construed as providing two nodes dominated by one unfixed node introduced by a single step of * Adjunction. Such asymmetry between (14) and (15) supports the claim\footnote{This claim is also made in Kempson, Kjaer, and Cann (2003).} that only one unfixed node of a type at a time is allowed in structure building. But at this juncture, given the restriction of not being able to introduce another unfixed node by a second application of * Adjunction, the only way to construe dative NP \textit{Yuna-ekey} is by using a step of Local * Adjunction, which will ensure that the node it decorates will be fixed to its local, here main, structure as an indirect-object argument for the predicate \textit{say}nggakha-'think', but this is again problematic, as the predicate \textit{say}nggakha-'think' can't take a dative NP as its argument.

4 Conclusion

In this paper, based on the on-line parsing framework in Dynamic Syntax, I have shown how structure building in Korean not only can, but must, be construed as local and incremental just like English (contrary to what is generally assumed, though see Aoshima, Phillips, and Weinberg (2003)). The success of this analysis turned on assuming that intonation plays a crucial role in disambiguating possible parse sequences.
References


Acquaintance resolution and belief *de re*

EMAR MAIER  
University of Nijmegen  
e.maier@phil.kun.nl

**Abstract.**  
I propose a way of semantically representing *de re* belief ascriptions that involves contextual resolution of the acquaintance relation between the attitude holder and the object about which the attitude is *de re*. A special case is that where the belief is about the believer herself. Here, we may discern two possibilities: the acquaintance relation is equality, in which case we end up with a *de se* belief, or, if the first option fails, we search the context for a different suitable relation of acquaintance between the believer and herself, like looking in a mirror or seeing yourself on TV. This second option leaves open the possibility that the believer herself is unaware of the fact that she's actually seeing herself, thereby accounting for the true reading (*de re/non-de se*) of "Lain believes she will win" in mistaken identity scenarios. To implement these ideas formally, I use a two-dimensionally modal extension of DRT, and second order binding and unification.

1 Introduction

This paper is concerned with the analysis of *de re* belief reports, i.e. sentences of the form ‘*x believes that φ*’ where φ describes a belief of *x* about some object in the common ground at the time and place of the report’s utterance. As a first approximation, consider the following traditional explication of *de re* belief based on (Kaplan 1969) and (Lewis 1979):¹

\[(1) \quad x \text{ believes } de \text{ re } of \text{ y that it is P iff} \]
\[a. \quad x \text{ is uniquely related to y via an acquaintance relation } R, \text{ and} \]
\[b. \quad x \text{ self-ascribes the property of bearing } R \text{ to something which is P} \]

Taking this definition of *de re* belief as a starting point, I develop a unified theory of *de re* and so-called *de se* attitude reports, in effect reducing both to a *de re* analysis along the lines of (1). Other attempts at reducing *de se* to *de re* based on some variant of (1) have been proposed in the literature, see for instance (von Stochow 1982)², the sequence-of-tense account proposed (and

¹Von Stochow (1982) used this synthesis of ‘Quantifying in’ and *de se*-as-property-ascription.
²Though not his later (2002) feature deletion account, which is inspired by Schlenker (1999).
rejected) by Abusch (1997), and the arguably reductionistic attempt of Reinhart (ms). My own reduction is formulated in a dynamic representational framework and it avoids some of the main defects of these earlier theories.

The paper is structured as follows: After explaining the framework of Layered DRT in section 2, we investigate *de re* reports where the attitude is ascribed to a third person, and is *de re* about someone referred to by means of a proper name or other referential expression, e.g.

(2) Lain thinks Arisu looks good

Then, in section 4 we look at cases where the subject of the embedded and the matrix clause are co-referential:

(2) Lain thinks she looks good

Such reports are interesting since they are true if the ascribee, Lain, has a *de se* attitude, i.e. a thought from a first person perspective, “I look good”, but, crucially, also in what I will call ‘mistaken identity scenarios’, in which she has a thought of the form “That girl here in this picture, she looks good” while in the actual world of the report she happens to point at her own picture. In such a scenario we say Lain’s thought is *de re* about herself.

Perhaps I should point out here already that my analysis is only weakly reductionistic in the sense that although the so-called preliminary sentence representations are uniform for *de re* and *de se*, after resolution the final representations (the output contexts in dynamic semantics jargon, see section 2) differ, which is as it should be given the differing truth-conditions for *de re* and *de se* and the existence of unambiguously *de se* reports.

## 2 Layered DRT with attitudes

This section reviews and extends a two-layered fragment of the LDRT framework presented in (Geurts and Maier ms) to represent and interpret different kinds of information expressible in a discourse. The basic idea behind LDRT is to keep apart the various levels of content (implicated, asserted, accommodated,...) by representing them at different *layers* in one LDRS, with discourse referents taking care of inter-layer communication, and using various layered notions of content. In the following we will restrict ourselves to only two layers, one labeled *fr* for frege, i.e. conditions that describe the actual propositional, asserted content of an utterance; and the other, labeled *k* for kripke-kaplan content, i.e. conditions to be interpreted rigidly with respect to the context. After giving a minimal syntax and intensional semantics, we turn to the treatments of presupposition, rigidity, and the semantics of attitude reports.

### Basic syntax & semantics

The primitive symbols of the LDRT language are the ones for DRT (discourse referents, first order predicates, and some logical constants (Kamp and Reyle 1993)) plus a set of layer labels, in our case consisting of just two elements, *fr* and *k*. An LDRS is a set of labeled discourse markers (the LDRS’s so-called universe) paired with a set of labeled conditions. A labeled discourse marker is a discourse marker subscripted with a layer label (e.g. *xfr*); labeled conditions are of the forms exemplified by ‘lovek(x,y)’ or ¬Frφ (φ denoting another LDRS). We write *U*(φ) for the universe
of $\varphi$, and $\text{Con}(\varphi)$ for its condition set, so $\varphi = \langle U(\varphi), \text{Con}(\varphi) \rangle$. A standard notational convention for (L)DRSs is illustrated in (3):

(3) \[ [x_k y]_{fr} \text{ asuka}_k(x) \text{ see}_{fr}(x,y) \text{ donkey}_{fr}(y) \]

Intuitively, (3) expresses that there is a contextually given individual, Asuka ($k$ layer), who is said to see a donkey ($fr$ layer). To make this precise we give the partial (semantic values may be undefined), intensional model-theoretic semantics (4), the basic notions of which are a set of individuals $D$, interpretation functions mapping $n$-place predicates to $n$-tuples of individuals, and embedding functions mapping discourse referents to individuals. We take possible worlds to be just interpretation functions so an intensional model is simply a pair $\langle D, W \rangle$ where $W$ is a set of interpretation functions.$^3$

(4) Let $\varphi$ be an LDRS, $m, l \in \{k, fr\}$, $\langle D, W \rangle$ a model, $w \in W$, and $f$ a partial embedding

a. definedness:

i. $[\varphi]_{l,w}^f$ is defined iff there is an embedding $g$, $\text{Dom}(g) = \text{Dom}(f) \cup \{x | x_l \in U(\varphi)\}$ and for all $\psi \in \text{Con}(\varphi)$: $[\psi]_{l,w}^g$ is defined

ii. $[P_m(x^1, \ldots, x^n)]_{l,w}^f$ is defined iff $\{x^1, \ldots, x^n\} \subseteq \text{Dom}(f)$

iii. $[-m \varphi]_{l,w}^f$ is defined iff $[\varphi]_{l,w}^f$ is defined

iv. $[\varphi \lor_m \psi]_{l,w}^f$ is defined iff $[\varphi]_{l,w}^f$ and $[\psi]_{l,w}^f$ are defined

b. If defined, the semantic values of conditions and LDRSs are:

i. $[\varphi]_{l,w}^f = \{ g | \text{Dom}(g) = \text{Dom}(f) \cup \{x | x_l \in U(\varphi)\} \}$ and for all $\psi \in \text{Con}(\varphi)$: $[\psi]_{l,w}^g = 1$

ii. $[P_m(x^1, \ldots, x^n)]_{l,w}^f = 1$ iff $m \neq l$ or $\langle f(x^1), \ldots, f(x^n) \rangle \in w(P)$

iii. $[-m \varphi]_{l,w}^f = 1$ iff $m \neq l$ or $[\varphi]_{l,w}^f = \emptyset$

iv. $[\varphi \lor_m \psi]_{l,w}^f$ iff $m \neq l$ or $[\varphi]_{l,w}^f \cup [\psi]_{l,w}^f \neq \emptyset$

v. for every labeled condition $\psi$: $[\psi]_{l,w}^f = 0$ iff $[\psi]_{l,w}^f$ is defined and $[\psi]_{l,w}^f \neq 1$

Intensional $l$-content is defined as the set of worlds for which there is a truthful embedding with respect to a layer $l$:

(5) a. $[\varphi]^f_l = \{ w \in W | [\varphi]_{l,w}^f \neq \emptyset \}$ if $[\varphi]_{l,w}^f$ is defined for some $w$ (otherwise $[\varphi]^f_l$ is undefined)

b. $[\varphi]_l = [\varphi]^0_l$

**Presupposition**

We follow van der Sandt (1992) and adopt the theory of presuppositions as anaphora; If a sentence triggers a presupposition, we mark the material triggered in the preliminary LDTRS by prefixing a $\partial$. The Resolution algorithm tries to bind or, if that fails, accommodate these presuppositions. Since we have two layers in our fragment, we have two kinds of presuppositions: normal descriptive presuppositions, recognizable by a $fr$ labeled discourse referent in their universe, and indexical or proper name presuppositions, whose universes contain $k$ labeled discourse referents. By saying

$^3$In (10) on p. 164 all this and more will be clarified in an example computation of a semantic value.
that accommodation consists in dropping presupposed material at the layer it is labeled with, we have combined presuppositional treatments of referential terms (Geurts 1997) with the classical theory of direct reference (Kripke 1972; Kaplan 1989), getting the best predictions of both worlds, provided we add a two-dimensional semantics for the k layer, which we will now do.

**Context-dependence: the k layer**

We define the Kaplanian content by implementing a second semantic dimension, the context-parameter (Kaplan 1989). We take contexts to be just possible worlds, i.e. in our framework, interpretation functions, but in order for Kaplanian content to be defined at such a context, that context often has to be a ‘small world’, i.e. one mapping predicates like ‘speaker’ and ‘joey’ to singleton sets, because we will need contexts to provide us with a unique embedding verifying the k layer. In our framework then, Kaplanian content at a context c ∈ W is defined as in (6). The working of this admittedly dense definition and the underlying semantics of (4)-(5) is illustrated by the example computations in (10) below (p. 164):

\[(6) \quad \text{a. If } \llbracket \varphi \rrbracket_{k,c} \text{ is a singleton, } \llbracket \varphi \rrbracket_{c} = \llbracket \varphi \rrbracket_{k,c}^g \text{ where } g \text{ is the unique element of } \llbracket \varphi \rrbracket_{k,c}. \text{ Otherwise undefined}
\]

b. \[\llbracket \varphi \rrbracket_{c}^g = \llbracket \varphi \rrbracket_{0,c}^g\]

In words: Kaplanian content, \(\llbracket \cdot \rrbracket_c\), as defined above, models the truth-conditional contribution of the \(fr\) layer against the background of an external anchor provided by the evaluation of the \(k\) layer at the context parameter. It is meant to correspond to the standard classical notion of static truth-conditional content. Although LDRT is capable of expressing arbitrarily many different types of contents, we will mainly stick with this classical one. This conservativity facilitates a better comparison of our results with other theories, since most research on indexicals, attitudes and attitude reports has been done in static classical frameworks by philosophers and semanticists.

**Diagonalization**

With \(\llbracket \varphi \rrbracket_{c}^{f}\) we can define the diagonal as expected, as a set of contexts:

\[(7) \quad \Delta_f(\varphi) = \{ c \in W | \llbracket \varphi \rrbracket_{c}^{f} \text{ is defined and } c \in \llbracket \varphi \rrbracket_{c}^{f}\}\]

Diagonalization ensures that the \(k\) labeled information, which normally, in \(\llbracket \varphi \rrbracket_{c}^{f}\), does not enter the actual truth conditions but only contributes to fixing the \(k\) discourse referents, part of the ‘proposition’ (construed for this purpose as a set of contexts). The notion of a diagonal proposition is an important ingredient for our semantics of attitude reports in the following.

**Syntax and semantics of belief**

We add a special kind of predicate ‘believe’ to the language and then extend the syntax with layered conditions of the form ‘believe\(_k\)(x) : \(\varphi\)’. The interpretation of ‘believe’, \(\text{bel} \in [D \times W \rightarrow \wp(W)]\), is added to our models and can be seen as mapping every individual at a world to the set of its belief alternatives. The semantics of the new condition is given in (8), which is modeled after Haas-Spohn’s (1994) Stalnakerian analysis of attitudes as self-ascriptions of diagonal propositions:
(8) if \( x \in \text{Dom}(f) \) and \([\varphi]_{m,w}^f\) is defined, \([\text{bel}(f(x),w) = \varphi]_{m,w}^f = 1 \) iff \( m \neq l \) or \( \Delta^f(\varphi) \supseteq \text{bel}(f(x),w) \)

To see the semantics in action, consider the following representation of a belief report:\(^4\)

(9) a. Lain believes the king of France likes donkeys
b. \( x_k \mid \text{bel}(x) : \varphi; \quad \left[ x_k \left| \begin{array}{c} \text{king of France} \in (u) \text{ like donkeys}_k(u) \end{array} \right. \right] \)

The computation of Kaplanian content, \( [(9b)]_k^f \), below shows how the semantics defined above assigns the intuitive truth-conditions of (9a) to the LDRS (9b). First we compute the interpretation of the embedded DRS, call it \( \varphi_1 \), which involves computing first the rigid \( k \) interpretation (10a.i), then the \( f \) contribution (10a.ii), and, combining these, the Kaplanian content (10a.iii) and diagonal proposition (10a.iv). Next, we repeat the first three steps for the embedding LDRS in (10b) to arrive, in (10b.iii), at the Kaplanian content of (9b):

(10) a. \( \varphi_1 := \left[ x_k \left| \text{king of France}_k(u) \text{ like donkeys}_k(u) \right. \right] \)
   i. \( [\varphi_1]_k^f = 1 \) iff there is one unique king of France in \( c \) i.e., \( c(\text{king of France}) \) is a singleton.
   ii. \( [\varphi_1]_f^w = 1 \) iff \( f(u) \) likes donkeys in \( w \)
   iii. \( [\varphi_1]_w^f = \{ w \in W \mid c(\text{king of France}) \text{ is a singleton and } c(\text{king of France}) \subseteq w(\text{like donkeys}) \} \); the set of worlds \( w \) in which \( c \)'s unique king of France likes \( w \)'s donkeys.
   iv. \( \Delta^f(\varphi_1) = \{ c \in W \mid [\varphi_1]_k^f \text{ is defined and } c \in [\varphi_1]_k^f \} = \{ c \in W \mid c(\text{king of France}) \text{ is a singleton and } c(\text{king of France}) \subseteq c(\text{like donkeys}) \} \)

b. \( \varphi_2 := \left[ x_k \mid \text{bel}(x) : \varphi_1 \right] = (9b) \)
   i. \( [\varphi_2]_k^f = 1 \) iff there is a unique individual called 'Lain' in \( c \)
   ii. \( [\varphi_2]_f^w = 1 \) iff \( \Delta^f(\varphi_1) \supseteq \text{bel}(f(x),w) \) iff every belief alternative of \( f(x) \) in \( w \) is such that it contains a unique king of France who likes donkeys.
   iii. \( [\varphi_2]_w^f = \) the proposition that (= the set of worlds in which) the person called 'Lain' in \( c \), believes to inhabit a world/context with a unique donkey-loving king of France.

For analysing \emph{de re} and \emph{de se} attitudes we need a way to represent the first person, the agent responsible for a speech act c.q. the experiencer of a thought or hope, to serve as the referent of an utterance of "I" in a speech or monologue intérieur. For this purpose we have the 1-place predicate 'center', which in a proper context \( c \in W \) is usually assigned a singleton extension.\(^5\)

With the 'center' predicate we can handle first person pronouns:

\(^4\)Note that we are focussing on the \emph{de dicto} reading for now, so Lain’s belief is about being king of France (regardless of whoever actually fills that role) and liking donkeys.
\(^5\)Because of the way we set up the partial semantics, it is strictly speaking unnecessary to lay down in advance which elements of \( W \) are 'possible worlds' and which are 'contexts', so we have omitted such a restriction of models. For more obvious compatibility with Kaplan (1989) we might as well define: \( C := \{ w \in W \mid w(\text{center}) \text{ is a singleton} \} \); a context is a 'centered world'.

164
(11)  a. I am a superhero
    b. \[ y_k \mid \text{center}_k(y) \text{ superhero}_R(y) \]

Consequently, we can represent ascriptions of first person (de se) attitudes as in (12a), which informs us that Lain had a thought of the form (11):

(12)  a. Lain believes to be a superhero
    b. \[ x_k \mid \text{lain}_k(x) \text{ believe}_R(x); y_k \mid \text{center}_k(y) \text{ superhero}_R(y) \]

The interpretation of (12b) is the proposition that all of Lain’s belief alternatives are such that they have a unique center/experiencer who is a superhero. Section 4 is concerned with the compositional derivation of representations like (12b) from a sentence’s surface structure. My aim is to give an analysis of belief reports that assigns them all a single uniform preliminary LDRS modeled after (1), the de-re-via-acquaintance-relations analysis alluded to in section 1, and in that sense unifying de re and de se reports.

3 Analyzing de re reports

This section is devoted to sketching the framework of acquaintance resolution by analysing a 3rd person belief-about-someone-else report:

(2)  Lain thinks Arisu looks good

We assume with Kaplan (1969) that in order for this report to be truly de re, the discourse context in which it is uttered should provide a suitable relation of acquaintance between Lain and Arisu. Say for instance that for the audience it is common ground that Arisu and Lain are best friends, in an LDRS:

(13)  \[ x_k \ y_k \mid \text{lain}_k(x) \text{ arisu}_k(y) \text{ best}_R(x,y) \]

According to Kaplan (1969), an acquaintance relation must fulfill two criteria: first, \( R(x,y) \) relates \( x \) to \( y \) via a causal chain, and, second, \( R(x,y) \) is a *sufficiently vivid name* of \( y \) for \( x \), i.e. it is one of the descriptions that \( x \) mentally associates with \( y \). Assuming that being best friends in this scenario fulfills these requirements, the reading Kaplan (1969) would assign to (2) in context (13) is one where Lain is reported to believe de re about Arisu, under the acquaintance relation ‘my best friend’, that she, the person she is thus acquainted with, looks good (cf. (1)). This reading we can easily represent, with its context, in LDRT:

(14)  \[ x_k \ y_k \mid \text{lain}_k(x) \text{ arisu}_k(y) \text{ best}_R(x,y); u_k \ y_k \mid \text{center}_k(u) \text{ best}_R(u,v) \text{ look}_R(v) \]

The question now is how to compute this as an ‘output’ context from input (13), plus some preliminary LDRS, compositionally constructed from the sentence (2). In my proposal, the preliminary representation contains \( k \)-presuppositions triggered by the proper names, in addition to another form of underspecification (\( R(z,w) \neq ? \)) for the acquaintance relation, requiring 2nd order
binding and unification for its resolution:

\[
\text{Prel}(2) = \begin{bmatrix}
\partial [z_k | \text{lain}_k(z)] R(z,w) = ? \\
\text{believe}_k(z); [u_k v_k | \text{center}_k(u) R_k(u,v) \text{ look_good}_k(v)] \\
\partial [w_k | \text{arisu}_k(w)]
\end{bmatrix}
\]

Merging the background (13) with the preliminary representation (15) and resolving the trivial proper name presuppositions \((z=x, w=y)\), gives:

\[
\begin{bmatrix}
x_k y_k & \text{lain}_k(x) \text{ arisu}_k(y) \text{ best_friend}_k(x,y) \\
\text{believe}_k(x); [u_k v_k | \text{center}_k(u) R_k(u,v) \text{ look_good}_k(v)]
\end{bmatrix}
\]

Now we have to determine what the acquaintance relation is in this context, we do this by finding a part of the current LDRS appropriately relating \(x\) to \(y\) and filling the question marked slot with it. This essentially anaphoric process we dub 2nd order binding, in analogy with the similar, but first-order, presupposition binding of (van der Sandt 1992; Geurts 1999). Here, a good binder candidate is given by the ‘best_friend’ condition. Note that this a purely formal, syntactic movement procedure, and that consequently \(\approx\) is a formal (not semantically interpreted, hence unlabeled) relation between parts of LDRSs, syntactic objects, not individuals: 

\[
\begin{bmatrix}
x_k y_k & \text{lain}_k(x) \text{ arisu}_k(y) \text{ best_friend}_k(x,y) \\
\text{believe}_k(x); [u_k v_k | \text{center}_k(u) R_k(u,v) \text{ look_good}_k(v)]
\end{bmatrix}
\]

We have now a proper \(\varphi \approx \psi\) condition, interpreted as ‘the lambda terms \(\varphi\) and \(\psi\) are equivalent (under \(\alpha/\beta/\eta\)-interconvertibility)’. The discourse referent ‘R’ here is like a free variable which we will eliminate by higher-order unification, cf. Dalrymple et al.’s (1991) resolution of the elided property in VP-ellipsis constructions. We first find a unifying substitution, i.e. a substitution

---

6 A lot remains to be said about the 2nd order binding process. First, how to define ‘part of’ an LDRS \(\varphi\)? A formal definition could be given, but note that it is a little more complicated than just taking a subDRS of \(\varphi\), or a subset of \(\text{Con}(\varphi)\). Second, although I don’t have a full specification of the 2nd order binding algorithm, it seems reasonable to assume that it could be implemented in, say, a unidirectional OT system featuring at least the following (soft) constraints:

- avoid circularity (don’t bind to a DRS part containing the bindee)
- be relevant (try to find something that with the same set of free (first-order) discourse referents as the bindee)
- avoid redundancy (take as little as possible)
- be local (try to find salient stuff available as close to the bindee as possible)
- final outcome should be pragmatically semantically acceptable (throw out candidates leading to nonsensical readings)

Note that these constraints are more or less the same as those constraining anaphora resolution (van der Sandt 1992).
for the free variable which will make the $=$-equality true. In this case there are several a priori possible unifiers, but the one we want is (18a), which we then apply to the whole LDRS, yielding (18b):

\[(18) \quad R \rightarrow \lambda s \lambda t. \text{best\_friend}(s, t)\]

\[\begin{bmatrix}
\text{lain}_k(x) \quad \text{aris}_u(y) \quad \text{best\_friend}_p(x, y) \\
\text{(}\lambda s \lambda t. \text{best\_friend}(s, t)) (x, y) \cong \text{best\_friend}(x, y)
\end{bmatrix}\]

\[\begin{bmatrix}
x_k \quad y_k \\
\text{believe}_p(x): \quad u_k \quad v_k \\
\text{center}_k(u) \\
\text{look\_good}_p(v)
\end{bmatrix}\]

After some $\beta$-reductions, this final LDRS is easily seen to be equivalent to the representation we wanted, (14), which concludes this illustration of my analysis of de re attitude ascriptions.

4 De se vs. de re-about-self

This section explores a more interesting type of reports, sentences of the form:

\[(3) \quad \text{Lain thinks she looks good}\]

As said in the introduction, such sentences have been argued to exhibit a de re/de se ambiguity, since they can be true if the ascribed thought is de se, i.e. Lain thinks “I look good”, but also if the thought were merely de re about herself, i.e. Lain thinking “That girl looks good” while pointing at, but not recognizing, a picture of herself. Lewis (1979) argues we can reduce de se to de re, since:

Self-ascription of properties is ascription of properties to oneself under the relation of identity. Certainly identity is a relation of acquaintance par excellence. So belief de se falls under belief de re. (Lewis 1979:156)

Since Lewis, various authors have argued against such a reduction construed as a unification of de re and de se reports (Chierchia 1989; Percus and Sauerland 2003; Schlenker 2003; von Stechow 2002), but, as I aim to show here, the present account of de re in effect constitutes an enhanced Lewisian reduction, differing from the earlier ones in that it takes seriously the context-dependence of the acquaintance relation, resolving it in context, rather than e.g. existentially quantifying over it.

The de se reading

Consider the preliminary LDRS on my account (cf. (15)) for (3):

\[(19) \quad \text{Prel}(3) = \begin{bmatrix}
\partial [z_k] \quad \text{lain}_k(x) \quad \text{R}(z, w) \cong \\
\text{believe}_p(z): \quad u_k \quad v_k \\
\text{center}_k(u) \quad \text{R}_k(u, v) \quad \text{look\_good}_p(v) \\
\partial [w_k] \quad \text{fem.3.sg}_k(w)
\end{bmatrix}\]

167
In an empty context, or one containing just Lain, this resolves to:

\[
\begin{align*}
(20) & \quad \exists x \ \text{lain}_k(x) \ R(x,x) \Leftrightarrow \\
& \quad \exists x \ \text{believe}_\ell(x) \ : \ u_k \ v_k \ \text{center}_k(u) \ R_k(u,v) \ \text{look}_\ell(v)
\end{align*}
\]

We cannot bind ‘\(R(x,x)\)’ to anything, so we add the always available harmless tautology ‘\(x = x\)’ to the representation,\(^7\) bind to it, (21a), and extract the unifier ‘\(R \rightarrow \lambda s \lambda t. s = t\)’, which we then apply, (21b), a more readable but equivalent representation of which is given in (21c).

\[
\begin{align*}
(21) & \quad \exists x \ \text{lain}_k(x) \ x = x \ R(x,x) \Leftrightarrow \\
& \quad \exists x \ \text{believe}_\ell(x) \ : \ u_k \ v_k \ \text{center}_k(u) \ R_k(u,v) \ \text{look}_\ell(v)
\end{align*}
\]

The truthconditions of this representation, according to the semantics of section 2, are, roughly: \([\text{(21c)}]^{\neg} = \) the set of worlds in which all of Lain’s belief alternatives feature a unique agent who looks good. In other words, we get the so-called de se reading.

The de re reading

Now, say we are in a context in which Lain is saliently known to be looking at a picture of herself:

\[
\begin{align*}
(22) & \quad \exists x_k \ \text{lain}_k(x) \ \text{look}_\ell \text{at}_\ell \text{picture}_\ell \text{of}(x,x)
\end{align*}
\]

Adding the same preliminary representation (19) we used to derive the de se reading above, and performing again our resolutions yields:

\[
\begin{align*}
(23) & \quad \exists x_k \ \text{lain}_k(x) \ \text{look}_\ell \text{at}_\ell \text{picture}_\ell \text{of}(x,x) \\
& \quad R(x,x) \Leftrightarrow \\
& \quad \exists x_k \ \text{believe}_\ell(x) \ : \ u_k \ v_k \ \text{center}_k(u) \ R_k(u,v) \ \text{look}_\ell(v)
\end{align*}
\]

\(^7\)The label of the equality is not important, \(k\) would have done just as well.
So we see that we can get a de re, non-de se, reading if and only if we give enough context. Note that if Lain were aware that the person in the picture was herself (and we knew this), we would be able to reduce (23d) to the de se reading (21c) by combining the belief that the person in the picture looks good with the belief that the person in the picture is ‘me’ (the current center), but if she mistakenly thinks it is someone else in the picture, we are stuck with (23d), which is exactly as it should be. That we don’t need any special context for deriving the de se reading, as shown in (19)-(21c), is one of the advantages of my approach, accounting for the observed preference for de se readings of reports like (3).8

Conclusions

The conclusion is that, on the one hand, we can maintain, with e.g. Kaplan (1989) but against e.g. Chierchia (1989), that (3) and its kin are not syntactically ambiguous, since our Prel assigns it only a single representation: (19). On the other hand, interpreting the sentence in a discourse context can give rise to two possible output contexts, (21c) and (23d), with different (de se and de re, respectively) truth-conditions. We might say that one of the main features of the analysis proposed here is that the preliminary representations of all belief reports are uniformly constructed from the surface structure, but are underspecified (in more ways than one), which delegates the determination of the actual interpretation to the resolution mechanism.

As a survey of ‘future research’, let me mention two important omissions due to lack of space.9 The first is how the present theory handles unambiguously de se reports, such as Chierchia’s (1989) gerunds and infinitives, “Lain believes to be on the winning side”, and related shifted first person examples from Amharic or Ancient Greek, glossable as “Laini believes that I am a hero” (cf. Schlenker 2003). The second is concerned with how to account for the intuitions about quantified belief reports put forward in the literature, i.e. intuitions about the truth of sentences like Percus and Sauerland’s (2003) “Only Lain believes she will win” and Zimmermann’s “Everybody here hopes he will win” (cf. Chierchia 1989) in contexts with a mixed de re/de se company.

Acknowledgements

I thank the anonymous reviewers for valuable comments.

---

8This preference I deduce from the reluctance encountered in convincing non-philosophers of the truth of (3) in mistaken identity scenarios.

9Preliminary extensions to the theory needed to account for these data can be found in a double-length version of this paper, at http://www.ru.nl/phil/\textasciitilde emar
References

Automatic Extraction of Subcategorization Frames for Bulgarian

Svetoslav Marinov
GSIT, Sweden
marin@isp.hias.se

Abstract.
Knowledge of verb’s valency or subcategorization is essential for many NLP tasks. The present paper describes an attempt to learn this kind of information from a corpus of parsed sentences of Bulgarian. Our program acquired the subcategorization information for 38 verbs and achieved 87.7% precision and 68.3% recall. We did not use predefined sets of frames but automatically induced such from a treebank.

1 Introduction

Subcategorization Frame (SF) is an expression of what kind of and how many syntactic arguments a verb takes. The external argument, i.e. the subject of the clause, is usually excluded from the SF and the emphasis is on the internal ones - objects, infinitives, that-clauses, participle clauses, prepositional phrases.

In this paper we will present our attempt to extract and learn SFs for certain verbs from a parsed corpus of Bulgarian sentences. The work has been done on TreeBank data, kindly provided by the BulTreeBank project\(^1\). The paper is organized as follows: Section 2 presents the task and reviews previous attempts and results in the area of Automatic SF extraction; section 3 gives a brief overview of some linguistic phenomena concerning verbs in Bulgarian; section 4 describes the TreeBank we have used; section 5 presents our system for extraction and learning SFs; section 6 gives the results and section 7 concludes.

2 Background

Manning (1993) and Briscoe and Carroll (1997), among others, discuss the necessities of automatically acquiring knowledge about verbs’ SFs rather than manually creating such lexicons. The authors agree on several issues, such as: 1) Manually compiled verb lexicons (i.e. lexicons

\(^1\) We would like to thank Kiril Simov for letting us use the corpus, which is still under development.
of verbs with their respective SFs) are time-consuming, demand a lot of human resources and often contain mistakes; 2) such lexicons do not exist for all languages, e.g. Bulgarian; 3) they seldom represent the up-to-date language use; 4) they are difficult to update and 5) they are not coherent. Automatically created lexicons have, according to the authors, the benefit of 1) being based on real data, i.e. text corpora; 2) having the possibility of being updated quickly; etc.

Brent (1991) is one of the earliest references to acquisition of verbs’ SFs. His program works on untagged text from the Wall Street Journal from which verbs are extracted first. In order to do this he uses simplistic, yet unambiguous, cues to detect the verbs. These are namely the positions of pronouns and proper names, i.e. the so called “case-marked” elements, which predict that a verb will be in the closest vicinity. After that the SFs are detected and statistically verified.

Brent’s best result is the discovery of 40% of the verbs taking direct object, but this with only 1.5% error rate and the worst result gives 3% of all the verbs taking infinitive, yet with 3.0% error rate.

Manning (1993) criticizes this approach in terms of waste of valuable data. He also starts with raw data but first tags it using Julian Kupiec’s stochastic part-of-speech tagger and then the output is sent to a finite state parser in order to extract the SFs. His system, in comparison to Brent’s, distinguishes among 19 different SFs, yet still uses a variant of Brent’s binomial filtering process to filter the much higher amount of false cues (i.e. initially discovered SFs). Precision and recall were measured, using Oxford Advanced Learner’s Dictionary as a benchmark. Thus the precision estimate for 40 verbs was 90% and the recall 43%.

Briscoe and Carroll (1997) presented a complete system that distinguishes among slightly more than 160 different subcategorization classes. The main distinctions with respect to Manning’s approach concern, according to the authors, the use of global instead of strictly local syntactic information and the use of a more complex (linguistically guided) filter on the extracted patterns. The system’s output was evaluated by comparing the results for randomly selected verbs to both manual analyses of the verbs in question and to their respective entries in dictionaries containing argument structure information. The results obtained comparing the system’s output to the manually made analyses gave a precision of 76.6% and a recall of 43.4%. One weak link of the system that Briscoe and Carroll themselves point out is the filtering of patterns extracted from the data, especially for low frequency SFs. The filtering process is built on the binomial distribution test originally introduced by Brent.

Sarkar and Zeman (2000) take a slightly different approach to extracting SFs. First, they use syntactically annotated data (i.e. The Prague Dependency Tree Bank) and hence the choice of language also differs from the previous approaches, all dealing with English. Czech is a free word-order language and much closer to Bulgarian than English. So the results and techniques discussed by the authors are quite relevant for us.

Sarkar and Zeman (2000) concentrate mostly on the filtering of adjuncts from the observed frames. They use 3 different statistical techniques to learn possible SFs for certain verbs, namely, Likelihood ratio test, T-scores and Binomial distribution. They check all possible subsets of observed frames in order to find the best match. Their best results were achieved using the Binomial distribution. Thus precision was measured 88% and recall 74%, and 137 SFs were learned.

The approach we take to SF-extraction for Bulgarian is similar to the one by Sarkar and Zeman (2000). Namely, we do not work with predefined set of frames (e.g. Manning (1993, ?)).

\footnote{The ANLT and the COMLEX dictionaries, see Briscoe and Carroll (1997)}
but rather acquire the frames from a treebank.\footnote{We thank an anonymous reviewer for noticing the lack of this information in the paper.}

Statistical processing of the initially discovered frames is almost inevitable\footnote{In the majority of cases the SF-extraction algorithms discover also false frames/arguments for a given verb. These have to be filtered away either by using some hand-written rules or on a statistical basis.}. Most of the authors employ similar techniques, Log likelihood ratio tests Sarkar and Zeman (2000, ?), T-score Sarkar and Zeman (2000), Binomial distribution Sarkar and Zeman (2000, ?, ?, ?), EM algorithm Carroll and Rooth (1998), Clustering algorithms Basili, Pazienza, and Vindigni (1997).

Of the above methods Binomial distribution has been used most widely and probably most successfully. Yet, a problem with it is that it assumes a uniform error likelihood of all verbs disregarding their rather zipfian-like distribution. It disregards the correlation between the conditional distribution of SFs given a predicate and the unconditional distribution independent of a specific predicate Korhonen (2002). Sarkar and Zeman (2000) also mention this and propose the use of a multinomial distribution.

3 Syntactic particularities of Bulgarian

Bulgarian is a South Slavic language and can be characterized as having a relatively free word order. Although Bulgarian does not have morphological case on nouns, unlike West and East Slavic languages, it possesses clitic pronouns marked for Dative and Accusative case. Whenever these clitics are present together with the nouns or noun phrases for direct and indirect object, i.e. the so called clitic doubling phenomenon, the full NPs\footnote{Actually the indirect object will be a PP} may appear in any order and precede the main verb. The subject NP could appear after the main verb. In addition, the pronominal subject of a clause can be omitted since the verbal morphology carries the necessary information about person and number, i.e. on a par with the pro-drop phenomenon in some Romance languages, e.g. Italian, Spanish. These particularities are explained in (1).

(1)  
\begin{enumerate}
  \item a. Az go vidyah.
      I him\textsubscript{ACC} saw\textsubscript{ip.sg}  
      'I saw him.'
  \item b. Vecra gi vidyah az.
      Yesterday them\textsubscript{ACC} saw I  
      'I saw them yesterday.'
  \item c. Knigata im ya dadoha na detsata.
      Book them\textsubscript{DAT} it\textsubscript{ACC} gave\textsubscript{ip.pl} to kids  
      'They gave the book to the kids.'
  \item d. Izyadoh az kiflata.
      Ate\textsubscript{ip.sg} I bun.  
      'I ate the bun.'
\end{enumerate}

The above examples clearly show the following word orders - SOV, OVS, O\textsubscript{Ind}O\textsubscript{Dir}V, VSO, in addition to the default SVO.

Examples like (1) show the difficulties one could meet when designing an automatic SF-learner for Bulgarian.
4 BulTreeBank

For the present purpose we chose to look at data from the Bulgarian Tree Bank project Simov, Popova, and Osenova (2002). We were offered a preliminary version of the parsed corpus which consists of 580 sentences, fully parsed in Head-Driven Phrase Structure Grammar (HPSG) formalism and each word carries a rich part of speech tag. The original xml file was transliterated for the sake of ease in processing and viewing under different operating systems.

However, since the corpus is still under development, information like subcategorization, is still missing from it. The standard for HPSG ARG-ST- or COMPS-features, that describe the argument structure of a verb, are not yet present in the annotated text. This renders our task non-trivial one and our results valuable for expanding the treebank.

The information present in the data we work with is just a parse tree with begin- and end-tags for each node, so that a sentence like (2a) looks like (2b), which for sake of visibility is presented as a parsed tree in (3):

(2) a. Ne mi e do smyah.
    Not me-refl is to laughter
    'I'm not in the mood to laugh.'

b. <S><VPC><V><T>Ne<ta>Tc</ta></T><<V><Pron>mi<ta>PP-dis-t</ta></Pron>
   <V>e<ta>Vx--f-r3s</ta></V>/<V>/V><<PP><Prep>do<ta>R</ta></Prep>
   <N>smyah<ta>Ncmsi</ta></N>/PP></VPC><pt>.</pt></S>

(3)

\[
S \\
|  \\
VPC \\
|  \\
V \\
|  \\
PP \\
|  \\
T \\
|  \\
N \\
|  \\
Pron \\
|  \\
V \\
|  \\
Pres \\
|  \\
Prep \\
|  \\
Ncmsi \\
|  \\
dp-dis-t, Vx--f-r3s

Beyond the node labels and the POS-tags no other information is present in our training corpus.

5 The Program

In this section we outline the initial implementation of a system for learning SFs for Bulgarian verbs. The system consists of three modules, all implemented in the Oz programming language van Roy and Haridi (2004). The data was initially preprocessed in order to remove some information that was not relevant for the task.
The main module is the SF-extractor. Unlike Briscoe and Carroll (1997) we decided to keep to strictly local syntactic information, i.e. the algorithm searches for possible arguments only within the verb phrase, so that from this parse

```
VPC
  V
    znæ
    V
      NPA
        M
          knows
          Mcmi
            two
            languages
```

the following information will be extracted:

- **znæ**: NPA

The next module is the lemmatizer. Since there is no such tool available for Bulgarian\(^6\), at least one that works with non-cyrillic input, we created our own. This was done in order to merge results found for different forms of the same verb.

The last of the modules is a statistical filter using the binomial log-likelihood ratio test to filter out SFs. Although, the literature is not very favourable of this method, our little and sparse data did not allow for the use of other statistical techniques\(^7\). However, our results show that this test is really fitted for low-frequency data, as argued already by Dunning (1993). We took the standard log-likelihood ratio test \((-2 \log \lambda)\):

\[-2 \log \lambda = 2 \left( \log L(p_1, k_1, n_1) + \log L(p_2, k_2, n_2) - \log L(p, k_1, n_1) - \log L(p, k_2, n_2) \right)\]

where:

\[\log L(p, n, k) = k * \log 2p + (n - k) * \log 2(1 - p)\]

and:

\[p_1 = \frac{k_1}{n_1}, \quad p_2 = \frac{k_2}{n_2}, \quad p = \frac{k_1 + k_2}{n_1 + n_2}\]

In the above equations \(k_1\) is the count of the number of times the test verb appears with the test frame\(^8\), \(n_1 - k_1\) is the count of the number of times that verb appears with any other frame, \(k_2\) is the count of the number of times that frame appears with any other verb and \(n_2 - k_2\) is the count of the number of times any other verb appears with any other frame. In other words, the log-likelihood ratio test describes the probability of the verb with a given frame, based on the overall likelihood of the frame.

---

\(^6\) Too late we learned about the existence of a morphological analyzer for Bulgarian, available at [http://www.bulterbank.org/Resources.html](http://www.bulterbank.org/Resources.html)

\(^7\) We intend, however, to check how other statistical techniques fare with respect to the sparse data set, e.g. Binomial distribution.

\(^8\) Here test verb and test frame mean the discovered verb and its discovered frame.

175
6 Results

We excluded all verbs (i.e., lemmas) that were seen less than 4 times in the corpus. Around 90% of the verbs were seen only once. Of the remaining verbs we decided to disregard ditransitives. The threshold was set to -2.0 and this gave us the final result: 130 verbs with their respective frames.

Since there is no electronic valency dictionary for Bulgarian, nor a printed one, we had to rely on our own judgement, as well as the corpus, for verifying the results. Precision and recall were calculated using the standard formulae - \( \frac{X}{X+Y} \) for the former and \( \frac{X}{X+Z} \) for the latter, where \( X \) is the number of correctly identified frames, 114 in our case, \( Y \) is the false frames, 16 in number, \( Z \) is additional correct frames not identified by the algorithm but present in the data, i.e. 53. Thus precision was 87.7% and recall 68.3% per frame.

The program learned 16 SFs for 38 different verbs. Although we worked with little data the overall result is comparable to previously described results, especially the one by Sarkar and Zeman (2000), since it deals with another Slavic language.

7 Conclusion

In this paper we looked at the area of automatic extraction of SFs for verbs. We reviewed why this is important and the previous attempts in this area. Our work differs from the previous ones in several respects. First we deal with Bulgarian, a Slavic language, whereas most other work is done on English. We addressed the issue how and why our work is important for extending the Bulgarian treebank. We also achieved results matching those employing the binomial distribution test, i.e. 87.7% precision.

This work will be continued in several directions. One is to devise better heuristics for SF extraction (we have to include ditransitives) and employing a different or at least two statistical tests for filtering the results. We also intend to adopt a tagger for Bulgarian and attempt SF learning only from tagged data.
8 Some Verbs with SFs

chuvam : nil
chuvam : CLCHE
chuvam : Pron
garantiram : N
garantiram : NPA
gledam : CoordP
gledam : NPA
gledam : PP
govorya : nil
govorya : Pron
idvam : nil
idvam : PP
idvam : Pron
imam : N
imam : NPA
iskam : CLDA V
iskam : CLDA VPC
iskam : CLDA VPS
iskam : N
izglezhdam : Adv
izglezhdam : CLCHE
izglezhdam : CLDA VPS
izglezhdam : CoordP
izlizam : Adv
izlizam : CoordP
izlizam : PP
kazvam : CLCHE
kazvam : N
kazvam : Pron
mislya : CLCHE
mislya : PP
mislya : Pron
moga : CLDA V
moga : CLDA VPA
moga : CLDA VPC
moga : CLDA VPS
nosya : NPA
References


Why Not Use Query Logs As Corpora?

Olena Medelyan
University of Freiburg, Germany
medelyan@coli.ing.uni-freiburg.de

Abstract.
Generally, every Web search engine logs the user sessions. These records, called query logs, contain valuable information about the behaviour of Internet users and their language. There are only a few experiments on mining query logs, but they confirm that query logs are very useful for designing natural language applications in Web retrieval. This paper shows how lexical and semantic information can be extracted from query logs using statistical methods. I first summarize approaches in query log processing and mining for different purposes. After a short description of the used query logs, I present new domain- and language-independent methods for generating a compound dictionary and extracting semantically similar terms. The evaluation will shed light on the quality of proposed methods and show that the results are good enough to be directly integrated in query processing and improve information retrieval on the Web.

Keywords: Web retrieval, query processing, text mining, information retrieval

1 Introduction

Since the Internet has become one of the most popular information sources and search engines are necessary for navigation in the World Wide Web, query logs are now extremely valuable for information acquisition. Simple statistics about the most frequent queries easily disclose the demands of users in all areas from technology development to the music industry. Deeper mining into queries can reveal more important information about search engine users and their language use. Despite the large interest in pattern extraction and statistics on query logs, results are still dissatisfactory. Interesting approaches like correlation analysis for extracting compounds and collocations (Silverstein et al. 1998), query clustering (Beeferman and Berger 2000), extraction of correlating terms for query extension (Cui et al. 2002) and transforming query phrases for better question answering on the Web (Agichtein et al. 2001) have shown that query logs can compete with conventionally used corpora like newspaper articles. Furthermore, they can be used

---

1Work partially done during an internship at exorbyte GmbH (http://www.exorbyte.com) in Winter 2003/04
for the automatic generation of natural language resources, which are perfectly adapted for Web retrieval.

The main part of this paper presents my experiments in query log mining. One of the most important steps of query processing is the segmentation of the query into terms. Many user queries contain compounds (e.g. "[green card] for a job in the [united states]"). Tokenizing the query without considering compounds and phrases leads to the loss of the query sense. In Section 4.1, I describe a technique for the automatic construction of a compound dictionary, leaned against decomposition for German and Dutch presented by Chen (2002). This dictionary will be used in the query preprocessing step of the approach in automatic detection of query term similarities. I developed a statistical technique, which uses co-occurrences of query terms in the query log to determine their semantic distance. Section 4.2 shows the creation of a similarity dictionary in detail. After the evaluation of the results in Section 5, I will discuss the advantages of query logs and the reasons why they should be better analyzed in the future (Section 5).

2 Approaches in Query Log Mining

Because the search engine operating companies do not want to disclose proprietary information, there are still very few publications about query log analysis. Nonetheless, statistical analysis on query logs is very important not only to understand how human use search engines to find information they are interested in, but also to reveal new information from the search requests. Some approaches have shown that even simple counting of queries and query terms can describe behaviour of search engine users (see Jansen et al. (1998), Silverstein et al. (1998) and Cacheda and Viña (2001)). E.g. Silverstein et al. (1998) demonstrate that user queries are very short (in average 2.35 terms per query), which makes sophisticated natural language analysis, as needed in standard IR (e.g. information requests at TREC), unessential for Web retrieval.

More detailed approaches were made using query data collected from large-scale engines like AltaVista (Silverstein et al. 1998), Lycos (Beeferman and Berger 2000), Encarta (Cui et al. 2002) and (Wen et al. 2002). Besides user queries, some of these query logs contain other components such as "clickthrough data" (query and the URL, which the user selected from among other offered candidates for this query), result screens and submitter information. These data allow graph-based techniques for query and document clustering (Beeferman and Berger 2000), (Cui et al. 2002): related query terms and URLs are identified by their co-occurrence in the clickthrough data. Another interesting approach for query clustering presented in Wen et al. (2002), combines cross-references between users’ queries and the documents they clicked on with similarities between query terms. The term similarities are computed with a cosine correlation function, applied to terms weighted with TF*IDF. Agichtein et al. (2001), de Lima and Pedersen (1999) and Silverstein et al. (1998) use pure user queries as a corpus, without considering additional features common corpora do not possess. The latter used a Chi-squared test for correlation analysis of the most frequent 10,000 query terms and yielded phrases such as “cindy crawford”, “visual basic” a. o. The combination of a part-of-speech tagger and a query grammar (a context free grammar with 300 rules) in de Lima and Pedersen (1999) detects phrases like “free java games”, “history of stock market” and “howard m. dean”. In the next step, these phrases were transformed into a short, more precise form (“free games”). The evaluation demonstrated that this technique can improve the average precision of top ranked results.

Thus, elaborated statistics applied on large query logs with millions of different queries can
be used to extract linguistic information about the language of the Internet users. These data enable the improvement retrieval methods.

3 Query Logs Used in this Project

In this project I used query logs in three languages from two different search engines. The first query log (henceforth query log EN) comes from a big commercial search engine in Great Britain and contains mostly English queries (total number of queries is 71 Mill. where 19,8 Mill. are unique). The other query logs stem from an international paid-listings-provider with different domains for Germany and the Netherlands. The German query log (DE) contains 5,8 Mill. queries where 874,000 are unique. The query log from the Netherlands (NL) contains mostly Dutch queries (totalling 14,7 Mill. queries, 2 Mill. unique). The reason for using query logs in different languages is to show that the presented approaches are language-independent. Unlike the query logs used in Cui et al. (2002), Silverstein et al. (1998), that contain several components (sessions cookies, submitter information etc.), I only had access to pure user queries, which were collected over a period of one month.

On each query log, I applied the statistical methods proposed in Silverstein et al. (1998). The results of these statistics (Table 17.1 and 17.2) demonstrate that the distribution of query length in terms in the query log EN has a similar structure to the Alta Vista query log described in Silverstein et al. (1998), while results for query logs DE and NL are similar to those presented in Wen et al. (2002). These differences can be explained by the nature of the search engines: The main purpose of paid-listings-providers and Encarta is to search for products or word definitions and facts. Common search engines like Alta Vista are more popular and are used to search for every day needs, which are mostly expressed in phrases. These three query logs have in common, that the distribution of query frequencies is in all three query logs very similar (Table 17.2). Furthermore, in all query logs a few queries cover a significant portion of the whole log. The 25 most frequent queries cover 3.07% of all queries, although these are only 0.00013% of all different queries (query log EN). These queries represent the most frequently asked categories or subject areas Internet users are interested in: search engines (google, yahoo, ask jeeves), email (hotmail, msn), flights (cheap flights, easyjet), sex (sex, porn), games, news (bbc, weather) etc.

Since English has become the world language, a lot of English words are internationally understood and used (e.g. weekend or software). It is interesting to see, that about 24% of queries in the query log DE contain English terms (“small business directories”, “beauty”, “steel”), while only 72% of queries consist of German terms. This ratio was computed on a sample of 500, randomly selected from the 10,000 most frequent queries.

Another interesting data ascertainments concerns the frequency of proper nouns in the query logs, such as brands and company names (“coca cola”, “e plus”), geographical names (“south african airways”, “volksbank hoogstede”), song, movie, tv-show titles (“once upon a time in mexico”, “wer wird millionär”, “gazz”) etc. These statistics were computed for the German sample set mentioned above and additionally for the sample set from query log EN, created in the same way. Independent from the sample size (100, 200 ... 500 queries), the portion of proper nouns vs. nouns remains constant. As shown in Figure 17.1, query log EN consists of an average of about 45% proper nouns, while the query log DE has noticeably more general nouns (an average of 68%).
4 Description of the Experiments

As shown in Section 2, there are a lot of possibilities for query log mining: query clustering, phrase detection, contextual analysis, pattern mining for information extraction or just simple statistical analysis about, for example, new trends in the mobile phone industry or newly emerging terms. The ways to achieve these aims are also very different. One can use only the query log itself or with help of an additional corpus (e.g., web pages). Statistical analysis can be supported by dictionaries or NLP-tools such as stemmers, taggers and chunkers, but most of them are not disposable and the development of new tools is expensive. The main goal of this work was to develop an automatic method for creating a similarity dictionary, which had to be integrated into the query processing step of a search engine. It had to be done in short terms, and it was also important not to use information sources other than the query log itself. Furthermore, the technique should be domain and language independent. For these reasons I mainly used statistical analysis.

Table 17.2: Query length in terms in Query Logs DE, NL and EN
4.1 Compounds Extraction

Almost every analysis of a query log needs a good technique for detecting the terms a query consists of. Simple splitting at blanks is disadvantageous, because query logs contain a lot of compounds or multi-word expressions\(^2\) such as “david blaine”, “world cup”, “bed and breakfast”, “who wants to be a millionaire” (cf. Section 3). Furthermore, new compounds like product names or song titles are invented every day. Hence, compounds can not be covered by a fixed dictionary and should thus be computed dynamically. Because query logs frequently contain queries in different languages, the method for automatic compound extraction should be language independent. The first trial was to include in the compounds dictionary all phrases consisting of two or more words, which exceed a chosen co-occurrence frequency threshold. This had several disadvantages:

1. Seldom, but high correlated phrases (e.g. “birmingham international airport” or “kingston upon hull”) were not found.
2. Instead, low correlated but frequent phrases (e.g. “matrix cheats”) were detected by mistake as compounds.
3. The actual length of a phrase was not considered, so that erroneous compounds were extracted:
   (a) “red hot chilli” - instead of “red hot chilli peppers”
   (b) “dido life for rent” - instead of “dido” and “life for rent”
   (c) “britney spears me against” - instead of ”britney spears” “me against the music”

That means that in some cases expanding (a), in some cases decompounding (b) or both (c) is necessary. Due to this, the following steps were applied in the final version of the algorithm:

1. **Expanding:** If a phrase P1, consisting of x words, is a subphrase of another phrase P2, consisting of x+1 words, P1 will be substituted by P2 in a collection of expanded phrases.
2. **Decomposing** (as shown in Chen (2002) and Martínez-Santiago et al. (2003)): All expanded phrases will be split according to the following rules:
   (a) If a phrase is in a base form dictionary, it will not be split.
   (b) The shortest decomposition (with a minimal number of base forms) will be chosen.
   (c) In case of several possible decompositions, the one with the highest probability will be chosen. The probability of a decomposition is the sum of probabilities of all base forms.

The base form dictionary used in the decomposing step consists of all terms of the query log, which utilises a hidden advantage of a query log: some users write high correlated phrases as one word (e.g. “matrixreloaded”, “thematrix”, ”enterthematrix”). Low correlated phrases (e.g. “matrix-cheats”) are seldom or never written as one word. Additionally, one can add known compounds extracted from any electronic dictionary (e.g. WordNet) to the base form dictionary to improve its quality. To keep my method language independent, I didn't use any additional sources.

\(^2\)The definitions for multi-word items used in the literature diverge. A very detailed overview and classification of such sequences is provided in (Guenthner and Blanco). The definition of compounds I suggest is described in Section 5.
4.2 Extracting Semantically Similar Word Pairs

In this section I present experiments in statistical extraction of semantically similar terms. The algorithm takes the 10,000 most frequent terms of the query log as input and estimates the similarity value for each term pair, due to their co-occurrence behaviour in the log. The output is a similarity dictionary: a ranked list of term pairs, which exceeded the similarity threshold. The method is based on the idea that semantic distance between two words depends on their contextual interchangeability (Miller and Walter 1991). The more contexts two words have in common, the shorter is the semantic distance between these words.

Each query with more than one term can be seen as a 2-tuple \{query term, subquery\}, where a subquery is the remaining part of the query and therefore the context of the given query term. E.g. a query “cheap car hire london” contains three tuples: \{cheap, X car hire london\}, \{car hire, cheap X london\}, \{london, cheap car hire X\}, where X is a placeholder for the query term. For each term pair, which co-occurs with a minimum of common subqueries, I analyse the total number of their common subqueries and the characteristics of these terms. This analysis includes estimating the relevance of each subquery, which is affected by several factors:

a) **Number of subquery terms**: The more query terms a subquery contains, the higher its contextual information content is.

b) **Occurrences in the query log**: The more distinctive terms co-occur with a subquery in the log, the lower its relevance is. Very frequent subqueries like free, online or download co-occur with a lot of terms from almost all areas of interest and are therefore irrelevant for similarity estimation.

c) **Terms of a subquery**: Not all terms definitely indicate to which subject area a query belongs. While a term like “flights” indicates an affiliation of the query to the domain \{flights, airlines, planes\}, subqueries like “information on” or “find me” are non-distinctive and do not support the assumption of similarity. The more such irrelevant suqueries could be found, the better the results are.

d) **Order of the terms**: Though the language used in a query log does not conform to grammatical rules, there are still many queries that form complete or partial noun phrases: “facts about britney spears”, “cheap flights”, ”property for sale in france” etc. Therefore, it is obviously good to consider the order of terms while analysing the contexts.

Due to these heuristics, the information content for each subquery in the query log was computed. Then the characteristic of each query term could be estimated as a sum of frequencies of all queries, where the term appear, weighted by the information content of the subquery. This value was used to determine the direction of the semantic relation between terms. In case if the relation is asymmetric (e.g. hypo- and hyperonymy), the more generic term always has a higher characteristic value than its species. I used the Dice-coefficient (Manning and Schütze 1999) to compute the overlap \(O\) of two terms \(A\) and \(B\) in the query log:

\[
O_{(A,B)} = \sum_x \min(f'(Ax), f'(Bx))
\]

(17.1)

\(f'(Ax)\) is here the frequency of the query \(Ax\), weighted by the information content of the subquery \(x\). The semantic distance or similarity penalty \(PEN_{A\to B}\) is estimated for both terms.
as:

$$PEN_{(A \rightarrow B)} = \frac{O_{(A, B)}}{F(A)}$$  \hspace{1cm} (17.2)

$F(A)$ is here the characteristic of the term $A$, as described above. This is the final cost function for substitution of the specific term by the generic term usable in a query extension application. The higher the overlap of $A$ and $B$ is, the lower the penalty will be. In case of exact synonymy or equivalence I expect to get $PEN = 0$ in both direction.

5 Evaluation

The most established evaluation criteria in the IR are recall and precision. They describe the ratio of relevant found results to the answer set (precision) or to all relevant results in the document set (recall). Computing recall means extraction of all possible compounds and semantically similar words from the query log manually, which is a very complex task. However, the shortening of the query log would dramatically lower the significance of the statistics. In order to compute the precision, I only need to evaluate word pairs returned by the algorithm, which is easier and more relevant for my purposes.

The complete evaluation was made on sample sets extracted from the top-ranked 1000 results. In order to evaluate the extraction of compounds from the query log EN and DE, two sample sets each composed of 300 compounds were rated by two native speakers. To compute the inter-rater reliability (IRR) in each set the randomly selected compounds were the same. Because the concept compounds is vague, all raters got brief instructions based on the definitions in Quirk et al. (1985). The following is the extract from these instructions:

**Instructions for the Evaluation of Compounds**

- Definition: A compound is a lexical unit consisting of more than one base and functioning both grammatically and semantically as a single word (concept). E.g.: washing machine, hot dog.
- Compounding can take place within any of the word classes resulting above all new nouns and, to a lesser extent, adjectives.
- Although both bases in a compound are in principle equally open, they are normally in a relation whereby the first is modifying the second. E.g.: pine tree, meat delivery, language teacher.
- In contrast to noun phrases, in English, compounds have primary stress on the first constituent. E.g.: a ’blackbird’ vs. the phrase a ’grey’bird.
- Consider all multi-word-expressions and proper names (locations, names, brands, song titles etc.) as good compounds. E.g.: las vegas, st petersburg, michael jackson, windows xp.
- Don’t pay attention to bad spelling. If an expression should be written in one word, it is also a good compound. The main idea is to extract all expressions, which may not be split.
- Judge each compound with 1 (good), 0 (bad) and -1 (unknown).
<table>
<thead>
<tr>
<th>Query log</th>
<th>Compound</th>
<th>R1</th>
<th>R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN</td>
<td>air conditioning</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>distance learning</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>self build</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>yu gi oh</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>DE</td>
<td>lueneburger heide</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>geld verdienen</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>x2 die bedrohung</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>uni duesseldorf</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>rund um</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>NL</td>
<td>vroom en dreesman</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>zone energie</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>s hertogenbosch</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 17.3: Examples from the evaluation table for extracted compounds

<table>
<thead>
<tr>
<th>Word 1</th>
<th>Word 2</th>
<th>Query</th>
<th>Question 1</th>
<th>Question 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>volkswagen</td>
<td>vw</td>
<td>X campervan sale</td>
<td>R1</td>
<td>1</td>
</tr>
<tr>
<td>erotic</td>
<td>sexy</td>
<td>X chinese girls</td>
<td>R2</td>
<td>1</td>
</tr>
<tr>
<td>cds</td>
<td>dvds</td>
<td>cheap blank X</td>
<td>R3</td>
<td>1</td>
</tr>
<tr>
<td>spares</td>
<td>accessories</td>
<td>kenwood chef X</td>
<td>R4</td>
<td>0</td>
</tr>
<tr>
<td>revision</td>
<td>coursework</td>
<td>a2 psychology X</td>
<td>R1</td>
<td>0</td>
</tr>
<tr>
<td>fiesta</td>
<td>escort</td>
<td>X body kit</td>
<td>R2</td>
<td>0</td>
</tr>
<tr>
<td>accessories</td>
<td>reviews</td>
<td>suzuki jimmy X</td>
<td>R3</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 17.4: Examples from the evaluation table for extracted similar words

The evaluation yields precision of 89.2 by IRR of 83.9 for English and precision of 87.9 by IRR=88.3 for German. Only small test with one native-speaker and two samples with 100 results each was made for the Dutch query log, where the precision of 94.6 was reached. To show the typical quality of the results, I present an extract of the evaluation in Table 17.3.

Unfortunately there is no sufficient test base for the field of semantic similarity estimation. Most authors compare their results with human judgements (only 30 word pairs) as published in Miller and Walter (1991). To evaluate my approach, I created an own test set consisting of 100 word pairs, which were randomly selected from the top-ranked 1000 similar pairs extracted from the query log EN. Four native English speakers (R1 - R4) were then asked to rate this set. Because the similarity depends on context, each word pair was provided with a randomly selected query (cf. Table 17.4). The rater had to answer the two following questions with yes (1) or no (0): “Is the sense of the query by using both words instead of X the same?” and “If you used the query with the first word, would you be interested to get results from the query with the second word?”. This interchangeability test refers to the works described in Miller and Walter (1991). To compute the precision I considered only those pairs as similar, which have passed the
test by more than two judges. The first question yielded 60% precision, and the average inter-rater reliability was here 78.2%. The second question was approved by 87% of word pairs, with the average IRR of 84%. I didn’t evaluate the German and the Dutch query logs, because as described in Section 3, only a few queries consist of more than one term. Therefore, they do not provide a good co-occurrences source as the English query log does and will probably yield worse results. Table 17.5 contains some examples in order to show the typical quality of my results as proposed in Lin (1998). Numbers in brackets are penalty scores on a scale from 0 (identical) to 30 (low similarity).

<table>
<thead>
<tr>
<th>Query log</th>
<th>Part of speech</th>
<th>Query term</th>
<th>Extracted similar words</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN</td>
<td>Noun</td>
<td>hotels</td>
<td>b&amp;b (2), guest houses (3), motels (4), bed breakfast (6), guest house (6), inns (7), resorts (14), accommodation (14), inn (18), flights (28), holidays (28), villas (29)</td>
</tr>
<tr>
<td></td>
<td>Noun</td>
<td>jobs</td>
<td>vacancy (3), careers (7), job vacancies (2), recruitment (16), employment (17), qualifications (17), agencies (20), courses (23), equipment (23), training (24), working (26), recruitment (28)</td>
</tr>
<tr>
<td></td>
<td>Adjective</td>
<td>naked</td>
<td>pictures (11), nude pics (13), topless (14), nude (15), pictures (19), pics (23), sexy (24)</td>
</tr>
<tr>
<td></td>
<td>Verb</td>
<td>apply</td>
<td>application (9), applying (10)</td>
</tr>
<tr>
<td>DE</td>
<td>Noun</td>
<td>ferienhaus</td>
<td>ferienwohnung [holiday flat] (16), ferienhau [summer cottages] (18), urlaub [holiday] (18), hotel (20), immobilien [real estate] (20)</td>
</tr>
<tr>
<td></td>
<td>Adverb</td>
<td>neuwertig</td>
<td>op [still in package] (10), top (14)</td>
</tr>
<tr>
<td>NL</td>
<td>Noun</td>
<td>vakantie</td>
<td>vakanties [vacations] (5), vliegreizen [air trip] (6), vakantiepark [holiday village] (8), last minute (9), bungalowpark (10), campings (10), vliegen [travel by air] (10), reizen [travel] (11), hotels (15)</td>
</tr>
<tr>
<td></td>
<td>Adjective</td>
<td>naakt [naked]</td>
<td>bloot [nude] (6), playboy (10), sex (10), naked (12), nude (13), naakte (15), geile [horny] (16)</td>
</tr>
</tbody>
</table>

Table 17.5: Examples for extracted similar words from all query logs

Another interesting way to evaluate the proposed approach for extraction of similar words would be by applying the same algorithm to a standard corpus, after it was tagged according to part of speech and split into phrases with a Chunker. This experiment is planned for further research.

6 The Advantages of Using Query Logs

The evaluation of methods for compounds extraction and creating a similarity dictionary shows that even simple techniques can yield good results in information extraction from a pure query log. This fact can be explained by considering the structure and the characteristics of query logs. Each query is a compressed formulation of the information request of the search engine
user. People represent their questions as concisely as possible and use casual terms. Applying context based methods, as described in Subsection 4.2, does not need any preparatory work to extract good and correct contexts: they are already provided by the query log itself. The next advantage is the good quality of resulting words, because they contain only common terms used in everyday language. After integrating methods in the query processing stage of a search engine, the updating of automatically created dictionaries using the most recently recorded query log is always possible. Furthermore, the results are language and domain independent, which is the most important advantage. Statistical methods can be easily used for query log data from different search engines. Using query logs from a domain-specific search engine will also return results from this particular domain. Although the query logs in other languages seem to contain about 25% English queries (as shown for the German query log in Section 3), the method yields usable results (cf. Section 17.5, where English and Dutch similar words were found for the Dutch word “naakt” [naked]).

7 Conclusion

In this paper I presented known approaches for query log mining and experiments in applying simple statistical methods for the extraction of important natural language resources, which yielded remarkably good results. Advantages of using query logs as corpora, summarized in previous section, demonstrate that query logs possess features, which are unusual for normal corpora, but very beneficial for information extraction. Unfortunately there are still very little research in the exploration of these features and using query log as corpus in general. Search engines are still restricted to a simple full text search, without making use of any natural language methods. Including automatically created dictionaries with frequently used phrases, similar words or other useful information in query processing, would be a step towards improving Web retrieval and making it more sophisticated.

Acknowledgements

I would like to thank all human testers, who helped me to evaluate the sample sets. A special thanks to my mentor Benno Nieswand and the company exorbyte GmbH, where I learned so much.
References


On Epistemic Temporal Strategic Logic

SIEUWERT VAN OTTERLOO
University of Liverpool
sieuwert@csc.liv.ac.uk

GEERT JONKER
Universiteit Utrecht
gertj@cs.uu.nl

Abstract.
ATEL is one of the most expressive logics for reasoning about knowledge, time and strategies. Several issues around the interpretation of this logic are still unresolved. This paper contributes to the ongoing discussion by showing that agents do not have to know a specific strategy for doing something in order to have a capability. Furthermore we claim that agents can possess so-called strategic knowledge that is derived from their knowledge of strategies being played. In order to prove these claims we present an alternative interpretation of ATEL over extensive game forms. For the definition of abilities we use strategy domination, and to deal with strategic knowledge we include strategy profiles in the model. We illustrate the interpretation issues mentioned using several small examples. Furthermore we show how perfect recall and perfect memory can be characterized.

1 Introduction

Logic and game theory are important for understanding language and communication. Logic because it can be used to formalise ideas and notions, and game theory because communicating agents often have goals to reach with their communication. In this paper we look at a logic called ATEL (van der Hoek and Wooldridge 2002), which is a logic to reason about knowledge, time and strategies. This logic can be applied to the formal analysis of a wide range of systems, for instance distributed protocols, synchronisation and security, but also to any issue that can be modeled as an extensive game, such as argumentation, auctions or language games. In this paper we are concerned with the interpretation of ATEL, and we use simple abstract games for the issues at hand.

The original interpretation of the language featured attractive computational properties, but suffered from some counterintuitive properties, which led to a number of refinements (van der
Hoek and Wooldridge 2003; Jonker 2003; Jamroga and van der Hoek 2003). In this paper we provide a new interpretation for ATEL. In order to keep things simple we do this for turn based systems, but we see no reason why the approach could not be extended to the more general class of concurrent systems.

The history of ATEL started with the definition of Computation Tree Logic, a logic for branching time models (Clarke and Emerson 1981). Using this logic one can express many temporal properties of distributed or concurrent systems, such as ‘all possible computations will reach this state’ or ‘this state eventually occurs in at least one computation’. It was discovered by Alur and others (Alur, Henzinger, and Kupferman 1997) that this logic could be extended to reason about multiagent systems without changing the complexity of model checking. They extended the language with coalition operators and called it Alternating-time Temporal Logic (ATL). Using ATL one could express properties like ‘This coalition can ensure that a state is reached’ or ‘This coalition cannot avoid that p always holds’. Van der Hoek and Wooldridge realized that it would be useful to reason about the knowledge of agents within ATL, and extended the language with a knowledge operator calling it ATEL (van der Hoek and Wooldridge 2002). They gave the knowledge operator a very intuitive interpreted systems semantics (Fagin, Halpern, Moses, and Vardi 1995) but did not alter the notion of a strategy used in the coalition operator. The result was that agents were assumed to always be able to make different choices in different states, even if the agent could not distinguish these states. Assuming that agents can make different choices in states they cannot distinguish is counterintuitive (Jonker 2003) and different semantics along the lines of imperfect recall ATL (Schobbens 2004) were developed by Jonker, Van der Hoek and Jamroga (Jonker 2003; Jamroga and van der Hoek 2003).

In plain English, ATEL originally assumed that a coalition of agents can achieve something if a strategy to achieve it exists. This condition has been strengthened in the sources cited:

- A coalition of agents can achieve X if they have a uniform strategy (or strategy-under-incomplete-knowledge) that achieves X (Jonker 2003)
- A coalition of agents can achieve X if they have a uniform strategy of which they know that it achieves X. (Jonker 2003; Jamroga and van der Hoek 2003)

In section 2 we define the notion of a uniform strategy formally, but it can be thought of as a strategy with extra restrictions so that it does not use facts that an agent is not supposed to know. One of the main points of this paper is that this latter condition is too strong. A coalition does not have to be able to identify a strategy of which they know it will be successful. A coalition of sensible agents will choose, if they cannot identify a foolproof strategy, the best strategy that they can come up with. More precisely, they will choose a strategy that is not dominated by any other strategy. We say that a coalition of agents can do X if any undominated strategy achieves X. We will define domination later in this paper.

A second point of this paper is that the knowledge of a coalition does not have to depend only on the state of a game or system, but also on the strategies they employ. It seems safe to assume that agents know what strategies they employ, and that this gives them extra information about the future. This phenomenon can be called strategic knowledge (Druiven 2002). The interpretation developed here addresses this issue by assuming all agents in a coalition know the strategy that a coalition uses. We stress that this is intended not as the final answer on this issue, but as a demonstration how one can incorporate some form of strategic knowledge.
The language CTL is in fact a syntactical restriction of the language CTL*. The same is true for ATL and ATL* and also for ATEL and ATEL*. The unstarred versions have received the most attention, because they have a low model checking complexity (van der Hoek and Wooldridge 2003). In this paper however we are more interested in the meaning of interpretations for the languages than in complexity. Therefore we prefer to work with ATEL*, the language without restrictions, rather than to define an interpretation only for ATEL.

In section 2 we present necessary definitions. Section 3 contains examples. In section 4 we define an interpretation for ATEL*. Section 5 uses the logic to analyse the examples of section 3. In section 6 two theorems are proven and section 5 is the conclusion.

2 Extensive Game Forms

Games are models for interaction between agents with different and possibly competing objectives (Osborne and Rubinstein 1994). An extensive game gives a detailed description of such interaction. It shows which decisions are made in order to reach an outcome. It can be represented in a game tree, where each leaf corresponds to an outcome and each node corresponds to a choice between options. The preferences of all agents are part of an extensive game. In many cases one wants to study the structure of a game independent of the preferences of the agents. In that case one can use the idea of a game form, which is an extensive game without preference function. The notion of an extensive game goes back to Kuhn (Kuhn 1953). We have adapted the next definition from Osborne (Osborne and Rubinstein 1994). Since the structure encodes the fact that agents may not be sure what exactly the current state is when making a decision, we speak of an extensive game with imperfect information.

We have adopted notation fashionable in game theory (Osborne and Rubinstein 1994) and notation used in coalition logic (van der Hoek and Wooldridge 2003). The set of all agents is denoted $\Sigma$ and $\Gamma$ is used for a subset of agents. Individual agents are denoted $X, Y$. For game forms and game form interpretations $F, G$ are used. Strategies are called $S, T$ or $S_\Gamma, T_\Gamma$, to indicate for which group the strategy is. A game form and a strategy together form a model, denoted by $M$. Formulas are denoted $\varphi, \psi$ and atomic propositions $p, q$. $P$ is a set of propositions. They are interpreted using a function $\pi$. Actions are $a, b$ and histories are called $h, h'$ or $j$. To improve readability, we sometimes abuse notation a little bit and write $XY$ for the set of agents $\{X, Y\}$.

**Game Form** A game form is a tuple $(\Sigma, H, Ow, \sim)$, where $\Sigma$ is a finite set of agents and $H$ is a non-empty set of histories. The set $H$ must be prefix-closed, which means that for any sequence $ha \in H$ also $h \in H$. We use the special symbol $\epsilon$ to denote the empty sequence. The set of all actions available after $h$ is denoted $A(h) = \{a | ha \in H\}$. A history $h \in H$ is terminal if $A(h) = \emptyset$. The set of all terminal histories of $H$ is denoted $Z(H)$.

The function $Ow(h) : H \setminus Z(H) \rightarrow \Sigma$ defines which player chooses the next action. Intuitively the agent $Ow(h)$ owns the history $h$, but we can also say that it decides $h$, controls $h$ or has the initiative in $h$. For each agent $X \in \Sigma$, the relation $\sim_X$ is an equivalence relation between histories, where $h \sim_X j$ expresses the fact that agent $X$ cannot tell the difference between having gone through history $h$ and history $j$. One condition applies: if $Ow(h) = X$ and $h' \sim_X h$ then also $Ow(h') = X$ and further $A(h) = A(h')$. This condition ensures that an agent knows when it can select a action and that it knows which actions are available. This definition is taken from Osborne and Rubinstein (Osborne and Rubinstein 1994), definition 200.1, where it is called
an extensive game form. We have extended the information sets such that agents also have information when they are not in charge, which is a common extension for logical purposes (van Benthem 2001; Bonanno 2004b).

**Game Form Interpretation** A game form interpretation is defined as a tuple $\langle \Sigma, H, O\omega, \sim, P, \pi \rangle$. The first elements $\langle \Sigma, H, O\omega, \sim \rangle$ are a game form. The set $P$ contains propositions and $\pi : H \rightarrow 2^P$ is a function that assigns to each history the set of propositions that are true in the final state of that history. The idea is that these propositions can be used to refer to certain histories, for instance to histories where an agent achieves a certain goal.

**Strategies** A strategy $S_\Gamma$ for a coalition $\Gamma$ is a function that takes a history $h$ such that $O\omega(h) \in \Gamma$ and returns a non-empty set of actions $S_\Gamma(h)$ such that $S_\Gamma(h) \subseteq A(h)$. This means that strategies can be non-deterministic. We sometimes call the strategy $S_\Gamma$ a strategy profile to indicate that it contains a strategy for every agent in $\Gamma$. There is no fundamental difference between strategies and strategy profiles in this paper. A strategy $S_\Gamma$ is uniform iff for all $h, j$ with $h \sim_{O\omega} j$ it is the case that $S_\Gamma(h) = S_\Gamma(j)$. A uniform strategy thus prescribes the same actions in histories that an agent cannot distinguish.

For the purposes of this paper, we think of a game form as a description of a commonly known protocol for interaction between autonomous agents. All agents have preferences and it is common knowledge that these are private, thus not known to other agents. It is also commonly known that agents with different objectives cannot communicate outside the structure of the game. If an agent adopts a certain strategy, for instance to reach a certain goal, then the agent itself knows which strategy it is playing, but other agents do not know this. If a coalition of agents has a goal, then it is assumed that agents have the right means of coordination in order to select a group strategy.

Mental capabilities are included in the definition of a game form. We assume that if an agent is for instance forgetful, that this has been encoded in the equivalence relation in the game form. The relations in the game form thus represent what an agent knows from all its sources, not just its observations. Under this assumption it makes sense to relate properties of agents, such as perfect recall and perfect memory in section 6, to properties of the equivalence relations.

### 3 Scenarios

To illustrate the principles of ATEL covered in this paper we will first have a look at some examples. The first game form we study is game form $G_1$ in figure 18.1. In this game agent $A$ can first choose between action 1 and action 2. Agent $B$ then chooses for either action 3 or action 4. The dashed line indicated that agent $B$ does not know what agent $A$ has done when $B$ has to choose. For this game form we are interested in the strategic abilities of agent $B$. In the original ATEL interpretation agent $B$ has a strategy in $t_0$ to satisfy $q$. If $A$ chooses action 1 it would choose action 3, if $A$ chooses action 2 it would choose action 4. However, at $t_1$ agent $B$ doesn’t know whether agent $A$ has played action 1 or 2. In terms of Jonker (Jonker 2003), agent $B$ doesn’t have a uniform strategy to satisfy $q$ in $t_0$. In $t_1$ things are a bit different. Agent $B$ does have a uniform strategy to achieve $q$, but doesn’t know which one. In terms of Jonker and Van der Hoek and Jannenga, it cannot identify the right strategy. If agent $B$ would want to satisfy $p$, it would be able to identify a uniform strategy in $t_1$: choosing action 3.
Figure 18.1: A simple game form $G_1$

Figure 18.2: Game form $G_2$

But what if agent $B$ would want to satisfy $p \land q$? At $t_1$ it wouldn’t be able to identify a uniform strategy, since it doesn’t know whether agent $A$ has played action 1 or 2. Nevertheless, it has a 'best bet' strategy: playing action 3. In the view of the agent, this strategy might make it reach its goal. There is no other strategy that is better than this one, so we call this strategy undominated. It would be dominated if there existed a strategy that performed better in some of the indistinguishable states, and equally in the others. We will define this more formally in section 4. Let us now assume that we are in the left situation, after action 1. We notice that if agent $B$ plays his 'best bet' strategy, it will achieve his goal. We therefore say that agent $B$ is able to satisfy $p \land q$, or that $p \land q$ is achievable for agent $B$.

Game form $G_2$ in figure 18.2 illustrates that an agent can have several undominated strategies. We assume the current situation is the one after action 2. Agent $B$ wants to make $p$ true in the next states. The agent has two actions to choose from, and it cannot identify which action is best for achieving $p$. In the left situation, the appropriate strategy would be action 4. In the right situation, the appropriate strategy would be action 5. For the middle situation however it does not matter which strategy the agent chooses. Either action leads to a desired state. Thus, the agent has two strategies that might make him reach his goal and to him it is not clear whether to use one or the other, they are both undominated. The best he can do is to randomly choose one of the two. Fortunately, since agent $A$ has played action 2, agent $B$ will reach his goal with any of his undominated strategies. Therefore, we say that agent $B$ is able to satisfy $p$, or that $p$ is achievable for agent $B$.

Both in game forms $G_1$ and $G_2$, the ability of agent $B$ to reach his goal was facilitated by agent $A$'s choice. In $G_1$, we saw that agent $B$ was able to satisfy $p$ and $q$ because agent $A$ had chosen to play action 1. We say that agent $A$ has the ability to give agent $B$ the ability to satisfy $p$ and $q$. The fact that agent $A$ moves before agent $B$ does, does not imply that agent $B$ cannot enable agent $A$ to reach certain goals. If agent $B$ decides beforehand to play action 3, it in a way gives agent $A$ the ability to satisfy (eventually) $p$ and $q$.

In figure 18.3 another game form is depicted. This game form is unusual since agents lose information, according to the relations indicated. Agent $A$ seems not to remember, after action 1 that it has chosen this action. Similarly, $B$ knows the difference between histories 1 and 2 but one step later it cannot distinguish 13 and 23. These peculiarities are discussed in section 3 and section 6.

The last scenario pictured in figure 18.4 deals with the question whether agents can lose abilities by choosing certain strategies. This is a minor issue, not very related to the previous issues. Nevertheless it illustrates one issue in the interpretation of ATEL. Assume that the leftmost state is the current state. The interesting property of this model is that the agent has

194
the ability to make $p$ true in the next state. It cannot get rid of this ability in one step. If we assume that agents can unconditionally commit to strategies, so that they cannot change later their strategy, then $A$ would be able, by committing to go to the rightmost state, to causing itself not to be able to achieve $p$. This is not what we think of as intuitive. Therefore we assume that agents can always change their own strategy at a later moment.

Finally, we turn back to game form $G_1$ again. Consider the scenario where agent $A$ has decided to play the strategy of choosing action $2$. At time $t_0$, this influences the knowledge of agent $A$. It now knows for example that agent $B$ won't be able to satisfy $p \land q$ anymore. This is called strategic knowledge (Druiven 2002). It is a kind of provisional knowledge; provided that an agent will follow the strategy it has committed to, it has more knowledge about the future. The same holds for a coalition; provided that they follow the same strategy and that this is common knowledge between them, their knowledge about the future increases. The next section shows how strategic knowledge, along with the other principles explained above, can be formalized using a new interpretation of $\text{ATEL}^*$.

4 Strategic Temporal Epistemic Logic

The language $\text{ATEL}^*$ is the smallest language $L$ such that for any formula $\varphi, \psi \in L$, any coalition of agents $\Gamma$ and any agent $X$ it is the case that:

\[
\begin{align*}
\varphi \lor \psi &\in L & \neg \varphi &\in L & \Box \varphi &\in L & \varphi \mathcal{U} \psi &\in L \\
\Diamond \varphi &\in L & K_X \varphi &\in L & \ll \Gamma \gg \varphi &\in L
\end{align*}
\]

This language is in fact a mixture between $\text{ATL}^*$ and epistemic logic (Fagin, Halpern, Moses, and Vardi 1995). The reader will be familiar with disjunction ($\varphi \lor \psi$, 'or') and negation ($\neg \varphi$, 'not'). The temporal operators $\Box$, $\mathcal{U}$ and $\Diamond$ say something about the future. $\Box \varphi$ means that $\varphi$ is true in all future states. The formula $\varphi \mathcal{U} \psi$ expresses the fact that at a certain point in the future $\psi$ becomes true, and until that time $\varphi$ is true. The next-state operator $\Diamond \varphi$ expresses the fact that in the next state $\varphi$ is true. The epistemic operator $K_X \varphi$ indicates that agent $X$ knows that $\varphi$ holds. The coalition operator $\ll \Gamma \gg \varphi$ expresses that the set of agents $\Gamma$ can ensure that $\varphi$ holds.

We interpret formulas $\varphi$ over a model $M$ and a history $h$ and write $M, h \models \varphi$ if the formula is true. Unlike previous interpretations we include strategies in the model. A model $M$ is a pair $(F, S_\Sigma)$ where $F$ is a game form with interpretation and $S_\Sigma$ a strategy for all agents.

195
We define the neutral strategy $S^0_\Gamma$ as the strategy which returns all available actions: $S^0_\Gamma(h) = A(h)$. Strategies for different coalitions can be combined using the function combine into a function for a larger coalition. The strategy $\text{combine}(\Gamma, S_\Gamma, T)$ is equal to $S_\Gamma$ for agents in $\Gamma$ and equal to $T$ for other agents:

$$\text{combine}(\Gamma, S_\Gamma, T)(h) = \begin{cases} S_\Gamma(h) & \text{if } O \in \Gamma \\ T(h) & \text{if } O \notin \Gamma \end{cases}$$

A model $M = (F, S_\Sigma)$ contains information about all strategies that the agents currently use. An agent only knows his own strategy. It knows it will follow that strategy, but assumes that nothing more is known about others than that they adhere to the neutral strategy $S^0_\Sigma$. Thus, when evaluating the knowledge of an agent about the future, we use the model that is the result of the agent using its strategy, while the others use the neutral strategy. This strategy is in fact the least restrictive strategy indistinguishable to the agent. We define it using the operator $K(M, X)$ which is defined as

$$K((F, S_\Sigma), X) = (F, \text{combine}(X, S_\Sigma, S^0_\Sigma))$$

The function $K$ is used in the definition of the knowledge operator.

In order to define the meaning of the strategic operator $\ll \Gamma \gg$, we use the idea of undominated strategies. Informally a strategy $S$ dominates $T$ if $S$ is strictly better than $T$ for reaching a certain goal. A coalition of rational agents, we assume, will not play a strategy if that strategy is dominated. To define domination, we first we need to define two other operators. The operator update is similar to $K$: it returns the model that represents the view of agents in coalition $\Gamma$ after the adoption of strategy $S_\Gamma$. Agents in $\Gamma$ adhere to $S_\Gamma$ but other agents can act in any way they want:

$$\text{update}(M, S_\Gamma) = (F, \text{combine}(\Gamma, S_\Gamma, S^0_\Sigma))$$

Furthermore, we call a strategy $S_\Gamma$ successful $\text{(success}(M, h, S_\Gamma, \varphi))$ in history $h$ (for $\Gamma$ with respect to $\varphi$) if and only if $\text{update}(M, S_\Gamma), h \models \varphi$. A strategy $S_\Gamma$ dominates a strategy $T_\Gamma$ with respect to a model $M$, a goal $\varphi$ and a history $h$ if two conditions are met: There is a history $h' \sim h$ such that $\text{success}(M, h, S_\Gamma, \varphi)$ but not $\text{success}(M, h, T_\Gamma, \varphi)$, and secondly there is no history $j$ such that $\text{success}(M, j, T_\Gamma, \varphi)$ but not $\text{success}(M, j, S_\Gamma, \varphi)$. This definition of dominance makes the domination relation transitive and asymmetric. These properties ensure that any nonempty, finite set of strategies contains at least one strategy not dominated by another strategy in the set. We say that achievable $\text{(achiev}e(M, h, \Gamma, \varphi)$ if for all undominated strategies $S_\Gamma$ it is the case that $\text{success}(M, h, S_\Gamma, \varphi)$. The reason that we quantify over all undominated strategies is that we imagine that a coalition picks randomly any undominated strategy, since it has no reason to prefer one undominated strategy over the other. Therefore success is only guaranteed if all undominated strategies are successful. Using all these notions we define the interpretation
of $\text{ATEL}^*$ as follows:

\begin{align*}
M, h \models p & \iff p \in \pi(h) \\
M, h \models \neg \varphi & \iff \text{not } M, h \models \varphi \\
M, h \models \varphi \lor \psi & \iff M, h \models \varphi \text{ or } M, h \models \psi \\
M, h \models K_X \varphi & \iff \forall h' : h' \sim_X h \implies k(M, X), h' \models \varphi \\
M, h \models \varphi U \psi & \iff \exists j \in Z(H(S, h)) \exists k : |h| \leq k < i \implies \\
& \quad M, j_0 \ldots j_k \models \varphi \land M, j_0 \ldots j_{k-i} \models \psi \\
M, h \models \Box \varphi & \iff \forall h' \in H(S, h) : M, h' \models \varphi \\
M, h \models \Diamond \varphi & \iff \forall a \in S(h) : M, ha \models \varphi \\
M, h \models \langle \Gamma \rangle \varphi & \iff \text{achievable}(M, h, \Gamma, \varphi)
\end{align*}

The set $H(S, h)$ contains all histories of $H$ that start with $h$ and are consistent (after $h$) with the strategy $S$. It can be defined recursively. $H(S, h)$ is the smallest set $H'$ such that $h \in H'$ and $\forall h' \in H', \forall a \in S(h') : h'a \in H'$.

5 Examples

In section 3 we have introduced four game forms. In this section we use these game forms to show the interpretation of example formulas. For all examples the model $M_i$ is defined as $(G_i, S^0_i)$. The empty history is denoted $\emptyset$. The first examples deal with temporal properties.

\begin{align*}
M_1, \emptyset \models t_0 \land \Diamond t_1 \land \Diamond t_2 & \quad \text{Initially $t_0$ holds, then $t_1$ and then $t_2$} \\
M_1, \emptyset \models \Box (t_2 \rightarrow \neg t_1) & \quad \text{If $t_2$ holds, then not $t_1$} \\
M_1, \emptyset \models \top U t_2 & \quad \text{Eventually $t_2$ holds}
\end{align*}

We have argued that in the game form $G_1$ after action 1, the agent $B$ can achieve $p$, but not $q$. It can also achieve $p \land q$ but it does not know that it can achieve this fact. The translations of these facts are given here. In model $G_2$ similar properties hold and these are also given.

\begin{align*}
M_1, 1 \models \langle \langle B \rangle \rangle \Diamond p & \quad B \text{ can make } p \text{ true in the next state} \\
M_1, 1 \models \neg \langle \langle B \rangle \rangle \Diamond q & \quad B \text{ cannot make } p \text{ true in the next state} \\
M_1, 1 \models \langle \langle B \rangle \rangle \Diamond (p \land q) & \quad B \text{ can make } p \text{ and } q \text{ true in the next state} \\
M_1, 1 \models \neg K_B \langle \langle B \rangle \rangle \Diamond (p \land q) & \quad B \text{ doesn't know it can make } p \text{ and } q \text{ true} \\
M_2, 1 \models \neg \langle \langle B \rangle \rangle \top & \quad B \text{ cannot make } p \text{ true in the next state} \\
M_2, 2 \models \langle \langle B \rangle \rangle \Diamond p & \quad B \text{ can make } p \text{ true in the next state} \\
M_2, 2 \models \neg K_B \langle \langle B \rangle \rangle \Diamond p & \quad B \text{ does not know it can make } p \text{ true}
\end{align*}

In game form $G_3$, agent $A$ does not remember the choices it has made in the past. Agent $B$ does not always know its previous observations. This is expressed in the next statements. The next model, $G_4$, shows that agents cannot in general commit themselves to act against their future preferences.
\[ M_3, \emptyset \models K_A \triangleleft A \triangleright \diamond p \quad \text{A knows it can make } p \ \text{true} \]
\[ M_3, \emptyset \models \neg \triangleleft A \triangleright \diamond K_A p \quad \text{A cannot know } p \ \text{in the next state} \]
\[ M_3, 1 \models K_B \diamond q \quad \text{B knows } q \ \text{is true in the next state} \]
\[ M_3, 1 \models \neg \diamond K_B q \quad \text{Next, B does not know that } q \]
\[ M_4, \emptyset \models \triangleleft A \triangleright \diamond \neg \triangleleft A \triangleright \diamond p \quad \text{A can make } p \ \text{true in 2 steps} \]
\[ M_4, \emptyset \models \neg \triangleleft A \triangleright \diamond \neg \triangleleft A \triangleright \diamond p \quad \text{A cannot ensure it cannot make } p \ \text{true} \]
\[ M_4, \emptyset \models \neg \triangleleft A \triangleright \neg \triangleleft A \triangleright \diamond p \quad \text{A cannot loose its ability} \]

Turning back to game form \( G_1 \), we give an example of how an agent can have strategic knowledge. Suppose that agent \( A \) has committed to the strategy of playing action \( 2 \). It then knows that agent \( B \) will not be able to achieve the satisfaction of \( p \land q \) anymore. Let \( S_{A,B} \) be the strategy profile consisting of the strategy of playing action \( 2 \) for agent \( A \) and the neutral strategy for agent \( B \). We can then represent the knowledge described above as:

\[ (G_1, S_{A,B}), \emptyset \models K_A \neg \triangleleft B \triangleright \diamond p \land q \quad \text{A knows B cannot make } p \land q \]

6 Perfect Recall and Perfect Memory

Agents have perfect recall if they never forget their previous observations and the actions they have chosen (Osborne and Rubinstein 1994). Similarly the have perfect memory (Koller and Megiddo 1992; Bonanno 2004a) if they do not forget their observations. Traditionally game theory has focused on perfect recall agents, but artificial agents in multiagent systems often do not have these properties. For the complexity of solving games, or model checking temporal formulas, it is relevant whether the systems have perfect recall of perfect memory (Koller and Megiddo 1992; Halpern, van der Meyden, and Vardi 2004). Therefore we present here two theorems that characterize whether a game form interpretation has perfect recall and perfect memory. Especially we want to illustrate the difference between perfect recall and perfect memory, since this difference does not appear in temporal logic without strategies, but does exist in games and strategic logics. Making use of the strategy profiles that we have included in the model is necessary for the proof of the perfect recall property. The authors are not sure whether a different theorem regarding perfect recall could hold for previous interpretations of ATEL”.

It can be argued that perfect recall is not a property of a game, but a property of a player in a game. However we think of the equivalence relations \( \sim_X \) in a game form \( (\Sigma, H, Ow, [\sim_a]_{a \in \Sigma}) \) as representing the knowledge of the agents. We thus assumed that the capabilities of the agents have been included in the equivalence relations. Thus we view perfect recall as a property that a game form can or cannot have. We define perfect recall in terms of observations; an agent has perfect recall if it remembers all its observations, including the actions it has chosen. Let the \( O_X(h) \) be the function returning the ordered list of all observations and actions chosen by agent \( X \) in history \( h \). Then an agent \( X \) has perfect recall if and only if \( h \sim_X j \iff O_X(h) = O_X(j) \). The function \( O_X \) can be defined recursively. The observation function of \( ha \) contains all observations of \( h \), plus maybe the action \( a \) (if \( Ow(h) = X \)) and the observation made in \( ha \). Using such recursive definition, it is not hard to show that the property of perfect recall is equivalent to the
next two properties.

\[ h a \sim_X j b \rightarrow h \sim_X j \]

\[ Ow(h) = X \land h a \sim_X j b \rightarrow a = b \]

This characterisation of perfect recall is the one we use in the next theorem.

**Theorem 1** Let \( F = (\Sigma, H, Ow, [\sim_X]_{X \in \Sigma}) \) and \( X \in \Sigma \). \( X \) has perfect recall in \( F \) if and only if for every \( P, \pi \), every \( \varphi \), every strategy \( S_\Sigma \) and every \( h \):

\[ ((F, P, \pi), S_\Sigma), h \Vdash K_X \circ \varphi \rightarrow \bigcirc K_X \varphi \]

Suppose that \( X \) has perfect recall in \( F \) and let \( P, \pi, \varphi, S_\Sigma, h \) be given. Let \( G = (F, P, \pi) \) and suppose that \( (G, S_\Sigma), h \Vdash K_X \circ \varphi \). Define \( M' = (G, S'_{\Sigma}) = k((G, S_\Sigma), X) \). Let \( a \in S_\Sigma(h) \) and \( j b \sim_X h a \). From the perfect recall properties we know that \( h \sim_X j \). From the definition of \( k \) one can see that for any \( h' \) it is the case that \( S_\Sigma(h') \subseteq S'_{\Sigma}(h') \). Using \( (G, S_\Sigma), h \Vdash K_X \circ \varphi \) we can conclude that \( M', j \Vdash \circ \varphi \). If \( Ow(h) = X \) then \( Ow(j) = X \) and we know that \( b = a \) and therefore \( b \in S_\Sigma(h) \subseteq S'_{\Sigma} \). If \( Ow(h) \neq X \) then \( S'_{\Sigma}(j) = A(j) \) and thus \( b \in S'_{\Sigma}(j) \). From \( M', j \Vdash \circ \varphi \) and \( b \in S'_{\Sigma}(j) \) we can conclude that \( M', j b \Vdash \varphi \). Since we have shown this for an arbitrary \( j b \sim_X h a \) we conclude that \( (G, S_\Sigma), h \Vdash \bigcirc K_X \varphi \).

For the second half, assume that for every \( P, \pi \), every \( \varphi \), every strategy \( S_\Sigma \) and every \( h \):

\[ ((F, P, \pi), S_\Sigma), h \Vdash K_X \circ \varphi \rightarrow \bigcirc K_X \varphi \]

Let \( G = (F, P, \pi) \) and let \( h a \in H \). Take \( j b \) such that \( h a \sim_X j b \). Let \( P = \{p\} \) and define \( \pi \) such that \( \pi(j' b') = \{p\} \) iff \( h \sim_X j' \) and \( b' \in A(j') \). Let \( S = S_\Sigma^0 \). This interpretation ensures that \( (G, S), h \Vdash K_X \circ p \). We can derive from the assumptions that \( (G, S), h \Vdash \bigcirc K_X p \). Thus for every \( a \in A(h) \) it is the case that \( (G, S), h a \Vdash K_X p \). Since \( h a \sim_X j b \), we conclude that \( (G, S), j b \Vdash p \). By definition of \( \pi(p) \) this gives us \( h \sim_X j \).

Let \( G = (F, P, \pi) \) and let \( h a \in H \) be such that \( Ow(h) = X \). take \( j b \in H \) such that \( h a \sim_X j b \). From the previous part we can already conclude that \( h \sim_X j \). Let \( P = \{p\} \) and let \( S = S_\Sigma \) be a strategy such that \( S(h) = \{a\} \). define \( \pi \) such that \( \pi(j' b') = \{p\} \) iff \( b' = a \) and \( j' \sim_X h \). These definitions ensure that \( (G, S), h \Vdash K_X \circ p \). We can derive \( (G, S), h \Vdash \bigcirc K_X p \). From the definition of \( \pi \) we know that \( (G, S), h a \Vdash K_X p \) and thus that \( (G, S), j b \Vdash p \). By definition of \( p \) we conclude that \( a = b \). This concludes the proof.

An agent with perfect memory remembers all its previous observations. Let \( M_X(h) \) be the function returning the ordered list of all observations by agent \( X \) in history \( h \) (excluding the actions it has chosen). We define that an agent \( X \) has perfect memory if and only if \( h \sim_X j \iff M_X(h) = M_X(j) \). Again, one can define the observation function \( M \) recursively. Using such recursive definition, it is not hard to show that the property of perfect memory is equivalent to \( ha \sim_X j b \rightarrow h \sim_X j \).

**Theorem 2** Let \( F = (\Sigma, H, Ow, [\sim_X]_{X \in \Sigma}) \) be a game form and \( X \in \Sigma \). \( X \) has perfect memory in \( F \) if and only if for every \( P, \pi \), every \( \varphi \), and every \( h \):

\[ ((F, P, \pi), S_\Sigma^0), h \Vdash K_X \circ \varphi \rightarrow \bigcirc K_X \varphi \]

199
In this theorem, instead of just any strategy we use the neutral strategy \( S^0 \). An important property of this strategy is that \( k((F, S^0), X) = (F, S^0) \).

For the first half of the proof, let \( X \) have perfect memory in \( F \) and assume that \( P, \pi, \varphi \) and \( h \) are given. Let \( M = ((F, P, \pi), S^0) \) and assume that \( M, h \models K_X \circ \varphi \). Let \( a \in A(h) \) and \( jb \) any history with \( jb \sim_X ha \). From the perfect memory property we know that \( j \sim_X h \). We can derive that \( k(M, X), j \models \circ \varphi \). Since \( M \) contains the neutral strategy, \( k(M, X) = M \) and thus \( M, j \models \circ \varphi \). This means that \( M, jb \models \varphi \). Since we have shown this for an arbitrary \( jb \) with \( jb \sim_X ha \) we can conclude that \( M, h \models \circ K_X \varphi \).

For the second part, assume that for every \( P, \pi, \varphi \) and \( h \): \( ((F, P, \pi), S^0), h \models K_X \circ \varphi \rightarrow \circ K_X \varphi \). Let \( ha \sim_X jb \) be given. Define \( P = \{ p \} \) and \( \pi \) such that \( \pi(h'a') = \{ p \} \) iff \( h' \sim_X h \) and \( a' \in A(h') \). This definition ensures that \( M, h \models K_X \circ p \). We can conclude that \( M, h \models \circ K_X p \). From this formula one can derive that \( M, ha \models K_X p \) and thus that \( M, jb \models p \). This means that \( \pi(jb) = \{ p \} \) and thus that \( j \sim_X h \). Therefore \( X \) has perfect memory in \( F \).

7 Conclusions and Further research

We have presented the logic \( \text{ATEL}^* \) and shown how this logic, because it deals with knowledge, time and strategies, can be used for reasoning about game situations and multi-agent protocols. There are some subtle issues in the interaction between knowing and being able to do something. By giving an interpretation based on the notion of dominated strategies, we have shown that knowing that you can do something is not necessary in order to be able to do something.

This idea has led us to present a new interpretation that considers strategies explicitly, and considers the knowledge that agents have about these strategies. We believe that this is an important issue in reasoning about games, because this is what allows agents to coordinate their actions and to reason more accurately about the future. Within this paper we have presented two technical results, namely that perfect recall and perfect memory of game forms can be characterised in terms of \( \text{ATEL}^* \). There is a lot more work to be done on this logic. In order to apply it to linguistic and game-theoretic domains it would be nice to understand which axioms are valid in this logic, and which strategic knowledge assumptions are the most natural for each domain. Another open problem is the computational complexity of this logic. If this problem is tractable, then \( \text{ATEL}^* \) could be applied to many multi agent verification problems, such as problems similar to the Russian Cards problem (van Ditmarsch 2003) and the Dining Cryptographers problem (van der Meyden and Su 2002). We conjecture that the interpretation presented in this paper may have made the model checking problem harder, but cannot present any results on this issue yet.
References


A Coreference Resolution Model on Excerpts from a Novel

OANA POSTOLACHE
Saarland University, Saarbrücken, Germany & University of Iaşi, Romania
oana@coli.uni-sb.de

CORINA FORĂSCU
University of Iaşi, Romania
corinafor@infoas.ro

ABSTRACT.
This paper describes a model for coreference resolution integrated within a general anaphora resolution engine. We focus on finding coreference chains in unrestricted text, considering all mentions of entities, expressed as noun-phrases, different types of pronouns (including demonstratives and wh-pronouns) and even numerals. We have experimented with texts from the bellettristic register. We have created a specialized corpus for coreference resolution research. While the interannotator agreement (in terms of measure) was, for some texts, as low as 60%, our system output achieves scores between 63% and 70%.

1 Introduction
Anaphora describes the language phenomenon of referring to a previously mentioned entity; anaphora resolution is the process of finding that previous item.

Anaphora resolution and the more complex problem of coreference resolution are very important for several key fields of Natural Language Understanding and Processing such as Information Extraction, Question Answering, Machine Translation and Text Summarization.

Due to its importance, many approaches have been developed using different techniques: feature-based (Lappin and Leass 1994), (Baldwin 1997), (Kennedy and Boguraev 1996), genetic algorithms (Orăşan 2000), machine learning (Aone and Bennet 1996), statistical methods (Kehler 1997), (Florian et al. 2004).

This paper describes a model for coreference resolution integrated within a general anaphora resolution engine. We focus on finding coreference chains in unrestricted text, considering all mentions of entities, expressed as noun-phrases, different types of pronouns (including demonstratives and wh-pronouns) and even numerals.
The rest of the paper is organized as follows. Section 2 describes our anaphora resolution engine into which we incorporate our coreference model. In Section 3 we discuss the corpus we have created for the purposes of coreference resolution research and which is used for experiments. Section 4 presents our coreference resolution model, its attributes, external knowledge sources, rules. Section 5 describes the evaluation metric and our results. We conclude in Section 6 and discuss future work.

2 The anaphora resolution engine

In (Cristea et al. 2002) a framework having the functionality of a general anaphora resolution (AR) engine and able to accommodate different AR models is proposed. This approach involves three layers: the text layer - populated with referential expressions (REs), the projected layer - where feature structures (Carpenter 1992) are filled-in with information from the text layer (in the rest of the paper we refer to the projected structures as PSs) and the deep semantic layer consisting of discourse entities (DEs). We say that a PS is projected from an RE and that a DE is proposed or evoked by a PS (Figure 19.1).

![Three-layer representation of two corefering referential expressions](image)

Figure 19.1: The three-layer representation of two corefering referential expressions

Any AR model is defined in terms of four components: a set of primary attributes that fill the PSs of the projection layer and their values are then propagated to the semantic layer, a corresponding set of knowledge sources that populate these attributes with values, a set of matching rules and heuristics used to decide whether an RE introduces a new DE or not and, if not, which of the existing DEs it refers to, and a set of heuristics that configure the domain of referential accessibility, thus, establishing the order in which DEs have to be checked.

The type of analysis supported by the framework is incremental, anaphoric references being processed from left to right (in left to right reading languages) reflecting the way humans read texts. Like in normal reading, anaphors are mostly resolved at the time of reading, but sometimes decisions are postponed until the acquisition of complementary information adds enough data to allow a disambiguation process. It is like when backwards eye movements reveal indecissions (Vonk 1985).

The set of attributes may be morphological (lexical number, lexical gender and person), syntactic (the head of the RE, the syntactic role of the RE in the sentence, the immediate parent and daughters in a dependency description, the quality of being in a subject, an apposition, or a predicative name position, etc.), lexico-semantic (lemma, name - for proper nouns, natural gender, either the sense of the RE’s head’s lemma as a WordNet synset index (Miller et al. 1993), (Fellbaum 1998) if a sense disambiguation source is available or just all the senses, semantic roles,
etc.), positional (the inclusion in a discourse unit), surface realisation (zero-pronoun, full-flagged pronoun, indefinite NP, definite NP, proper noun, etc.). Using Wordnet senses allows to investigate over conceptual hierarchies. Also, features as animacy, natural gender and concreteness could be considered simplified semantic tags derived from a conceptual hierarchy. Out of semantic roles, selectional restrictions, inferential links, pragmatic limitations, semantic parallelism and object preference can be verified.

The AR-engine works by scanning left-to-right the raw text in search of REs to be processed. REs are processed sequentially, one at a time and the processing involves three compulsory phases and an optional fourth one. (1) The projection phase, builds a PS on the projection layer out of the information, centered on the current RE, obtained from the text layer and with the contribution of the knowledge sources available. (2) The proposing/evoking phase, is responsible of matching the PS towards one DE, either by proposing a new one or by deciding on the best candidate from the existent ones. This process involves firstly checking whether the current PS is a first mention by looking at the kind of determiner the corresponding RE has. If it is the case of an indefinite determiner than a new DE is created. Then, if this is not the case, the set of rules defined by the AR-model are runned. The rules are applied in the following order:

- **certifying rules** which, if evaluated to 'true' on a pair (PS, DE), certify without ambiguity the DE as an antecedent of the PS.

- **demolishing rules** which rule out a possible DE as candidate of a PS (and its corresponding RE). These rules lead to a filtering phase that eliminates, from among the candidates, those discourse entities that cannot possibly be referred to by the RE under investigation.

- **promoting/demoting rules**, which increase/decrease a resolution score associated with a pair (PS, DE). These rules have attached a weight that represents the degree of relevance of each rule.

In the end, either an existing DE was firmly identified by a certifying rule or matching scores between the current PS and a set of antecedent DEs were computed. Based on these scores, three possibilities arise:

- all candidate DEs' scores are under a predefined threshold (\(\text{threshold}_\text{min}\)): a new DE is build.

- the best DE score is above \(\text{threshold}_\text{min}\) but there are more than one DE with a score over \(\text{threshold}_\text{min}\) and in the range \(\text{threshold}_\text{diff}\): not enough evidence for choosing among the top candidates. The decision about the current PS is postponed in the hope that successive processing would provide additional clues for its resolution.

- just one score is placed in the \(\text{threshold}_\text{diff}\) range: the best candidate strongly identifies itself from the remaining candidates and is therefore chosen as the antecedent.

In (3) the completion phase, the information from a PS, for which either an antecedent DE was found or a new DE was created, is integrated in the corresponding (possibly newly created) DE. Then the PS is removed from the projection layer. Only bi-directional links between the RE and the identified DE are kept. Note that a DE can be linked to many REs, when all these REs refer to the same entity, and for each attribute, the DE comprises all the different values gathered
from all REs. Finally, (4) the optional re-evaluation phase is triggered if postponed PSs remained on the projection layer, with the intent to apply matching rules again on all of them (as the new solved PS might have brought new useful information).

The AR model is responsible for setting values to $threshold_{min}$ and $threshold_{diff}$ and also for establishing what kind of anaphors are to be resolved, namely what parts of the text are considered referential expressions (REs).

3 A corpus for coreference resolution research

For our experiment we have used four chapters (approx. 19,500 words), from the original English version of the novel “1984” (Orwell 1949).

The text was first POS-tagged, then FDG-tagged (Järvinen and Tapanainen 1997) and then the NPs were extracted automatically from the FDG structure (all structures dominated by a head noun or pronoun). NP heads were also automatically extracted. To correct and complete the list of referential expressions (markables), a group of annotators eliminated manually the errors from the automatic detection of the NPs and marked new REs, other than NPs or pronouns. Our markables are generally conformant with the MUC-7 (Hirschman and Chinchor 1997) and ACE (ACE 2003) criteria, although there are some differences.

The types of mentions that we marked as referential expressions are: noun phrases: definite (the principle, the flying object), indefinite (a book, a future star) or undetermined (sole guardian of truth), singular and plural, and including names (Winston Smith, The Ministry of Love), dates (April), currency expressions ($40) and percentages (48%); pronouns: personal (I, you, he, him, she, her, it, they, them), possessive (his, her, hers, its, their, theirs), reflexive (himself, herself, itself, themselves) and demonstrative (this, that, these, those); wh-pronouns: relative pronouns which, who, whom, whose and that when they replace an entity; when they are part of an interrogative sentence they are not marked as REs; numerals, when they refer to entities (four of them, the first, the second), but not when they have an adjectival or adverbial function (He was 29, three books).

Some important aspects that have to be mentioned are that our markables do not include relative clauses, each term of an apposition is taken separately ([Big Brother], [the primal traitor]), conjoined expressions are annotated individually ([John] and [Mary], [hills] and [mountains]), and modifying nouns appearing in noun-noun modification are not marked ([glass doors], [prison food], [razor blades]). Table 19.1 shows some statistics we investigated on our corpus.

4 The coreference model

The engine presented in Section 2 allows for the integration of different models of anaphora resolution. It can be used for approaches which deal with specific types of anaphors (for example, pronoun resolution) or, if a proper model is integrated, it can solve referential relations, different than strict coreferentiality (Cristea and Postolache 2004), (Postolache 2004).

In this paper, we investigate a model for coreference resolution on unrestricted text. Our aim is to find all the coreference chains of the entities in a document. We consider every mention of an entity to be a referential expression (RE). Mentions can be expressed by noun phrases, pronouns
<table>
<thead>
<tr>
<th></th>
<th>Text 1</th>
<th>Text 2</th>
<th>Text 3</th>
<th>Text 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of sentences</td>
<td>311</td>
<td>175</td>
<td>169</td>
<td>328</td>
<td>983</td>
</tr>
<tr>
<td>No. of REs</td>
<td>1942</td>
<td>914</td>
<td>916</td>
<td>1702</td>
<td>5472</td>
</tr>
<tr>
<td>Average no. of REs per sentence</td>
<td>6.2</td>
<td>5.2</td>
<td>5.4</td>
<td>5.1</td>
<td>5.4</td>
</tr>
<tr>
<td>Pronouns</td>
<td>645</td>
<td>281</td>
<td>362</td>
<td>614</td>
<td>1902</td>
</tr>
<tr>
<td>No. of DEs</td>
<td>921</td>
<td>520</td>
<td>464</td>
<td>863</td>
<td></td>
</tr>
</tbody>
</table>

Table 19.1: Statistics on the corpus

(including demonstratives and *wh*-pronouns) and even numerals. In the following subsections we describe the components of the coreference model we propose.

4.1 Attributes of the model

The model contains the following attributes: *lemma* (of the head of the RE), *number*, *pos*, *role* (the syntactic function of the head), *femaleName* (takes the value YES, if lemma is a female name), *maleName* (takes the value YES, if lemma is a male name), *familyName* (YES, if lemma is a family name), *Ref* or *They* (the probability of a noun phrase to be referred to by the pronouns *he*, *she*, *it* or *they*), *includes* (containing a vector of IDs of REs nested in the current RE, possibly empty), *indefinite* (takes the value YES if the RE is an indefinite determined NP and NO if the RE is definite determined or undetermined), *predName* (contains the ID of the subject when the current RE is a predicative name of a form of the copular verb *to be*), *apposition* (contains the ID of the RE with which the current RE is in an apposition relation), *synonyms* (the list of the WordNet synonyms of the lemma, regardless of the sense), *hyponyms* (the list of the hypernymic synset IDs in WordNet, no matter what the sense is). There is also an attribute *wh* whose value (fetched by a special knowledge source) is YES in the case of REs such as *which*, *who*, *whose*, *whom* and *that*.

4.2 The knowledge sources

For every attribute defined within the model, we developed an external knowledge source that fetches the corresponding values. Some of these sources (those corresponding to *lemma*, *number*, *role* and *pos*) are straightforward. They make use of the output of the POS-tagger and the FDG parser used for preprocessing the input. The attributes *includes* and *indefinite* take only non-null values only for referential expressions denoted by noun-phrases. To find out whether an RE is indefinite we look at the left corner of the noun-phrase, and if it is *a* or *an* then this attribute takes the value YES. Also to find the value of the *includes* attribute, the corresponding knowledge source looks for embedded NPs within the NP which denotes the RE for which the value must be fetched.

For the *female/male/familyName* attributes we used large lists of male and female first names and also a separate list with family names. The same name could be both a male and female first

206
name (e.g., Chris, Kim, Nicola). In such cases, both attributes `femaleName` and `maleName` hold
the value YES, leaving the rules that use these attributes to take an appropriate decision.

The `predName` attribute takes a non-null value for those REs which have the syntactic role of
predicative name of the copular verb *to be* (at any time or form). The value is the ID of the NP
which stays in the subject position. This attribute will be used by a certifying rule which will
find the predicative name to refer to the DE to which the subject refers.

The `apposition` attribute takes non-null values in those cases of REs which are appositives
of other REs. Detecting automatically the appositions in a text is not a trivial task. They are
seldom marked by commas, used before and after them. However, the punctuation is not a very
reliable criterion, because it can trigger wrong decisions (for example, in the case of an enumeration),
or it can fail to find some appositives un-marked by commas, as in *The last year champion Panagioti Kouzi won the title again*. Our knowledge source responsible for this task takes into
consideration not only the presence of the commas before and after the presumed apposition, but
also imposes some syntactical and semantic restrictions. For example, the restriction of accordance in number of the two NPs would rule out cases as in *He didn’t trust her smile, her eyes.*

We also look for a hypernymic or hyponymic relation between an NP and its appositive, according
to the Wordnet hierarchy, in order to eliminate some cases of enumeration, as in *In spite of his isolation, his helplessness, and his doubt he continued to fight for his cause.* Of course, there
are cases in which appositives are metaphors and Wordnet doesn’t indicate a hyper/hyponymic
relation between the two NPs (in which case we fail to mark the apposition accordingly). But
detecting metaphoric appositions is beyond the scope of this paper.

A special knowledge source which we developed and which has a great importance in the
process of coreference resolution is the one that fetches values for the `HeSheItThey` attribute.
While in English nouns don’t have lexical gender, they have a natural gender - namely, they refer
to female persons, male persons, animals, physical things or abstract things. The `HeSheItThey`
attribute takes as a value an array of normalized counts of how often a RE is referred to by the
pronouns *he*, *she*, *it* or *they*. For REs expressed by pronouns the computation of the corresponding
value is straightforward; for example a *he* will have the value `HeSheItThey="1.0,0.0,0.0,0.0,0"`,
meaning that there is 100% chances to be referred by another *he* and 0% chances to be referred
by either *she*, *it* or *they*. Another example, for an RE expressed by *which* the value will be
`HeSheItThey="0.0,0.0,0.5,0.5"`, meaning that there are 50% chances to be further referred by *it*
and 50% chances to be referred by *they*.

For REs expressed by NPs we used the Wordnet hierarchy to compute these values. We first
find all the synsets corresponding to the head of the NP; we note their number with *n*, then we
count how many of these are hyponyms of the synsets *(female, female person - (a person
who belongs to the sex that can have babies))* - we note this number with *f*, *(male, male
person - (a person who belongs to the sex that cannot have babies))* - we note this
number with *m*, or *(person, individual, someone, somebody, mortal, human, soul - (a
human being; there was too much for one person to do))* - we note this number with *p*.
We note with *t* the number of the rest of the synsets which are not hyponyms of any of the above
mentioned synsets.

We note the normalized counts of how often a RE is referred to by a *he*, *she*, *it* or *they* with
*P_he*, *P_she*, *P_it* and *P_they* and we compute these values as follows. If the lexical number of the RE
is plural, then *P_they*=1.0 and the rest 0.0. Else, *P_he* = \( \frac{m+\frac{1}{2}(p-m)}{n} \), *P_she* = \( \frac{f+\frac{1}{2}(p-f)}{n} \), *P_it* = \( \frac{n-p}{n} \) and
*P_they*=0.0.

207
Finally, we have a knowledge source which detects the referees of the relative pronouns who, which, whose, whom and that (we call them wh-pronouns). We have developed a separate phase for these cases, because they always refer to an entity mentioned in the same sentence, and more, usually refer the entity mentioned just before them. This knowledge source will fetch values for the wh attribute, values that are the IDs of the corresponding antecedents of the wh-pronouns. We have implemented some simple heuristics that allow us to find the antecedents of the wh-pronouns. The first heuristic is that, if preceded by a non-imbricated NP, a wh-pronoun coresfers with it. Examples of this kind are (we underline the anaphor and display the antecedent in bold):

(1) He sat down at a small table that stood to the left of the telescreen.

(2) There was a shallow alcove in which Winston was now sitting.

When the preceding NP of a wh-pronoun is imbricated (contains other NPs in its structure) than the rule is as follows: the first included NP that is definite determined from right to left, if exists, will be the antecedent of the wh-pronoun (see example (3)); if such an NP doesn’t exist then the outer NP is set to be the antecedent (see example (4)). Here are some examples (we mark the imbricated NP and the included NPs by putting them in square brackets, and we bold the antecedent):

(3) They were [the homes of [the four Ministries]] between which the entire apparatus of government was divided.

(4) It was because of [the atmosphere of [hockey-fields] and [cold baths] and [community hikes] and [general clean-mindedness]] which she managed to carry about with her.

4.3 The matching rules

As we mentioned in Section 2, the AR-engine allows to define three types of rules: certifying, demolishing and promoting/demoting. Our model implements a single demolishing rule, named IncludingRule, which takes as arguments a PS and a DE. This rule implements the heuristics that an RE cannot corefer with an embedded RE.

The certifying rules contained by the model are the following: ProperNameRule, which certify that a PS denoting a name of a person refers to that DE which already has been referred by a mention with the same name. This rule uses the attributes femaleName, maleName and familyName and also lemma. Another certifying rule is WhRule, which uses the wh attribute. For a current PS, for which the rule is applied, and a candidate DE, if the value of the wh attribute of the PS (namely the ID of the antecedent detected by the previously mentioned knowledge source) matches the ID of one of the REs that constitute the DE in question, then this rule certifies that this DE is the one to which the PS (and its corresponding RE) refers to. The last two certifying rules that we have implemented are PredicativeNameRule and AppositionRule, which using the corresponding attributes, certify that a predicative name refers to the same DE which is referred to by the subject and, respectively, that an apposition of an RE corefers with that RE. The responsibility of choosing in which cases of predicative names or appositions there exists a coreference relation is taken by the corresponding knowledge sources that fetch the values for the predName and apposition attributes.
The rest of the rules implemented in our model of coreference resolution are promoting/demoting rules: they also have as arguments a PS and a DE and they return a score that corresponds to that specific DE from the list of all DE candidates. The scoring rules are the following:

- **NumberRule** returns 1 if the PS and the DE have the same value for the attribute number, else it returns 0. We have to mention that a DE may have a set of different values for an attribute, because it incorporates the values triggered from all the REs that refer to it. So, actually, the scoring rule tests the inclusion of the value of a PS attribute within the set of values of the corresponding DE attribute.

- **LemmaRule** - is similar to the **NumberRule**, only that it looks at the lemma attribute and also at the other components of the referential expression.

- **RoleRule** - this rule was implemented in order to apply it in cases of very close scores between antecedents. It returns values between 0 and 1 according to the syntactic role of the antecedents, where the role ranking is conformant with that applied in the Centering Theory (Grosz et al. 1995), namely subject > direct object > indirect object > attributive.

- **He/She/TheyRule** - takes as arguments a PS and a DE and returns values in the range 0-1. To elaborate the way these values are computed, we will use the notations $PS_{he}$, $PS_{she}$, $PS_{it}$ and $PS_{they}$ for the percentages corresponding to the PS (as described above). The DE has as a value a set of this kind of (different) percentages. If we note with $X$ an element of this set, then its corresponding percentages are denoted by $X_{he}$, $X_{she}$, $X_{it}$ and $X_{they}$. The value returned by this rule is calculated as follows: $\max_{X \in DE_{He/She/They}} [\min(PS_{he}, X_{he}), \min(PS_{she}, X_{she}), \min(PS_{it}, X_{it}), \min(PS_{they}, X_{they})]$.

- **SynonymyRule** - using the lemmas of its arguments (a PS and a DE), it investigates the Wordnet in order to find a synset that contains both lemmas. If such a synset is found, the rule returns 1, else 0. Of course, it is possible that in the text, the two lemmas have different senses (in which case the PS does not refer to the DE); because we don’t have a WSD module to help us disambiguate the senses, this rule is associated with a relatively low weight, in order not to influence drastically the final score. Most of the benefit of this rule comes when it fires simultaneously with other rules.

- **WordnetChainRule** - looks in Wordnet for a hyponymic or hypernymic chain between the lemma of its first argument, the PS, and each of the lemmas of its second argument, the DE. If such a chain is found, the rule returns 1, else 0. For the same reason as in the case of the **SynonymyRule**, the **WordnetChainRule** was also associated with a low weight.

5 Evaluation

As we mentioned in Section 4, we have used for our experiments, four texts extracted from a novel. Each text was assigned to a team of two annotators. The annotators were instructed to mark the coreference relations and they had to annotate their assigned parts individually. Considering that the chosen texts belong to the belletristic register there were pairs of annotators for which the interannotator agreement was as low as 60%. It is important to note that we
attempt to resolve coreference between all REs not only certain semantic types as in the case of the Automatic Content Extraction (ACE) task (LDC 2003) where only 5 types are considered (person, organization, geo-political entity, location, facility). (LDC 2003) have reported interannotator agreement 87.8% (ACE value) and systems can get 88.0% (ACE value) on true mentions (also the evaluation metric is different from what we use).

The metric used to evaluate the coreference model was the success rate (the ratio between the number of correctly resolved anaphors and the number of all anaphors) defined in (Mitkov 2002). We have considered each of the REs that we have marked as a potential anaphor and instead of deciding whether a certain anaphor was correctly resolved or not, for each anaphor we assign a value of correctness between 0 (meaning incorrectly solved) and 1 (meaning correctly solved). The values are computed as follows: Taking into consideration the fact that our output consists of coreference chains (sets of anaphors which refer to the same entity), we must compare the golden set of chains (obtained from the manually annotated corpus) with the system output chains (we will call it the test set). For each anaphor:

- If, in the golden set, it belongs to a chain that doesn't contain any other anaphor, then we look in the test set to see if it belongs to a similar chain, in which case it will get the value 1; else (the test chain corresponding to the current anaphor contains more than one anaphors) it will get the value 0.

- If, in the golden set, the anaphor belongs to a chain containing other n anaphors, then we count how many of these n anaphors belong to the chain corresponding to the current anaphor in the test set (we note this number with m). The ratio m/n will be the value assigned to the current anaphor.

As stated in Table 19.2, for the four texts we obtained values lying in the range 63% to 70%.

<table>
<thead>
<tr>
<th></th>
<th>Text 1</th>
<th>Text 2</th>
<th>Text 3</th>
<th>Text 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interannotator agreement</td>
<td>60%</td>
<td>68%</td>
<td>65%</td>
<td>64%</td>
</tr>
<tr>
<td>Success Rate</td>
<td>63%</td>
<td>70%</td>
<td>69%</td>
<td>65%</td>
</tr>
</tbody>
</table>

Table 19.2: Success Rate values of our system

6 Conclusions and future work

In this paper we describe a model for coreference resolution integrated in a general AR engine. We aim to find all coreference chains in a text, considering all types of referential expressions. The model does not rely heavily on large amounts of world or domain knowledge; it uses syntactic and morphological information and exploits the Wordnet for finding minimal semantic features (e.g., natural gender). The coreference task is not trivial for any type of input text, and it proved to be quite hard for texts of the bellettristic register. A special problem that we encountered while doing experiments on excerpts from Orwell’s novel is that the author uses a large amount of new (non-existing) words as: *ingsoc, newspeak, minitrue, minilove, doubleplusgood* etc, which
were difficult to tackle (they don’t appear in any dictionary or in Wordnet). Also, the texts were large enough and, as our results show, the larger the texts were, the weaker the performance of the engine was. Still, considering that even the annotation process was by itself difficult (the interannotator agreement and the success rate was measured using the same metric), a major contribution of our work is to have a system that performs better than humans for texts in the bellettristic register.

The next step that we intend to do in our future work is, besides improving the model with new (more sophisticated) attributes and rules, to test it on less problematic types of input (for example texts from newspapers, utilization manuals etc.) and, as our reviewers also suggested, to evaluate our approach against alternate stochastic and knowledge-based methods.
References


LDC (2003). http://www.ldc.upenn.edu/Projects/ACE.


Bounded Model Property
for Multi-Context Systems

FLORIS ROELOFSEN
University of Twente, the Netherlands

ABSTRACT.

Our general interest is to understand the inherent complexity of contextual reasoning. In this paper we establish the bounded model property for propositional multi-context systems with finite sets of bridge rules: the number of local models needed to satisfy a set of formulas in such systems is bounded by the number of contexts addressed by formulas in that set plus the number of bridge rules of the multi-context system. Utilizing this result we achieve NP-completeness and a tractable encoding of contextual satisfiability into propositional satisfiability, which paves the way for the implementation of contextual reasoners based on already existing propositional satisfiability solvers.

1 Introduction

A solid paradigm for contextual knowledge representation and contextual reasoning is of paramount importance for the development of sophisticated theory and applications in AI. McCarthy (1987) profusely pleaded for a formalization of context as a possible solution to the problem of generality, whereas Giunchiglia (1993) emphasized the principle of locality: reasoning based on a large knowledge base can only be effectively pursued if confined to a manageable subset (context) of that knowledge base.

Contextual knowledge representation has been formalized in several ways. Most notable are the propositional logic of context (PLC) developed by McCarthy, Buvač and Mason (1993, 1998), and the multi-context systems (MCS) devised by Giunchiglia and Serafini (1994), which later became associated with the local model semantics (LMS) introduced by Ghidini and Giunchiglia (2001). Recently, Bouquet and Serafini (2004) have proven MCS/LMS to be both conceptually and technically more general than PLC.

Contexts were first implemented as microtheories into the notorious common sense knowledge base CYC (Lenat and Guha 1990). However, while in CYC the notion of local microtheories was a choice, in contemporary settings like that of the semantic web the notion of local, distributed knowledge is a must. Modern architectures impose highly scattered, heterogeneous knowledge
fragments, which a central reasoner is no longer able to deal with. This engenders a high demand for distributed, contextual reasoning procedures.

Nonetheless, a general approach towards capturing the inherent complexity of contextual reasoning has so far rarely been pursued. NP-completeness has been established for both PLC (Massacci 1996) and MCS/LMS (Serafini and Roelofsen 2004). But these approaches are founded on and applicable to a very specific algorithm and/or formalism only. In this paper we establish a semantical property of multi-context systems, which yields a rather general insight into the computational complexity of contextual reasoning. Moreover, whereas the analysis presented here only regards MCS/LMS, in (Roelofsen and Serafini 2004) our results are shown to be equally well applicable to PLC.

We proceed as follows. After defining MCS/LMS and explicating the contextual satisfiability problem, we directly establish the central result of the paper: the number of local models needed to satisfy a set of formulas \( \Phi \) in a multi-context system MS is bounded by the number of contexts addressed by formulas in \( \Phi \) plus the number of bridge rules in MS. Utilizing this result, we achieve NP-completeness and a refined upper-bound for the time complexity of MCS/LMS-based reasoning. Moreover, we provide an encoding of contextual satisfiability problems into purely propositional ones, providing for the implementation of contextual reasoning systems based on already existing SAT solvers. We conclude with a concise recapitulation of our achievements, and some pointers to future research avenues.

2 Multi-Context Systems

A simple illustration of the intuitions underlying MCS/LMS is provided by the so-called “magic box” example (Ghidini and Giunchiglia 2001), which is depicted in Figure 20.1.

![A magic box](image_url)

Figure 20.1: A magic box.

**Example 1** Mr.1 and Mr.2 look at a box, which is called “magic” because the observers cannot make out its depth. Both Mr.1 and Mr.2 maintain a local representation of what they believe about the box. These representations must be coherent – if Mr.1 believes the box to contain a ball, for instance, then Mr.2 may not believe the box to be empty.

We demonstrate how such interrelated local representations can be captured formally. Our point of departure is a set of indices \( I \). Each index \( i \in I \) denotes a context, which is described by a corresponding formal (in this case standard propositional) language \( L_i \). To state that a propositional formula \( \varphi \) in the language \( L_i \) holds in context \( i \) we utilize so-called labeled formulas of the form \( i: \varphi \) (when no ambiguity arises we will simply refer to labeled formulas as formulas). Formulas that apply to different contexts may be related by so-called bridge rules. These are expressions of the form:

\[
i_1: \varphi_1, \ldots, i_n: \varphi_n \rightarrow i: \varphi
\]

(20.1)
where \( i_1, \ldots, i_n, i \in I \) and \( \varphi_1, \ldots, \varphi_n, \varphi \) are formulas. Notice that our language does not include expressions like \( \neg(i : \varphi) \) and \( (i : \varphi \land j : \psi) \). In bridge rule (20.1), \( i : \varphi \) is called the \textit{consequence} and \( i_1 : \varphi_1, \ldots, i_n : \varphi_n \) are called the \textit{premises}. We write \text{cons}(br) and \text{prem}(br) for the consequence and the set of all premises of a bridge rule \( br \), respectively.

**Definition 1 (Propositional Multi-Context System)** A propositional multi context system \( \langle \{L_i\}_{i \in I}, \mathcal{BR} \rangle \) over a set of indices \( I \) consists of a set of propositional languages \( \{L_i\}_{i \in I} \) and a set of bridge rules \( \mathcal{BR} \).

In this paper, we assume \( I \) to be (at most) countable and \( \mathcal{BR} \) to be finite. Note that the latter assumption does not apply to MCSs with \textit{schematic} bridge rules, such as provability - and multi-agent belief systems (Giunchiglia and Serafini 1994). The question whether our results may be generalized to capture these cases as well is subject to further investigation.

**Example 2** The MCS that formalizes the situation in example 1 consists of two contexts 1 and 2, described by \( L_1 = L(\{l, r\}) \) and \( L_2 = L(\{l, c, r\}) \), respectively. Intuitively, atomic proposition \( l \) \((c, r, \text{respectively})\) corresponds to the existence of a ball in the left \((\text{center}, \text{right}, \text{respectively})\) section of the box. The constraint that Mr.2 may not believe the box to be empty if Mr.1 believes it to contain a ball, can be captured by the following bridge rule:

\[
1 : l \lor r \quad \rightarrow \quad 2 : l \lor c \lor r
\]

Let \( M_i \) be the class of classical interpretations of \( L_i \). Each element \( m \) of \( M_i \) is called a local model of \( L_i \). Interpretations of an entire MCS are constructed from sets of local models.

**Definition 2 (Chain)** A chain \( c \) over a set of indices \( I \) is a sequence \( \{c_i\}_{i \in I} \), where each \( c_i \subseteq M_i \) is a set of local models of \( L_i \). We say that \( c \) is \( i \)-consistent if \( c_i \) is nonempty, and \( J \)-consistent, for some \( J \subseteq I \), if it is \( j \)-consistent for all \( j \in J \). A chain \( c \) is point-wise if \( |c_i| \leq 1 \) for all \( i \in I \), and set-wise otherwise.

A chain can be thought of as a set of “epistemic states”, each corresponding to a certain context (or agent). The fact that \( c_i \) contains more than one local model amounts to \( L_i \) being interpretable in more than one unique way. So, set-wise chains correspond to partial knowledge, whereas point-wise chains indicate complete knowledge.

**Example 3** Consider the situation depicted in Figure 20.1. Both agents have complete knowledge, which corresponds to the following point-wise chain:

\[
\begin{align*}
\{\{l, r\}\}, \\
\{\{l, \neg c, \neg r\}\}
\end{align*}
\]

We can imagine a scenario however, in which Mr.1 and Mr.2’s views are restricted to the right half and the left-most section of the box, respectively, as depicted in Figure 20.2. In this case, both Mr.1 and Mr.2 have only partial knowledge; their observations may be interpreted in different ways. This is reflected by the set-wise chain:

\[
\begin{align*}
\{\{l, \neg r\}, \{\neg l, \neg r\}\}, \\
\{\{l, \neg c, \neg r\}, \{l, c, r\}, \{l, \neg r\}, \{l, c, r\}\}
\end{align*}
\]

216
Figure 20.2: A partially hidden magic box.

The epistemic states that a chain consists of concern one and the same situation. Therefore, arbitrary sets of local models may not always constitute a “sensible” chain. The somewhat vague conception of “sensibility” is captured by the more formal notion of “bridge rule compliance” specified below.

**Definition 3 (Compliance and Satisfiability)** Let \( c \) be a chain, \( \varphi \) a formula over \( L_i \), and \( BR \) the set of bridge rules of a multi-context system \( MS \).

1. \( c \) satisfies \( i : \varphi \) if \( m \models \varphi \) for all local models \( m \in c_i \). We write \( c \models i : \varphi \).
2. \( c \) complies with \( BR \) if for all \( br \in BR \) either \( c \models \text{cons}(br) \) or \( c \not\models i : \xi \) for some \( i : \xi \in \text{prem}(br) \).
3. \( i : \varphi \) is consistently satisfiable in \( MS \) if there is an \( i \)-consistent chain \( c \) that satisfies \( i : \varphi \) and complies with \( BR \).

The contextual satisfiability problem, then, is to determine whether or not a set of labeled formulas \( \Phi \) is consistently satisfiable in a multi-context system \( MS \).

**Example 4** Consider an MCS with contexts 1 and 2, described by \( L(\{p\}) \) and \( L(\{q\}) \), respectively, and subject to the following bridge rules:

\[
egin{align*}
    1 : p & \rightarrow 2 : q \\
    1 : \neg p & \rightarrow 2 : q
\end{align*}
\]

The formula \( 2 : \neg q \) is consistently satisfied in this system by the following chain:

\[
\{ \{p\}, \{\neg p\} \},
\{ \{\neg q\} \}
\]

This example reflects that an MCS cannot be encoded into propositional logic by simply labeling its propositions with the index of their associated context – such an encoding of the above system would be inconsistent. In section 4 we show that a more subtle encoding of MCS into propositional logic does exist. In order to do so, however, we must first pursue a rather more general insight into the contextual satisfiability problem and its complexity. Hereafter, we will consistently refer to the set of bridge rules of \( MS \) as \( BR \), and to the set of contexts involved by formulas in \( \Phi \) as \( J \).

## 3 Bounded Model Property

In this section we establish the bounded model property for propositional multi-context systems. This property implies that, to consistently satisfy a set of formulas \( \Phi \) in a multi-context system
MS, it should suffice to construct a chain, which consists of at most \(|J| + |BR|\) local models. This result significantly restrains the amount of time required for non-deterministically settling contextual satisfiability. Namely, in order to do so, it is sufficient to “guess” a chain with only \(|J| + |BR|\) local models, and then to check whether this chain consistently satisfies \(\Phi\) in MS.

Let us first introduce some notation and terminology. The size of a labeled formula \(i : \varphi\) is denoted by \(|i : \varphi|\). Let \(P(i : \varphi)\) and \(P(\Phi)\) be the set of propositional atoms appearing in a formula \(i : \varphi\) or a set of formulas \(\Phi\). Let \(G(i)\) be the number of local models contained by (the \(i^{th}\) component of) a chain. Let \(\Xi(br)\) and \(\Xi(BR)\) consist of the premises and the consequence(s) of a bridge rule \(br\) or a set of bridge rules \(BR\). Finally, let \(N\) be the total size of the formulas in \(\Phi\) and \(\Xi(BR)\):

\[
N = \sum_{i : \varphi \in \Phi} |i : \varphi| + \sum_{i : \xi \in \Xi(BR)} |i : \xi|
\]

**Theorem 1 (Bounded Model Property)** A set of formulas \(\Phi\) is consistently satisfiable in a multi-context system MS if and only if there is a \(J\)-consistent chain that contains at most \(|J| + |BR|\) local models and satisfies \(\Phi\) in compliance with \(BR\).

**Proof.** Take any \(J\)-consistent chain \(c\) that satisfies \(\Phi\) in compliance with \(BR\). Let \(BR^* \subseteq BR\) be the set of bridge rules whose consequences are not satisfied by \(c\). Every \(br \in BR^*\) must have a premise which is not satisfied by some local model \(m_{br}\) in \(c\). On the other hand, for every \(j \in J\), there must be at least one local model \(m_j \in c_j\) that satisfies all those formulas in \(\Phi\) that apply to context \(j\). The chain \(c^*\) obtained from \(c\) by eliminating all local models except for:

\[
\bigcup_{j \in J} m_j \cup \bigcup_{br \in BR^*} m_{br}
\]

is \(J\)-consistent, satisfies \(\Phi\) in compliance with \(BR\) and contains at most \(|J| + |BR^*| \leq |J| + |BR|\) local models. \(\square\)

We use this result to prove contextual satisfiability to be NP-complete and to establish a refined upper bound for the amount of time it requires. In order to do so we need the following lemma:

**Lemma 1** Model checking, that is, the problem of determining whether a given chain \(c\) consistently satisfies a set of formulas \(\Phi\) in a multi-context system MS, can be performed deterministically in time:

\[
O\left(\sum_{i : \varphi \in \Phi \cup \Xi(BR)} G_i \times |\varphi|\right)
\]

**Proof.** Model checking can be split into three sub-tasks:

1. Checking whether \(c\) satisfies \(\Phi\);
2. Checking whether \(c\) complies with \(BR\);
3. Checking whether \(c\) is \(J\)-consistent.

218
First consider sub-task 1. Checking whether a particular formula \( i : \varphi \in \Phi \) is satisfied by \( c \) can be done as follows. Let \( \varphi_1, \ldots, \varphi_k \) be an ordering of the subformulas of \( \varphi \), such that \( \varphi_k = \varphi \) and if \( \varphi_i \) is a subformula of \( \varphi_j \), then \( i < j \). Since \( \varphi \) has at most \( |\varphi| \) subformulas, we have \( k < |\varphi| \). By induction on \( k' \) we can label each local model \( m \) in \( c_i \) with either \( \varphi_j \) or \( \neg \varphi_j \), for \( j = 1, \ldots, k' \), depending on whether or not \( m \models \varphi_j \), in time \( O(G_i \times k') \). As a result, checking whether \( c \) satisfies \( \Phi \) can be carried out in time \( O(\sum_{i \in \mathbb{E}} G_i \times |\varphi|) \). Sub-task 2 takes time \( O(\sum_{i \in \mathbb{E}} G_i \times |\mathbb{E}|) \), as in the worst case it involves checking whether all the consequences and premises of every bridge rule in \( \mathbb{E} \) are satisfied or not. Sub-task 3 merely consists in checking whether \( c_j \) is nonempty, for \( j \in J \). This can be done in \( O(|J|) \) timesteps. The result follows directly. \( \square \)

**Theorem 2** Contextual satisfiability is NP-complete. It can be settled in non-deterministic time:

\[ O(|J| + |\mathbb{E}|) \times N \]

**Proof.** We have already observed that contextual satisfiability is NP-hard. To determine satisfiability we first non-deterministically appoint a set \( Cons \) of bridge rule consequences, and a set \( Prem \) of bridge rule premises, such that for every \( br \in \mathbb{E} \), either \( br \)'s consequence is in \( Cons \), or one of \( br \)'s premises is in \( Prem \). Let \( J, I_{Cons}, \) and \( I_{Prem} \) be the set of contexts involved by \( \Phi \), \( Cons \), and \( Prem \), respectively. Furthermore, let \( \Phi_i, Cons_i, \) and \( Prem_i \) be the set of \( i \)-formulas contained by \( \Phi \), \( Cons \), and \( Prem \), respectively. Now we construct a chain \( c \), such that:

- For all \( i \in I_{Prem}, c_i \) contains \( |Prem_i| \) local models;
- For all \( i \in J \setminus I_{Prem}, c_i \) contains exactly one local model;
- For all \( i \notin J \cup I_{Prem}, c_i \) is empty;
- For all \( i \in I \), each \( m \in c_i \) evaluates the propositional atoms not appearing in \( \Phi_i \cup Cons_i \cup Prem_i \) to \( True \).

The only “guessing” involved in constructing \( c \), apart from the choice of \( Cons \) and \( Prem \), are the truth values to which each local model in \( c_i \) should evaluate the propositional atoms in \( P(\Phi_i \cup Cons_i \cup Prem_i) \). Notice that \( c \) contains at most \( |J| + |Prem| \leq |J| + |\mathbb{E}| \) local models, which are distributed over those components \( c_i \) of \( c \) with \( i \in J \cup I_{Prem} \); all the other components of \( c \) are empty. Consider a local model \( m \) contained in \( c_i \) for some \( i \in J \cup I_{Prem} \). The number of atomic propositions \( |P(\Phi_i \cup Cons_i \cup Prem_i)| \) that \( m \) should “explicitly” evaluate is clearly bounded by \( N \). We must appoint at most \( |J| + |\mathbb{E}| \) such explicit valuations (one for each local model in \( c \)), so \( c \) can be constructed in non-deterministic time \( O((|J| + |\mathbb{E}|) \times N) \).

It remains to be checked whether \( c \) is \( J \)-consistent, satisfies \( \Phi \), and complies with \( \mathbb{E} \). By lemma 1 this requires deterministic time \( O((|J| + |\mathbb{E}|) \times N) \). Theorem 1 assures that, if \( \Phi \) is consistently satisfiable in \( MS \), then guessing a chain as described above is bound to result in a suitable one. Thus, consistent satisfiability of \( \Phi \) in \( MS \) can be determined in non-deterministic polynomial time \( O((|J| + |\mathbb{E}|) \times N) \). \( \square \)

### 4 Encoding Into Propositional Satisfiability

As contextual satisfiability is NP-complete, it must be tractably reducible to propositional SAT. In this section we provide such a reduction. In doing so we may loose the particular structure of
our problem, but do lay the groundwork for the implementation of purely Sat-based contextual reasoners, which could benefit from existing, well-advanced techniques.

A propositional representation of contextual satisfiability problems may be acquired by exploiting the understanding we obtained while establishing the bounded model property in the previous section. The key insight there was that any chain $c$, which consistently satisfies a set of formulas $\Phi$, can be reduced to a chain $c^b$ such that:

- For every formula $i : \varphi \in \Phi$, $c^b_i$ contains at least one local model $m$ that satisfies $\varphi$.
- For every bridge rule $br \in BR$ whose consequence is not satisfied by $c$, there is a premise $j : \xi$ of $br$, such that $c^b_j$ contains at least one local model $m$ that satisfies $\neg \xi$.

Notice that to meet these requirements, the number of local models in each component of $c^b$ can be kept down to $|BR|$ (we assume that $|BR| \geq 1$). Also, if a non-empty component of $c^b$ contains less than $|BR|$ local models it can be extended to comprise exactly $|BR|$ models, simply by adding duplicates of already existing models. So we may say that $\Phi$ is satisfiable in MS iff it is satisfied by a chain $c^*$ all of whose components are either empty or contain exactly $|BR|$ local models.

Now, we construct a propositional formula $\psi$, which is satisfiable iff such a chain $c^*$ exists. We express this formula in a language which contains an atomic proposition $p^k_i$ for each $p \in L_i$, and each $k = 1, \ldots, |BR|$. Intuitively, the truth value assigned to $p^k_i$ by a propositional model of $\psi$ corresponds to the truth value assigned to $p$ by the $k^{th}$ local model in $c^*_i$. The language also contains an atomic proposition $e_i$ for each index $i \in I$. Intuitively, $e_i$ corresponds to $c^*_i$ being empty.

Let us write $K = \{1, \ldots, |BR|\}$. For any formula $\varphi$, $i \in I$ and $k \in K$ let $\varphi^k_i$ denote the formula that results from substituting every atomic proposition $p$ in $\varphi$ with $p^k_i$. Furthermore, let us write $\varphi^K_i = \bigwedge_{k \in K} \varphi^k_i$. Now, the translation of a labeled formula reads:

$$(i : \varphi)^* = e_i \lor \varphi^K_i$$

For bridge rules we have:

$$(i_1 : \varphi_1, \ldots, i_n : \varphi_n \rightarrow i : \varphi)^* =$$

$$(i_1 : \varphi_1)^* \land \cdots \land (i_n : \varphi_n)^* \lor (i : \varphi)^*$$

And a $j$-consistency constraint is captured by:

$$(j\text{-cons})^* = \neg e_j$$

**Theorem 3** There is an assignment $V$ to the propositions $\{p^k_i \mid i \in I, k = 1, \ldots, |BR|\} \cup \{e_i \mid i \in I\}$ that satisfies:

$$\psi = \bigwedge_{i : \varphi \in \Phi} (i : \varphi)^* \land \bigwedge_{j \in J} (j\text{-cons})^* \land \bigwedge_{br \in BR} (br)^*$$

if and only if there is a $J$-consistent chain $c^V$ that satisfies $\Phi$ and complies with $BR$.  

220
Proof (⇒) From $V$ construct a chain $c^V$, such that each component $c^V_i$ is empty if $V(e_i) = True$ and contains exactly $|\mathbb{BR}|$ local models otherwise. In the latter case, let the $k^{th}$ local model of $c^V_i$ evaluate each atomic proposition $p \in L_i$ to $True$ iff $V(p^k_i) = True$. It is easy to see that $c^V$ is $J$-consistent and satisfies $\Phi$ in compliance with $\mathbb{BR}$.

(⇐) If there is a $J$-consistent chain $c$ that satisfies $\Phi$ in compliance with $\mathbb{BR}$, there must also be a $J$-consistent chain $c^*$ each of whose components is either empty or contains exactly $|\mathbb{BR}|$ local models, and which still satisfies $\Phi$ in compliance with $\mathbb{BR}$. From $c^*$ we obtain an assignment $V$ as follows. To an atomic proposition $e_i$, $V$ assigns $True$ iff $c^*_i$ is empty. To an atomic proposition $p^k_i$, $V$ assigns $True$ iff the $k^{th}$ local model of $c^*_i$ satisfies $p$, and any truth value iff $c^*_i$ is empty. It is easy to see that $V$ satisfies $\psi$.

5 Conclusion

We established the bounded model property for propositional multi-context systems with finite sets of bridge rules. This result entails a significant augmentation of our insight into the inherent complexity of contextual reasoning. We proved contextual satisfiability to be NP-complete and established a refined upper bound for the amount of time it requires. Also, we provided a tractable encoding of contextual satisfiability problems into purely propositional ones. In doing so, we laid the groundwork for SAT-based implementations of contextual reasoning systems. Future work will indeed encompass experimenting with both native and SAT-based contextual reasoners. Also, we are interested to what extent our results may be generalized so as to apply to multi-context systems with schematic bridge rules as well.

Acknowledgements

This work has ensued from a M.Sc. graduation project carried out at IRST in Trento, Italy. I am very grateful to Luciano Serafini for having inspired and guided me throughout the project. Also, I am indebted to Johan van Benthem, Roberto Sebastiani, and Alessandro Cimatti for their comments on earlier drafts and presentations of this and preceding work.
References


Time and Focus: The Case of German *gerade*

Gerhard Schaden
Univensité Paris 8, UMR 7023
gerhard.schaden@yahoo.fr

Abstract.
The article deals with the temporal readings of German focus particle *gerade*. First, it will be shown that there are two temporal readings, "immediate-anteriority" and "progressive" respectively, depending on the aspectual properties of the sentence. A scalar treatment of aspect in its interaction with *gerade* will be outlined. Second, *gerade* will be compared with *schon* (‘already’). The question of the presuppositional structure will be discussed in order to account for some effects of *gerade*.

1 The Problem

As some other much better investigated focus particles – for instance German *schon* ‘already’ or *noch* ‘still’ – the German focus particle *gerade* (~ ‘straight’) displays temporal uses. In one of these temporal uses, *gerade* has been identified by Dahl (1985) as the German expression of the progressive, which seems very plausible considering examples like (1):

(1) a. Otto isst gerade Schokolade. [=Präsen$^1$]
   Otto eats GERADE chocolate.
   ‘Otto is eating chocolate (in this very moment).’

   b. Als das Feuer ausbrach setzte Otto gerade seinen Helm auf.
   when the fire outbroke put Otto GERADE his helmet on.
   [= Präteritum]
   ‘When the fire started, Otto was putting on his helmet.’

$^1$This article has greatly benefitted from the suggestions and comments of and the discussions with Nisrine Al-Zahra, Nora Boneh, Patricia Cabredo Hofherr, Ekkehard König, Brenda Laca, Laurent Roussarie, Benjamin Spector, Alice ter Meulen and three anonymous reviewers of ESSLLI. I’d also like to thank Jules Gouguet for correcting my English. For any error or omission, of course, I take entire responsibility.

For the sake of better comprehension, I will indicate in square brackets the tense of the verb *gerade* applies to.
Were *gerade* omitted, (1a) might assert not an actual event of Otto eating chocolate at the time of utterance (TU), but an individual-level disposition or habit (he isn’t disgusted by chocolate; he eats it). In (1b), if *gerade* were omitted, the preferred reading would be that Otto put on his helmet after the fire started and *because* the fire started (say, because he is a firefighter). However, (1b) without *gerade* could also have the (pragmatically strongly dispreferred) ‘progressive’ reading exemplified in the translation above. This is to say that the readings triggered by *gerade* are also available for the sentences not containing it: ‘progressive’ *gerade* merely disambiguates a sentence.

However, there is a second temporal use of *gerade* that rules out any straightforward identification with an English-style progressive: *gerade* is also commonly used to indicate the immediate anteriority of an event to some contextually given point of reference R:

(2) a. Anna hat *gerade* einen Brief geschrieben. [=Perfekt]
   Anna has *GERADE* a letter written.
   ‘Anna has just written a letter.’

   b. Als Anna *gerade* alle Beweise beseitigt hatte, stürmte die Polizei ihre
   when Anna *GERADE* all proofs destroyed had, assaulted the police her
   Wohnung. [Plusquamperfekt]
   flat.
   ‘When Anna had just destroyed all proofs, the police took her flat by assault.’

In (2), *gerade* forces to situate the event described immediately before R; in (2)a), the writing of the letter might have taken place at any time in the past, and in (2)b), Anna’s destroying of the proofs might have similarly taken place at any time preceding the assault of the police in the absence of *gerade*.

The distributional pattern of *gerade* to some extent recalls the English focus particle *just*, which appeared in the glosses of (2), and which situates the event in the same way immediately before R (cf. (3)a)). *Just* can also occur in a sentence with the progressive (cf. (3)b)).

(3) a. Anna? – I’ve just seen her in the hall.
   b. I’m just cleaning my room.

   (3)b) has a modal meaning; as opposed to the same sentence in German, it does more than merely state the fact that at TU, the speaker is engaged in the activity of cleaning his/her room.\(^2\) Contrary to what *gerade* does, *just* doesn’t take away a possible interpretation, but adds a modal value instead.

   Nevertheless, (1)-(2) do not show what is just an idiosyncratic behaviour of German *gerade*, as Russian *kak ruz* (‘wh- time’) seems to behave in a quite similar way.²

---

²König (1991, pp. 121ff.) states furthermore that *just* belongs to the subclass of exclusive focus particles – that is, particles that discard any alternative focus value different from the one asserted – whereas *gerade* doesn’t belong to this class. This is correct, I think, but it is quite tricky to show it by opposing them directly, as it seems that *just* and *gerade* don’t apply (as non-temporal focus particles) to the same type of arguments. Note, however, that you can generally paraphrase *just* in its non-temporal uses by ‘nothing but’, which is impossible with *gerade*.

³Thanks to Ora Matushansky for pointing that out to me and for providing me with example (4).
2 The Role of Aspect

We have seen that with *gerade* the Präsens (1a) and the Präteritum (1b) trigger a progressive reading, whereas the Perfekt (2a) and the Plusquamperfekt (2b) trigger an immediate-antiority reading.

To explain this, it will be enlightening to adopt a neo-Reichenbachian framework as Klein (1995).\(^4\) Therein, temporality is split in two domains: tense, which is the relation between TU and a topic time (TT; the time for which an assertion holds),- and aspect, which is the relation between TT and the time of the situation (T-Sit). I will assume with Smith (1991) that there is a sort of underspecified or ‘neutral’ aspect, whose interpretation is much less constrained than that of perfective or imperfective aspect.

For what will follow, I will assume that the temporal system of Standard German can be captured by the following schema:\(^5\)

<table>
<thead>
<tr>
<th></th>
<th>Neutral</th>
<th>Retrospective</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TT ⊆ T-Sit</td>
<td>TT &gt; T-Sit</td>
</tr>
</tbody>
</table>

Present: TU ⊆ TT | Präsens | Perfekt
Past: TU > TT    | Präteritum | Plusquamperfekt
Future: TU < TT   | Futur | Futur Perfekt

We have already seen in (1)–(2) which effects *gerade* produces with the present and the past

---

\(^4\) Readers familiar with neo-Reichenbachian analyses might be puzzled by the fact that I use at the same time the neo-Reichenbachian TT and its historic ‘ancestor’, the Reichenbachian R. In Reichenbach (1966), R performs two basic tasks: It is responsible for the temporal coherence of a discourse, and contributes to the temporal location of E (i.e., T-Sit). More recent analysis prefer to split those two tasks: Neo-Reichenbachian TT took over only the second one; and in Klein (1995 or 2000), there is nothing comparable to the first task of R. The Standard DRT’s (cf. Kamp and Reyle (1993)) conception of R on the contrary is the successor precisely of the first task of Reichenbach’s R, whereas the aspeclual component is taken care of by other mechanisms. In this article, I will use ‘TT’ in the neo-Reichenbachian tradition, which means that it is (a) an interval and (b) given by the tensed verb (at least in German). ‘R’ is conceived of as it is in DRT: It represents (a) a point in time and (b) may be contextually given. As we will see later (cf. (6a) and (7)), we need such a contextually given point in time to describe the behaviour of *gerade* in an adequate way.

\(^5\) This schema is an oversimplification in two respects. First, at least in the Southern dialects, where the Präteritum has been eliminated from spoken language by the Perfekt, the Perfekt is ambiguous between the reading indicated in (5), a retrospective present tense, and a neutral past tense, that is, it also expresses the value represented by the Präteritum in (5). Second, the representation of neutral viewpoint aspect by ⊆ is merely meant to indicate underspecification; for a different formalization, cf. Pancheva (2003).
tenses. Let’s now examine what happens with the future tenses:

(6) a. Anna wird gerade Schokolade essen. [= Futur]
    Anna becomes GERÄDE chocolate eat.
    ‘Anna is probably eating chocolate.’

b. Wenn wir (morgen) abfliegen, wird Otto gerade im Büro sitzen.
    when we (tomorrow) off-fly, becomes Otto GERÄDE in-the office sit.
    [= Futur]
    ‘(Tomorrow) When we fly away, Otto will be sitting in his office.’

c. Wenn wir (morgen) abfliegen, wird Otto gerade gefrühstückt haben.
    when we (tomorrow) off-fly, becomes Otto GERÄDE breakfast-ed have.
    [= Futur Perfekt]
    ‘When we fly away (tomorrow), Otto will just have had his breakfast.’

(6)a) shows that – if there is no contextually given R – the Futur with gerade is interpreted as an epistemic modal present, probably because R is equated to TU by default. If there is a contextually given R, the effects are exactly the same as in the past tenses. From this, we may conclude that gerade is interpreted as a progressive when applied to an aspectually neutral tense, and as an expression of immediate anteriority when applied to a retrospective tense.6

There are, however, some differences between past and future tenses: whereas (6)a) gives an epistemic modal present, its past tense equivalent is simply odd out of context, as it lacks an accessible R to relate to:

(7) ??Anna aß gerade Schokolade. [= Präteritum]
    Anna ate GERÄDE chocolate.
    ‘Anna was eating chocolate.’

Note that this is the same sort of strangeness one associates with ‘out of the blue’ imparfait sentences in Romance or past progressive sentences in English which lack an ‘anchoring’ temporal adverbial or temporal clause.7

Let’s take one step back. We have seen that gerade eliminates all but one of the possible

---

6 Two things do not fit neatly into this picture: one can get an immediate anteriority reading with the Präterita of sein (to be), and also with haben (to have) at X place (eg., in my hand). This is probably correlated to the fact that for some reason, the Präteritum forms of those verbs are strongly preferred to their corresponding Perfekt forms, even in the Southern dialects (where these verbs constitute the last remnants of the Präteritum).

7 One of the reviewers suggested that gerade simply might ensure that TT was included in T-Sit in case of neutral aspect, and that in case of retrospective aspect, gerade operates on the “result state” of the event encoded, such that TT is included in the result state. I understood this as a proposal to analyze gerade as an operator on viewpoint aspect, which would be concise and elegant. However, I don’t think that we should adopt such an approach.

If gerade were an aspecual (viewpoint) operator, I don’t see how one could deal with sentences like the following, without any inflected verb (and therefore possibly without any functional categories of time it could operate on):

(i) Gerade noch in seiner Werkstatt, jetzt auf unserer Showbühne!
    GERÄDE still in his workshop, now on our show-stage!
    ‘One moment ago still in his workshop, and now on our stage!’

226
readings with both retrospective and neutral aspect. On a purely descriptive level, we can say that *gerade* in its immediate anteriority reading assures that we get something like ‘contiguity’ between *TT* and *T-Sit* (cf. (8)b) when combined with retrospective aspect, instead of having the ‘normal’ retrospective semantics (cf. (8)a)):

(8) \[ \lambda P \exists e [ P(e) \land \tau(e) < TT ] \]
\[ P(e) \land \tau(e) < TT \land \exists I, I', I'' [ I' \subset TT \land I'' \subset I \land I'' \subset I \land I' \oplus I'' = I ] \]

Based on Smith (1991), we can formalize in the same way neutral (which I take to be under-specified, cf. (9)a)) and progressive viewpoint aspect (cf. (9)b)):

(9) \[ \lambda P \exists e, t_i, t_j [ P(e) \land t_i = I(e) \land t_j = F(e) \land \forall t [ t \in TT \rightarrow t \geq t_i \land t \leq t_j ] ] \]
\[ P(e) \land t_i = I(e) \land t_j = F(e) \land \forall t [ t \in TT \rightarrow t > t_i \land t < t_j ] \]

What is particular about those formalizations\(^{10}\) is that (8)b entails, but is not entailed by, (8)a). Thus, (8)b asymmetrically entails (8)a). The same relation holds between (9)b) and (9)a). We can put it the other way round: (9)b) and (8)b) are more specific forms of (9)a) and (8)a).

Now, such relations of asymmetrical entailment are used in neo-Grecian pragmatics to establish what are called “Horn-scales”, in order to explain the phenomenon of scalar implicatures.\(^{11}\) Take the positive general quantifiers, for instance. They form a scale:

(10) some < many < most < all.

All asymmetrically entails most and the other quantifiers. So, all is more informative than most, many, or some. The (strongly abbreviated) pragmatic reasoning is the following: if I hear somebody say some *N*, when it would be pertinent for me to know all *N*, I conclude that all *N* doesn’t hold.

Although the situation is not exactly the same for *gerade* and the aspects, there are good reasons why one might want to adopt such a scale based treatment of this focus particle. When focusing on adjectives or DPs, *gerade* quite clearly displays scalar uses (cf. (11)). So, one needs a way to treat scales *gerade* applies to (or creates?) anyhow.

(11) Otto ist nicht gerade verrückt, aber …\(^{12}\)
Otto is NEG GERADE mad, but …
Otto is not exactly mad, but …

\(^{10}\)Showmaster Rudi Carell announced in this way the apparition on stage of candidates that were introduced first at their working place before they performed live on stage.

\(^{11}\)Furthermore, one doesn’t really see how this analysis of *gerade* as an aspectual operator could be linked with its being a focus particle, that is: with the fact that such particles order the alternative focus values in a certain manner, as Krifka (2000) asserts.

\(^{8}\)\(\tau(e)\) is the temporal trace of the event *e*. Put this way, (8)b) only works if we assume discrete time. The second part of the formula makes sure that \(\tau(e)\) and TT are contiguous.

\(^{9}\)\(t_i\) is the beginning of the event, and \(t_j\) is the end of the event.

\(^{12}\)It is true that if one takes the formalization of Pancheva (2003), these implications do not hold. But it can be shown for independent reasons that Pancheva’s formula of the neutral aspect is not adequate, at least for German.
Thus, we may say that *gerade* eliminates in its temporal uses the less informative reading associated to an aspect. This move gets us the (approximatively) correct truth conditions of *gerade* with both of the aspects of German. However, (8) will have to be integrated in a more general picture of (at least German) Perfect semantics, as the two readings in (8) are not able to capture the three or four Perfect values generally assumed in the literature (cf. Alexiadou, Rathert, and von Stechow (2003)).

3 *Gerade vs. Schon*

Previous work by Löhner (1989, 1999) and Krifka (2000) on *schon* (‘already’) and its relatives has been able to clearly establish four major facts about those focus particles: first of all, like quantifiers, they are organized in duality groups: \(\neg \textit{schon} \varphi (\neg \textit{already} \varphi)\) being equivalent to *noch* \(\varphi\) (still \(\varphi\)). Second, they have to be evaluated on (relevant) intervals, not on points in time. Third, they trigger presuppositions, and fourth, they display a monotonic mapping between times and focus values.\(^{13}\)

I will not take up the issue of the possible duals of *gerade*, but I will address the other three points, which are all more or less intertwined.

Let us consider the following sentences:

(12) a. Anna hat noch [drei]_{FOC} Bonbons in ihrer Tasche.
   ‘Anna has still three candies in her pocket.’

   b. Anna hat schon [drei]_{FOC} Bonbons in ihrer Tasche.
   ‘Anna has already three candies in her pocket.’

   c. Anna hat gerade [drei]_{FOC} Bonbons in ihrer Tasche\(^{14}\).
   ‘Anna has GERADE three candies in her pocket.’

Following the analysis of Löhner and Krifka with respect to *schon* and *noch*, one can say all sentences in (12) assert the same: at TU, there are three candies in Anna’s pocket. They differ, however, in their presuppositions: For (12)a) to be true, there must have been at least one moment in the relevant interval where she had more than three candies. For (12)b) to be true, she must have had less than three candies. If we set the relevant interval for evaluation from \(t_0\) to \(t_4\), TU being \(t_4\), (12)a) would need a development like (13)a), and (12)b) a development like in (13)b):

\(^{12}\)Those scalar interpretations with adjectives appear mostly under negation. It seems to me that this is due to the fact that, without negation, one would rather interpret *gerade* in (11) in a temporal way, even if the result is very odd.

\(^{13}\)That is, the focus value must be either steadily increasing or steadily decreasing with time. Or, defined in a formal way, taken from Krifka (2000, p. 5): \(\leq_T\) and \(\leq_A\) are aligned with respect to \(f : [T \to A]\) iff \(\forall t, t' \in T \forall X, X' \in A [f(t) = X \land f(t') = X' \rightarrow [X <_A X' \rightarrow t < T t']]\).

\(^{14}\)In this sentence, as opposed to the other two, the focus with which the focus particle associates might also be candies. Nevertheless, in the following analysis, I will only consider the focus indicated by the brackets in (12)c).
(13)  
  a. \( \langle t_0, 8 \rangle \langle t_1, 6 \rangle \langle t_2, 5 \rangle \langle t_3, 4 \rangle \langle t_4, 3 \rangle \)  
  b. \( \langle t_0, 0 \rangle \langle t_1, 1 \rangle \langle t_2, 2 \rangle \langle t_3, 2 \rangle \langle t_4, 3 \rangle \)  
  c. \( \langle t_0, 2 \rangle \langle t_1, 6 \rangle \langle t_2, 0 \rangle \langle t_3, 4 \rangle \langle t_4, 3 \rangle \)  

As one can easily check, (13)a-b display a monotonic mapping between times and the number of candies, whereas (13)c doesn’t. It is impossible to assert (12)a or b truthfully in a context like (13)c, whereas (12)c can be truthfully uttered under (13)a-c. This could be taken to mean that the truth conditions of a sentence with *gerade* depend only on the state of the world at a single moment.

But this is clearly not the case. Sentences like (14) are odd, although they describe truthfully the current reality (I assume):

(14)  
  a. ??/‡Löwen haben *gerade* vier Pfoten.  
      lions have GERADE four paws.  
      ‘Lions have got four paws (in this very moment).’
  b. *3 ist gerade* eine Primzahl.  
      3 is GERADE a prime-number.  
      ‘3 is a prime number (in this very moment).’

*Gerade* is extremely bad with generic sentences or eternal truths. Some people (for instance, those believing in an extremely capricious omnipotent god) could be willing to accept (14)a, but I do not see how one could save (14)b.

Furthermore, *gerade* is very odd with non-reversible (resultant) states:

(15)  
  a. ??/‡Der Apfel ist *gerade* gegessen.  
      The apple is GERADE eaten.  
      ‘The apple is eaten (for the moment).’
  b. ??/‡Otto ist gerade tot.  
      Otto is GERADE dead.  
      ‘Otto is being dead (for the moment).’

If the apple is not self-regenerating, and if Otto isn’t someone who rises regularly from the death, sentences (15) are not acceptable.

So, we may want to add a presupposition to sentences with *gerade*, too, one that expresses that the circumstances described must be able to change.\(^\text{15}\) According to Löbner (1989), for *already \( \varphi \)* (or *schon \( \varphi \)*) to be true at \( t_c \), there must be an interval \( I \) preceding \( t_c \) where \( \neg \varphi \) is true. Consider (16):

(16)  
  Anne is already in New York (# and she never has left this city).\(^\text{16}\)

\(^{15}\)It is very difficult to test if *gerade* triggers a presupposition in the standard, straightforward way. As there are multiple interferences of scope phenomena with negation or counterfactuals, any answer to what is happening in these constructions presupposes a working theory of *gerade*, precisely what we are trying to develop.

\(^{16}\)A sentence like this would only be possible in the case of an inherited perspective. For a discussion, cf. Löbner (1989, p. 183).
The presupposition of noch (‘still’) is – again according to Löbner – that for noch \( \varphi \) to be true at \( t_c \), there must be an interval following \( t_c \) where \( \varphi \) doesn’t hold. Taking this description as a starting point, and in order to account for the similarity of gerade with the progressive, I propose the following formula as a description for the presupposition triggered by gerade:

\[
\text{(17) } \text{GERADE}(t,w,\varphi) \\
\text{presupposition: } \exists t_1, t_2 [t_1 < t \land \neg \varphi(t_1,w) \land \exists w' \in W_{\text{Inr}}(t,w) | t < t_2 \land \neg \varphi(t_2, w')]] \]

which means that gerade \( \varphi \) at \( t \) in \( w \) presupposes that there is a moment \( t_1 \) preceding \( t \) such that \( \neg \varphi \) is the case at \( t_1 \). The second part states that there is at least one world \( w' \) which is a member of the set of inertia worlds of \( w \) up to \( t \) (that is, \( w \) and \( w' \) are identical up to \( t \)), such that \( \neg \varphi \) is the case at a moment \( t_2 \) posterior to at least \( t \) in this world \( w' \).

(17) seems nice because we can derive it for free from two things we can safely assume: first of all, the fact that gerade is a non-exclusive focus particle, that is, it allows for alternative focus values. Secondly, that time is branching into the future. Given these two assumptions, and if we take gerade to associate points in time \( t \neq t_c \) with alternative focus values \( a \) to form ordered sets of pairs \( \langle t_0, a_0 \rangle \ldots \langle t_n, a_n \rangle \), (17) would be what we get. However, the only use of (17) is to get us rid of undesirable generic readings, as its adequation for the immediate-anteriority reading is trivial.

If we consider (17) and compare them with the presuppositions given by Löbner (1989) for schon and noch, we see that (17) is roughly the combination of the presuppositions of those two focus particles. One would expect therefore that, whenever either noch or schon fail to apply felicitously to a sentence because of presupposition failure, gerade isn’t acceptable either. This is indeed the case:

\[
\text{(18)} \\
a. \text{ Otto ist schon/*noch/*gerade alt.} \\
\text{Otto is already/*still/GERADE old.} \\
b. \text{ Anna ist *scon/noch/*gerade jung.} \\
\text{Anna is *already/still/GERADE young.} \\
c. \text{ Otto ist schon/noch/gerade im Urlaub.} \\
\text{Otto is already/still/GERADE on holiday.}
\]

As our knowledge of the world tells us that ‘being old’ is not followed by a phase of ‘not being old’, noch cannot apply to (18)\( \text{a} \), and neither can gerade. In (18)\( \text{b} \), schon is out because there is no phase of ‘not being young’ preceding ‘being young’, and gerade isn’t felicitious either. In (18)\( \text{c} \) however, schon and noch are fine, and so is gerade.

A further point is that (17) doesn’t exclude \( \varphi \) being a habit. In fact, gerade does not impose an interpretation as a single actual event:

\[
\text{(19)} \\
\text{Anna raucht gerade Smart.} \\
\text{Anna smokes GERADE Smart.} \\
\text{‘Anna smokes Smart (for the moment).’}
\]

If Anna changes the brand she smokes from time to time, and for TU the brand she prefers
is Smart, (19) is perfectly adequate, though she may not actually have a burning cigarette in her mouth in the moment (19) is uttered.

There are however examples that do not seem to be compatible with (17), as one of the reviewers pointed out:

(20) Jedes Kind, das gerade in diesem Krankenhaus war, wurde mit dem Virus
Every child, who GERADE in this hospital was, became with the virus
infiziert.
infected.
‘Every child that happened to be in this hospital was infected with the virus.’

(20) is in fact compatible with there being one or more children having been born in the hospital and having been infected with the virus. This, however, is excluded by (17). The reviewer suggests therefore to modulate the presupposition on both sides, or to reduce the presupposition to a conversational implicate.

Let’s take a closer look at (20): if it lacked gerade, one would obtain at least one reading that is excluded by the focus particle: Being a child and having spent a period of time in that hospital would be the necessary and sufficient conditions for getting infected with the virus. (20) as it is, however, needs a R in the preceding discourse for gerade to relate to. Something like (21)a will not do, as it does not provide any event; one would need something like (21)b to obtain a coherent discourse.

(21) a. J.F Scurf Hospital was notorious for its bad hygiene conditions.
b. Last August, there was a power failure in J.F. Scurf Hospital.

This may lead us to reconsider the presupposition attributed to gerade on the base of sentences like (15). Indeed, English sentences (22) are odd in a quite similar way to those in (15), although one would not be willing to say that today or in this very moment trigger a presupposition. One would rather go for a pragmatic explanation, saying that there is a conversational implicate.

(22) a. ??King Arthur is dead today/in this very moment.
b. ??3 is a prime number today/in this very moment.

Nevertheless, the effects on such non-reversible states don’t seem to be exactly the same as the ones triggered by gerade, and abandoning the idea of a presupposition for a focus particle at this early stage would be a step with quite far-reaching consequences. It seems to be a safer option to modulate the presupposition on both sides of the assertion. Therefore the presupposition has to be restated as (23):

(23) \[ \text{GERADE}(t,w,\varphi) \]

premise: \( \exists t_1, t_2, w' \in W_{tr}(t, w)[t_1 < t < t_2 \land \neg \varphi(t_1, w') \land \neg \varphi(t_2, w') \] where
\[ \forall w'[w' \in W_{tr}(t, w) \rightarrow \forall x[P(x)]^{w',t} = [P(x)]^{w,t}] \]

This weaker version of (17) doesn’t allow for any inference on \( \neg \varphi \) in \( w \) at \( t_1 \) and is thus compatible with (20), without allowing at the same time for i-level predicates and unchangeable

---

\[^{18}\text{This defines inertia worlds before and after } t, \text{ an inertia world of } w \text{ at } t \text{ being a world } w' \text{ that is identical to } w \text{ in any respect at } t.\]
resultant states to occur. However, one cannot derive it as easily as (17) from general and generally accepted properties of time and focus.

4 Conclusion and Perspectives

In this article it has been shown that the distinction between progressive or immediate-antiority readings depends crucially on the syntactic aspect of the verb the focus particle applies to. It has been argued that gerade triggers a presupposition, and a scalar treatment of aspect has been sketched which might prove applicable to other, non-temporal uses of gerade.

However, we are still far away from having an unified analysis of this focus particle, so that we see in the end, as Brecht:

[...] der Vorhang zu, und alle Fragen offen.\textsuperscript{19}

\textsuperscript{19}[...] the curtain closed, and all questions open.
References

Mouton de Gruyter.
University of Texas at Austin.
Löbner, S. (1999). Why German schon and noch are still duals: A reply to Van Der Auwera.
Linguistics and Philosophy 22, 45–107.
Signaling games and non-literal meaning

Merlijn Sevenster
Institute for Logic Language and Computation, Universiteit van Amsterdam
sevenstr@science.uva.nl

Abstract.
Sally (Sally 2003) proposed a game-theoretical framework to study the use and conventionalization of non-literal language use. We have objections against this model from a conceptual and a methodological point of view. Instead, we introduce a 'Super Conventional signaling game' to account for non-literal use of language. Such a game is an elaboration of Lewis' (Lewis 1969) 'signaling game', that preserves and makes use of a strict separation between literal and non-literal interpretation of utterances. We argue that, following Sally now, real people play Super Conventional signaling games as they use non-literal speech.

1 Introduction

If one says “George W. Bush is a pig”, one does not mean to say that George W. Bush is a pink animal with a tail, although this is implied by the literal meaning of the sentence. Instead, the speaker of this sentence expects the hearer to give the sentence its non-literal meaning. Sally (Sally 2003) recognizes the use of messages that are intended to have a non-literal meaning as ‘risky speech’, as the hearer might not recognize the speaker’s intention and respond “Oh, that comes as a surprise to me. I thought he was the president of the US”. Although more risky, experimental results (for references see (Sally 2003)) show that non-literal speech has stronger cognitive and social impact than literal speech.

Lewis (Lewis 1969) defined the notion of a signaling game in order to explain the conventionalization of meaning of language without assuming any pre-existing relation between messages and meaning. Playing a signaling game optimally (in a sense that will be made explicit below) results in a unique meaning for every message. Lewis used signaling games as a philosophical reply to Quine’s objection against conventionalism, that any linguistic convention could only be formed using another linguistic convention. The last couple of years however, scholars use signaling games to come to an understanding of pragmatisical issues; e.g. Horn’s division of pragmatic labor (van Rooij 2004).

We gratefully acknowledge Robert van Rooij for guiding remarks, his continuing enthusiasm and for proof-reading this paper several times. We thank Peter van Emde Boas for correcting big mistakes. We thank the anonymous referees for numerous hints to improve the paper.

Proceedings of the Ninth ESSLLI Student Session
Laura Alonso i Alemany and Paul Égré (editors)
Chapter 22, Copyright © 2004, Merlijn Sevenster
A signaling game is a cooperative game amongst two players: a sender S and a receiver R, whose shared goal it is to let R perform an action that is appropriate with respect to the state S and R are in. This state is only observed by S though, and S communicates it by means of a meaningless message. Think of a couple of agents, the one looking out for hungry predators and the other searching for food on the ground. Both players are in the same state — there is a predator approaching or there is not —, but only S knows which state they are in. S signals the state by means of a message that has no pre-defined meaning — say, ‘buh’ or ‘bah’. In turn, R hears the message and is free to perform any action of his liking, but every state has a most appropriate action. E.g., if there is a hungry predator approaching, R should flee, otherwise R should keep on searching. The best thing for S to do is to say ‘buh’ if there is a hunting predator and ‘bah’ otherwise; and for R to flee when he hears ‘buh’ and not to in case he hears ‘bah’. (It is equally good of course to do the same, but with ‘buh’ and ‘bah’ interchanged.) If S and R play the game in such an optimal way, the meanings of the messages ‘buh’ and ‘bah’ are created in the play of the game. Game-theoretically, playing in a way that makes the messages meaningful amounts to coordinating on a ‘separating Nash equilibrium’ (to be defined below).

In this paper, we are concerned with conventionalization of non-literal meaning. As to non-literal meaning, we distinguish, following Searle (Searle 1979), between what a sentence means and what a speaker means by uttering this sentence. The non-literal meaning of a sentence is its ‘speaker’s utterance meaning’. In the hearer’s process of attaching the speaker’s utterance meaning — according to Searle, and we follow him —, the hearer first has to recognize the defectiveness of the utterance’s literal meaning. For this recognition to be possible, it thus must be the case that the literal meaning is common knowledge. This insight we mark (♠). As to the conventionalization of non-literal meaning, we propose a signaling game à la Lewis that takes the conventional meaning as parameter to model conventionalization of non-literal meaning; a proposal that is conceptually different from the one proposed by Sally (Sally 2003). In his general signaling games’ namely, Sally does not take the state of the players as primitive, but rather the speech act assigned to S. We feel that this does not do right to the notion of speech act, as it neglects the performative side of playing signaling games. We will come back to Sally’s model, after we have defined Lewisean signaling games and discuss our objections with respect to this model.

Although we conceptually disagree with Sally’s signaling game, we are more than happy to adopt his insights on how people play signaling games in experimental settings and how these findings can be used to explain under what conditions people are more likely to use non-literal speech.

Sally namely, taking for granted that “people play the language game in a way that is consistent with their play in all games” ((Sally 2003), pg. 1232), introduces empirically justified rules of thumb that describe how people play coordination games. The central notion in these rules appears to be ‘sympathy’ and ‘common ground’, as the rules roughly state that players are more likely to play risky if they are more sympathetic to each other.

The element of risk is introduced by Sally’s pay-off functions that reward successful non-literal communication higher than successful literal communication; and that — in case of unsuccessful communication — punish the player (possibly both) deviating from the convention more severely than if he would have stuck to the convention. For these pay-off functions empirical justification

---

\(^2\)We are afraid that this is closest we will get to defining the linguistic notion of ‘non-literal meaning’. Eventually, we will give a formal definition.
is given, but they are not defined technically. Sally shows that playing his game riskily amounts to using messages non-literally. As a consequence it follows that the more sympathetic people are, the more likely it is for them to communicate non-literally: A finding that itself matches our intuition and empirical reports.

In our approach, we take the notion of literal meaning for granted and use it in a higher-level signaling game to account for conventional, non-literal meaning. We call this game a ‘Super Conventional signaling game’ (from now on SC signaling game), being a game that has a convention of literal meaning parameterized. We will prove that also in our Super Conventional signaling games playing (more) risky resembles communicating (more) non-literally.

In Section 2 we formally define signaling games in strategic normal form and introduce the game theoretical solution concepts of pay-off dominant and risk dominant Nash equilibria. The latter will be used to formalize the notion of meaning, the former to formalize the notion of playing risky. In Section 3 we will introduce Sally’s model in greater length, point out our objections, and state Sally’s rules of thumb as to how game players actually play coordination games and see how these are used to explain under what conditions people use non-literal speech. In Section 4 we will define the notion of a Super Conventional signaling game. We will prove that this game has multiple pay-off dominant and one risk dominant Nash equilibrium, using parameter values inspired by Sally’s findings. We will conclude by showing that also in Super Conventional signaling players play more risky, if they are more sympathetic. Section 5 fulfills its role as a conclusion.

2 Signaling games

Let $T$ be the set of states, $M$ be the set of messages and $A$ be the set of actions such that $|T| = |A| = |M|$. Let $f : T \rightarrow A$ be the bijective function, that adds the appropriate action $f(t) \in A$ to every type $t \in T$. Then $S$ ($R$) plays the signaling game following strategy $s$ ($r$), that is a function from $T \rightarrow M$ ($M \rightarrow A$). In cheap talk signaling games, successful communication of state $t$ (thus in case $f(t) = r(s(t))$) is rewarded with 1, whereas unsuccessful communication (thus in case $f(t) \neq r(s(t))$) is rewarded with 0, independent of the state $t$ and the message $s(t)$:

$$u_S(f, t, s, r) = u_R(f, t, s, r) = \begin{cases} 1, & \text{if } f(t) = r(s(t)); \\ 0, & \text{if } f(t) \neq r(s(t)). \end{cases} \quad (22.1)$$

We assume that Nature picks the state according to some probability distribution $P$ over $T$.

The utility function for $S$ is the expected utility relative to the probability distribution $P$ over $T$:

$$U_S(s, r) = \sum_{t \in T} P(t) \cdot u_S(f, t, s, r) \quad (22.2)$$

and the utility function for $R$ is the expected utility provided that $S$ uses $s$, so that $S$ is of a type from $T_s(t) = \{ t' \in T \mid s(t) = s(t') \}$:

$$U_R(s, r) = \sum_{t \in T} P(t) \cdot \sum_{t' \in T_{s(t)}} P(t') \cdot u_R(f, t', s, r). \quad (22.3)$$

$^3$We assume that $P(t) > 0$, for every $t \in T$. 

236
Finally, we define a *cheap talk signaling game* \( G \) as a tuple \( \langle \{ S, R \}, \{ S, R \}, \{ U_5, U_R \} \rangle \), where \( S \) is the set of strategies \( s : T \to M \) for player \( S \); \( R \) is the set of strategies \( r : M \to A \) for player \( R \); and \( \{ U_5, U_R \} \) contains both players’ utility functions as defined in (22.2) and (22.3). \( G \) is called 'cheap talk' because \( u_S \) and \( u_R \), simultaneously defined in (22.1) are called this way. Our SC signaling game will use other functions.

A pair of strategies \( \langle s^*, r^* \rangle \) forms a *Nash equilibrium*\(^4\) in \( G \) iff neither \( S \) nor \( R \) gains from unilateral deviation:

\[
U_5(s^*, r^*) \geq U_5(s, r^*) \quad \text{and} \quad U_R(s^*, r^*) \geq U_R(s^*, r),
\]

for all \( s \in S \) and \( r \in R \). We will omit to mention the game \( G \), if no confusion arises. We call a Nash equilibrium \( \langle s^*, r^* \rangle \) *pay-off dominant* in \( G \) iff \( U_5(s^*, r^*) \geq U_5(s, r) \) and \( U_R(s^*, r^*) \geq U_R(s, r) \), for all Nash equilibria \( \langle s, r \rangle \) in \( G \). Conversely, we say that \( \langle s, r \rangle \) is pay-off dominated by \( \langle s^*, r^* \rangle \).

To strengthen our intuition, consider the following utility matrix:

<table>
<thead>
<tr>
<th></th>
<th>( r_1 )</th>
<th>( r_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td>2, 2</td>
<td>0, 2</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>2, 0</td>
<td>3, 3</td>
</tr>
</tbody>
</table>

Strategy pairs \( \langle s_1, r_1 \rangle \) and \( \langle s_2, r_2 \rangle \) are Nash equilibria; and \( \langle s_2, r_2 \rangle \) is the payoff dominant equilibrium. Although \( \langle s_1, r_1 \rangle \) is pay-off dominated, \( S \) and \( R \) have their incentive to choose \( s_1 \) and \( r_1 \), respectively: both strategies guarantee a utility of 2 and are therefore considered less risky. Formally, we call a Nash equilibrium \( \langle s^*, r^* \rangle \) *risk dominant* in \( G \) in the sense of (Harsanyi and Selten 1988) iff

\[
(U_5(s^*, r^*) - U_5(s, r^*)(U_R(s^*, r^*) - U_R(s^*, r)) \geq (U_5(s, r) - U_5(s^*, r))(U_R(s, r) - U_R(s, r)),
\]

for all Nash equilibria \( \langle s, r \rangle \) in \( G \). As the reader can check, \( \langle s_1, r_1 \rangle \) is risk dominant.

For instance, consider a signaling game where \( T = \{ t_1, t_2 \} \), \( M = \{ m_1, m_2 \} \) and \( A = \{ a_1, a_2 \} \) and \( x = P(t_1) > P(t_2) = y \). Then \( S \) and \( R \) both have four different strategies, hence there are 16 strategy-pairs of which we have the players' utilities \( U_5 \) and \( U_R \) below.

This cheap talk signaling game has four Nash equilibria, being \( \langle s_1, r_1 \rangle \), \( \langle s_2, r_2 \rangle \), \( \langle s_3, r_3 \rangle \) and \( \langle s_4, r_1 \rangle \). As the reader can check, only in \( \langle s_2, r_2 \rangle \) and \( \langle s_3, r_3 \rangle \) communication takes place: these are precisely the pay-off dominant equilibria. Lewis calls such equilibria 'signaling systems'. Technically, \( \langle s, r \rangle \) is a signaling system iff \( f(t) = r(s(t)) \), for every \( t \in T \). Necessary condition for \( \langle s, r \rangle \) be a signaling system is that \( s \) and \( r \) are bijective functions.

\(^4\)In fact, we are dealing with a *Bayesian* Nash equilibrium. For definitions consult (Osborne and Rubinstein 1994).
\[
\begin{array}{cccc|cccc}
  t_1 & t_2 & m_1 & m_2 & r_1 & r_2 & r_3 & r_4 \\
 s_1 & m_1 & m_1 & r_1 & a_1 & a_1 & & \\
 s_2 & m_1 & m_2 & r_2 & a_1 & a_2 & & \\
 s_3 & m_2 & m_1 & r_3 & a_2 & a_1 & & \\
 s_4 & m_2 & m_2 & r_4 & a_2 & a_2 & & \\
\end{array}
\]

3 Sally's model and four rules of thumb

Sally gives a sketchy account of a signaling game that models the use of non-literal speech (and conventionalization of non-literal meaning). Contrary to Lewis, Sally takes speech acts (SA) as primitives rather than states. So Nature assigns S a speech act sa \( \in SA \), and it is up to R to interpret S's message \( m \in M \) as the intended speech act. Sally (Sally 2003) defines a 'general signaling game' \( G' \) as a tuple \( \{ S, R \}, \{ S', R' \}, \{ U_S, U_R \} \), where S and R are the two players as we had them before; \( S' (R') \) is the set of strategies \( SA \rightarrow M (M \rightarrow SA) \) for player S (R); and the utility function \( U_S \) and \( U_R \) capture the setting, the needs and the wants of the participants, their relationship [and] their prior statements (Sally 2003, pg. 1232). Sally (Sally) has shown that every strategy-pair \( (s', r') \) such that \( sa = r'(s'(sa)) \) for all \( sa \in SA \) is a Nash equilibrium.

Other than in cheap talk signaling games two separating Nash equilibria \( (s', r') \) and \( (s'', r'') \) do not yield equivalent utilities per se; i.e. possibly \( U_S(s', r') \neq U_S(s'', r'') \) and \( U_R(s', r') \neq U_R(s'', r'') \). This differing pay-off models the differing cognitive and social impact\(^5\) of the messages used to communicate a certain speech act. In particular, Sally gives empirical evidence for the fact that successful communication by means of a non-literally intended message has greater impact than a literally intended message. On the other hand, in the case of unsuccessful communication S sent a non-literally intended message and/or R interpreted the received message non-literally. Sally gives empirical evidence that this situation is more painful for the player (possible both) who deviated from the literal meaning. Moreover, Sally claims that the penalty for the player deviating from the literal meaning in case of unsuccessful communication is greater than the reward the player experiences in case of successful non-literal communication. Intuitively we agree with this: One can call one's friend a wanker, but how extremely embarrassing is the conversation wherein the supposed friend takes it as an insult!

Our criticism to Sally's general signaling game is twofold. Firstly, by taking the speech acts as primitive objects, Sally neglects the performative aspect of signaling games. It is in Lewisean signaling games the message \( s(t) \) together the act of playing the game according to \( s \) that causes \( s(t) \) become a speech act in itself. As such, Lewisean signaling games have more atomic primitives than general signaling games. Secondly, we sense that pay-off functions \( U_S \) and \( U_R \) in their current form take so many parameters into account that any formal elaboration is doomed to fail. Furthermore, the utility-functions do not reflect any philosophical considerations as to how non-literal language is opposed to literal language. E.g. is the notion of literal language incorporated in the "setting" or the interlocutors' "prior statements"? In the next section we will rigorously

\(^5\)People have shown to remember non-literal utterances better than literal ones. Furthermore, use of non-literal speech implicitly stresses the common ground of the interlocutors. For references see (Sally 2003).
define our signaling games, that take the states as primitives and have the literal meaning of the message parameterized.

On the other hand, we will adopt Sally’s insights under what condition people tend to communicate non-literally. Sally namely takes for granted that “people play the language game in a way that is consistent with their play in all games”. We think that he is right in doing so, and like to quote Rubinstein in favor of this stance: “[...] if game theory is to shed light on real life phenomena, linguistic phenomena are the most promising candidates. Game theoretical solution concepts are most suited to stable life situations which are “played” often by large populations of players”. Hence, the condition that causes players to coordinate on such-and-such an equilibrium also causes interlocutors to communicate in a way that corresponds to that type of equilibrium. In particular, Sally links pay-off dominant Nash equilibria to a convention of non-literal meaning that is more risky than the convention of literal meaning.

Below we have the rules stated as to how people play coordination games and as to under what conditions interlocutors communicate non-literally. We will use them in the same way Sally does.

Rules of thumb – Coordination games Although Harsanyi and Selten (Harsanyi and Selten 1988) argued otherwise, people have been shown to coordinate on the risk dominant Nash equilibria by default. Sally summarized these results ((Sally 2003), pg. 1229–1231, for references to empirical evidence (Sally 2003)):

Rule 1: “In a game with one outcome risk dominant and another ‘modestly’ pay-off dominant, the former is more likely to be chosen.”

Rule 2: “As sympathy between the players increases, a pay-off dominant, risk dominated equilibrium is more likely to be realized.”

Rules of thumb – Non-literal language usage In the literature it is argued that non-literal speech can only be used if the common ground of speaker and hearer is big enough. For otherwise the receiver can not make up whether the speaker intended the utterances literally or metaphorically. We summarize the results as follows (for references to empirical evidence see (Sally 2003)):

Rule i: “If the common ground is minimal, people are more likely to speak literally than non-literally.” We take this rule for granted.

Rule ii: “The more the common ground of interlocutors come to overlap, the more they will use non-literal language.”

It is not too daring an assumption that the more people are sympathetic to each other, the more their common ground overlaps. In the next section, we will define the notion of a SC signaling game. In Section 5 we will predict which equilibrium will be satisfied under what conditions assuming that players play this game following Rules 1 and 2.

4 Super Conventional Signaling Games

Intuition behind SC signaling games says that S and R play a signaling game, having common knowledge of the fact that \( \langle cs, cr \rangle \) is the conventional signaling system. Thus, we denote the
conventional sender and receiver strategy by means of $cs$ and $cr$, respectively. It is these strategies that model the literal meaning. As such they have agreed on the conventional meaning of messages in

$$M' = \{ m \in M \mid \text{there exists a } t \in T \text{ such that } s(t) = m \},$$

that contains the messages that convey the to-be communicated types. Since $s$ is a function, $|M'| = |T|$. In accordance with Searle (Searle 1979), it is only the messages in $M'$ that can have non-literal meaning.

Typically, non-literal utterances have it that the sentence taken literally means something different than was intended by the speaker. In formal terms, although $S$ is of type $t$ he uses a message $m \neq cs(t)$, and $S$ wants and expects $R$ not to perform $cr(m)$ but $f(t)$.

Following Sally we assume that it pays off to use non-literal speech instead of conventional. We, however, will model the extra gain by a parameter $\epsilon > 0$. On the other hand we argued that unsuccessful communication is more ‘painful’ for the interlocutor (possibly both) who deviated from the convention. So if $S$ uses message $m \neq cs(t)$ (e.g. $m$ =“wanker”) to communicate $t$ (e.g. $t$ =“Hey friend”), but $R$ performs $a = cr(cs(t))$ (e.g. hitting $S$ in the face), $S$ and $R$ did not successfully communicate. We hold that this is more painful for $S$ than it is for $R$: $R$ gets 0 as usual, $S$ gets $-\epsilon'$, for some parameter value $\epsilon' > 0$. From our discussion above it follows that $\epsilon' > \epsilon$.

This brings us to the main definition of this paper. A Super Conventional signaling game $G_{\langle cs, cr \rangle}$ is a standard signaling game $G$ equipped with a convention

$$\langle \{ S, R \}, \{ S, R \}, \{ U_S, U_R \}, \langle cs, cr \rangle \rangle,$$

where $U_S$ and $U_R$ are equal to (22.2) and (22.3), except for $u_S$ and $u_R$ as defined in (22.1), respectively. Instead of (22.1) we take

$${u_S}^e_{\epsilon'}(f, t, s, r) = \begin{cases} 
1 + \epsilon, & \text{if } f(t) = r(s(t)) \text{ and } s(t) \neq cs(t); \\
1, & \text{if } f(t) = r(s(t)) \text{ and } s(t) = cs(t); \\
0, & \text{if } f(t) \neq r(s(t)) \text{ and } s(t) = cs(t); \\
-\epsilon', & \text{if } f(t) \neq r(s(t)) \text{ and } s(t) \neq cs(t); 
\end{cases}$$

and

$${u_R}^e_{\epsilon'}(f, t, s, r) = \begin{cases} 
1 + \epsilon, & \text{if } f(t) = r(s(t)) \text{ and } r(s(t)) \neq cr(s(t)); \\
1, & \text{if } f(t) = r(s(t)) \text{ and } r(s(t)) = cr(s(t)); \\
0, & \text{if } f(t) \neq r(s(t)) \text{ and } r(s(t)) = cr(s(t)); \\
-\epsilon', & \text{if } f(t) \neq r(s(t)) \text{ and } r(s(t)) \neq cr(s(t)). 
\end{cases}$$

In the remainder of this section we will prove that every signaling system is a Nash equilibrium; furthermore we give characterizations of payoff dominance and risk dominance in SC signaling games with respect to signaling systems (and not Nash equilibria) to trigger Rules 1 and 2.

**Fact 1** Let $T$, $M$ and $A$ be sets such that $|T| = |M| = |A|$. Let $G_{\langle cs, cr \rangle}$ be a SC signaling game. If $\langle s, r \rangle$ is a signaling system in $G_{\langle cs, cr \rangle}$, then $\langle s, r \rangle$ is a Nash equilibrium in $G_{\langle cs, cr \rangle}$.
Proof The crux of the proof is the fact that both s and r are injective functions, assuming that \( \langle s, r \rangle \) is a signaling system. Any unilateral deviation will result in a function that communicates at least one \( t \in T \) incorrectly. Therefore, \( S \) nor \( R \) can gain from unilateral deviation. \( \square \)

The converse does not hold. But we point out that every Nash equilibrium \( \langle s, r \rangle \), that is not a signaling system, can be extended to a signaling system in the following sense: Let \( S \subset T \) be the set of types that is successfully communicated through \( \langle s, r \rangle \): \( S = \{ t \in T \mid f(t) = r(s(t)) \} \). Then, there exists a signaling system \( \langle s^#, r^# \rangle \) such that \( s^#(t) = s(t) \), for every \( t \in S \). Furthermore, we claim that every signaling system \( \langle s^#, r^# \rangle \) thus obtained, pay-off dominates \( \langle s, r \rangle \). The signaling systems characterized in Fact 2 (below) count as a special case of this claim.

As we saw in the proof of Fact 1, there exist equilibria that communicate some types literally and some non-literally. A signaling system \( \langle s, r \rangle \) divides \( T \) in two disjoint subsets

\[ L_{\langle s, r \rangle} = \{ t \in T \mid s(t) = cs(t) \text{ and } f(t) = r(s(t)) \} \]

and

\[ M_{\langle s, r \rangle} = \{ t \in T \mid s(t) \neq cs(t) \text{ and } f(t) = r(s(t)) \} \]

That is, \( L_{\langle s, r \rangle} \) is the set of types that are communicated literally and \( M_{\langle s, r \rangle} \) contains all other types — the ones communicated non-literally. We call the number of types that are communicated literally the rank of an equilibrium, relative to the game \( G_{\langle cs, cr \rangle} \) of course. Formally, the rank of a signaling system \( \langle s, r \rangle \) equals \(|L_{\langle s, r \rangle}|\). It is an easy exercise to prove that for all \( r \in \{0, \ldots, |T|\} \) there exists a signaling system with rank \( r \). The pay-offs of a signaling system can be expressed in terms of these sets \( L_{\langle s, r \rangle} \) and \( M_{\langle s, r \rangle} \). In a signaling system \( \langle s, r \rangle \) namely, \( S \) gains 1 or \( 1 + \epsilon \) per \( t \in T \) depending on whether \( t \in L_{\langle s, r \rangle} \) or \( t \in M_{\langle s, r \rangle} \). Therefore,

\[ U_S(s, r) = \left( \sum_{t \in L_{\langle s, r \rangle}} P(t) \right) + \left( \sum_{t \in M_{\langle s, r \rangle}} (1 + \epsilon)P(t) \right) \]

Since \( \langle s, r \rangle \) is a signaling system, \( s \) is injective; hence the conditional probability that \( t \) is selected, provided that message \( s(t) \) is sent, is 1. Consequently, \( U_S(s, r) = U_R(s, r) \) for all signaling systems \( \langle s, r \rangle \).

Rule 1 and 2 mention pay-off and risk dominant Nash equilibria. To see what strategy-pairs have this property in SC signaling games we give characterizations of both notions in Fact 2 and 3. Fact 3 only characterizes risk dominant signaling systems; from our above discussion concerning extensions of equilibria we hope to have made clear that in fact it is the signaling systems that are most interesting anyhow.

Fact 2 Let \( \langle s, r \rangle \) be a Nash equilibrium in \( G_{\langle cs, cr \rangle} \). Then \( \langle s, r \rangle \) is a pay-off dominant Nash equilibrium in \( G_{\langle cs, cr \rangle} \) iff \( \langle s, r \rangle \) is a signaling system of rank 0.

Proof Realize that the maximum pay-off is \( 1 + \epsilon \). \( \square \)

Fact 3 Let \( \langle s, r \rangle \) be a signaling system in \( G_{\langle cs, cr \rangle} \). If \( \epsilon' > \epsilon \), then \( \langle s, r \rangle \) risk dominates all other signaling systems in \( G_{\langle cs, cr \rangle} \) iff \( s = cs \) and \( r = cr \).
Proof \((\Leftarrow)\) We observe that \(L_{(cs, cr)} = T\). We will prove that for any other signaling system \(\langle s, r \rangle\) in \(G_{(cs, cr)}\), we have that

\[
A = (U_S(cs, cr) - U_S(s, cr))(U_R(cs, cr) - U_R(cs, r)) \geq (U_S(s, r) - U_S(cs, r))(U_R(s, r) - U_R(s, cr)) = B.
\]

If a conventional strategy, say \(cs\), is played against a non-literal one, say \(r\), communication fails in case Nature selected a type \(t\) that is non-literally interpreted by \(r\); that is, \(t \in M_{\langle s, r \rangle}\). We denote

\[
\left( \sum_{t \in L_{\langle s, r \rangle}} P(t) \right) \text{ and } \left( \sum_{t \in M_{\langle s, r \rangle}} P(t) \right)
\]

by \(p\) and \(1-p\), respectively. The respective pay-offs can be obtained from (1), (2), (4) and (5). Filling those in yields: \(A = (1 - (p - \epsilon'(1-p))^2\) and \(B = ((1 + \epsilon)(1-p))^2\). And, as required, by rewriting we learn that \(A > B\), if \(\epsilon' > \epsilon\).

\((\Rightarrow)\) Suppose \(\epsilon' > \epsilon\) and \(\langle s, r \rangle\) is not risk dominant. Then this signaling system is not equal to the conventional signaling system, since this one was risk dominant.

If we conceive of a SC signaling game as a real game, played by real people in a lab, we can apply Rule 1 and 2 from Section 3 to predict the players' behavior. That is, following Rule 1, people by default coordinate on a risk dominant equilibrium. Applying Fact 3 yields that this people by default communicate by means of the conventional signaling system. This matches Rule i. Now, if sympathy increases between people (and their common ground increases) a pay-off dominant, risk dominated equilibrium is more likely to be realized. From Fact 2 we learn that realizing a pay-off dominant equilibrium is equal to realizing a signaling system that communicates non-literally. Again this matches Rule ii.

5 Conclusion

In this papers we have introduced a model for non-literal language usage, that builds on Lewisian signaling games. Furthermore, it is consistent with Searle's conception of non-literal language. On the other hand, the model meets the rawest desiderata set by experimental literature on coordination games and language use.

For future research we suggest to embed our game-theoretical model in a broader pragmatical context. A more detailed study of the role of sympathy within the model as well as a more detailed account of different kinds of non-literal speech acts would be interesting.


Expectation Reasoning: A Computationally Grounded Theory of Agency

Bình Vũ Trân
School of Computer Science and Information Technology, RMIT University, Australia
tvubinh@cs.rmit.edu.au

James Harland
School of Computer Science and Information Technology, RMIT University, Australia
jah@cs.rmit.edu.au

Margaret Hamilton
School of Computer Science and Information Technology, RMIT University, Australia
mh@cs.rmit.edu.au

Abstract.
The computational grounding problem is a well known problem within the agent research community. For years, it has been believed that obscure ontological status is the principal cause. This acute problem hampers the speed of agent oriented development. In this paper, from our formulation of this gap, we propose an alternative way for modelling intelligent agents through the concepts of observation and expectation.

1 Introduction
The view of agents as intentional systems by Dennett (1987) has been dominant for many years. Mental attitudes such as knowledge, belief, desires, hopes, fears, etc. have been formally analysed to predict the intelligent behaviour of an agent. Following Hintikka (1962), this study is usually carried out using modal logics with possible-worlds semantics. In such theories, modal operators are added to represent an agent’s mental attitudes. However, the three questions raised by Wooldridge in his thesis (Wooldridge 1992) about the ontological status of possible worlds remain improperly answered: “Do they really exist? If so, where are they? How do they map onto an agent’s physical architecture?” This is usually referred as the computational grounding problem of agent theories.

To readdress these questions, our work is divided into two main parts: the syntactical problem and the semantic problem. In the first part – the syntactical problem – we investigate what
hampers agent developers from mapping agent theory onto an agent physical architecture. In other words, we investigate whether there exists a one-to-one correspondence between agent theory and its abstract interpreter. One approach (Wooldridge 2000; Fagin, Halpern, Moses, and Vardi 1995; Rosenschein 1985; Wooldridge 1992; Wooldridge and Lomuscio 2001) denies the existence of such relationship with the assumption that “in general, there is no relationship between models mod(Ł) for (a modal logical language) Ł (representing a theory of agency) and computations C.” For example, Wooldridge opts the approach in (Fagin, Halpern, Moses, and Vardi 1995; Rosenschein 1985; Wooldridge 1992; Wooldridge and Lomuscio 2001) to directly use a computational model for deriving formulae of the specification language. Consequently, this approach classifies a number of other useful theories of agency such as Cohen-Levesque’s theory of intention (Cohen and Levesque 1990) and Rao-Georgeff’s Belief-Desire-Intention (BDI) logics (Rao and Georgeff 1991; Rao and Georgeff 1998) as ungrounded.

On the other hand, the approach led by Rao (1996) accepts such existence, but they are unable to increase expressivity of modal language. The resulting languages of these researches are usually a fragment of first-order logic. Following this approach, we argue that the relationships between models (including computational models) preserving all properties of modal languages have been very well studied under the notion of bisimulation (Blackburn, de Rijke, and Venema 2001) (cf. p-relations (van Benthem 1983, Definition 3.7)). Bisimulations are many-to-many relations. Hence, the problem is not because of the non-existing relationship. It is a selection problem: which corresponding model is the most appropriate one? The reason revealed by the study of bisimulation is that modal languages are not expressive enough to define different properties of a possible worlds frame.

In our work, following Blackburn (2000b), first, we trace back to what is known as the asymmetry problem of modal logic (Blackburn 2000b): although possible worlds are crucial to Kripke semantics, nothing in modal syntax is able to represent them. This results in inadequacy and difficulty of using modal logic as representation formalism and reasoning systems. By adopting Blackburn et al’s hybrid logics (Areces, Blackburn, and Marx 2001; Blackburn 2000a; Blackburn and Tzakova 1999), we show that it is possible to construct a corresponding computational model for any agent theory specified by a hybrid language. By uniquely tagging a label to each world, a bisimulation between two hybrid models ensures that not only points named by the same label are linked to each other and but importantly only to each other. Hybrid bisimulation therefore becomes isomorphic (Areces, Blackburn, and Marx 2001). Crucially, the increase of expressive power comes with no cost: the satisfiability problem remains decidable in PSPACE-complete (Areces, Blackburn, and Marx 2001). This approach is exactly what we, human beings, do when we get lost in a jungle or in a maze by using landmarks or observing path-turning angles respectively. This assists us in envisaging the world structure in our mind as precisely as the real world that we observe.

In the second part of our work, we identify the semantic problem. The approach and theory of knowledge summarised by Wooldridge (2000) are essentially based on sensory observations and the relationships between them. A crucial assumption in this approach is that “there is no uncertainty about the result of performing an action in some state”. Hence if there exists a difference between a belief and a sensory information, the belief finds no ground in this framework though the sensory information can be incorrect or imperfect. Although Wooldridge also claims “dropping this assumption is not problematic,” there is no pivotal work showing how useful the introduction of uncertainty would provide for the grounds of mental attitudes such as beliefs and
desires.

So, what should be the grounds for mental attitudes such as beliefs, desires? Jakob Fries conceived the existence of non-intuitive grounds of knowledge and Leonard Nelson advocated this idea in (Nelson 1949). Close to this approach, Karl Popper (1969, p. 47) proposed a more specific notion: the notion of expectations. Each agent is born with expectations, the psychologically or genetically a priori, i.e. prior to all observational experience. The crucial point of this approach is: once an expectation is disappointed by observations, it creates a problem. The process of error elimination using critique continuously generates new expectations and also new problems. The growth of knowledge proceeds from old problems to new problems, by means of conjectures and refutations (Popper 1972, pp 258, 259).

The goal of our work is to introduce a new formal reasoning system based on expectations which does not suffer the fate of the computational grounding problem. Section §2 introduces a possible-worlds model where every possible world has a corresponding ground which can be translated to a computational model. In section §3, we introduce the observation refutation method as a fundamental tool in knowledge evolution. Section §4 discusses further an approach to integrate criticism into the process.

2 The Ground of Knowledge

2.1 Expectation and Observation

In answering the questions about the ontological status of possible worlds: “Do they really exist? If so, where are they?” Wooldridge (1992, 2000), following (Fagin, Halpern, and Vardi 1992), takes computational states as the grounds of knowledge. The grounds correspond to an agent’s sensory experience of the real world — intuitive immediate knowledge. This approach offers a powerful analytical tool in various problems such as knowledge-based protocols. Unfortunately, intuitive immediate knowledge cannot be grounds for mental states such as beliefs, which can hold false statements according to sensory observations. For example, a belief that “David Copperfield cannot fly” may well become false when one sees his body floating in the air. A ground for such statements is discovered, in the Kant-Friesian school of philosophical theory, as non-intuitive immediate knowledge. According to Nelson (1949), proof, demonstration and deduction are three possible ways to ground a proposition: 1) proof provides justification using logical derivation; 2) demonstration verifies judgements by pointing out the intuition on which they are grounded. The task is done through sensory observations; 3) deduction provides justification on a non-intuitive ground — the immediate knowledge of pure reason. This task remains within the limits of mental self-observation.

Through this analysis, the realm of observation emerges as an important means for judgements. It represents the relation between the material world $\mathcal{G}$, on which intuitive ground lies, and the mind of each agent $a_i$, on which non-intuitive ground lies. Herein, we call the mental state corresponding to the occurrence of each observation expectation. Formally,

**Definition 1** An observation relation $\mathcal{O} \subseteq \mathcal{G} \times \mathcal{E}_i$ is a relation between the real world $\mathcal{G}$ and a subset of mental states called expectation set $\mathcal{E}_i$, where $i \in \mathcal{I}$ is the identity of an agent $a_i$.

Sensory observations are obtained via sensors (e.g. the eagle’s eyes $\varsigma$) where mental self-observations are expressed through effectors (e.g. wings $\varepsilon$). An important note here is that
sensory observations are full in the sense that they fully associate the states of the material world and the states of mind. However, since mental observations are non-intuitive, they do not contain the association with it. Hence, to complete a mental (effective) observation, it must be associated with at least a sensory observation. Let $S = \bigcup_{i \in I} S_i$ and $E = \bigcup_{i \in I} E_i$ be respectively the sensor and effector sets of an observation system, where $S_i$ and $E_i$ are respectively the sets of sensors and effectors of an individual agent $a_i$.

The formation of these primitive observations is called an observation method. An observation method which contains only one primary sensor or effector is called primitive observation method $\mathcal{M}_0$ (e.g. $e_i, e \in \mathcal{M}_0$). A more complicated set of observation methods $\mathcal{M}_k$ would arrange the $k$ expectations of other observation methods in some order to generate new expectations about the world. These expectations are also associated with global states to form more complex observations.

**Definition 2 (Observation methods)** An observation method family is a set of observation method sets $\mathcal{M} = \{\mathcal{M}_k\}_{k \in \mathbb{N}}$ where $\mathcal{M}_k$ is a set of observation methods of arity $k$ for every $k \in \mathbb{N}^+$. $\mathcal{M}_0 = S \cup E$ is called primitive observation method set. $\mathcal{M}_k$ is inductively defined as follows:

- $e \in \mathcal{E}$ for all $e \in \mu_0, \forall \mu_0 \in \mathcal{M}_0$
- $\mu_k(e_1, \ldots, e_k) \subseteq \varphi(\mathcal{E})$ for all $\mu_k \in \mathcal{M}_k$ and $e_1, \ldots, e_k \in \mathcal{E}$

### 2.2 Expectation Model

Before defining a modal logical language which can also describe possible worlds in its syntax, in this section, we first describe the set of possible worlds $\mathcal{G}_i$ of an agent $a_i$. Each possible world $g$ carries the agent $a_i$'s information about its environment. There are two closely related sources of information: from the agent's set of sensors $S_i$ and from the results of the agent's effectors $E_i$. By organising the sources of information, an agent will obtain certain information about its environment. We call each possible world (each possible way of organising information sources) an observation. On the one hand, similar to epistemic logic, the grounds of sensory observations are information from sensors. On the other, following philosophers Fries, Nelson and Popper, we ground effective observations on non-intuitive grounds, inborn expectations. Inborn expectations are genetically given to an agent at its birth. We call the valid set of formulae at an observation its expectations.

The expectation language $\mathcal{L}$ is similar to the language of propositional logic augmented by the modal operator $\mathcal{E}_i$ and the observation operators $\mathcal{G}_s$, where $s$ is an observation label i.e. an atomic proposition which is true at exactly one possible world in any model.

**Definition 3 (Expectation language)** Let $\Phi$ be a set of atomic expectation propositions. Let $\Xi$ be a nonempty set of observation labels disjoint from $\Phi$. An expectation language $\mathcal{L}$ over $\Phi$ and $\Xi$ where $p \in \Phi$ and $s \in \Xi$ is defined as follows:

$$\varphi := s \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi \rightarrow \varphi \mid [\mathcal{E}_i] \varphi \mid [\mathcal{G}_s] \varphi \mid \mathcal{G}_s \varphi.$$  

Two observations are said to be related if one can be obtained by changing (adding/removing) the information sources (sensors/effectors) of the other. In other words, it is possible to reach the other observation by changing the information sources from the current one. Let $\sim \subseteq \mathcal{G}_i \times \mathcal{G}_i$ be
the set of such related observations. The pair \( \mathcal{F} = \langle G_i, \sim \rangle \) is called an \textit{observation frame}. The interpretation of an agent \( a_i \)'s expectations is defined by the function \( \pi : \Phi \cup \Xi \rightarrow \wp(\mathcal{F}) \). The crucial difference from orthodox modal logic in this definition is that for every observation label \( s \in \Xi, \pi \) returns a \textit{singleton}. In other words, \( s \) is true at a unique observation, and therefore tags this observation (Blackburn 2000a). The triple \( \mathfrak{M} = \langle G_i, \sim, \pi \rangle \) is called an expectation model.

\textbf{Definition 4} \textit{The semantics of expectation logic \( \mathcal{L} \) are defined via the satisfaction relation \( \models \) as follows}

1. \( \langle \mathfrak{M}, g \rangle \models p \iff g \in \pi(p) \) (for all \( p \in \Phi \))
2. \( \langle \mathfrak{M}, g \rangle \not\models \neg \varphi \iff \langle \mathfrak{M}, g \rangle \not\models \varphi \)
3. \( \langle \mathfrak{M}, g \rangle \models \varphi \lor \psi \iff \langle \mathfrak{M}, g \rangle \models \varphi \) or \( \langle \mathfrak{M}, g \rangle \models \psi \)
4. \( \langle \mathfrak{M}, g \rangle \models \varphi \land \psi \iff \langle \mathfrak{M}, g \rangle \models \varphi \) and \( \langle \mathfrak{M}, g \rangle \models \psi \)
5. \( \langle \mathfrak{M}, g \rangle \models \varphi \rightarrow \psi \iff \langle \mathfrak{M}, g \rangle \not\models \varphi \) or \( \langle \mathfrak{M}, g \rangle \models \psi \)
6. \( \langle \mathfrak{M}, g \rangle \models \langle E_i \rangle \varphi \iff \langle \mathfrak{M}, g' \rangle \models \varphi \) for some \( g' \) such that \( g \sim g' \)
7. \( \langle \mathfrak{M}, g \rangle \models \langle \bar{E} \rangle \varphi \iff \langle \mathfrak{M}, g' \rangle \models \varphi \) for all \( g' \) such that \( g \sim g' \)
8. \( \langle \mathfrak{M}, g \rangle \models s \iff \pi(s) = \{g\} \), for all \( s \in \Xi, g \) is called the \textit{denotation} of \( s \)
9. \( \langle \mathfrak{M}, g \rangle \models \#_s \varphi \iff \langle \mathfrak{M}, g_s \rangle \models \varphi \) where \( g_s \) is the \textit{denotation} of \( s \).

\textit{where} 1 – 7 are standard in modal logics with two additions of hybrid logics in 8 and 9.

\section{3 Reasoning about Unexpectedness – Observation calculus}

We now come to the heart of the work. To illustrate why expectation reasoning is useful, imagine a hungry eagle is chasing an agile sparrow in a maze which neither of them know how to get out in advance. Suddenly, the sparrow darts behind a wall and disappears from the eagle's field of view. Naturally the eagle would very rarely stop the chase and rest. It continues by making conjectures, \textit{its expectations}, whereabouts the sparrow is, and acts accordingly. However, the eagle's expectations are often violated. The sparrow may not reappear at the end of the wall as expected or it may get into a dead-end. By reasoning about such unexpectedness critically, the eagle would be able to eliminate erroneous expectations and also generate new ones finding a way to catch the sparrow and get out of the maze. Throughout all of his well-known books about scientific knowledge, Popper (1959, 1969, 1972) summarised the above reasoning process by the following schema

\[ P_1 \rightarrow TT \rightarrow EE \rightarrow P_2 \]

Here \( P \) stands for problem, \( TT \) stands for tentative theory, and \( EE \) stands for error-elimination. \textit{The first problems} are created when an agent's \textit{inborn expectations} are disappointed by some
observation. The ensuing growth of knowledge may then be described as the process of correcting and modifying previous knowledge through new observations using refutation methods.

Analytic semantic tableaux methods are refutation methods that have received much attention recently in automated theorem proving. A technique by Fitting (Fitting 1996), which defers the choice of free-variables until more information is available, has been used to reduce search space and the non-determinism inherent in automated proof search. This technique resembles the ability to use expectations as assumptions to delay a current obstructed observation until justified as well as to use expectations as conjectures to find path in a maze in the above example. Among different approaches using free variables in the labels of semantic modal tableaux, Beckert and Goré's string matching technique (Beckert and Goré 1997) can be used to describe the connection between sensory observations and effective observations. Before introducing our refutation methods, we now elaborate a construction for the observation frame in §2.2 as follows.

Firstly, an observation (a possible world) is described in detail by a sequence $\sigma$ of sensors and effectors. For example, the eagle's first observation, looking forward using its eyes $\varsigma \in S_i$, is represented by the string $\sigma = \varsigma$. Its next observation by subsequently using its wings $\varepsilon$ is represented by $\sigma = \varsigma.\varepsilon$. The brackets denote that the resulting effect of $\varepsilon$ is non-intuitive and this effect can be verified using the eagle's eyes $\varsigma$. We use $[\varsigma]$ to denote a generic observation (either sensory or effective observation). An assumption can be made by substituting a free variable $x$ of the variable set $\mathcal{A}$ at some position of the sequence. Each of this sequence can be named using the set observation labels $\Xi$ by the naming function $N$.

**Definition 5 (Linear observation method syntax)** Let $\mathcal{A} = \{x, y, \ldots\}$ be a set of assumptions which are originally not bound to any sensors or effectors. Let $\Gamma$ be a set of observation sequences. A linear observation method can be expressed by a string $\sigma \in \Gamma$ defined inductively as follows:

i. $\varsigma$ is an observation sequence for all $\varsigma \in S_i$;

ii. If $\sigma$ is an observation sequence, then so are $\sigma.\eta$ (if $\eta \in S$) and $\sigma.(\eta)$ (if $\eta \in \mathcal{E}$);

iii. If $\sigma$ is an observation sequence, then for all $x \in \mathcal{A}$ $\sigma.(x)$ is also an observation sequence but not $\sigma.x$. $\sigma.(x)$ stands for all possible successors of the observation sequence $\sigma$.

iv. $\text{prefix}(\sigma) = \{\tau \mid \sigma = \tau.\theta\}$ is a function which returns a set of all prefixes of an observation sequence $\sigma$.

v. The function $N : \Xi \to \Gamma$ assigns each label in $\Xi$ to an observation sequence.

By this novel representation, effective observations and assumptions have not only found their non-intuitive grounds but also intertwined closely with sensory observations. With the refinement of observations and observation methods defined above, we can describe our refutation method. An expectation $@s.\varphi$ can be refuted by constructing a tableau proof with $@s.\varphi$ at its root, where $s$ is a name for the observation from a built-in sensor $\varsigma$ and $\varphi$ is the unexpected information perceived by $\varsigma$.

Due to space restrictions, we omit the tableau construction rules here. However, there are some important points to discuss here. Though all rules look standard in any KE system (D'Agostino and Mondadori 1994), the tableau modality expansion rules have some distinctive significance to be discussed here. Firstly, it is important to note that the only branching rule in this rule set is
PB (principle of bivalence), which considerably reduces the search space. This rule further insists that the choice to pursue a path is dependent on the truth value of the expectations themselves not on the connectives that link together. This also impacts how the nature of an assumption changes. Normally, a universal assumption \( x \) is introduced into an observation sequence \( t \) by the reduction of universal rules (\( \neg \epsilon \) or \( \Box \)-rule). For example, the eagle may have the expectation “All sparrows are agile”. Hence, the eagle can assume that in any further observation it would see agile sparrows. The assumption is no longer universal (or being freed), when the eagle is able to observe some sluggish sparrow in one of its further observation. A system with PB rule can easily show this change. In classic methods such as Beckert and Goré’s free-variable tableau method (Beckert and Goré 1997), it would not be possible to represent it since the branching rule is based on disjunction rule. By the integration of hybrid logics (Areces, Blackburn, and Marx 2001; Blackburn 2000a; Blackburn and Tzakova 1999), free-variable modal tableaux (Beckert and Goré 1997), and the KE system (D’Agostino and Mondadori 1994) we still obtain the soundness and strong completeness results for this system.

4 Reasoning about Unexpectedness: Applications

4.1 Criticism

Besides refutation method which provides a useful way to determine an appropriate theory, criticism plays a significant role in knowledge evolution. Originating from Kant (1893), criticism searches for contradictions and their elimination.

For example, in the bit transmission problem (Halpern and Zuck 1992), there is no guarantee that any messages sent by either agent are correctly received. Halpern and Zuck’s insight was sending acknowledgement messages carrying the knowledge state of the sender of the message.

However, when the bit takes more than one cycle to be delivered, this approach can be inefficient since the sender has to wait until the bit is received. The problem becomes more interesting when the channel is working properly at certain times, while failing at others. Worse, the failures at different times may be totally different. The ability of agents to ignore insignificant errors (since they can correct them from what they know) and to construct new optimised protocols in an unstable environment thus becomes more desirable in such kind of environment.

This creates another problem: what is an appropriate justification? In other words, how to make further conjectures? We are currently investigating the possibility of using the triadic structure: Thesis-Antithesis-Synthesis by Hegel (1929). Unlike classical logic, where a double negation ("\( A \) is not \( \neg \neg A \)"") would simply reinstate the original thesis, Hegel suggested synthesis as the third emerging element for a higher rationality. Hence, the contradiction \( @_i \text{ recack} \) and \( @c \text{ recack} \) when the justification \( @_i a \) is made, can be analysed in a synthetic observation \( c \) where \( @c @_i \text{ recack} \) and \( @c @_c \text{ recack} \). This suggests a way to represent the justification strategy.

4.2 Disappointment and Regret

Research in economics such as Bell (1982, 1985) and Loomes-Sugden (1982, 1986) reveals that, if a person were put in uncertain situations, one’s decision would also depend upon how that person would feel when comparing the decision consequences together and with prior expectations. The difference between expectations and actual observations usually results in some mental states.
which would strongly effect the person’s future decision. Such mental attitudes are regret/rejoice and disappointment/elation.

Unlike the approach usually taken in reasoning about uncertainty (Halpern 2003), hybrid languages allow us to assign an arbitrary set of labels to possible worlds and not just numerical probabilities. We hence can extend further regret and disappointment theory by Bell (1982, 1985) and Loomes-Sugden (1982, 1986) to deal with situations where statistics are not available or it is impossible to calculate probabilities for the possible worlds. We can then further show different properties of observation frame for avoiding bad emotions (regret/disappointment).

5 Concluding remarks

In this work, we have elaborated the computational grounding problem by showing its anomalies and different approaches towards the problem. The expectation and observation framework has some significant advantage not just in giving more expressive power, but also in extending an evolutionary knowledge based system whilst dealing with unexpectedness and uncertainty. Our remaining work is to continue to prove the generality of this framework in comparison with others. Further work research questions include a number of areas such as more efficient tableau construction and assigning resources into possible worlds.
References


252
The indefinite; an extra-argument-slot analysis

HIROYUKI UCHIDA
University College London
uchida@ucl.ac.uk

ABSTRACT.
The domain restriction analysis proposed by Schwarzchild (2002) is promising in that it does not require an exceptional scope taking of indefinites. However, the intermediate scope reading and the functional reading of indefinites suggest that the domain restriction should be allowed to be dependent on another quantificational element in the sentence. I argue that an indefinite has an inherent argument slot that can be bound by a c-commanding quantifier in the same sentence, which marks the dependency of the domain restriction to this quantifier.

1 Introduction
It has been observed that the scope of the universal quantifier cannot cross a tensed clause.

(1) a. Some/ a teacher said that every student smoked at school.
b. *∀x[student′(x) → ∃y[teacher′(y) & say′([smoke′(x)]y)]].
c. *For each student, there is a possibly different teacher involved.

(1a) does not have a reading in which the universal quantifier in the embedded that-clause takes the wide scope over the indefinite in the matrix clause. The impossibility of the reading (1b) is often explained in terms of the impossibility of a covert quantifier movement out of the tensed clause, as in May 1977.

(2) a. Some/ a teacher said that every student smoked at school.
b. LF: *[Every student]_1 [some teacher said [CP that t_1 smoked at school]].
c. *every > some [: α > β means that α takes wide scope over β.]

In (2b), a covert movement of the DP every student at LF is blocked by the tensed that-clause, which makes the reading in (1b) impossible.

1I ignore tense in the logical forms I use.
However, if we put an indefinite in a tensed *that*-clause, as in (3a), the sentence does have the wide scope reading of the indefinite over the universal in the matrix clause.

(3)    a. Every teacher said that *some/a student* smoked at school.
       b. \( \exists x [\text{student}'(x) \land \forall y [\text{teacher}'(y) \rightarrow \text{say}'(\text{smoke}'(x)),(y)]]. \)
       c. *There is only one and the same student involved for all the teachers.*

If we assume that a covert quantifier movement at LF also explains this exceptional scope taking of indefinites, then we need to assume that only indefinites can covertly move out of the tensed clause, as in (4b). But this makes it difficult to formulate a quantifier raising as a uniform syntactic movement.

(4)    a. Every teacher said that *some/a student* smoked at school.
       b. LF: \([\text{Some student}]_1 [\text{every teacher said } [CP \text{ that } t_1 \text{ smoked at school}]]. \)
       c. *some > every*

Reinhart (1997) and Winter (1997, 2001) leave the indefinite within the tensed clause while introducing a choice function that picks out an individual member out of the nominal restriction set of the indefinite.

(5)    a. Every teacher said that *some/a student* smoked at school.
       b. LF: \( \exists_{(ct)c}[\text{CF}'(f) \land [\text{every teacher said } [\text{that } f(\text{student}) \text{ smoked at school}]]. \)
       c. \( \exists_{(ct)c}[\text{CF}'(f) \land \forall x [\text{teacher}'(x) \rightarrow \text{say}'_{t(c)}([\text{smoke}'(f(\text{student}'))]),(x)]]. \)

The rough LF representation (5b) or the more accurate logical form (5c) means that there is a choice function \( f \) such that for all the teachers \( x \), \( f \) picks out an individual out of the student set and \( x \) said that the chosen individual smoked at school\(^2\). Because the existential quantifier that binds the function variable has wide scope over the universal quantifier, the choice function picks out the same member for all the teachers. The indefinite noun phrase does not move out of the tensed *that*-clause at LF and so we do not need an exceptional covert movement of the indefinite.

However, this analysis introduces the function variable \( f \), the choice function property \( \text{CF}' \), and an existential closure applied to the function variable. These extra logical items make the paring of the phonological string and the logical form less systematic. Also, just as the covert quantifier raising analysis of indefinites in (4b) requires a movement that is not blocked by the tensed clause, the existential quantifier in (5b) binds the function variable in the embedded tensed clause from the top of the matrix clause. The structurally unconstrained nature of this operation is no less worrying than an exceptional quantifier raising possibility of indefinites in the quantifier raising analysis.

It would be better if we can assume that the scopes of both the universal quantifier and the existential quantifier are clause bound. The domain restriction analysis by Schwarzschild enables this, but the analysis in turn has some problems, as we see shortly. In order to solve these problems, I propose a formal modification of the theory. In section 2, I explain Schwarzschild’s domain restriction analysis and some of its problems. I also informally explain my proposal in

\(^2\)Winter uses a higher order choice function of type \(((\text{et})(\text{et}t))\) to deal with a case in which the nominal restriction set of the indefinite is empty. But in this paper, I use the type \(((\text{et})e)\) for simplicity.
that section. In section 3, I show how we can formalize this idea in the compositional derivation of an interface logical form. Section 4 is my conclusion.

2 The Domain Restriction

Schwarzschild (2002) claims that the exceptional wide scope reading of an indefinite is not a matter of its structural wide scope taking but a result of a pragmatic domain restriction. Consider the example (6a).

(6) a. Every teacher reported that some student smoked (at school).
    b. ∃x[student′(x) ∧ ∀y [teacher′(y) → report′_r,ェ_ル(x))(y)]].
    c. ∀y[teacher′(y) → report′_r,ェ_ル(∃x[student′(x) ∧ smoke′(x)])(y)].

In (6a), when the domain of the set of students is pragmatically restricted to a singleton set, an assertion is made about one and the same student, who is the only member of the set. This means that we can pragmatically derive the reading equivalent to the wide scope reading in (6b) from the narrow scope meaning encoded with the string, given in (6c). The scope of the indefinite stays within the embedded clause, so we do not need an exceptional covert movement of the indefinite or a structurally unconstrained existential closure applied to a choice function variable.

On the other hand, a challenge for the domain restriction analysis is the intermediate scope reading of indefinites as in (7). Ruys (1992:101-102) and Abusch (1994: 84 - 88) argue that the lexical ambiguity analysis of indefinites in Fodor and Sag (1982) cannot explain this intermediate scope reading because it predicts that an exceptional wide scope reading of an indefinite is always its widest scope reading. If the domain restriction analysis always gives the widest scope reading when the domain is restricted to a singleton, the analysis is subject to the same criticism.

(7) Every student discussed every analysis that solved a (certain) problem in Chomsky 1995. (cf.Reinhart 1997: 346)

(8) a. Every student admired a (certain) teacher - his homeroom teacher.
    b. A woman that every man loves is his mother. (cf. Winter 2003:1)

(7) has a reading that says that for each student x, there is a possibly different problem y in Chomsky 1995, and x discussed all the analyses that solved y. If the domain restriction to a singleton set is insensitive to other quantificational elements in the sentence, we make a wrong prediction such that whenever the domain is restricted to a singleton, a certain problem has to denote one and the same problem for all the students.

One way to solve this problem is to assume that an indefinite has an inherent argument slot on which the domain restriction is dependent.

(9) ∀x [student′(x) → ∀y [[analysis′(y) ∧ ∃z[sg′(question′)(z)(x) ∧ solve′(z)(y)]]] → discuss′(y)(x)]]

I explain the operator sg′ in more detail in section 3.2. In (9), sg′ maps the question set onto a function that maps an individual z to a singleton question set for z. The argument slot z is bound
by the universal quantifier \( \forall x \) and so we can restrict the domain differently for each student. In this way, we can pick out a different specific question \( z \) for each boy.

The same applies to (8a). In (8a), each student can admire a possibly different specific teacher, and if this specificity is the result of a domain restriction to a singleton set, the domain restriction has to be done in a possibly different way for each student. The extra argument slot of an indefinite enables us to do this.

The so-called functional reading gives another argument for this inherent argument selection of indefinites. The co-indexed pronouns in (8a) and (8b) give a problem to a structural analysis of pronoun binding because these pronouns are not within the surface c-command domains of the universal quantifiers. But if we assume that an indefinite has an inherent argument slot that can be formally linked to another quantifier, then we can claim that in (8a) (or in (8b)), the equality of the functional relation holding between the universal quantifier and the indefinite on the one hand and the functional relation between the universal quantifier and the noun phrase containing the pronoun on the other justifies this use of the pronoun\(^3\). In (8a), the sentence means that the function mapping each student to a singleton teacher-set for him is the same as the function mapping each student to a singleton homeroom-teacher-set for him. I regard the noun phrase his homeroom teacher as definite description and assume its linguistic meaning is a singleton set containing one homeroom teacher as its unique member. (8b) means that the function mapping each man to a singleton woman-set for him is the function mapping each man to the singleton set containing his mother as its unique member.

From these considerations, I argue that the indeterminates in (7), (8a) and (8b) have an inherent argument slot that can be bound by another element in the sentence, which makes the domain restriction dependent on this element. Schwarzschild assumes that the dependency of the domain restriction is pragmatically derived without a linguistic specification, but I assume that an indefinite noun phrase has an extra argument slot as lexical information, so that we can derive a dependency relation compositionally in a syntactic derivation of a logical form.

The phrase a certain changes the set into a singleton set. This leads to the specific reading, but the specificity can be relativized because of the inherent argument position of the indefinite. An indefinite a teacher without the word certain still has this inherent argument slot, but there is no linguistic singleton set requirement. We can still pragmatically restrict the domain into a singleton, and then the identity of this singleton set can be dependent on the inherent argument slot of the indefinite. However, because of the existence of the more specific expression a certain teacher, some native speakers have difficulty pragmatically restricting the domain into a singleton set to get the specific reading with the ordinary indefinite a teacher.

In the next section, I formalize this idea in a compositional derivation of a logical form as the meaning representation of a simple English sentence.

\(^3\)Winter (2003: 4-13) uses a similar argument to support his Skolem function analysis of indefinites. An example in Jacobson (1999) suggests that this identity of functional relations justifies a binding relation across a question-answer pair as well:

3 Formal Analysis

3.1 Categorial Grammar and the derivation of a logical form

Following a Categorial Grammar framework as in Jacobson (1999), I assume a Grammar derivation pairs a pronounced string with a logical form. More specifically, each lexical item has three entries:

(10)  \(</\text{phonological form}; \text{syntactic category}; \text{logical expression}>\)

For example, the lexical item \(\text{boy}\) is: \(</\text{boy}; \text{N}; \text{boy}_{et}>\). Lexical items are successively merged based on their syntactic categories, and at the end of a derivation, a phonological string is paired with a logical form.

I follow Jacobson (1999) for her notation of syntactic categories. The category \(X/RY\) selects the category \(Y\) to the right and the result category after the merge is \(X\). The category \(X/LY\) is merged with the category \(Y\) to the left, and the result is the category \(X\).

I adopt Jacobson’s variable-free framework, but like her, I use variables bound by lambda operators in my notations. In this way, we can follow a derivation more easily than in an alternative notation using only constants and combinators. As long as we do not use a free variable at any stage of a derivation, we can also represent the same derivation without using any variables (cf. Steedman 2000: Chapter 8).

3.2 A Sample Derivation

I show a derivation of a simple sentence in (11) as an example of a compositional derivation of a logical form.

(11)  Every boy loves a certain girl.

(12)  a. \(\text{girl}: </\text{girl}; \text{N}; \lambda x.\text{girl'}(x) [\text{or girl'}_{et}>].\)

b. \(\text{boy}: </\text{boy}; \text{N}; \lambda x.\text{boy'}(x) [\text{or boy'}_{et}>].\)

c. \(\text{love}: </\text{love}; ((S/LNP)/RNP) [\text{or TV}]; \lambda x.\lambda y. \text{love'}(x)(y) [\text{or love'}_{et}]>.\)

d. \(a: </a; N^U/RN; \lambda A_{et}. \lambda u_r. \lambda x_e. a'(A)(u)(x) [\text{or } a'_{et}(\tau(\tau))]>.\)

e. \(\text{a certain}: </a \text{certain}; N^U/RN; \lambda A_{et}. \lambda u_r. \lambda x_e. sg^f(A)(u)(x) [\text{or } sg^f_{et}(\tau(\tau))]>.

f. \(\text{every} (\text{Nom}): </\text{every}; (S/R(S/LNP))/RN; \lambda A_{et}. \lambda B_{et}. \forall x_e [A(x) \rightarrow B(x)]>.\)

g. \(\text{some}^* (\text{Acc}): </\text{some}; ((S/LNP)/LTV)/RN; \lambda A_{et}. \lambda P_{et}(\lambda x_e. \exists y_e[A(y) \land P(y)(x)]>.\)

Each lexical item has three entries as in (10). For some of the logical expressions, I give the variable free notations in the brackets [ ]. The determiner \(\text{some}^*\) in (12g) has a null phonological entry. This item is inserted into a syntactic derivation as a sister of an indefinite \(\text{a (certain)}\) \(\text{girl}\). The reason I do not give the existential logical expression as in (12g) to the indefinite article

---

4Jacobson’s variable-free framework is linked to the direct mapping between a phonological string and a logical form that represents model theoretic meaning without assuming an extra level of representation like LF. It also enables us to get rid of variable assignment functions out of the model theory. I do not commit myself to the obviation of LF, but the variable free framework is preferable for a compositional derivation of a logical form solely based on the information encoded with the lexical items.

5TV (for a transitive verb) is used for notational convenience only. It is not an extra syntactic category. See Dowty (1988: 165).
is that an indefinite noun phrase can be interpreted non-existentially, like as predicate in a copula construction or as generic.

Whether we should associate the inherent argument slot with a (certain) or some* depends partly on whether we can get the dependent specific reading in a non-argument position as well.

(13) a. Every boy mistakenly believed that Mary was a certain woman.
     b. Every boy mistakenly believed Mary to be a certain woman.

Can we pick out a different woman for each boy in these predicative positions? Though the judgment is subtle, I understand that the identity of the woman can co-vary with each boy. So I associate an inherent argument position with a (certain) rather than with some*.

(12f) and (12g) are for the subject quantifier and the object quantifier respectively. In the lexicon, these should be represented as a uniform set of three entries for both the positions. I do not show how I can do this, but assume that these specific entries can somehow be derived from a uniform entry using an under-specified category and a logical form.

The expression a certain in (12e) is of the semantic type (et)(τ(et)) and the encoded logical expression sg' is a constant. It maps an input set to the corresponding singleton set, adding an extra argument slot u of type τ. The type τ is some basic semantic type. τ is normally instantiated as type e, but I keep it under-specified, so that the τ slot can be filled out by an expression denoting a tense argument or an event or a world (/situation) argument. The expression sg'(girl') is of type (τ(et)) and this denotes a function that maps an entity u to a singleton girl set for u. The singleton girl set can co-vary with u, but the function denoted by sg'(girl') itself is the same function for all the boys, so we cannot simply map u to whichever singleton set it is for each boy. This explains why one fixed relation has to hold between each boy and the specific girl for him even though we can pick out a different specific girl for each boy in (10).

The extra argument slot u corresponds to the superscript U in N^U / R^N in the syntactic category. U should normally be NP, but it might also be a category for an expression denoting a tense or a world.

The indefinite article a on its own is of type (et)(τ(et)) and has an inherent argument slot u, but unlike sg', the expression a' does not assign a singleton requirement to the input set. Only when the context pragmatically restricts the domain to a singleton, the expression a'(woman')(u) is interpreted as a singleton, relativized to an element u.

Some explanation is required for the syntactic category with a superscript N^U. Jacobson (1999) assumes that a pronoun like he or she has the semantic type (ge): λx.x, denoting an identity function from individuals to individuals. An expression containing this pronoun as its part inherits the underlined argument slot e till a later stage of derivation. The syntactic

---

6I assume that an indefinite in a predicative position denotes a set of entities without a quantificational force. This means that the phonologically null item some* is lacking in this position.

7See Steedman 2000: 71, as an example of an under-specified category.

8This might extend the definition of a constant, when we consider that the hearer usually does not know what function the constant sg' represents in the world. However, once we accept a higher order constant like every', which denotes a function from sets of individuals to sets of sets of individuals, I do not think that it is out of the question to assume a constant function that maps sets to singleton sets.

9For example, sg'(girl') might be instantiated as the function that maps each boy to his girl friend, though this does not have to be the case.
category of a pronoun is $NP^N$ and the syntactic category of an expression containing a pronoun is $XP^N$. $XP^N$ normally behaves just like a standard $XP$ when it is merged with another expression. For example, the category $XP^N$ cannot be merged with the category $NP$ directly, even if the semantic types match. This superscript category is inherited till a later stage of derivation by Jacobson’s geach rule:

\[ \text{(14) a. Syntax: } g(B/A) = B^C/A^C. \]

\[ \text{b. Semantics: If } f \text{ is a function of type } (a,b) \text{ then } g(f) \text{ is a function of type } ((c,a),(c,b)), \text{ where } g(f) = \lambda V_{(c,a)}[\lambda C_c. [f(V(C))]_0]. \text{ (Jacobson 1999: 138)} \]

Jacobson keeps the superscript category under-specified as $C$, which is normally instantiated as $NP$. Because I uniformly defined verbs as argument of quantificational noun phrases (QNP), I modify Jacobson’s geach rule $g$, so that it can be applied to QNP.

\[ \text{(15) a. Syntax: } g^q((X/R(X/LNP))/RNP)) = X^C/R(X/LNP))/RNP^C, \]

\[ g^q((X/L(X/RNP))/RNP)) = X^C/L(X/RNP))/RNP^C, \]

$X$ is either $S/(L\ldots)$ or $S/(R\ldots)$, and $C$ is some category.

\[ \text{b. Semantics: } g^q(\lambda A_{et}, \lambda P_{e1\cdot\cdot\cdot(en,t)}, \lambda x_{e1}\cdot\cdot\cdot\lambda x_{en}. \exists x_e [A(x) \land P^n(x)(x^1)\ldots(x^n)]) = \lambda A^1_{\tau(e)}, \lambda P_{e(e)}. \lambda v_{e}, \lambda x_{e1}\cdot\cdot\cdot\lambda x_{en}. \exists x_e [A^1(x)(v) \land P^n(x)(x^1)\ldots(x^n)] \]

(*$e1\cdot\cdot\cdot en$ and $x^1\cdot\cdot\cdot x^n$ can be lacking in $P$, as in the case of a subject QNP.)

For example, *some* in the object position is mapped to the following:

\[
\text{Syntax: } g^q(((S/LNP)/LTV))/RN) = ((S/LNP)^C/LTV))/RNP^C, \\
\text{Semantics: } g^q(\lambda A_{et}, \lambda P_{e(e)}, \\text{amboy}_e. \exists x_e [A(x) \land P(x)(y)]) = \lambda A^1_{\tau(e)}, \lambda P_{e(e)}, \lambda v_{e}, \exists x_e [A^1(x)(v) \land P(x)(y)].
\]

This quantifier entry percolates an extra argument slot in its first input argument of the category $C$ across a verb category (e.g., $TV$) onto the output category, when the QNP is merged with this verbal category. In this respect, even though I percolate the extra argument slot of the nominal restriction through the quantificational determiner category, the operation still preserves Jacobson’s original $g$ operator, which compositionally transmits this argument slot through the $TV$ category. Notice that the superscript category $C$ does not appear on argument $TV$; the superscript passes through this $TV$ argument.

Actually, I could have used Jacobson’s original $g$ in (14b) both on the verb and the quantificational determiner, coupled with a type lifting on the verb so that the verb can take in the QNP that has an extra argument slot as an argument, while percolating this extra argument slot to the output of the function application. In this way, I could also have used Jacobson’s original $z$ operator I explain later on the verb category without modifying it. However, the original intuition behind the inherent argument slot $C$ is that the category $XP^C$, which carries the superscript category $C$, behaves exactly like the category $XP$ in its merging possibilities with another category.

---

10A recursive geach rule is required to combine a function containing a pronoun with an argument containing another pronoun, like combining his teacher with loves her husband in John said that [his\_ teacher\_2 loves her\_ husband]. In my treatment of the indefinite, this corresponds to a sentence like Every boy who hates a certain woman will have a certain problem. I do not deal with a complex example like that in this paper.
except for the operations required to fix the binding/dependency relation between the lexical item that introduces this C category and the category that acts as the binder of this extra argument position. g and z operators are used specifically for fixing the binding/dependency relation, so it is architecturally understandable that the existence of an inherent argument position triggers the use of these operators. But using a type lifting to change the argument - functor relation between the verb and QNP just because of this C position seems a little odd to me. On the other hand, my modifications of the g operator above and the z operator below do not fundamentally change the original definitions of these operators. The basic idea of fixing the binding relation between the subject position and the object position through the mediating verb is preserved in my definition. In that sense, the modified operators are just applicational variants of the original operators.\footnote{11}

The superscript category C can technically be any syntactic category but in this paper, I limit it to a category that is originated with some lexical item as superscript, like NP in N\(^N\)P with a pronoun he or U in N\(^U\)/R\(_N\) with the indefinite a (certain). Correspondingly, I limit the under-specified semantic type for an extra-argument slot to \(\tau\), (which is for the variables \(v\) and \(u\) in this paper). That is, an extra argument slot is always of type \(\tau\), which is usually instantiated as type e.

With the modified geach rule in (15b), the object QNP can then be merged with a normal transitive verb category and carry the extra argument slot over until the VP level category gets merged with the subject QNP. I re-formulate Jacobson\’s binding operator z accordingly.

\begin{align*}
(16) & \quad \text{a. Syntax: } z^g(S/R(S/L\mathbf{NP})) \equiv S/R(S/L\mathbf{NP})^U. \\
& \quad \text{b. Semantics: } z^g((\mathbf{NP})/(\mathbf{NP})) \overset{\text{def}}{=} \lambda Q(\mathbf{ET}), \lambda R^1(\mathbf{ET}), Q(\lambda x. R^1(x)(x)). \\
& \quad \quad \text{e.g. we can get } [\lambda R^1(\mathbf{ET}). \forall x [A_{\text{et}}(x) \rightarrow R^1(x)(x)]] \text{ as an output.} \\
& \quad \text{c. Syntax: } z^0((\mathbf{NP})/(\mathbf{NP})) \equiv (\mathbf{NP})/R\mathbf{NP}^N. \\
& \quad \text{d. Semantics: } z^0((\mathbf{NP})/(\mathbf{NP})) \overset{\text{def}}{=} \lambda R^1(\mathbf{ET}), \lambda f_{\mathbf{ee}}, \lambda x_{\mathbf{ee}}, R^1(f(x))(x). \quad \text{(cf. Jacobson 1999:132).} \\
\end{align*}

(16c) and (16d) are Jacobson\’s original z, which is applied to a verb category. \(z^0\) is for a transitive verb when there is one bound pronoun in the object NP. In the quantificational analysis presented here, the (ee) function: \(f_{\mathbf{ee}}\) is not used at any stage of derivation. So I modify z for the subject QNP\footnote{12} as in (16a) and (16b). When this \(z^0\) operator is applied to a subject QNP, the result can only be merged with a VP as its argument that carries an extra e (or \(\tau\) slot associated with the superscript category NP (or U). This means there is either a bound pronoun or an indefinite a (certain) in the nominal restriction of the object (Q)NP.

I show a derivation for (11): Every boy loves a certain girl.\footnote{13} I show the syntactic derivation first.

\footnote{11}{An alternative analysis is to type-lift a transitive verb entry whenever the internal argument is a QNP even if the object QNP has no extra argument slot. Then we can use Jacobson\’s g and z as they are.}

\footnote{12}{In order to allow the first object to bind a pronoun in the following object position in a dis-transitive verb construction, I would need to define z for an object QNP as well, while still disallowing an object QNP to bind a pronoun in the subject. The current definition correctly prohibits the subject quantifier from binding a pronoun in its own nominal restriction through z operation.}

\footnote{13}{I do not show the derivation of the phonological form.}
Syntax:
\[
\text{Every boy} \quad D \quad \overset{\text{loves}}{\frac{g'(\text{some}^*)}{\text{a certain girl}}} \quad \text{loves} \quad \text{a certain girl}
\]

D at the end of the top-left line means that I have omitted the derivation from the lexical level up to that level. \(g^q\) and \(z^q\) signifies the use of the two operators I explained. \(f_a\) is a forward application and \(b_a\) is a backward application. On the last line, the under-specified category \(U\) is instantiated as \(NP\), and so the concatenation is successful. Next, I show the semantic derivation up to the object QNP.

Semantics:
\[
\lambda A_{r(\text{et})}, \lambda P_{e(\text{et})}, \lambda \nu, \lambda x. \exists y[A(v)(y) & P(y)(x)] \quad [\lambda P_{e(\text{et})}, \lambda \nu, \lambda x. \exists y[s g'_e(\text{et}))(g i r'_l)(v)(y) & P(y)(x)]](\text{et}(\text{et}))(\nu(\text{et}))(\lambda x(\text{et}))
\]

The under-specified type \(\tau\) corresponds the syntactic type \(U\). Notice the inherent argument position with \textit{a certain girl} is inherited as \(v\) after the concatenation with the phonologically null existential quantifier, because of the geach rule \(g^q\).

Next, we merge this result with a transitive verb \textit{loves}, ignoring the tense.

Semantics:
\[
\text{loves} \quad \frac{g'(\text{some}^*)}{\text{a certain girl}} \quad \overset{\text{loves a certain girl}}{\text{loves}}
\]

Lastly, we let the subject quantifier bind both the \(v\) and \(x\) positions, by using \(z^q\) rule.

Semantics:
\[
\text{Every boy} \quad D \quad \overset{\text{loves a certain girl}}{\frac{\lambda B_{\nu(\text{et})}, \forall m[\text{boy}')(m) \rightarrow B(m)(m)]}{\forall m[\text{boy}')(m) \rightarrow (\exists y[(\text{sg'}(g i r'_l))((v)(y) & \text{love'}(y)(x))]])(m)(m)]](\text{et}(\text{et}))(\nu(\text{et})](\nu(\text{et})))}
\]

\[
\forall m[\text{boy}')(m) \rightarrow (\exists y[(\text{sg'}(g i r'_l))((v)(y) & \text{love'}(y)(x))])))((m)(m)](\text{et}(\text{et}))(\nu(\text{et}))(\lambda x(\text{et})]
\]

\[
\frac{\forall m[\text{boy}')(m) \rightarrow B(m)(m)]}{\lambda B_{\nu(\text{et})}, \forall m[\text{boy}')(m) \rightarrow B(m)(m)]}
\]

\[
\frac{\lambda B_{\nu(\text{et})}, \forall m[\text{boy}')(m) \rightarrow B(m)(m)]}{\forall m[\text{boy}')(m) \rightarrow (\exists y[(\text{sg'}(g i r'_l))((v)(y) & \text{love'}(y)(x))])))((m)(m)](\text{et}(\text{et}))(\nu(\text{et})](\nu(\text{et})))}
\]

262
The $\beta$ reduced logical form in the bottom line says that for each boy $m$, there is a possibly different singleton girl set, and $m$ loves the singleton member $y$ of that set. Notice $m$ in the expression $(\text{sg'(girl')})(m)(y)$ makes the singleton domain restriction dependent on the universal quantifier $\forall m$.

4 Summary

In this paper, I explained the meaning of an indefinite in an argument position of a verb. I adopted Schwarzschild's domain restriction analysis. When the domain of an indefinite nominal restriction set is restricted into a singleton set, we get the impression that the utterance is about a specific individual. But this specific individual can co-vary with some other element in the sentence. In order to derive the intermediate scope reading and the functional reading of the indefinite, I argued that the phrase a (certain) girl has an extra argument selection of an under-specified semantic type $\tau$. If this slot is bound by a universal quantifier in every boy, the domain is restricted in a different way for each boy, which leads to a relativized specific reading.

By using Jacobson's $g$ and $z$ operators, I showed how this extra argument slot of an indefinite can compositionally transmitted through a syntactic derivation and then get bound by another element in the sentence without using a free variable at any stage of the derivation.

The exact definition of a logical constant at the level of cognitive linguistic meaning and the issue of exactly how $g$ and $z$ operators should be used to deal with an indefinite are left for further research.
References

Stable Categories as a Basis for Concepts

MATTHIAS WEISGERBER
Konstanz University, Germany
matthias.weisgerber@uni-koeln.de

Abstract.
Categorization is a basic tool for storing and communicating information. In human categorization, there are different kinds of categories – starting from perceptual categories, up to highly abstract ones. A special kind of categories are ‘ad-hoc categories’ (Barsalou 1983), which are only constructed to fulfil one special goal and are not stored. We call these categories ‘unstable’ – and, after exploring how (in)stability of a category should be defined, propose a formal quantitative measure. The framework we choose is Prototype Theory.

1 Introduction

"Imagine there is no categorization ...” – many books and articles start with a scenario like this. Always with the same intention: to recall that categorization is a fundamental tool wherever information has to be dealt with. Recognizing patterns and objects (like humans, wild beasts, faces of our friends), and situations, as well as communicating using a language or simply finding something to eat, are built on one ability: the ability to simplify information by grouping it into parts which ‘belong together’ in some way.

However, categorization as a technique to simplify amounts of data by grouping would be of no use for communication if every ‘categorization device’ (let it be a human brain, a computer, an AI creature) categorizes with its own rules – despite lots of effort spent in a ‘good’ and efficient outcome. With other words, categorization needs an aspect that provides a common ground for all those who want to share and communicate information: Despite the ability to spontaneously create new categorizations, participants of a communication must rely on common parts shared by their categorization systems.

Human categorization, together with human language, is a wonderful example of such a system. There are several kinds of categories, starting with perceptual categories, which are claimed to be hard-wired in the human brain (take as examples the perception of colours, or the auditory system, especially the architecture of the cochlea; both have been widely addressed in literature), semantic categories (where only the semantic description justifies belonging together of objects), up to very abstract categories. And, there are categories that are only constructed
for one special purpose: for the reason to fulfil goals (e.g. the category THINGS TO CLIMB ON IN ORDER TO CHANGE A LIGHT BULB), for which Barsalou (1983) chose the term ad-hoc-categories.

In this paper, we want to address this last kind of categories, since they are candidates for categories, that are not common to all participants. As a categorization framework, we will use Prototype Theory (PT), which we will introduce in a short way. After that, we will start out from an informal definition of what we will call stability and discuss some examples. In order to obtain a formal measure of stability, we present some measures concerning PT proposed in last years and examine their use for measuring stability.

1.1 Prototype categories

The question of how human brains categorize has been addressed in numerous theories and experiments – and is still ‘under investigation’. Some time ago, the problem seemed to be solved: In the Aristotelian view (Smith and Medin (1981) called it the ‘classical view’), an object either clearly belongs or clearly belongs not to a category – a decision which is made via a number of defining features.

The view has changed – or, at least, has been broadened.\(^1\) With the beginning of cognitive science in the 20th century (see Gardner (1985), for example, for a description of the birth and growth of cognitive science) the classical view got into trouble: there have been more and more examples of things that should get a place in a category despite having or despite having not certain features.\(^2\) Eleanor Rosch and her team’s work in the 1970s\(^3\) finally laid the groundwork for a new view of categories and concepts: the Prototype Theory. Members of a category are grouped round a prototype; the existence of common features which are shared by all members of a category is no longer a necessary feature. Directly from that follow the basic properties of prototype categories.

Only this ‘fundamental compromise’ provides categorization with the ability to deal with reality: Given the great variation of things in our world and the need to represent them in a relatively small number of concepts which can be efficiently handled by our brains, we need a mapping between things and classes of things, that can only be done with a fuzzy architecture of categories and concepts (see Murphy (2002) for a discussion).

1.2 Stable categories?

Prototype Theory’s architecture brings about a whole bundle of effects concerning categorization. First, categorization fundamentally depends on measuring similarity to a prototype. Subjects’ similarity ratings, which form a basis for measuring, have been shown to vary with age, expertise, knowledge, context, and many other factors. However, for the purpose of this discussion, we will focus on the concept of stability, which is defined as the consistency of a category’s membership across different contexts.

1Note that despite our argumentation in the following we do not want to totally abolish the classical view.

2One of the first mentioned examples for a semantic category is the category BIRD and it’s member ‘penguin’, which does not fly. As a consequence, ‘fly’ can either not be taken as a defining feature of BIRD or the definition of ‘defining feature’ has to be changed.

3See Rosch (1975, Rosch et al. (1976), Mervis and Rosch (1981) and many other papers by Rosch and her team; see Smith and Medin (1981) for a description of that ‘downfall of the classical view’; see Murphy (2002) for another detailed overview.

4Following Murphy (2002, p. 5), we define concepts as mental representations of classes of things (where things are all kinds of entities, including abstract ones) and categories as the classes themselves.
environment, method of presentation, cultural context, and many other aspects (see Goldstone (1994) for a review). Another class of effects supports the view that categorization depends on factors other than similarity, like theories (as systems of knowledge) and background knowledge (Goldstone 1994). Furthermore, there are categories that are dependent on context (Barsalou 1982; Barsalou 1987).

Finally, there is a class of categories that are constructed dependent on goals and perspectives – they are constructed only for the reason to fulfil goals: THINGS TO CLIMB ON IN ORDER TO CHANGE A LIGHT BULB and THINGS TO BE CARRIED OUT OF A BURNING HOUSE. Barsalou (1983) calls them ‘ad-hoc-categories’. They allow only few inferences and are not memorized in long-term memory.

Categorization may be unstable due to several reasons. However, given that humans have concepts in mind which make them able to talk about things and abstract entities to other humans which (normally) understand them, there must be concepts represented in our mind which are independent from contextual and other influences and must last over time without change. Although categories may be constructed in a ‘spontaneous’ fashion, there must be a stable core in categorization. To find a measure which enables us to decide how stable categories are is our plan in the following. In order to find such a measure, we give an overview of measures proposed in categorization literature, and evaluate their limits. Finally, we discuss our own suggestion to measure stability together with a small empirical test.

2 Trying to catch (in)stability

In this section, we first will get an idea what instability is, then start searching for a measure by systematically looking at the properties of Prototype Theory and some measures that have been proposed within prototype theory so far. In the next section, then, we will propose our own measure.

2.1 A closer look on instability

In order to define which features or effects make a category stable or unstable, respectively, we will discuss some examples and derive, step by step, a definition of stability.

Example 1. Berlin and Kay (1969) interviewed and tested speakers for 20 languages to show that the perceptual category of COLOURS is a prototype category. It turned out that speakers of the same language almost agreed on ‘focal colours’, which form the prototypes, and produce variation only at the boundaries. Only the number of focal colour terms varies across languages and cultures.

This (famous) experiment was one of the first experiments to show prototypical effects in perceptual categories. We call that categories stable, since different speakers agree on the prototypes and only vary in boundary cases (perceptual categories are claimed to be hard-wired in brain, indeed), and we call the effect speaker independence.

We do not mention learning here, that could be defined here informally as a 'slow and linear change in a concept's structure'.

267
Example 2. Labov (1973) presented pictures of ‘small containers’ to students, which had to classify them in one of the categories CUP, BOWL and VASE. Clear cases were classified consistently, whereas vases or bowls with a handle produced inconsistent answers. When one of the tasks ‘imagine the object with coffee in it’ or ‘... with flowers in it’ or ‘... standing in the shelf’ was added, more objects were classified as cups or vases, respectively. Labov concluded that the categories CUP, BOWL and VASE are organized in a prototypical way, where a prototypical cup is as wide as high, has a bowl and is used for drinking hot drinks, while prototypical vases are higher and smaller and to be used for putting flowers in. Unclear cases lead to inconsistent answers, that shows the fuzziness of the borders of between the categories.

In this example, we have a triple of prototype categories with fuzzy boundaries. One of the prototype features (to be filled with coffee or flowers, resp.) is a ‘goal’ feature. A change of the situation in which the objects are presented changes the actual classification of unclear cases – the prototype remains unchanged, however. Also, if we would ask a subject to describe a ‘cup’ or a ‘vase’, we would obtain a thing looking like a prototypical cup, or vase. We observe that changes of the situation have only influence on the classification of unclear cases, but not on the prototype. The second observation is that the membership decision is a systematic one; since subjects agree on the description of the prototype, they are able to systematically categorize new objects. We call the categories stable, and we call the first observed effect situation independence, the second systematic membership decision.

Example 3. Consider now the category INN SIGN. In our opinion, objects like inn signs are interesting cases; the category is very restricted, the objects are constructed to fulfil one goal – but, nevertheless, we claim the category INN SIGN to be both a prototypical category and stable: A prototype may for example be defined by the goal feature ‘hangs over the entrance of an inn’, by the type ‘is a sign’ and by a description of the prototypical shape. The category can be called stable, since it fulfils all of the effects discussed so long: it is speaker independent, situation independent, the membership decision can be performed systematically. Since the features that describe a prototypical inn sign are quite exclusive (nothing else hangs over inn entrances so often), we want to add the effect feature exclusiveness.

Despite the effects are ‘only’ derived from examples, we judge to be basic effects able to define stability – since the examples we chose are representative of whole classes of similar examples. We now define

Definition 3 (Stability)
The stability of category is a gradual feature. The more of the following claims hold, the more stable is a category A:

- **Speaker independence.** Speakers agree on the description of the prototype. They only vary in category boundary cases.

- **Situation independence.** Changes of situation have only influence on the classification of unclear cases, not on the description of the prototype itself.

- **Systematic membership decision.** The decision of category membership can be performed systematically, for example via checking features or via comparison to a prototype.
• Feature exclusiveness. A category is described by features that are exclusive for that category.

• Feature Description and Prototype. There is a prototype that can be described via features.

Example 4. To test the definition, consider two categories that we want to call unstable. We will check the effects listed in the definition. We choose two of the ad-hoc categories mentioned by Barsalou (1983), quoted above and repeated here: Things to Climb on in order to change a light bulb and Things to be Carried out of a Burning House.

To carry out of a burning house, one would choose —things that are lying around in the way to the exit or —precious things that are small enough to carry or —personal things that one can never replace by buying them anew. To climb on in order to change a light bulb, one would choose anything that —is in a reachable distance, —has a suitable height and a flat top and —is stable enough to climb on. One could choose ladders, chairs, boxes, a stack of old phonebooks, or even a horse or an elephant (if available).

We see, the categories do not fulfil the features of stability: (1) they are not speaker independent, everybody would choose category members in a different way; (2) they are not situation independent; (3) there are little systematic features to describe the prototype; (4) there are no exclusive features that define the category and contrast categories; (5) there is no feature description at all, since the category is only constructed to fulfil a goal in a special situation. It follows that the categories are not stable.

To conclude: We have gained an impression of how a category must look like that we may call it stable or unstable, respectively. Let us go one step further now: We will search for a suitable quantitative measure for category stability.

2.2 (In)stability and basic properties of prototype structure

• All features Graded structure, Fuzzy boundaries and Graded membership are orthogonal to stability. Example: The category of Blue squares is fuzzy (since blue is a vague term), however, it is pretty stable (due to our definition).6

• Prototypes as best examples and reference points and Category membership is a matter of similarity to the prototype support a stability decision: There are prototypical members that are consistently considered the best examples of the category. If there are objects which are similar to that prototype (according to a suitable measure7), than they altogether form a stable category.

• No set of necessary conditions and Family resemblance are orthogonal to stability.

6We want to thank the reviewer for this example important for clarity.
7Goldstone (1994) assigns similarity an important but not exclusive place in categorization: On the one hand, it has been shown by many researchers that similarity is not sufficient for categorization, on the other hand, it “does play an important role in establishing many of our categories” (p. 151f). A suitable measure is a measure (in a suitable architecture) that accounts for all kind of features and correlations
2.3 (In)stability and common measures

Category utility / feature predictability. Corter and Gluck (1992) propose the category utility measure, which they claim to be "a normative justification for the existence of basic levels on the basis of an account of the usefulness of categories to the categorizer [p. 292]" (see also Gluck and Corter (1985)). However, although their measure offers a metric for evaluating the "goodness of categories [p. 300]", it is orthogonal to the stability measure we are looking for.

Naturalness. Berlin and Kay (1969) and Rosch (1973) give the impression that cultures do not differ (too much) in what they call a natural concept. Osherson (1978) calls the concepts humans favour to organize their experience ‘natural’. He offers three conditions on the naturalness of concepts (one of them is the predictability tree). Gärdenfors (2000) correlates naturalness with convexity of the geometrical space representing the concept – hence, he does not provide formal advice how to implement concept representations. However, naturalness and stability seem to be orthogonal: a stable category, that is not natural, would be the evidence.

Cue Validity (CV) and category validity (catV). Rosch et al. (1976) introduced the notion of the basic level categories of categorization (the level which cognitively operates faster than higher or lower levels; the highest level at which category members are similar with respect to features like ‘shape’). They proposed Cue Validity (the conditional probability of an object of being in a category \( \mathcal{A} \), given that it possesses some feature \( \alpha \)) as a measure for finding basic-level categories (Rosch: CV becomes maximal there). Later, formulas to calculate CV have been proposed (see, for example, Blutner (1985)) – but Murphy (1982) pointed out that “cue validity of a single feature increases (or, at best, remains the same) with category inclusiveness (p. 175)”. Category validity (the conditional property of an object having some feature \( \alpha \) given that it is in some category \( \mathcal{A} \)) suffers from the same effect in the opposite direction, as Medin (1983) pointed out.

Hence, both CV and catV lost their only application, picking out basic levels. – However, it will turn out in the next chapter that CV can be turned into a measure for a stability (given a discrete features architecture).

3 A model

Murphy (1982) suggested a new and better use for Cue Validity (CV): “Cue Validity might be roughly proportional to certainty of classification [p.177, emphasized by Murphy]. Let us recall, on the other hand, our definition of stability in chapter 2.1 on page 269. The definition can be 'condensed' into two main features: stability: the category is not bound to one explicit context; instability: the category is constructed within one explicit context, often goal oriented, and there are too less features that are in contrast with neighbouring categories. The last one exactly reminds of ‘certainty of classification’ – what due to Murphy can be calculated via CV. Therefore, in the rest of the paper, we develop a formalism to calculate the stability of a category

---

8Cruse contradicts the assumptions needed for tessellation since around category borders, there was happening more than a linear decrease of similarity to one prototype and a linear increase of similarity to a different prototype (personal communication and (Croft and Cruse 2003).

9Definition quoted from Corter and Gluck (1992), who give a good and short overview.
via its CV. This already defines the strategy we will have to use: Based on empirical data, which we obtain in a Rosch-like test, we will define membership in a category, subjective and average probability for a feature to be empirically mentioned for an object, the Cue Validity (CV) of a feature as its relevance for the member / not member decision. Following from that, we will finally define stability of a category via the CVs of its members.

3.1 Getting empirical data ‘to play with’

In order to have a data set to ‘play with’ and to formalize in the end, we did exactly what Rosch did: ask subjects. In order to be sure not to overestimate data (especially since we use a rather small set of data), we will not use it to formulate hypotheses about certain categories – we will only formulate formal ways to deal with category judgements, and give some examples.

In order to test which are the typical features that subjects assign to objects we chose three well known animal categories: DOG, BIRD, and INSECTS. The method of testing is directly derived from a characteristic property of prototypes found by Rosch: Prototypes are those objects which subjects mention before other members of a category. Therefore, the subjects (students at Konstanz University) of the first test are instructed to write down all members of a category they can think of. For each round of 30 seconds, subjects get exactly 60 seconds, followed by a break of 30 seconds. After that, the 6 objects mentioned most were chosen out of each category. Only these were used as input material for the second test, where subjects were instructed to list features of the object they saw mentioned on paper. Again, subjects get 60 seconds in each round. The test has been performed in German language, using German object- and category names.

3.2 Formalizing

Obviously, features differ systematically in the respect of ‘how good’ they are to assign an object to a category. Consider as an example the features $\alpha =$ ‘has four legs’ and $\beta =$ ‘makes barking sounds’. The feature $\alpha$ is true for quite many physical objects – therefore it will be of less use to categorize than $\beta$. We will now formulate a theory that calculates for a single feature the centrality of its role is in categorizing. Let us start with a definition:

Definition 4 (subjective probability)

Let $|\alpha(a)|_e$ be the number of mentioning the feature $\alpha$ for an object $a$ (where $| \ldots |_e$ is an empirical measure); let $|VP(a)|$ the number of subjects that have been asked about object $a$. The probability for mentioning the feature $\alpha$ for object $a$ by subjects is then calculated as

$$\text{prob}(\alpha, a) = \frac{|\alpha(a)|_e}{|VP(a)|}.$$

Additional, let be $n(A)$ the number of objects in the category $A$. Then the subjective probability for feature $\alpha$ in the category $A$ is

$$\text{prob}(\alpha, A) = \frac{\sum_{i=1}^{n(A)} \text{prob}(\alpha, a_i)}{n(A)}.$$  \hspace{1cm} (25.1)
Example 5.
prob\(\text{fliegen, Amsel}\) = \(3/6 = 0.50\), prob\(\text{fliegen, Bussard}\) = \(0/3 = 0.00\),
prob\(\text{fliegen, Specht}\) = \(3/4 = 0.75\), prob\(\text{fliegen, Spatz}\) = \(5/6 \approx 0.83\),
prob\(\text{fliegen, Vogel} = \frac{1+3/6+3/6+4+3/4+5/6}{9} \approx 0.60\),
prob\(\text{bellen, HUND} = \frac{2+3/2+3/2+2+2+2+2+2+1/6}{9} \approx 0.37\). …

Note that, in this test scenario, ‘fliegen’ (fly) is more often mentioned for Vogel (bird) than
‘bellen’ (bark) is mentioned for Hund (dog). However, ‘bellen’ is a feature that (almost) only
occurs with dogs, while ‘fliegen’ occurs with many physical things like insects, aircrafts, and
whatever. Exactly this is the contribution of the sum in the next formula, where the Cue Validity
\((CV)\) of a feature will be defined:

Definition 5 \((CV)\) Let be \(A_1 \ldots A_k\) the categories mentioned. The \(CV\) of a feature \(a\) for a
category \(A_i\) increases with \(\text{prob}(a, A_i)\) and decreases with the probabilities of the contrast categories
\(A_j\) \((i \neq j)\):

\[
CV(a, A_i) = \frac{\text{prob}(a, A_i)}{\sum_{j=1}^{k}\text{prob}(a, A_j)}.
\] (25.2)

Example 6.

\[
CV(\text{fliegen, Vogel}) \approx \frac{0.60}{0.60 + 0 + 0.38} \approx 0.51,
\]

\[
CV(\text{bellen, Hund}) \approx \frac{0.38}{0 + 0.38 + 0} = 1.
\]

Interpretation: as discussed above, a bird with the feature ‘fliegen’ is less distinctively described
as a dog via the feature ‘bellen’ (bark), because not only birds have the feature ‘fly’.

Using the CVs of the features, we are finally able to define the stability of a category:

Definition 6 \((\text{stability of a category})\) Let \(m(A)\) be the number of features that appear in cat-
gory \(A\). The \(\text{stability of a category} A\) is calculated as the sum of the CVs of the \(m(A)\) features
divided by the number of features:

\[
\text{stab}(\mathcal{A}) = \frac{\sum_{i=1}^{m(\mathcal{A})} CV(A_i, \mathcal{A})}{m(\mathcal{A})}.
\] (25.3)

Interpretation: the stability of a category increases with the CV of the defining features.

\(^{10}\)Because \(\text{prob}(a, A_i)\) of the own category \(A_i\) is included in the division, the denominator cannot become
0. The outcomes are between 0 and 1.

272
3.3 Measuring instability

Let us, finally, try this formalization for one of the discussed unstable categories: THINGS TO CLIMB ON TO CHANGE A LIGHT BULB. Consider, we choose some typical objects out of that category. The Cue Validity of ‘solid’ for that category will be calculated as

\[
CV(\text{solid, thingsToClimbOn...}) = \frac{\text{prob}(\text{solid, thingsToClimbOn...})}{\sum_{j=1}^{b} \text{prob}(\alpha, A_j) \ (\text{all other categories } A_j, i \neq j)}
\]

(25.4)

which gives a low CV value, since there are many solid things. That means that the stability \(\text{stab(THINGS TO CLIMB ON...)}\) will have a low value. That fits with our argumentation, that the mentioned category is not stable.

3.4 Discussion

The main aspect of what we did in this section is not that we obtained some data – but that we found an affirmation for our suggestion that CV can be used in order to get a quantitative measure for category stability as defined above. There are some weak points left in the formalization section, some of which shall be discussed here:

When Murphy (1982) critically discusses CV, he raises three objections against Rosch-like empirical testing: 1. “Attributes of categories may often be unknown, vague, or logically dependent on one another”. The problem remains – but it only concerns obtaining data in an empirical way. Indeed, our formalization uses this kind of data here; but it does not depend on it. It can be made working in different applications, in ontologies, for example, were features are given and the ‘world’ is finite. 2. “One would need unobtainable quantitative facts about many object categories”. The problem remains. It thus can be solved for cases were it is sufficient to estimate (in)stability, since it can be guessed if there are few or many things around in the world that have a certain feature – as we did in the last example. Again, inside an ontology, where all features of all objects are known, the problem dissolves. 3. There is an influence of “experiences and knowledge of individual speakers as well as cultural knowledge”. This, indeed, is a systematic effect of categorizing: since categorization as such depends on personal knowledge and on culture, why should that not be reflected in the calculation of the stability of a category? If there is a people where everybody is used to climb on things in order to change light bulbs (or, take experts like a concierge of an University as a better example), then the mentioned category is a stable category for that community. The formula will come to exactly the same claim.

Nevertheless, the method can be improved. Where we ask for features of objects and then sum up to obtain features of a category, we could directly ask for features of that category. We decided for the way we used because we think it is more explicit to let the subjects first decide for prototypical exemplars and then deal with the features of these exemplars. Indeed, since we cut the list by a time limit, we get prototypical exemplars due to Rosch’s claim ‘prototypical exemplars are mentioned first’. However, the other way would be technically the same.

4 Conclusion

This paper’s main focus is stability of categories and the question how Barsalou’s ad-hoc categories fit into the picture of categorization. Discussing examples, we managed to define stability in a
way that is compatible with Barsalou's notion of ad-hoc categories. Choosing Prototype Theory as a framework and discrete features as an architecture, we finally proposed a stability measure based on Cue Validity.

However, since it is not the formalization of the measure alone but the notion of (in)stability as such that forms the centre of this paper, the formalization could have gone other ways. There are other (and, perhaps, better) architectures and frameworks, like geometrical models (for geometrical frameworks, it should be possible to find a correlation between the possibility of partitioning the feature space, called Voronoi Tessellation, and the stability of the represented category), attribute-value models, frame models, models that are able to deal with categories defined by goal or with highly complicated relations between events (as it is the case for certain verbs).

Further work can be to carry the notion of (in)stability into many areas of representation, modelling and implementation, thereby improving the theory and getting more and more insight into the phenomenon of unstable ad-hoc categories.

Acknowledgements

I want to thank Prof. Peter Pause and Wilhelm Geuder (both Konstanz University) for numerous discussions about the role of categorization in linguistics and its correlation to semantics and lexicalization. I want to thank the anonymous reviewers for very useful hints, and the proceedings' editors for their work and patience. Finally I want to thank the 'Sonderforschungsbereich SFB 471' for financial support.
References

Using Generalized Linear Regression to Train a Novel Sentence Detection System

YANG YE
University of Michigan, Ann Arbor, MI 48109, USA
yye@umich.edu

ABSTRACT.
The current work seeks to build a system of novel sentence detection using Generalized Linear Regression Model. Based on the assumption that the novelty of a sentence can be predicted by several different features and the relation between the novelty and the features is parametric, we regressed the novelty score of a sentence on four features and trained a novelty classifier on 50 pairs of training data from news-streams. The evaluation was done on 30 pairs of testing data, which showed a general precision of 72.4% with an increase of 37.9% from the baseline, and an expected distance\(^1\) between the true value and system prediction of 0.858 with a reduction of 0.555 from the baseline.

1 Introduction
Novelty detection is an issue that has begun to catch researchers’ attention in natural language processing during the past decade. Given the newness of this research topic, the scope of the literature on novelty detection is fairly small. It has been recognized that this is closely related to many other tasks in natural language processing, information retrieval and information summarization. One of the first investigations of novelty detection goes back to a report from a summer workshop at the Center of Language and Speech Processing at Johns Hopkins University. It addressed the Topic-based Novelty Detection that is triggered by a document-clustering task. It attempted to investigate First Story Detection (FSD) (Allan, Jin, Rajman, Wayne, Gild, Lavrenko, Hoberman, and Caputo 1999) and New Information Detection but ended up focusing on the former one. The methodology was the classical Vector-Space approach, where the determination of novelty of the story is at the level of lexical similarity metrics, although some stemming had been done to reduce single words to their root forms and words appearing on a stop-word list were removed. After that, novelty detection had been investigated under the paradigm of

\(^1\)Here “expected distance between the true value and system prediction” means how far is the system prediction from the true value. We use this to evaluate the accuracy degree of the system.
adaptive information filtering (Zhang, Callan, and Minka 2002), where the methodologies include Set Difference, Geometric Distance, Distributional Similarity and a Mixture Model, all of which use word-frequency patterns to access the novelty of the documents. Novelty detection as an independent research area was roughly equivalent to lexical mapping and was rarely done on the sentence level. However, novelty detection on the sentence level interests researchers of summarization, especially multiple-document summarization, where the most important sentences are expected to be identified and included in the summaries. There is a novelty reranker in summarization systems (Radev, Teufel, Saggers, Lam, Blitzer, Qi, Celebi, Liu, and Drabek 2003) that prevents a highly-similar sentence to previous sentences from being included in the resulting summaries. But the novelty reranker follows the tradition and only takes care of lexical similarities. To conclude, novel sentence detection has not been looked at as an independent problem and "novelty" is regarded (methodologically-speaking) as lexical novelty. This paper aims to go beyond the lexical level by incorporating the predicate-argument structures in sentence novelty detection.

2 Novel Sentence Detection

With novelty, the concerns are usually facts in the real world hence novelty detection is fundamentally a semantic problem. Lexicon is the most direct hence the easiest way to access the facts of the text; lexical statistics are also easier to formalize and compute. Measuring the overlapping of lexicons has been used by researchers because of its convenience. The work in this paper aims to push novelty detection toward the semantic structure of the text. We focus on novel sentence detection firstly because it could be directly used to improve the performance of many tasks in NLP. The second reason that we chose to investigate sentence novelty is because sentence is the most appropriate level to look at semantics.

3 Potential Applications of Current Work

Many tasks in NLP could benefit from current work, among which Information Summarization and Question Answering are the most immediate ones. In Information Summarization, especially in Multi-Document Summarization (MDS) where a set of related documents on the same topic is converted to a short summary containing the most salient facts on the topic, a novelty reranker, in addition to other rerankers, will improve the summaries by reducing the redundancy in the summaries. In Question Answering, novelty detection will help with fact tracking in evolving texts. Novel sentence detection could also be used to detect plagiarism effectively.

4 Capturing Sentence Novelty through Features

Ideally, a novel sentence detection system should be able to handle sentences paraphrasing the same facts. But the system in this paper only seeks to deal with events with the same phrasing. The task that we address in the paper is defined as follows: building a system that can tell us automatically how novel the incoming sentence is by using several features. The input of the system is a pair of some old sentences and an incoming sentence. The output of the system is the probabilities of the incoming sentence being different scales of novelty (redundant, half redundant
and half novel, slightly novel and novel). There has been no official definition of novelty in the literature. Since the novelty we are evaluating in this paper is aimed to improve the performance of summarization whose concern is real facts, we define that being novel means the facts conveyed by the sentences have not been mentioned before, and we do not separate relevance evaluation and novelty evaluation, namely, sentences that are not relevant are also eligible for novelty evaluation. Facts could be reflected in events and we try to combine event extraction with lexicon-based similarity measuring. A sentence is possibly not novel even when it is using new words and vice versa. The aim of this paper is to integrate linguistic representation into a statistic model of language by building a system where novelty can be both explained and predicted by a statistical model. Novelty evaluation is made for a given pair of inputs, namely, the sentence of interest and the pool of old sentences. While we cannot do full-fledged fact extraction to a sentence at this stage, we hope to include in the model the most relevant features of a sentence that are factual cues. Up to date, lexical overlapping in one dimension is the only intrinsic feature used in novelty evaluation. (Radev et. Al, 2003) uses six features for the Trek novelty evaluation task: Centroid, Length, Position, QueryCosineNoIdf, QueryNarrativeWordOverlap and QueryTitleWordOverlap. Four of these six features are based on lexical overlapping. In the system in current paper, we go beyond the lexical overlapping measure and have a predictor which reaches into the semantic structure of sentences, on the other hand, since we are currently working with sentence by sentence novelty evaluation, we are not interested in Position features.

4.1 The Experimental Features

We tried four features for our novelty evaluation:

- Feature 1: word probability difference between the incoming sentence and the old sentences
- Feature 2: standardized difference of length between the incoming sentence and the old sentences
- Feature 3: overlapping of the nominals in the incoming sentence and the old sentences
- Feature 4: the predicate-argument structure difference between the incoming sentence and the old sentences

The first feature is based on the classical similarity measurement with slight modification. Since the texts we are looking at are not big enough, especially for the incoming sentence which only has a few words, we want to dampen the probability of the words in the sentence to better represent the salience of the word. The idea is that a word which appears 3 times is more important to the content of the sentence than a word appearing just one time, but not as three times as important. Functions like $f(tf) = \sqrt{tf}$ or $f(tf) = 1 + \log(tf)$ can make the term frequencies better capture the importance of the terms. We choose the latter function in our regression model. Once the term frequencies of the words in the sentences are ready, we take the probabilities of the words in the old sentences and of the ones in the incoming sentence as two vectors and use cosine similarity to measure the difference between them (where $Ni$ is the word frequency of a word in the incoming sentence, $Oi$ is the word frequency of the corresponding word in the old sentences, $n$ is the length of the union of the words in the old sentences and the incoming sentence, $Ni$ and $Oi$ is 0 when the word does not appear):
\[ CosineSimilarity = \frac{\sum_{i=1}^{n} N_i \cdot O_i}{(\sum_{i=1}^{n} N_i^2)^{\frac{1}{2}} (\sum_{i=1}^{n} O_i^2)^{\frac{1}{2}}} \]

The second feature is based on the idea that the longer the incoming sentence is, the more possible that it contains new facts that have not been conveyed in the previous sentences. We calculate the standardized difference as follows (where \( Y \) is the length of the incoming sentence, \( \mu \) is the average length of the old sentences, \( X_i \) is the length of each of the old sentences, \( n \) is the number of old sentences):

\[ StandardDeviation = \frac{|Y - \mu|}{\sqrt{\frac{\sum_{i=1}^{n} (X_i - \mu)^2}{n}}} \]

Our third feature looks at the overlapping of the nouns in the old sentences and the incoming sentence. We see the fact that the nominals greatly represents the semantic aboutness of a text and the degree of overlapping between two texts reflects the semantic redundancy between them. In order to edit out the effect caused by the original length difference between the old sentences and the incoming one, we normalize the overlapping by the ratio of the nouns in the old sentences and the nouns in the incoming sentence. Hence, for the old sentences, we find out all of the nouns, and then for each of them, we look up in wordnet and group all of the corresponding synonyms, hyponyms and hypernyms with it\(^2\), consequently, we have a number of groups with each of them packing up a series of nouns. Then for each of the nouns in the incoming sentence, we search through these groups, keep track of the number of nouns that do not have match in the groups of nouns from the old sentences, the ratio of this number over the total number of nouns in the incoming sentence will be the final value of the predictor.

The fourth feature endeavors to capture the semantics of the sentences. In order to calculate the predicate-argument structure difference between the incoming sentence and the old sentences, we need to extract the predicate-argument structures of a sentence, and we rely on a naive First Order Predicate Calculus (FOPC) to realize this. An atomic formula of FOPC is structured as:

\[ \text{Predicate symbol } (term_1, term_2, \ldots, term_n) \]

Atomic formulas can be connected by logical connectives to become molecular formulas. Currently, we do not use a full-fledged FOPC representation, the reason is that the dependency tree representation we rely on to extract the predicate structure does not provide ample temporal and logical information, which will be discussed later in the paper. The fourth predictor, as a new attempt on top of the previous endeavors in novelty detection features, will be discussed in more details in next session.

4.2 Extracting Predicate-Argument Structure of Sentences

To extract the predicate-argument structure of a sentence, we need a tree representation which can tell us about the dependency structure of the sentence. Pilot efforts of tree-to-tree mappings have been made in Generative Machine Translation community which can convert the output of a

\(^2\)This way we essentially treat a noun and its more specific concept as being redundant, the same holds for a noun and its more general concept.
syntactic parser to a shallow semantic representation tree. In this conversion, a constituent-based tree such as a syntactic tree can be transformed to a dependency-based tree through a process called "headification"\(^3\). The precision of this conversion based on the Wall Street Journal data shows that incorrectly attached nodes vary from 5.1% to 9.4% and incorrectly assigned functors vary from 16.0% to 22.6%\(^4\). For our extraction of the predicate-argument structure, we start from the input raw sentence and through several steps advance to the dependency tree representation\(^5\) from where we extract the predicate-argument structure of a sentence. The dependency tree representation is of the Prague Dependency Tree bank (PDT) (Böhmová, Hajič, Hajicová, and Hladká 2001) format. The tectogrammatical level of annotation is a key intermediate step in this general flow of FOPC generation. The functors in the tectogrammatical tree indicate the thematic roles of the corresponding nodes and hence the cues to extract the predicate-argument structures. This process is illustrated in figure 1.1:

![Diagram](image)

**Figure 26.1: Flowchart of the fourth predictor**

### 4.3 Generating Predicate-argument Structures for Sentences

We use a single script from the 2002 workshop at Johns Hopkins University to convert Charniak parser output to dependency tree representation. We see that the generated tree structures have many problems the main ones of which are summarized as follows:

---

3. This headification process was done by Jason Eisner at Johns Hopkins University.

4. This is based on the difference between automatically and manually generated trees of 5 files from Wall Street Journal corpus.

5. Since that script was designed for conversions from PennTreebank annotation, which also contains functional markup (i.e. suffixes like -SBJ, -TMP, -LOC), conversions from the Charniak Parser are not very reliable.
The distinction between verb arguments and adjuncts is murky and information is lost because this dependency tree representation does not provide a solution to different subtypes of PP; tense is not represented in a consistent way; the "BE" verb is not represented correctly. While we have no control over the errors occurring in syntactic parsing and have not yet tried to tackle the mechanism of conversion from Penn Treebank tree to Dependency Tree, we can make use of the accurate information in the Dependency Tree representation and extract the predicate-argument structures of the events. The extraction of FOPC representation from the Dependency Tree structures involves two steps. The first is the recognition and extraction of all the verbs in the sentence which is essentially events recognition. The second step is filling into the argument positions of the verbs. We wrote 12 ad hoc rules to extract predicate-argument structures and convert all of the sentences in the training data to the format of predicate-argument structures as shown in the following example:

<s> Leading German dailies from across the political spectrum are unanimous in their view that the showing by Chancellor Gerhard Schroeder’s Social Democrats in Sunday’s state elections in Lower Saxony and Hesse constitutes a landmark defeat. </s>

sentence 1 (VBZ): ISA(e,constituteing) constituteee(e,defeat)

The script of the FOPC extraction extracts the predicate structure with a 90% precision at identifying the predicates and around 60% at identifying the arguments. Other than Prague Textogrammmatics, there are two other resources with similar endeavor of pursuing predicate-argument structure: Propbank and Framenet. The automatic labeling accuracy of Propbank is 57.7% for precision and 50% for recall and Framenet similar. We think given the room for improvement for the conversion from syntax tree to PDT, the current pursuit is more promising.

5 Training the system

The method that we use to train the system is Proportional Odd Linear Regression Model. There are two reasons for this choice: proportional-odds model meets the categorical and ordinal nature of the response; the probability outputs give users the flexibility to make cut-off probability if they have special needs of the novelty degree. The link function is a logistic link: \( p = \frac{e^\theta}{1+e^\theta} \), the model outputs the cumulative probabilities up to and including a response category, in our system, the different novelty degrees. For the 50 pairs of training data, we ended up merging absolute redundancy and slight redundancy as one category since the corresponding data points are too sparse. The inter-judge agreement for the novelty score of the training data is about 80%. We also trained a model with only the first three predictors as a backup model.

---

6 This murkiness is reflected in the low precision of the identification of the arguments of the predicates in our extraction of the fourth predictor.

7 The 60% precision of identifying arguments is not good, but fortunately it should not distort our model too much because in most cases, the scoring is ended at the predicate identification level. This relatively low precision is greatly owing to the inaccuracy of the dependency tree representation as discussed earlier in the paper.
6 Evaluation

For the 30 pairs of testing data, we asked three human judges to judge the novelty degree. The inter-judge agreement for the test data is 96.6%. We took the novelty degree that is agreed by two or three judges as the true value. Since proportional-odds linear regression model outputs the probabilities of the different scales of novelty degree of the sentence. In order to evaluate the system, we took the novelty degree with highest probability to be the final system output. This automatically causes some degree of uncertainty. So while we evaluated the final system output, we also calculated the distance between the system prediction and the true value of final system outputs. Whenever the fourth predictor is not available, we fall back to the smaller model for the novelty evaluation. Hence the evaluation is based on the combined model. The baseline of the general precision of novelty degree judgments is set to be judging all of the testing pairs to be the most frequent novelty degree in the training data, which in our training data is pure novel. The baseline's precision is 37.9%. The system yields a 72.4% general precision of novelty degree evaluation with an increase of 34.5%. The baseline of the average distance between the system prediction and the true value is set to be a uniform probability distribution with a 0.25 probability for each of the four categories. The baseline is 1.41379. Our system yields a distance of 0.858 with a reduction of 0.555 from the baseline. A system without the fourth predictor only yields a general precision of 41.4% indicating that the system benefits an increase of 31% from including the fourth predictor.

7 Discussion and Future Work

We see that in the system output, there are quite a number of sets with relatively uniform probabilities. We believe this is because our FOPC predictor is still very naive at this point. It awaits to be more finely developed to include temporal and logical resolutions. Also it needs to be pointed out that right now the t value of the FOPC mismatch is not significant yet, I believe that there are three reasons for this: firstly, the training data size is not big enough; secondly, the dependency tree representation is not accurate enough hence the extracted predicate structures are not yet the accurate representations of the predicate structures of the sentences; thirdly, the effect of this predictor might vary across different types of languages. English is more noun-oriented, hence the predicate structure is not as representative as it is in some verb-oriented languages (e.g. Chinese).

Sentence novelty detection, while mixing human knowledge, the language world and the real world, and hence having the reputation of being notoriously difficult to keep track of, is solvable as long as we can give an appropriate definition to the problem depending on what we need. For the task of novelty evaluation, there are four things that need to be defined in a very clear way before the task could be addressed and appropriately solved: firstly, a clear definition of novelty is needed; secondly, the reference of the novelty evaluation should be explicitly defined; thirdly, the most relevant predictors that are responsible for a sentence's novelty should be identified; lastly, the truth value of novelty should be able to be found for evaluation, which means that we should provide the human judges with a clearly-defined scheme.
References
