5th International Workshop on Constraints and Language Processing

Jørgen Villadsen
Henning Christiansen

ESSLLI 2008
20th European Summer School in Logic, Language and Information
4–15 August 2008
Freie und Hansestadt Hamburg, Germany

Programme Committee. Enrico Franconi (Bolzano, Italy), Petra Hendriks (Groningen, The Netherlands), Michael Kaminski (Haifa, Israel), Benedikt Löwe (Amsterdam, The Netherlands & Hamburg, Germany) Massimo Poesio (Colchester, United Kingdom), Philippe Schlenker (Los Angeles CA, United States of America), Khalil Sima'an (Amsterdam, The Netherlands), Rineke Verbrugge (**Chair**, Groningen, The Netherlands).

Organizing Committee. Stefan Bold (Bonn, Germany), Hannah König (Hamburg, Germany), Benedikt Löwe (**chair**, Amsterdam, The Netherlands & Hamburg, Germany), Sanchit Saraf (Kanpur, India), Sara Uckelman (Amsterdam, The Netherlands), Hans van Ditmarsch (**chair**, Otago, New Zealand & Toulouse, France), Peter van Ormondt (Amsterdam, The Netherlands).

`http://www.illc.uva.nl/ESSLLI2008/`
`esslli2008@science.uva.nl`

INSTITUTE FOR LOGIC,
LANGUAGE AND COMPUTATION

ESSLLI 2008 is organized by the Universität Hamburg under the auspices of the *Association for Logic, Language and Information* (FoLLI). The *Institute for Logic, Language and Computation* (ILLC) of the *Universiteit van Amsterdam* is providing important infrastructural support. Within the Universität Hamburg, ESSLLI 2008 is sponsored by the Departments *Informatik*, *Mathematik*, *Philosophie*, and *Sprache, Literatur, Medien I*, the *Fakultät für Mathematik, Informatik und Naturwissenschaften*, the *Zentrum für Sprachwissenschaft*, and the *Regionales Rechenzentrum*. ESSLLI 2008 is an event of the *Jahr der Mathematik 2008*. Further sponsors include the *Deutsche Forschungsgemeinschaft* (DFG), the Marie Curie Research Training Site GLoRiClass, the European Chapter of the Association for Computational Linguistics, the *Hamburgische Wissenschaftliche Stiftung*, the Kurt Gödel Society, Sun Microsystems, the Association for Symbolic Logic (ASL), and the European Association for Theoretical Computer Science (EATCS). The official airline of ESSLLI 2008 is Lufthansa; the book prize of the student session is sponsored by *Springer Verlag*.

Jørgen Villadsen and Henning Christiansen

5th International Workshop on Constraints and Language Processing

Proceedings. 20th European Summer School in Logic, Language and Information (ESSLLI 2008), Freie und Hansestadt Hamburg, Germany, 4–15 August 2008

Proceedings of the 5th International Workshop on

Constraints and Language Processing

(CSLP 2008)

Jørgen Villadsen
Henning Christiansen
(Editors)

MAY 2008 ROSKILDE UNIVERSITY COMPUTER SCIENCE RESEARCH REPORT #122

Computer Science
Roskilde University
P. O. Box 260
DK–4000 Roskilde
Denmark

Telephone:	+45 4674 3839
Telefax:	+45 4674 3072
Internet:	http://www.ruc.dk/dat_en/
E-mail:	datalogi@ruc.dk

ISSN 0109–9779

This research report constitutes the proceedings of the 5th International Workshop on Constraints and Language Processing (CSLP 2008) which is part of the European Summer School in Logic, Language, and Information (ESSLLI 2008), Hamburg, Germany, August 2008.

Research reports are available electronically from:

http://www.ruc.dk/dat_en/research/reports/

Part of ESSLLI 2008
European Summer School in Logic, Language, and Information
Hamburg, Germany, 4–15 August 2008

5th International Workshop, CSLP 2008

Constraints and Language Processing

Hamburg, Germany, 11–15 August 2008

Proceedings

Editors:

Jørgen Villadsen
Henning Christiansen

Preface

The present volume contains the proceedings of CSLP 2008, the 5th International Workshop on Constraints and Language Processing, which takes place in Freie und Hansestadt Hamburg, Germany, 11–15 August 2008, as part of the European Summer School in Logic, Language, and Information (ESSLLI 2008), 4–15 August 2008. We want to thank the organizers of ESSLLI 2008, especially Rineke Verbrugge and Benedikt Löwe, for hosting the workshop.

The CSLP 2008 workshop addresses the question of constraints and language processing from an interdisciplinary perspective. Constraints are widely used in linguistics, computer science, and psychology. How they are used, however, varies widely according to the research domain: natural language processing, knowledge representation, cognitive modelling, problem solving mechanisms, etc. The purpose is to pursue a paradigm, unifying the different approaches into a common framework capable of explaining how constraints play a role in representing, processing and acquiring linguistic information, and this from a formal, technical, and cognitive perspective. The topics include, but are not limited to, constraint-based linguistic theories, constraints in human language comprehension and production, context modelling and discourse interpretation, acquisition of constraints, probabilistic constraint-based reasoning, constraint satisfaction technologies and constraint logic programming.

We are honoured to present our invited speakers, Helen de Hoop (Radboud University Nijmegen, Netherlands), who will talk on speaker's and hearer's constraints on object fronting, and Gérard Huet (INRIA, France), who will talk on a syntax-semantics interface for Sanskrit using constraint processing of semantic roles.

This volume contains papers accepted for the workshop based on an open call, and each paper has been reviewed by three or four members of the program committee. As editors, we want to thank the other members of the organizing committee, Philippe Blache and Veronica Dahl, whose involvement has been important for the establishment of the forum around the CSLP workshops.

Selected and extended papers will be included in a volume of Studies in Computational Intelligence published by Springer. This will involve a separate reviewing round. Revised papers from the 1st CSLP were published as Springer Lecture Notes in Artificial Intelligence (volume 3438).

We want to thank the program committee listed below, the invited speakers, and all researchers who submitted papers to the workshop and all participants in the CSLP workshops 2004 (Roskilde, Denmark), 2005 (Sitges, Spain; with ICLP), 2006 (Sydney, Australia; with COLING/ACL) and 2007 (once again in Roskilde; with CONTEXT).

We expect to continue the CSLP series, which to us is a stimulating research forum concerning an important, interdisciplinary field, possibly collocating with central conferences for an increased exchange of knowledge.

The workshop is supported by the CONTROL project, CONstraint based Tools for RObust Language processing, funded by the Danish Natural Science Research Council.

Lyngby / Roskilde, May 2008

Jørgen Villadsen
Henning Christiansen

Organizing Committee

Philippe Blache
Henning Christiansen, co-chair
Veronica Dahl
Jørgen Villadsen, co-chair

Program Committee

Philippe Blache (Provence University, France)
Henning Christiansen (Roskilde University, Denmark), co-chair
Veronica Dahl (Simon Fraser University, Canada)
Denys Duchier (University of Orléans, France)
John Gallagher (Roskilde University, Denmark)
Claire Gardent (CNRS/LORIA, Nancy, France)
Barbara Hemforth (Provence University, France)
Jerry Hobbs (University of Southern California, USA)
M. Dolores Jiménez-López (Tarragona, Spain)
Lars Konieczny (Freiburg University, Germany)
Shalom Lappin (King's College, UK)
Detmar Meurers (Ohio State University, USA)
Véronique Moriceau (LIMSI-CNRS, Orsay, France)
Gerald Penn (University of Toronto, Canada)
Kiril Simov (Bulgarian Academy of Sciences, Bulgaria)
Jørgen Villadsen (Technical University of Denmark), co-chair
Eric Villemonte de la Clergerie (INRIA, France)

Contents

Speaker's and Hearer's Constraints on Object Fronting

Helen de Hoop

Radboud University Nijmegen
Netherlands

Abstract. Communication in general requires a process for (a) producing an optimal form given a certain meaning, and recovering that meaning given the optimal form, and (b) arriving at an optimal interpretation given a certain form, and reproducing that form given the optimal interpretation (Blutner et al. 2006). Hence, optimal communication involves more than the sum of two unidirectional processes of optimization. I will discuss this hypothesis for the case of object fronting in Dutch. Object fronting is rare but grammatical in Dutch. Usually, object fronting is considered to be a type of topicalisation. Therefore, we expect object fronting to occur when the object is the topic of the sentence, and since animate noun phrases are better topics than inanimate noun phrases, we can expect object fronting to happen more often when the object is animate. This would explain the speaker's tendency to start a sentence with an animate noun phrase, irrespective of its grammatical function. But from the hearer's perspective a fronted object is in fact more easily recognizable as an object when it is inanimate. I will argue that indeed both the speaker's and the hearer's perspectives constrain object fronting, and that not only does the speaker take into account the hearer's perspective, but also the other way around.

A Syntax-Semantics Interface for Sanskrit Using Constraint Processing of Semantic Roles

Gérard Huet

INRIA
France

Abstract not available at time of printing.

Introduction to Constraints and Language Processing

Henning Christiansen

Research group PLIS: Programming, Logic and Intelligent Systems
Department of Communication, Business and Information Technologies
Roskilde University, P.O.Box 260, DK-4000 Roskilde, Denmark
E-mail: henning@ruc.dk

Abstract. Interpretation of language, by human or machine, can be seen as a process of resolving a bunch of constraints, expressing syntactic, semantic and pragmatic properties. We (as humans) process many levels in parallel: a lexical ambiguity, for example, may be resolved using constraints coming from the discourse context, which again is produced from semantic and pragmatic considerations which presupposes the lexical analysis. Contextual constraints learned so far delimit the interpretations of the continued discourse.

Ideally, computerized models of interpretation should behave in the same way, and constraint solving techniques, especially constraint logic programming seem to have potentials for this. These technologies permit declarative language specification at a high level of abstraction, which ideally may be similar or at least inspired by to theories developed by linguistic and psycholinguistic researchers.

While this goal may be a naive future dream, we show how models based on constraint logic programming can be used to describe and analyze context-sensitive linguistic phenomena in fairly straightforward ways. We show examples from our own research based on Constraint Handling Rules and Prolog provide a natural ways of using abduction and other hypotheses-based principles in interpretation. No deep knowledge or experience in logic programming is assumed from the audience.

References

1. H. Christiansen. CHR Grammars. *Int'l Journal on Theory and Practice of Logic Programming*, 5(4-5):467–501, 2005.
2. H. Christiansen and V. Dahl. HYPROLOG: A new logic programming language with assumptions and abduction. In M. Gabbrielli and G. Gupta, editors, *ICLP*, volume 3668 of *Lecture Notes in Computer Science*, pages 159–173. Springer, 2005.
3. H. Christiansen and V. Dahl. Meaning in Context. In A. Dey, B. Kokinov, D. Leake, and R. Turner, editors, *Proceedings of Fifth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-05)*, volume 3554 of *Lecture Notes in Artificial Intelligence*, pages 97–111, 2005.
4. H. Christiansen, C. T. Have, and K. Tveitane. From use cases to UML class diagrams using logic grammars and constraints. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, and N. Nikolov, editors, *RANLP 2007, International Conference: Recent*

Advances in Natural Language Processing: Proceedings, pages 128–132. Shoumen, Bulgaria: INCOMA Ltd, 2007.
5. H. Christiansen, C. T. Have, and K. Tveitane. Reasoning about use cases using logic grammars and constraints. In H. Christiansen and J. Villadsen, editors, *Proceedings of the 4th International Workshop on Constraints and Language Processing, CSLP 2007*, volume 113 of *Computer Science Research Report*, pages 40–52. Roskilde University, 2007.
6. Henning Christiansen. Executable specifications for hypotheses-based reasoning with Prolog and Constraint Handling Rules. *Journal of Applied Logic*; to appear, 2008.

Acknowledgement: This work is supported by the CONTROL project, funded by Danish Natural Science Research Council.

A Quantification Model of Grammaticality

Philippe Blache[1] and Jean-Philippe Prost[1 2]

[1] Laboratoire Parole & Langage, CNRS, Université de Provence,
29 Avenue Robert Schuman, 13621 Aix-en-Provence, France
blache@lpl-aix.fr
[2] Centre for Language Technology, Macquarie University
Sydney NSW 2109, Australia
jean-philippe.prost@lpl-aix.fr

Abstract. The traditional binary notion of *grammaticality* is more and more often replaced by intermediate levels of acceptability, also called *gradience*. This paper aims to provide a numerical account of syntactic gradience. It introduces and investigates a numerical model with which acceptability can be predicted by factors derivable from the output of a parser. Its performance is compared to other experiments, and the fit of each model is evaluated. Our model shows a good correlation with human judgement of acceptability.

1 Introduction

Grammaticality is not binary (see [Chomsky75]) but rather a scale phenomenon: some sentences are judged as more grammatical than some others. This question has been recently described in terms of *gradience* ([Pullum01], [Fanselow06], [Aarts07]) and replaces *grammaticality* by intermediate levels of acceptability. Gradience phenomena have different sources. One is the violation of rules or constraints of a given language. These rules playing a more or less important role in the linguistic structure, it is possible to grade them, as it is done in probabilistic approaches (see [Manning99]) or, in a more theoretical perspective, in the Optimality Theory ([Prince93]). As a side effect, grading the syntactic properties makes it possible to explain the differences in acceptability judgements by human. For example, the repetition of a determiner as in (1a) can be considered as less problematic than a problem of word order as in (1b):

(1) a. *Buildings burn in the the Kenyan town of Eldoret.*
    b. *The adopted board the regulations.*

Another source of gradience in the evaluation of grammaticality comes from the syntactic structure itself. A same constraint violation is more or less acceptable according to its context or its location in the structure. For example, (2b) seems to be more acceptable than (2a). In both cases, the same linearity constraint is violated. The difference comes from the fact that the constituent affected by the violation is more deeply embedded in (2b) than in (2a). In the second sentence, the acceptability judgment is more affected by parsing difficulty than constraint violation.

(2) a. *Paul phoned father his.*
b. *Paul came back into the house that was built by father his.*

This example illustrates the fact that a violation can be, to a certain extent, balanced by the context, or more generally by information born by other domains, including prosody or pragmatics.

In this paper we propose a model for syntactic gradience,which enables quantifying the phenomenon. This paper is organised in three parts: we first propose to make precise the notion of gradience, its theoretical description and what are the needs for its quantification. In the second section we introduce a model, which is a refinement of the one described in [Blache06a]. In the last part, we validate the model using a parser's output. The results are evaluated first by comparison with those of a psycholinguistic experiment, then over a large corpus of unrestricted text.

2 Linear Optimality Theory

Several experiments have shown the effect of constraint violation on grammaticality. In particular, preliminary works by [Legendre90] (anticipating Optimality Theory) have proposed to quantify this effect. More recently, [Keller00] explored this question through in-depth description of different constructions. This work showed in particular the cumulativity effect, that led to the *Linearity Hypothesis* indicating that the "*grammaticality of a structure is proportional to the weighted sum of the contraint violations it incurs*". Starting from this hypothesis, Keller elaborates the *Linear Optimality Theory* (see [Keller06]). Linearity makes it possible to give to the notion of harmony a particular definition: the harmony of a structure it the opposite of the sum of the violated constraint weights. The grammaticality of different structures can be then evaluated and compared. The following figure, taken from [Keller06], illustrate the process. It shows 4 different candidate structures (S_1 to S_4) and three different constraints (C_1, C_2, C_3), with the respective weights 4, 3, and 1. Constraint violations are indicated by *. The figure gives the harmony of the different structures.

	C_1	C_2	C_3	
Structure	4	3	1	*Harmony*
S_1		*	*	-4
S_2		*	**	-5
S_3			*	-1
S_4	*			-4

Fig. 1. Structures, constraint violations and harmony

This formal examples illustrates the cumulativity effect: the multiple constraint violation of the structure S_2 makes its harmony worst than that of S_4

in spite of the fact that this last structure violates a strongest constraint. Moreover, the harmony values show that S_1 and S_4 are at the same level in terms of violation weights, in spite of the fact that they violate different types and number of constraints.

The structure hierarchy is then $S_3 > \{S_1, S_4\} > S_2$ (where the hierarchy in classical OT would be $S_3 > S_1 > S_2 > S_4$). Several experiments (see [Keller00]) has shown the validity of the linearity hypothesis.

However, several questions can not be addressed in this framework. First, this view of cumulativity is purely quantitative and does not take into consideration the structure itself. Concretely, it would not predict any acceptability variation between (2a) and (2b).

Second, OT was designed in order to compare several structures by means of constraint violation. Subsequently, two structures violating the same constraint can not be discriminated. The following example illustrates this phenomenon. Both *NP* violate the same constraint (gender agreement violation between the noun and the past participle). However, the same violation embedded into a relative clause in (3b) renders the NP more acceptable than (3b).

(3) a. *La maison détruit par l'explosion*
The house-fem destroy-PPast-masc by the explosion
b. *La maison qui a été détruit pas l'explosion*
The house-fem that has been destroy-PPast-masc by the explosion

Last, and more importantly, it is important to compare utterances that do not violate any constraint without being at the same level in terms of acceptability. In the following examples, the different sentences do not bring the same quantity of information. The different constructions instantiate differently the direct object. The first sentence is a cleft, there is a strict an unambiguous interpretation of the *NP* cleft as a direct object, the function being marked by the accusative mark of the relative pronoun "*que*". The example (5b) is a dislocation with a resumptive accusative pronoun "*la*", possibly referring to the dislocated *NP*. Finally, the last sentence is also a kind of dislocation, without any coreference phenomenon, the status of the extracted *NP* being ambiguous (direct object, vocative, etc.)

(4) a. *C'est Marie que je supporte pas.*
It is Mary that-Acc I can't stand.
b. *Marie je la supporte pas*
Mary I can't stand her.
c. *Marie je supporte pas.*
Mary I can't stand.

This example are graded: the first example is unambiguous because of the weight of its syntactic information: several morpho-syntactic and syntactic constraints are satisfied there. The second example contains less information, it is ambiguous (dislocation vs. vocative interpretation). In this case, the only visible relation concerns the possible agreement between the NP and the embedded clitic. Finally, the last example is even more ambiguous: Marie can receive a vocative, an accusative or even a dative interpretation there, without any morphology or syntactic mark.

These last examples show the necessity of taking into account, on top of the violated constraints, also the satisfied one: the quantity and the weight of satisfied constraints can play a role, as the violated on, in the ranking process.

3 Property Grammars

The Property Grammars (PG) approach [Blache05a] is purely constraint-based: all syntactic information is represented by means of constraints; no external device such as the *Gen* function in OT or the generation of the dependency network in Constraint Dependency Grammar [Maruyama90] is required. Several constraint types are used: *Constituency, Uniqueness, Precedence, Obligation, Requirement* and *Exclusion.* They can be specified in terms of constraints over directed graphs, as presented in figure (2).

- $Const(A, B) : (\forall x, y)[(A(x) \wedge B(y) \rightarrow x \lhd y]$
Classical definition of constituency, represented by the dominance relation indicating that a category B is constituent of A.

- $Uniq(A) : (\forall x, y)[A(x) \wedge A(y) \rightarrow x \approx y]$
If one node of category A is realized, there cannot exists other nodes with the same category A. Uniqueness stipulates constituents that cannot be repeated in a given construction.

- $Prec(A, B) : (\forall x, y)[(A(x) \wedge B(y) \rightarrow y \not\prec x)]$
This is the linear precedence relation: if the nodes x and y are realized, then y cannot precedes x

- $Oblig(A) : (\exists x)(\forall y)[A(x) \wedge A(y) \rightarrow x \approx y]$
There exists a node x of category A and there is no other node y of the same category. An obligatory category is realized exactly once.

- $Req(A, B) : (\forall x, y)[A(x) \rightarrow B(y)]$
If a node x of category A is realized, a node y of category B has too. This relation implements cooccurrence restrictions.

- $Excl(A, B) : (\forall x)(\nexists y)[A(x) \wedge B(y)]$
When x exists, there cannot exist a sibling y. This is the exclusion relation between two constituents.

Fig. 2. *Constraint types in PG*

The following example illustrates some constraints describing the *NP* and the *AP* (in which a dependency constraint has been added):

`Cx_NP` : *Prec(Det, N)* ∧ *Oblig(N)* ∧ *Req(N, Det)* ∧ *Excl(N, Pro)* ∧ *Dep(Det, N)* ∧ *Dep(AP, N)*

`Cx_AP` : *Prec(Adv, Adj)* $\wedge$ *Oblig(Adj)* $\wedge$ *Dep(Adv, Adj)*

A grammar is then a set of constraints. Parsing an input consists of evaluating this constraint system for a given assignment (i.e. the set of categories corresponding to the input words). The outcome is a description of the input, which is made of the set of the evaluated constraints. Depending on the form of the input this set may contain both satisfied and violated constraints.

In PG constraints thus play a double role (as it is the case in constraint programming): they can rule out structures as well as instantiate values. Such an approach is well-suited to modeling gradience because the constraints are independent from one another. Indeed the basic assumptions in PG, unlike in OT, stipulate that constraints are *unranked*, *local* and *violable*:

Unranked: A limited set of constraint types is used, each one with its own operational semantics. A constraint bears information of a unique type, representing a single and homogeneous piece. A constraint is thus atomic and can be evaluated independently from the other ones. Subsequently, by being mutually independent and never specified with respect to others. The weighting mechanism used in PG is not an order relation over the set of constraints, but a property specific to each constraint.

Violable: Allowing constraint violation is a pre-requisite when dealing with partial or non-canonical inputs. In PG, all constraints are violable, but not necessarily. In OT, as in *Weighted CDG* (see [Schröder02]), the number and the type of constraint violations is used to compare two structures. Violation is then necessary, and no optimal candidate can be selected when all constraints are satisfied. In PG, constraint violation is not required; it only introduces flexibility.

Local: Universality is not a required property of PG. On the contrary, all constraints are local to a construction. It is an important difference with OT, not only theoretically, but also in the way of designing and using constraints. As indicated above, OT can only compare two structures with respect to constraint violation. It explains the fact that constraints must be specified in a very general and imperative way: the more general a constraint is, the more frequently it is violated. Universality must then be understood in OT as a mechanism which favors constraint violation. In PG, no universality is required; constraints can be specified at any level.

4 A Computational Model for Syntactic Gradience

The existing accounts of gradience previously mentioned rely on several properties:

- *Constraint violation*: this is a pre-requisite. Constraints must be be defeasible in order to describe any kind of input, whatever its form.
- *Constraint weighting*: a comparison necessarily relies on the possibility of measuring the impact of the different constraints (see [Foth05]).
- *Cumulativity*: the effect of constraint violation was shown to be cumulative [Keller00].

However, these properties do not capture all phenomena that have been presented in the previous section. First, we have to explain the effects of a relative counterbalance between satisfied and violated constraints: constraint violation can be in some cases attenuated by the importance of satisfied constraints. Moreover the form of the syntactic structure (flat or deep) and the location of constraint violation in the structure also have importance (cf. examples 3 and 4). Thus we propose to complete the list of properties needed to describe gradience by the following ones:

- *Constraint counterbalance*: cumulativity must take into account both violated and satisfied constraints.
- *Violation position*: the embedded level of the violation site in the syntactic structure carries consequences on acceptability.

An approach integrating these different properties offers, on top of a precise description of gradience, the possibility of calculating an index for any input, and quantifying, at least partially, its grammaticality. In the next section we detail and organise the basic information on top of which our model is built.

4.1 Basic information

in this section we present, in the form of different postulates, the kind of information required to build a computational model of gradience.

Failure Cumulativity As in other approaches, we postulate that acceptability is impacted by the amount of constraints it violates. We note N_c^- the amount of constraints violated by the constituent c (this factor corresponds to cumulativity in LOT).

Success Cumulativity Gradience is also affected by successful constraints. That is, an utterance acceptability is impacted by the amount of constraints it satisfies. We postulate that some form of interaction between satisfied and violated constraints contributes to a gradient of acceptability. We note N_c^+ the amount of constraints satisfied by the constituent c, and $E_c = N_c^+ + N_c^-$.

Constraint Weighting We postulate that constraints are weighted according to their influence on acceptability. The question of whether such weights are proportional to the importance of either constraint success or failure is addressed in assuming that a given constraint is of same relative importance either way in absolute value. We note W_c^+ (respectively W_c^-) the sum of the weights assigned to the constraints satisfied (respectively violated) by the constituent c.

A weight may be of different *scope* and *granularity*. The *scope* has to do with how widely a weight applies (to a constraint type or to an individual constraint). *Granularity* concerns the level a weight applies at (the grammar vs. the construction level). Scope and granularity can then be combined in different ways: all constraints from the same type at the grammar level, or all constraints from the same type at the constituent level, or individual constraints at the constituent level, or individual constraints at the grammar level—the difference between the last two possibilities assuming that a same constraint may occur in the specification of more than one category. Although the more fine-grained and the narrower the scope, the more flexible and accurate the influence on gradience, a too fine granularity and a too narrow scope (as in [Schröder02]) are also quite complex to manage. Therefore, we opted for a fair compromise, where the weighting scheme is restricted to the constraint types at the grammar level, which means that all constraints from the same type in the grammar are assigned same weight. For examples, all constraints of linearity (i.e. word order) are weighted 20, all constraints of obligation (i.e. heads) are weighted 10, and so on.

Constructional Density Acceptability is also impacted by the *density* of the constituent structure (*i.e.*, the quantity of information born by the structure). This notion is measured by the amount of constraints specifying a category in the grammar. We note $T_{\mathcal{C}}$ the total amount of constraints specifying the category $\mathcal{C}$ of the constituent c. The underlying idea is to balance constraint violations by the amount of specified constraints: without such a precaution one violation in a rather non-complex construction, such as AP—only specified by 7 constraints in our grammar—would be proportionally much more costly than one violation in a rather complex construction, such as NP—specified by 14 constraints.

Propagation Acceptability also depends on that of its nested constituents. Therefore, we postulate that acceptability is propagated in the constituent structure through the relationship of dominance. We note Z_c the number of nested constituents in c.

Constraint violation affects both realization, but in a lesser extent when the violation is deeply embedded (example 5b). Subsequently the models we investigate are recursive functions of their constituents' score.

4.2 Rating Models

A rating model for gradience aims to place an item along a scale by assigning it a score (rate).

Scoring Terms The various scoring components presented here aim to capture the factors postulated above. Each component is meaningful as such, but not sufficient when considered alone.

Satisfaction/Violation Ratio The *SRatio* ϱ_c^+ (resp. VRatio ϱ_c^-) is defined for the constituent c as follows:

$$\varrho_c^+ = \frac{N_c^+}{E_c} \qquad \varrho_c^- = \frac{N_c^-}{E_c}$$

The SRatio and violation ratio (VRatio) capture the postulates of Success and Failure Cumulativity respectively.

Completeness Index

The *Index of Completeness* for the constituent c of category $\mathcal{C}$ is defined as the following ratio, T being the total number of constraints describing the category in the grammar:

$$\mathcal{E}_c = \frac{E_c}{T_{\mathcal{C}}}$$

This score contributes to implement the postulate of Constructional Density, which suggests that the complexity of a constituent influences its acceptability.

Quality Index

The *Index of Quality* for the constituent c is defined as the following ratio:

$$\mathcal{W}_c = \frac{W^+ - W^-}{W^+ + W^-}$$

The *quality* of a constituent implements the postulate of Constraint Weighting, which suggests that all constraints do not have same importance with respect to acceptability, and therefore must be weighted accordingly.

Precision Index

The *Index of Precision* for the constituent c is defined as the following ratio:

$$\mathcal{P}_c = k \cdot \mathcal{W}_c + l \cdot \varrho_c^+ + m \cdot \mathcal{E}_c$$

These *adjustment coefficients* (k, l, m) are used as variable parameters for tuning up the model.

We observed that the SRatio in use in the Precision score seems to overemphasise the role of success cumulativity, that is, the role of the successful constraints characterising a constituent. Therefore, we define an index of *anti-precision*, where the SRatio term in the precision index is replaced by the VRatio as a negative term. *Anti-Precision Index*

We define the *Index of Anti-Precision* for the constituent c as the following ratio:

$$\widetilde{\mathcal{P}}_c = k \cdot \mathcal{W}_c - l \cdot \varrho_c^- + m \cdot \mathcal{E}_c$$

Compared to the precision score, the anti-precision rather emphasises the factor of Failure Cumulativity.

Rating Functions A rating function combines different scoring terms into a single score. Among the numerous functions investigated, the following ones more particularly draw our attention for the significance of their results. Grammaticality Index

The *Index of Grammaticality* (g) for the constituent c is defined recursively as follows (where c_i is a nested constituent of c):

$$g_c = \mathcal{P}_c \cdot \overline{g_{c_i}} = \mathcal{P}_c \cdot \frac{\sum_{i=1}^{Z_c} g_{c_i}}{Z_c}$$

Next to the g–model we define below a new model, in order to run a comparative investigation of the two. The index of *coherence* is similar to the one of grammaticality, except that it relies on anti-precision rather than precision.

Coherence We define the *Coherence* of a constituent c recursively as follows:

$$\gamma_c = \widetilde{\mathcal{P}}_c \cdot \overline{\gamma_{c_i}} = \widetilde{\mathcal{P}}_c \cdot \frac{\sum_{i=1}^{Z_c} \gamma_{c_i}}{Z_c}$$

A comparison with LOT We have seen that cumulativity makes it possible in LOT to rank the structures with respect to their constraint violation. In the figure (3), we can compare the ranking by LOT with the obtained by our model. We can see that our ranking makes it possible to precise the result of LOT in providing an intermediate ranking between S_1 and S_4. As expected, S_4 satisfying more constraints than S_1, its precision index is higher.

	C_1	C_2	C_3		
Structure	4	3	1	*Harmony*	Precision index
S_1		*	*	-4	0.537
S_2		*	**	-5	0.379
S_3			*	-1	0.842
S_4	*			-4	0.620

Fig. 3. Structures, constraint violations and harmony

The following table recapitulates the rankings obtained by the different models (OT, LOT and GP) :

OT: $S_3 > S_1 > S_2 > S_4$
LOT: $S_3 > \{S_1, S_4\} > S_2$
GP: $S_3 > S_4 > S_1 > S_2$

5 Experimental Validation

We investigate to what extent the models of syntactic gradience presented above fit acceptability judgement by human standards.

5.1 Psycholinguistics experiment

[Blache06a] reports an experiment set up with psycholinguists. It shows a correlation between the Grammaticality Index (GI) (γ–model, above) and acceptability judgements provided by subjects.

The experiment relies on a set of sentences in which constraint violation was controlled. 20 types of sentence were designed, in which at most two constraints are violated. Several base sentences were created, each one generating the 20 types. As a result, 60 sentences from the different types were presented for evaluation to 44 subjects. The subjects were asked to rate the sentences, using *Magnitude Estimation* (see [Bard96]).

Next to this evaluation, GIs were calculated semi-automatically for each sentence: a generic syntactic structure (*i.e.* a syntactic tree) was associated to the phrase types, together with its characterisation (*i.e.* the set of satisfied and violated constraints) of each constituent. Figure (2) shows example sentences along with their GI.

The subjects' judgements were then compared to a rating relying on grammaticality indexes. A very good correlation (coefficient $\rho_1 = 0.76$) was observed between GI and acceptability judgement. An even better correlation with a coefficient $\rho_2 = 0.87$ is reported on a smaller sample of corrected data.

5.2 Computational Validation

For the work we present here we have experimented the two models (g and γ) in using the output from parsers based on PG. We have tested two different parsers, both robust: the first one is a chart parser using dynamic programming [Prost06], and is a direct interpretation of PG. It explores the entire search space, and relies on constraint satisfiability in order to build the set of structures. The second parser [Blache06b] is non-deterministic and relies on control heuristics consisting in selecting construction types by means of precedence and constituency constraints (corresponding to a left-corner like strategy). The parsers using different techniques, strategies and level of analyses, they may build different solutions for the same input (especially due to the non-determinism of one of them).

Both parsers show comparable results with respect to gradience quantification (described in this section). The robust parser has been used in the evaluation over the large corpus (next section). The first evaluation, presented here, consists of replicating automatically the experiment described in the previous section. The goal is to show a correlation between the predictions from the model and the subjects' assessment.

We progressively tune up the models by assigning values to the different parameters (i.e. adjustment coefficients and constraint weights). The problem consists of finding out the right order of magnitude among the different parameters in order to obtain the best possible correlation with the values of acceptability. Different combinations were attempted. A sample of the correlations obtained is reported in table 1. The best correlation ($\rho = 0.5425$) is obtained for record

No violations	
11. Marie *a emprunté un très long chemin pour le retour*	0.465
NP-violations	
21. *Marie a emprunté très long chemin un pour le retour*	-0.643
22. *Marie a emprunté un très long chemin chemin pour le retour*	-0.161
...	
VP-violations	
51. *Marie un très long chemin a emprunté pour le retour*	-0.56 *
54. *Marie emprunté un très long chemin pour le retour*	-0.322 *
...	

Fig. 4. Acceptability Results

♯	Adjust.			Weight						Correlation		
	k	l	m	wl	wo	we	wr	wd	wu	Max	g	γ
8	4	2	1	20	3	5	4	2	10	0.4658	0.3932	♮**0.4658**
11	4	2	1	5	3	2	2	0	2	0.4945	0.3891	♮**0.4945**
12	4	2	1	5	3	2	2	1	2	0.4946	0.3805	♮**0.4946**
17	4	2	1	20	10	5	4	3	2	0.5425	0.4745	♮**0.5425**
										0.5425		

Table 1. Calibration of adjustments and constraint weights. *The weights are those assigned to the different constraint types: Linearity (wl), Obligation (wo), Exclusion (we), Requirement (wr), Dependency (wd), and Uniqueness (wu); col.* ♯ *is a record Id; col.* Max *contains the maximum correlation for each record.*

♯17, for the γ–model. The scatter plot from fig. 5 illustrates how the γ–model fits acceptability judgement.

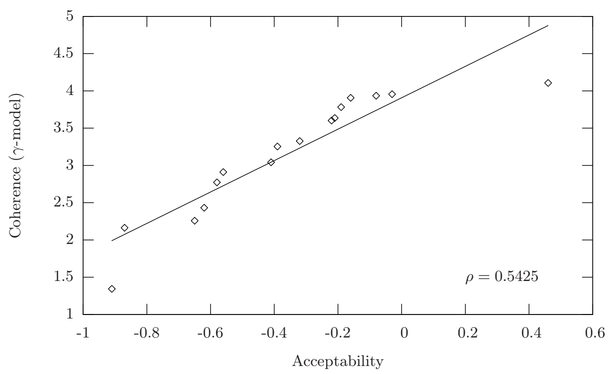


Fig. 5. Correlation Acceptability vs. Coherence

The constraint weights are ranked $wu < wd < wr < we < wo < wl$. It is important to emphasise that unlike in [Keller00], the constraints are not only ranked according to how much unacceptability they entail, but according to how important they are in *absolute value* with respect to acceptability.

Whatever the combination of parameters, γ always outperforms g, which confirms that this latter model is over-emphasising the role success cumulativity compared to the role of failure cumulativity.

The best performing scheme of parameters (rec. ♯17) grants a great deal of importance to Linearity (a factor 10 between $wl = 20$ and the minimum $wu = 2$, and a factor 2 between wl and its next follower $wo = 10$), then to Obligation (a factor 5 between $wo = 10$ and wu, and a factor 2 between wo and it very next follower $we = 5$). Then follow the remaining weights, ranging over $[2 \ldots 5]$. This observation of two constraint types (namely Linearity and Obligation) on one hand, and the other ones on the other hand, tends to confirm the hard vs. soft dichotomy discussed by Keller.

Reduced data sample from the psycholinguistics experiment:

In order to perform a more accurate comparison between our results and that reported in [Blache06a], we ran a series of experiments using the same data sample, which is a subset of the full corpus. The results are reported in table 2.

	Adjust.			Weight						Correlation		
♯	k	l	m	wl	wo	we	wr	wd	wu	Max	g	γ
2	4	2	1	5	3	2	2	0	2	0.6017	0.5408	♮**0.6017**
3	4	2	1	5	3	2	2	1	2	0.6017	0.5246	♮**0.6017**
4	4	2	1	20	10	5	4	3	2	0.6427	♮**0.6427**	0.6024
										0.6427		

Table 2. Correlations on the reduced data sample.

The best correlation (rec. ♯4) is obtained for the same parameter scheme as the best one from table 1, but surprisingly this time g outperforms the other two models. It confirms the crucial influence of Linearity on acceptability, but the roles of Uniqueness and Obligation is still unclear, though they are seemingly preponderant.

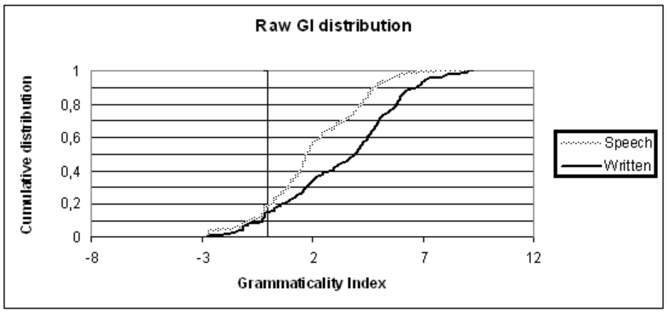
This first evaluation confirms the validity of our gradience model in showing a strong correlation with subjects predictions. Moreover, and this is the most important result, these predictions have been obtained automatically (at the difference with [Blache06a]), which opens the possibility of large scale experimentation, as presented in the next section.

5.3 Large Corpus Validation

We ran our model on a large French corpus, made from different sources: newspapers (184,367 words), spontaneous spoken language (14,065 words) and radio broadcasts transcriptions (84,685 words).

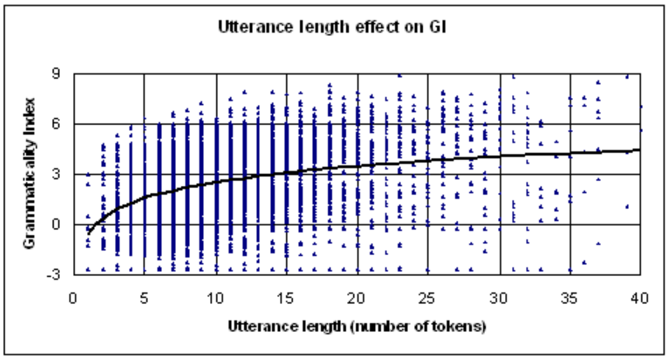
Some general figures can be given. Figure (1) illustrates the (unsurprising) difference between oral and written text: we observe that a greater proportion of sentences in oral production with low grammaticality index. As illustrated in figure (1), the mean index for spoken corpora is 2.46 whereas that of written is 3.36. What is more interesting is the repartition of the index values. There are only few differences between the two types: low and high index values are very similar, which means that some written sentences have a very low grammaticality index whereas some spoken (including spontaneous) can be very high.

As for our postulates, these results confirm the relevance of success cumulativity. This effect is illustrated in the following examples (from the corpus): the longer sentence (6) obtains a higher score than the short one.



(5) j'aime la cuisine (2,1385)
I like cooking
(6) donc c'est pas évident parce qu'il y a des jours où il y a pas de boulot il y a des jours où il y a du boulot comme partout (6,7266)
So it is not easy because there are days where there is no work there are days where there is work, like everywhere

A the corpus scale, we can illustrate this phenomenon in correlating grammaticality index with sentence size. The following figure shows the repartition of sentences with respect to the size. The average of the indexes in function of size match the curve $\langle \rho \rangle = \ln(size)$.



In other words, as expected, the number of words can increase the grammaticality index level. Again, as observed in other works, cumulativity remains less important than the importance (i.e. the weight) of violated constraints. This aspect is illustrated in the following examples: a precedence constraint has been violated in (8), explaining the lower score than in (7):

(7) des foyers genre foyers ce qu'on appelle foyers de jeunes filles ou non mixte quoi (0,8089)

Boarding houses sort of boarding houses what is called boarding houses for girls or not mixed like

(8) non ça m'a fait vraiment pas mal cogiter mais mais bien quoi c'est (-1,201)

No it really made me think hard but but right that's

Other assumptions can be confirmed by a detailed analysis of the results. At this level, we can see that most of constraint violation concern uniqueness, requirement and obligation.

6 Conclusion

We have propose in this paper a specification of the needs for a precise account of syntactic gradience. On this basis, we have specified a new computational model, taking advantage of a fully constraint-based syntactic representation as proposed in Property Grammars. This model has been evaluated first in replicating automatically a previous experiment showing the correlation between the scores given by the model and subjects acceptability judgements. We have then validate the approach in experimenting it on larger and unrestricted corpora.

An automatic account of gradience, such as the one presented here, can have many applications. In terms of parsing, it constitutes an efficient heuristic, helping in the selection of the construction types. Other applications can be imagined, for example in second language learning systems, helping the user in evaluating its productions. At a theoretical and cognitive level, this model shows the relevance of constraints in modelling language production and perception: the rating functions can help in explaining sentence complexity.

References

[Aarts04] Aarts B. (2004) "Modelling Linguistic Gradience", *Studies in Language*, 28(1):1–49.

[Aarts07] Aarts. B (2007) *Syntactic Gradience. The nature of Grammatical Indeterminacy*, Oxford University Press.

[Bard96] Bard E., D. Robertson & A. Sorace (1996) "Magnitude Estimation of Linguistic Acceptability", Language 72:1.

[Blache05a] Blache P. & J.-P. Prost (2005) "Gradience, Constructions and Constraint Systems", in H. Christiansen & al. (eds), *Constraint Solving and NLP*, LNCS, Springer.

[Blache05b] Blache P. (2005) "Property Grammars: A Fully Constraint-Based Theory", in H. Christiansen & al. (eds), *Constraint Solving and NLP*, LNCS, Springer.

[Blache06a] Blache P., B. Hemforth & S. Rauzy (2006) "Acceptability Prediction by Means of Grammaticality Quantification" , in proceedings of *COLING-ACL 06*

[Blache06b] Blache P. (2006) "A Robust and Efficient Parser for Non-Canonical Inputs", in proceedings of *ROMAND-06*,

[Bresnan01] Bresnan J., S. Dingare & C. Manning (2001) "Soft Constraints Mirror Hard Constraints: Voice and Person in English and Lummi", in proceedings of *LFG'01.*

[Chomsky75] Chomsky N. (1975) *The Logical Structure of Linguistic Theory*, Plenum press.

[Fanselow06] Fanselow G., C. Féry, R. Vogel & M. Schlesewsky (eds.), *Gradience in Grammar: Generative Perspectives*, Oxford University Press.

[Foth05] Foth K., M. Daum & W. Menzel (2005) "Parsing Unrestricted German Text with Defeasible Constraints", in H. Christiansen & al. (eds), *Constraint Solving and NLP*, LNCS, Springer.

[Fouvry2003] Fouvry F. (2003), *Robust processing for constraint-based grammar formalisms*, Ph.D. thesis, Graduate School, University of Essex, Colchester, United Kingdom, April.

[Haji-Abdolhosseini05] Haji-Abdolhosseini M. (2005) *Modularity And Soft Constraints: A Study of Conflict Resolution in Grammar*, Ph.D. thesis, University of Toronto.

[Keller00] Keller F. (2000) *Gradience in Grammar. Experimental and Computational Aspects of Degrees of Grammaticality*, Phd Thesis, University of Edinburgh.

[Keller06] Keller F. (2006) "Linear Optimality Theory as a Model of Gradience in Grammar", in Gisbert Fanselow, Caroline Féry, Ralph Vogel, and Matthias Schlesewsky (eds.), *Gradience in Grammar: Generative Perspectives*, Oxford University Press.

[Legendre90] Legendre G., Y. Miyata & P. Smolensky (1990) "Harmonic Grammar - A Formal Multi-level Connectionist Theory of Linguistic Well-formedness: Theoretical Foundations", in *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society.*

[Legendre06] Legendre G., A. Sorace & P. Smolensky (2006), "The Optimality Theory-Harmonic Grammar connection", in P. Smolensky & G. Legendre (eds.) *The Harmonic Mind: From Neural Computation To Optimality-Theoretic Grammar*, MIT Press.

[Manning99] Manning C. & H. Schütze (1999), Foundations of Statistical Natural Language Processing, MIT Press.

[Maruyama90] Maruyama H. (1990), "Structural Disambiguation with Constraint Propagation", in proceedings of *ACL'90.*

[Menzel98] Menzel W. & I. Schroder (1998) "Decision procedures for dependency parsing using graded constraints", in proceedings of *ColingACL Workshop on Processing of Dependency-based Grammars.*

[Prost06] Prost J-Ph. (2006) "Numbat: Abolishing Privileges when Licensing New Constituents in Constraint-Oriented Parsing", in proceedings of *CSLP-06.*

[Prince93] Prince A. & Smolensky P. (1993) *Optimality Theory: Constraint Interaction in Generative Grammars*, Technical Report RUCCS TR-2, Rutgers Center for Cognitive Science.

[Pullum01] Pullum G. & B. Scholz (2001) "On the distinction between model-theoretic and generative-enumerative syntactic frameworks", in processdings of *LACL-01.*

[Schröder02] Schröder I. (2002) *Natural Language Parsing with Graded Constraints.* PhD Thesis, University of Hamburg.

On Semantically Constrained Property Grammars

Veronica Dahl[1,2] and Baohua Gu[1]

[1] School of Computing Science
Simon Fraser University
Burnaby, B.C., Canada
email{veronica, bgu}@cs.sfu.ca

[2] Research Group on Mathematical Linguistics
Universidad de Rovira e Virgili
Tarragona, Spain.

Abstract. *Property Grammars*, or PGs, view linguistic constraints as properties between sets of categories, and can handle incomplete or erroneous text, thus endowing parsers with that important aspect of human cognitive abilities. However the main focus so far has been on syntax. In this work we extend PGs with concept and relation extraction abilities, and we present a new parsing methodology for the resulting SPG formalism, which uses the new semantic component to guide the parse along semantically acceptable lines, and to abduce non-explicit parts of a sentence together with their meaning representations. Our parser is built in terms of HYPROLOG — an extension of Prolog with assumptions and abduction based on CHR (Constraint Handling Rules).

Keywords: Semantic Property Grammars, semantic constraints, cognitively realistic parsing abilities, concept extraction, disambiguation, assumptions, abduction, Constraint Handling Rules, HYPROLOG, long distance dependencies.

1 Introduction

Property Grammars (PGs) [2] [3] belong to a new wave of linguistic formalisms which view a grammar as a set of constraints, and parsing as a constraint satisfaction problem. Antecedents such as HPSG [18] [20], minimalism [4] or the Optimality Theory [19] had already been moving away from rigid hierarchical parsing, towards flexible mechanisms that can treat incomplete, ambiguous or ungrammatical input, and which are thus more adequate for new developments such as recognizing speech, processing the ever growing volume of internet texts, or parsing controlled languages (e.g. an interlingua in which employees from different countries communicate within a geographically distributed enterprise despite possible errors and imperfections). Even outside such specialized needs, a parser ideally should model human cognitive abilities for parsing by being able to extract meaning from text produced in real life conversation, which typically is incomplete, often not perfectly grammatical, and sometimes erroneous. Imperfections can result from normal human error in actual speech, or be introduced by machines, as in the case of text produced from speech recognition systems, which, while evolved enough to be usable, are renowned for their error-proneness.

Many of the new approaches to flexible parsing sacrifice completeness: so-called shallow parsing [1], for instance, identifies syntactic phrases or *chunks* (e.g. noun phrases) derived by flattening down a sentence's parse tree, but typically loses much of the connection among chunks which a parse tree would exhibit.

In PGs, syntactic structure is expressed by means of relations between categories rather than in terms of hierarchy. For instance, a PG parse of the noun phrase "every blue moon" results in a set of satisfied properties (e.g. *linear precedence* holds between the determiner and the noun, between the adjective and the noun, and between the determiner and the adjective; the noun's requirement for a determiner is satisfied, etc.) and a set of unsatisfied properties, which is empty for this example. In contrast, "Every moon blue" would yield a violation of linear precedence between the adjective and the noun, indicated by placing this relationship in the set of unsatisfied properties.

In its original formulation, Property Grammars already provided full rather than shallow parsing, but produced no parse tree — just a list of satisfied and unsatisfied properties which together, characterized the sentence fully. The first directly executable rendition of Property Grammars [6] provided parse trees as well, for further user-friendliness to linguists, and to extend the model to be able to check linguistic constraints that refer to trees.

In this paper we make two further contributions to flexible parsing with PG: a) we extend property based parsing to include semantic information, so that selected phrases can be automatically extracted which incorporate syntax and semantics as a side effect of parsing, and b) we provide a straightforward implementation of the new model in terms of HYPROLOG [5]. This parsing model reaches farther than previous ones in that it allows to relate long-distance constituents quite economically through abduction, and in that it can disambiguate expressions on the basis of types, as well as represents semantic constraints that prune away syntactically correct while nonsensical sentences. It achieves this by extracting concepts and relations from the sentences it parses, and inferring their semantic types from the consultation of appropriate ontologies through our named entity recognition module [13]. We exemplify this for biomedical texts for which we use the GENIA Ontology and the Gene Ontology.

Section 2 presents background information on property grammars. Section 3 extends the Property Grammar model to include semantics in view of information extraction. Section 4 presents our parsing methodology, after an intuitive presentation of the programming tool used: HYPROLOG. Section 5 presents our discussion and conclusions.

2 Background on Property Grammars

Property based Grammars [2] define any natural language in terms of a small number of properties: **linear precedence** (e.g. within a verb phrase, a transitive verb must precede the direct object); **dependency** (e.g., a determiner and a noun inside a noun phrase must agree in number), **constituency** (e.g. a verb phrase can contain a verb, a direct object,...), **requirement** (e.g. a singular noun in a noun phrase requires a determiner), **exclusion** (e.g., a superlative and an adjectival phrase cannot coexist in a noun phrase), **obligation** (e.g. a verb phrase must contain a verb), and **unicity** (e.g. a prepositional phrase contains only one preposition). The user defines a grammar through these properties instead of defining hierarchical rewrite rules as in Chomskyan based models. In addition, properties can be relaxed by the user in a simple modular way. For instance, we could declare "precedence" as relaxable, with the effect of allowing ill-formed sentences where precedence is not respected, while pointing out that they are ill-formed (this feature is useful for instance in language tutoring systems).

The result of a parse is, then, not a parse tree per se (although we do provide one, just for convenience, even in the case of ill-formed input), but a list of satisfied and a list of unsatisfied properties. (see the table below as a toy example).

Input	`les cellules endothéliales immunotoxines peptides proapoptotiques (the endothelial ... cells)`
Output	`cat(np, [sing, masc], sn(det(les), n(cellules), ap(adj(endothéliales)), n(immunotoxines), n(peptides), n(proapoptotiques)), [prec(det,n), dep(det,n), requires(n, det), exclude(name, det), excludes(name, n), dep(sa, n), excludes(name, sa), excludes(sa, sup)], [unicity(n)])`

3 Semantic Property Grammars

The original Property Grammar formalism focusses on syntactic information. Only one of its properties — dependency — is meant to include semantics, but there is no clear specification of how, or of how the semantics included could serve to construct appropriate meaning representations for sentences being parsed.

Our proposed addition is simple but enough for our purposes here: we construct, in addition to the lists of satisfied and unsatisfied syntactic properties corresponding to a category being analysed, a list of *Semantic Properties* associated with the category. We take advantage of the argument structure of verbs and predicate nouns in order to compositionally build typed predicate representations from lexical semantic information plus an ontology of the application domain. These could be used for instance to disambiguate on the basis of expected semantic type compatibilities, perhaps along the lines proposed in [10]. Our main uses of them in this article are: (a) to allow us to reconstruct the meaning of non-overt constituents from the meaning of those overt ones they relate to, perhaps long-distance, (b) to allow us to express semantic constraints that can rule out sentences that are nonsensical in the given domain.

We next introduce this extension by means of examples taken from biomedical text — one of the applications we are currently working on. It assumes a type hierarchy of concepts, or ontology, that the parser can consult (we use the Genia Ontology[3] and the Gene Ontology[4]).

Associating terms with their semantic Types:

The predicates obtained from parsing noun phrases and verb phrases can be made more informative by consulting type hierarchies of the domain we are dealing with. For instance, in our biomedical domain, we consult a simplified adaptation of the GENIA ontology and the Gene Ontology by extracting for the terms that concern us only the IS-A part of these ontologies (see Fig. 1).

```
+------+-source----+-natural----+-cell type
       |           |            +-tissue
       |           |            +-organism
       |           |            +-other (natural source)
       |           |
       |           +-artificial-+-cell line
       |                        +-other (artificial source)
       |
       +-substance-+-compound-+-organic-+-amino acid-+-protein
       |           |          |         |            +-peptide
       |           |          |         |            +-other (amino acid)
       |           |          |         |
       |           |          |         +-nucleic acid-+-DNA
       |           |          |         |              +-RNA
       |           |          |         |              +-nucleotide
       |           |          |         |              +-other (nucleic acid)
       |           |          |         |
       |           |          |         +-carbohydrate
       |           |          |         +-other (organic compounds)
       |           |          |
       |           |          +-inorganic
       |           +-atom
       |
       +-bio process-+-immune system process-+-leukocyte activation
       |             |                       +-other (immune system process)
       |             |
       |             +-metabolic process
       |             +-localization
       |             +-cellular process
       |             +-regulation of bio_process-+-negative regulation of leukocyte activation
       |             |                           +-positive regulation of leukocyte activation
       |             |                           +-other (regulation of bio_process)
       |             +-other (bio_process)
       |
       +-other (biological terms)
```

Fig. 1. A simplified ontology of biological terms.

Extracting terms and relations within noun phrases: A noun is interpreted by our parser as the relational symbol of a semantic relationship to be

[3] available at http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/.
[4] http://www.geneontology.org/

extracted, and the arguments of this relation are constructed from the noun's various complements, appropriately typed after consultation of the domain-dependent (in this case, biomedical) ontology. For instance, the noun phrase:

```
The activation of NF-kappa-B via CD-28
```

parses semantically into the list of properties:

```
[protein('NF-kappa-B'), gene('CD-28'),
 activation('NF-kappa-B', 'CD-28')]
```

which shows the relationship obtained, i.e., `activation('NF-kappa-B','CD-28')`, together with the types that our concept hierarchy consultation associates with each of the arguments of the relationship (i.e, `'NF-kappa-B'` is of type *protein*, whereas `'CD-28'` is of type *gene*.

Extracting terms and relations within verb phrases: Just as nouns induce relationships, verbs also induce relationships whose arguments are the semantic representations of the verb's syntactic arguments. For instance, in the sentence:

```
Retinoblastoma proteins regulate leukocyte activation.
```

The verb *regulate* marks a relation between two concepts — *retinoblastoma proteins* and *leukycyte activation*. Our parser constructs a list of properties which identifies the semantic types of the relationship's arguments as well as the relationship itself, just as was done for noun phrases.

```
[protein('retinoblastoma proteins'), process('leukocyte activation'),
 regulate('retinoblastoma proteins', 'leukocyte activation')]
```

Domain-dependent semantic constraints: Some sentences that are syntactically correct should still be ruled out because they do not make sense in the domain of application. For example, in our biomedical domain, "it would not make sense to have a term of the form 'regulation of protein name', e.g. regulation of actin, because a protein is neither a process nor a function, so there is no indication of what biological activity is being regulated. "[5]. Our system accepts the codification of such semantic constraints in terms of CHR rules, as shown in Appendix A1.

The same technique can be used as well to disambiguate on the basis of semantic types, by adapting our proposal of [10]. While lexically ambiguous terms are all produced, their inclusion as part of a typed predicate's argument structure will be conditioned to type-compatibility.

4 Our Parsing Methodology

4.1 Background: HYPROLOG

Our parser's programming tool, HYPROLOG, is an extension of Prolog with assumptions and abduction, useful for hypothetical reasoning and other applications, running on top of Sicstus Prolog, from which it can use all features and

[5] http://www.geneontology.org/GO.process.guidelines.shtml

libraries, including Constraint Handling Rules or CHR [11]. Here we describe HYPROLOG in its three main components, in as intuitive a fashion as possible.

Assumptions: Assumptions are a logic programming incarnation of linear and intuitionistic implications [12] which includes as well a new type called *timeless* assumptions, which have been found particularly useful, among other things, for parsing. For further or more formal details, see [9].

Assumptions and consumptions are similar to the Prolog primitives "assert" and "retract", except that they are available during the entire continuation of the computation, and that they can be backtracked upon. Assumptions are facts[6] that are added to a Prolog program dynamically, making them available from the point in which they are called, and during the entire continuation. Linear assumptions can be used at most once[7], whereas intuitionistic assumptions can be used as many times as needed. Notation-wise, linear assumptions (the ones we would use the most) are preceded by '+', and their consumptions by '-'. Other types of assumptions and examples are shown in Appendix A2.

Constraint Handling Rules, or CHR:

A CHR program is a finite set of rules of the form {`Head ==> Guard|body`} where `Head` and `Body` are conjunctions of atoms and `Guard` is a test constructed from built-in predicates; the variables in `Guard` and `Body` occur also in `Head`; in case the `Guard` is the local constant "true", it is omitted together with the vertical bar. Its logical meaning is the formula ∀(`Guard` → (`Head` → `Body`)) and the meaning of a program is given by conjunction.

A *derivation* starting from an initial state called a *query* of ground constraints is defined by applying rules as long as it adds new constraints to the store. A rule as above *applies* if it has an instance (`H==>G|B`) with `G` satisfied and `H` in current store, and it does so by adding `B` to the store.

It is to be noted that if the application of a rule adds a constraint c to the store which already is there, no additional rules are triggered, e.g., `p==>p` does not loop as it is not applied in a state including `p`.

There are three types of CHR rules:

- **Propagation rules** which add new constraints (body) to the constraint set while maintaining the constraints inside the constraint store for the reason of further simplification.
- **Simplification rules** which also add as new constraints those in the body, but remove as well the ones in the head of the rule.

[6] They can also be full clauses, but the subset containing only facts is quite enough for our purposes in this paper

[7] The type of linear assumption we use is more rigorously called *linear affine implication* in the literature, and it differs from linear implication proper in that it can either be consumed once or not at all, whereas linear implication proper must be consumed exactly once

– **Simpagation rules** which combine propagation and simplification features, and allow us to select which of the constraints mentioned in the head of the rule should remain and which should be removed from the constraint set.

Abduction: Abduction is the unsound but useful rule of inference which concludes (or abduces) a from the knowledge of b and the rule that a implies b.

Abductive capablilities can be simply incorporated through CHR by declaring as abducibles certain predicates, which when generated and not resolvable, will simply remain in the constraint store. E.g. if every time it rains I go to the cinema, and going to the cinema has been declared as abducible, when querying: cinema, there being no definitions for it, it will remain in the constraint store, marked as abduced thanks to the declaration which states it is an abducible predicate. More details can be found in [9]. An example of using abducibles within our parser is shown in the next section.

4.2 A HYPROLOG Parser for Semantic Property Grammars

The Parsing Engine A HYPROLOG program is written as a Prolog program with additional declarations of assumptive and abductive predicates. In this paper we neglect to write all declarations, since it is apparent from our discussion which predicates are assumptions and which are abducibles.

Our parser consists of a parsing engine that specializes into a given parser by parser-specific definitions of a predicate `apply_rules/0`. Appendix A3 shows our parsing engine and Appendix A4 shows a sample parser.

Our SPG parser

a) New Category Inference

To use the parsing engine given in Appendix A3 for SPGs , we record all categories (including lexical categories) as assumptions, and define rule application such that it combines two categories (one of which is a phrase or a phrase head) after checking the properties between them, and constructs a new category from both of these, also recorded as an assumption.

Our syntactico-semantic categories are described in the notation:

```
+cat(Cat,Features,Graph,Sat,Unsat,Semantics,Start,End)
```

where `Cat` names a syntactic category stretching between the start and end points `Start` and `End`, respectively; `Features` contains the list of its syntactic features, such as gender and number, `Graph` represents the portion of parse tree corresponding to this category and is automatically constructed by the parser; `Sat` and `Unsat` are respectively the list of syntactic properties respectively satisfied and unsatisfied by this category; and `Semantics` holds the Semantic Properties list.

For example the lexical entry for the word 'NFkappaB' would in DCG notation look like:

```
name(protein-'NFkappaB') --> ['NFkappaB'], {ner('NFkappaB'), protein}.
```

where `ner` stands for our consultation of the GENIA Ontology and the Gene Ontology. More lexical entry examples can be found in Appendix A5. In the present system, this should compile either manually or mechanically into:

```
+cat(name,[sing],name('NFkappaB'),[],[],protein-'NFkappaB',Start,End)
```

The sentence's words and related information, thus expressed as assumptions, can be viewed as resources to be consumed in the process of identifying higher level constituents from them.

Lists of satisfied and unsatisfied properties are created by our single rule, so that incorrect or incomplete input is admitted but the anomalies are pointed out from the list of unsatisfied properties. The main rule for inferring the new category is given in Appendix A6.

Note that the parser's single rule consumes two resources before creating a new one, so each rule application decreases the (initially finite) number of resources available by one. The process stops when no more rule applications are possible, leaving if successful a category "sentence" stretching between the start and end points of the input sentence, and containing its full characterization (satisfied and unsatisfied properties, semantics, etc.)

b) Abducing semantic relationships among constituents

Using HYPROLOG allows us to raise the (admittedly simple) parsing engine shown to fairly sophisticated levels, by complementing it with CHR rules involving abducibles.

For instance, we can upon recognizing a relative pronoun, abduce a non-overt (i.e., with identical start and end point) noun phrase whose meaning coincides (unifies) with that of its antecedent. Noun phrases thus should be declared as abducibles, and their meaning (i.e., their typed argument structure) passed on from the antecedent to the non-overt noun phrase being abduced, which although syntactically empty, thus produces nevertheless the required meaning to be further combined with other relevant constituents.

To illustrate, consider the two relative clauses "The house that fell" and "The house that Jack built" . Both involve a missing noun phrase whose meaning should coincide with that of their antecedent "the house", but whereas in the first sentence the missing noun phrase is the subject inside the relative clause ("the house fell"), in the second sentence it is the verb's direct object ("Jack built the house").

When combining the determiner "the" with its noun "house", our incremental parser postulates a noun phrase already in the initial substring "the house", and constructs a semantic representation for it. The following CHR rule exploits this fact and the presence of a relative pronoun to abduce a missing noun phrase of same meaning. We only show explicitly the values of those arguments relevant to the present discussion:

```
+cat(np,_,_,_,_,SemNP,_,P), +cat(rel_pronoun,_,_,_,_,_,P,_)
==> np(_,_,_,_,SemNP,P1,P1).
```

Note that the missing np is abduced instead of being assumed through `cat/8` (i.e. it is not yet promoted to the rank of a proper category, though it already carries in particular its meaning, gleaned from its antecedent), and that its start and end points are- although their value remains as yet unkown- forced to unify, thus expressing its non-overt quality.

We can now check for the existence of an abduced noun phrase at points where a noun phrase would be needed but is not there. E. g., the following rule relativizes on the subject (as in "The house that fell"), by materializing the abduced np into a proper category placed after the relative pronoun (by unification through `P1`), with the same semantics `SemNP` as those of its antecedent:

```
+cat(rel_pronoun,_,_,_,_,_,_,P1), +cat(vp,_,_,_,_,_,P1,P2),
np(_,_,_,_,SemNP,P1,P1) ==> +cat(np,_,_,_,_,SemNP,P1,P1).
```

while the following rule relativizes on the object (as in "the house that Jack built"), by materializing from an abduced `np` a proper (although empty) `np` category at a point `P1` where it is expected and where some other category occurs instead:

```
+cat(tr_verb,_,_,_,_,_,_,P1), +cat(C,_,_,_,_,_,P1,P2),
np(_,_,_,_,SemNP,P1,P1)
==> diff(C,np) | +cat(np,_,_,_,_,SemNP,P1,P1).
```

This example illustrates how highly economical our methodology is for many types of long distance dependencies: missing constituents are abduced at the point in which their meaning and non-overtness can be anticipated, and from this abduction, materialize as empty constituents with the appropriate meaning at the point where they are missing. This keeps the core parsing engine extremely simple while concisely managing long distance dependencies through CHR rules that access the abduced elements when appropriate.

This syntactico-semantic treatment through assumptions, CHR constraints and abducibles can in general serve to express further linguistic constraints in the flexible way needed by many contemporary applications.

5 Discussion and Conclusions

The idea of throwing away the traditional, hierarchical parsing scheme in favour of a view of parsing which involves properties on categories rather than rewriting schemes was first proposed by G. Bes as the 5P formalism, which later evolved into Property Grammars (cf. [2], [3]).

One other formalism that shares the aims and some of the features of Property Grammars are Dependency Grammars (cf. [22] and on this point [15]), a purely equational system in which the notion of generation, or derivation between an abstract structure and a given string, is also absent. However, whereas in Dependency Grammars, as their name indicates, the property of dependence plays a fundamental role, in the framework we are considering it is but one of the many properties contributing to a category's characterization. Perhaps the work that most relates to ours is Morawietz's [16], which implements deductive parsing [21] in CHR, and proposes different types of parsing strategies (including one for Property Grammars) as specializations of a general bottom-up parser. Efficiency however is not addressed beyond a general discussion of possible improvements, so while theoretically interesting, this methodology is in practice unusable due to combinatorial explosion. Moreover, it produces all properties that apply for each pair of categories without keeping track of how these categories are formed in terms of their subcategories, so there is no easy way to make sense of the output in terms of a complete analysis of a given input string.

We have provided a novel and minimalistic (while fully operational) parsing scheme for our semantic extension of PGs, which can be used as well for parsing other grammatical frameworks where the focus is on flexibility and cognitive skills: in some cases, a sentence's characterization only contains satisfied constraints, but it can also be the case that some constraints can be violated, especially when parsing real life corpora. In most cases, such violations do not have consequences on the acceptability of the input.

Our parser, as we saw, keeps track of the output in the lists of properties (syntactic and semantic) that it constructs, and addresses the problem of combinatorial explosion since every rule application condenses two assumptions into one. Purely CHR implementations, on the other hand, would merely consult constraint predicates, without "consuming" them. We work with just one rule at the heart of parsing, but unlike minimalism, need not filter candidate structures. Our use of HYPROLOG, as we have seen, allows us an economical and modular methodology for dealing with long distance dependencies and other linguistic constraints. Our use of semantic constraints serves to block semantically inappropriate sentences and to disambiguate on the basis of type considerations.

With this work we hope to stimulate further research into the many ramifications of the proposed formalism and parsing methodology.

References

[1] Abney, Steven. Parsing By Chunks. In: Robert Berwick, Steven Abney and Carol Tenny (eds.), Principle-Based Parsing. Kluwer Academic Publishers, Dordrecht. 1991.

[2] Bès G & P. Blache. Propriétés et Analyse d'un Langage. In actes de *TALN'99.*

[3] Blache, P. Property Grammars: A Fully Constraint-Based Theory. In H. Christiansen et al. (eds), Constraint Solving and NLP, LNCS, Springer, 2005.

[4] Chomsky, N. The Minimalist Program. MIT Press, 1995.

[5] Christiansen, H. and Dahl, V. HYPROLOG: A New Logic Programming Language with Assumptions and Abduction. In Proceedings of International Conference on Logic Programming (ICLP), 2005.

[6] Dahl, V. and Blache, P. Directly Executable Constraint Based Grammars. In Proceedings of Journees Francophones de Programmation en Logique avec Contraintes, 2004.

[7] Dahl, V. and Blache, P. Extracting Selected Phrases through Constraint Satisfaction. In Proceeding of ICLP workshop on CSLP, 2005.

[8] Dahl, V. and Tarau, P. Assumptive Logic Programming. In Proceedings of *ASAI 2004*, Cordoba, Argentina, 2004.

[9] Dahl, V., Tarau, P. and Li, R. Assumption Grammars for Natural Language Processing. In *Lee Naish (ed.)* Proceedings of the Fourteenth International Conference on Logic Programming, pages 256-270, MIT Press, 1997.

[10] Dahl, V. and Gu, B. A CHRG Analysis of ambiguity in Biological Texts. In Proceedings of the 4th International Workshop on Constraints and Language Processing (CSLP), Denmark, 2007.

[11] Frühwirth T. Theory and Practice of Constraint Handling Rules. In *Journal of Logic Programming*, 37:1-3, 1998.

[12] Girard, J.-Y. Linear Logic. Theoretical Computer Science, 50:1-102, 1987.

[13] Gu, B. Recognizing Named Entities in Biomedical Texts. PhD thesis (to be defended in June 2008), School of Computing Science, SFU.

[14] Kakas, A.C., Kowalski, R.A., and Toni, F. The role of abduction in logic programming, Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 5, pp. 235¨C324, 1998.

[15] Igor Mel'čuk. *Dependency Syntax*, SUNY Press, 1988.

[16] Morawietz F. Chart Parsing as Contraint Propagation. In Proceedings of *COLING-2000*.

[17] M. Osborne. MDL-based DCG Induction for NP Identification. In Proceedings of *CoNLL-99*.

[18] Pollard C. & I. Sag. *Head-driven Phrase Structure Grammars*, CSLI, Chicago University Press, 1994.

[19] Prince A. & Smolensky P. *Optimality Theory: Constraint Interaction in Generative Grammars*, Technical Report RUCCS TR-2, Rutgers Center for Cognitive Science, 1993.

[20] Sag I. & T. Wasow. *Syntactic Theory. A Formal Introduction*, CSLI, 1999.

[21] Shieber S., Y. Schabes & F. Pereira. Principles and Implementation of Deductive Parsing. In *Journal of Logic Programming*, 24(1-2):3-36, 1995.

[22] Tesniere L. *Elements de syntaxe structurale*, Klincksieck, 1959.

Appendix

A1. Sample Semantic Constraints

The following semantic constraints express the biological knowledge that it does not make sense to regulate a protein or a gene, or in general a biological source or substance, while it does make sense to regulate either a biological function or process. Their implementation through CHR ensures early failure when failure is due, for added efficiency.

```
regulation(Regulator, Regulatee) ==> protein(Regulatee)       | fail.
regulation(Regulator, Regulatee) ==> gene(Regulatee)          | fail.
regulation(Regulator, Regulatee) ==> bio-source(Regulatee)    | fail.
regulation(Regulator, Regulatee) ==> bio-substance(Regulatee) | fail.
regulation(Regulator, Regulatee) ==> bio-process(Regulatee)   | true.
regulation(Regulator, Regulatee) ==> bio-function(Regulatee)  | true.
```

A2. Assumptions - Notation, Examples

To add, or call, an assumption, all we need to do is to precede them with the "+" sign if linear, or the "∗" sign if intuitionistic, at the point where we are adding them or calling them. To use, or consume, either type of assumption, we just precede it by the "−" sign.

For instance, a call to `example1` below will succeed, binding X to "the"; whereas `example2` will fail, since linear assumptions once consumed are no longer there to be consumed again, and a call to `example3` will succeed, binding both `X` and `Y` to "the", since the fact that there is a word "the" at point 0 has been assumed intuitionistically and can therefore be reused as many times as needed:

```
example1 :- +word(the,0), -word(X,0).
example2 :- +word(the,0), -word(X,0), -word(Y,0).
example3 :- *word(the,0), -word(X,0), -word(Y,0).
```

A3. The Parsing Engine

In what follows, the topmost call is to the predicate `recognize`, and `all_consumed` is a system predicate that checks that no assumptions remain unconsumed at the end of the computation:

```
% Parsing Engine

  recognize :- input, apply_rules, all_consumed.

  apply_rules :- apply_rule, !, apply_rules.
  apply_rules.
```

A4. Sample Parser Using the Parsing Engine

The following simple example illustrates the engine's workings for recognition of sentences in the scrambled $a^n b^n c^n$ language, where the input "words" are entered as linear assumptions. :

```
% Domain specific component of a parser for scrambled {a^n b^n c^n}
  % Input string: a b b c c a
  input :- +a, +b, +b, +c, +c, +a.

% Rule application:
  apply_rule :- -a, -b,-c.
```

The problem-dependent definition of `apply_rule` in this case simply consumes one `a`, one `b`, and one `c`. After the second iteration no more assumptions remain, so the second rule of `parse` triggers, and given that all assumptions have been consumed, the program stops with success.

A5. Lexical Rules

```
% name(Type-Entity) --> [Entity], {ner(Entity, Type)}.

name(protein-IL2)         -->['IL2'],        {ner('IL2'),protein}.
name(protein-NFkappaB)    -->['NFkappaB'],   {ner('NFkappaB'),protein}.
name(dna-bcl2)            -->['bcl2'],       {ner('bcl2'),dna}.
name(dna-promoter)        -->['promoter'],   {ner('promoter'),dna}.
name(rna-mRNA)            -->['mRNA'],       {ner('mRNA'),rna}.
name(celltype-monocytes) -->['monocytes'], {ner('monocytes'),celltype}.
name(celltype-leukocytes)-->['leukocytes'],{ner('leukocytes'),celltype}.
name(cellline-HL60)       -->['HL-60'],      {ner('HL60'),cellline}.
```

Here ner stands for our consultation of the GENIA Ontology and the Gene Ontology.

A6. New Category Inference

The main rule for new category inference is shown in Fig. 2.

```
apply_rule :-
    -cat(Cat,Features1,Graph1,Sat1,Unsat1,Sem1,Start1,End1),
    -cat(Cat2,Features2,Graph2,Sat2,Unsat2,Sem2,End1,End2),
    xp_or_obli(Cat2,XP), ok_in(XP,Cat),
    precedence(XP,Start1,End1,End2,Cat,Cat2,Sat1,Unsat1,SP,UP),
    dependency(XP,Start1,End1,End2,Cat,Features1,Cat2, Features2,SP,UP,SD,UD),
    build_tree(XP,Graph1,Graph2,Graph,ImmDaughters),
    unicity(Start,End2,Cat,XP,ImmDaughters,SD,UD,SU,UU),
    requirement(Start,End2,Cat,XP,ImmDaughters,SU,UU,SR,UR),
    exclusion(Start,End2,Cat,XP,ImmDaughters,SR,UR,Sat,Unsat),
    semantics(Sem1,Sem2,Sem),
    +cat(XP,Features2,Graph,Sat,Unsat,Sem,Start1,End2).
```

Fig. 2. New Category Inference

Frequency-based Constraints on Reflexive Forms in Dutch

Petra Hendriks[1], Jennifer Spenader[2], and Erik-Jan Smits[1]

[1] Center for Language and Cognition
University of Groningen
(p.hendriks|e.j.smits)@rug.nl
[2] Artificial Intelligence Department
University of Groningen
j.spenader@ai.rug.nl

Abstract. Can frequency-based constraints predict the choice between marked and unmarked reflexive forms? Dutch has two reflexive pronouns, the unmarked *zich* and the marked *zichzelf*. Two recent typology papers on how frequency can influence the choice of marked and unmarked forms, Ariel (2008) and Haspelmath (2004), make claims testable on corpus data that if a verb is frequently used to express self-directed meaning, an unmarked reflexive form like *zich* will be preferred. Smits, Hendriks, and Spenader (2007) actually tested this prediction on Dutch using a parsed version of a Very Large Corpus, the CLEF corpus (72 million words) and were able to account for 21% of the data. However, sparse data was a problem, and only one model was tested. The three proposals all argue reflexive form choice is correlated with the frequency of coding self-directed activities, but they differ in what they consider the relevant set of predictive actions. Smits, Hendriks, and Spenader (2007) used only third person actions, Ariel (2008) would predict all actions with the verb would be relevant, and Haspelmath (2004) suggests only the ratio of object pronouns to reflexives is relevant. These three models plus an additional model are tested on data from a parsed version of the Twente News Corpus (TwNC, 300 million words). Linear regression analysis performed on the TwNC showed Haspelmath's model is able to account for 45.6 % of the data. Because the results suggest that competition between related forms and frequency information play a major role in form choice, these results present a challenge for constraint-based models of grammar that try to retain locality as a guiding characteristic.

1 Marked and unmarked reflexives

Dutch, like German and Swedish, has two reflexive pronouns: *zich* and *zichzelf*, often called SE and SELF reflexives respectively. *Zichzelf* is generally considered the marked form because of its morphological complexity, its ability to bear focus and its tendency to occur with atypical self-directed events while *zich* is considered the unmarked reflexive form. For a group of verbs called *accidental* *reflexives*[3] both *zich* and *zichzelf* are possible, e.g.(1), and both variants are generally interpreted as having the exact same meaning.

(1) Ann kleedt zich/*zichzelf* aan.
Ann dresses SE/SELF
'Ann dresses herself'

Recent research explains the choice between *zich* and *zichzelf* as related to the expectation of the event associated with the verb to be self- or other-directed.

In this paper we review the recent typological results of Haspelmath (2004) and Ariel (2008) on the influence of expectations for actions to be self-directed or other-directed on the choice between marked and unmarked reflexive arguments. We point out some predictions based on their claims that are verifiable by looking at corpus data. We also review the empirical work of Smits, Hendriks, and Spenader (2007) which found a linear correlation between the frequency with which accidental reflexive verbs were used with self-directed events in a large corpus, and the frequency of the choice of *zich* for a reflexive argument. We identify a number of shortcomings in their methodology, improve on their work by using a substantially larger corpus and further, test the predictions made by the typological work of Haspelmath (2004) and Ariel (2008). The results show that frequency of event types plays a major role in reflexive form choice. Given that locality has been argued to be a core characteristic of constraint-based grammars, results showing this type of frequency based economy effect on linguistic forms present a challenge for constraint based frameworks. We conclude with a short discussion of this issue, a number of additional observations and suggestions for future work.

2 Structural and Functional Proposals

2.1 Structural Approach

An influential structural account of accidental reflexive argument choice is found in Reinhart and Reuland (1993). According to this proposal, accidental reflexive verbs that can take both *zich* and *zichzelf* as a reflexive object argument are actually the case of two different verbal lexical entries mapping to the same surface form. One lexical entry is inherently reflexive while the other is simply transitive and can be used to describe self-directed and other-directed events. Thus when a verb such as e.g. *aankleden* (dress) takes *zich* as an argument it is the inherently reflexive lexical entry being expressed. When the same surface form of the verb takes *zichzelf* or any other argument the lexical entry is the non-reflexive form of the verb. When it is used to code a self-directed event, it needs *zichzelf* to coerce the meaning into a self-directed interpretation.

Reinhart and Reuland (1993) accounts for why some verbs can only occur with *zich*: these only have a reflexive entry. Further, they can explain why some

[3] There is also a group of 'inherent reflexives' that only take *zich*, and for very atypical reflexive actions, *zichzelf* would be required.

verbs can only have a reflexive interpretation with *zichzelf*: these verbs have no reflexive lexical entry. However, there are several major problems with this analysis. First, it claims that the mental lexicon contains double entries for the same surface form, an untestable claim that suggests the lexicon is quite inefficient. Second, it can't help predict when given examples will occur with *zich* or *zichzelf* or what is likely for a given verb because the relevant features are hidden. Accidental reflexives also differ in their tendency to occur with *zich* or *zichzelf.* A simple search of the CLEF corpus shows this: percentage of reflexive uses with *zich* were 78% for *bijt*, (bite), 46% for *help*, (help), and 26% for *ontdek*, (discover). Reinhart and Reuland (1993) 's approach has no explanation for this. Third, it has no explanation for why a language would maintain two reflexive forms.

Note also that many of the problems with the analysis made by Reinhart and Reuland (1993) stem from their basic assumption that language is rule based, when rules apply without exception. Thus in order to account for what seems to be variability, a surface ambiguity is assumed.

2.2 A Functional Approach

Haspelmath (2004), Ariel (2008) and Smits et al. (2007) take a very different approach. They argue that frequency-based constraints can explain reflexive form choice: in a nutshell, if a verb is usually used to express other-directed meaning, use *zichzelf* as the reflexive object when using the verb to express self-directed meaning. The difficulty lies in defining what 'usually' means. This explanation essentially equates speech frequency with markedness, and following standard assumptions about markedness and economy,: highly frequent speech events are believed to be coded with unmarked forms, and infrequent speech events with marked forms, essential a Zipfian (Zipf 1935) explanation of grammatical patterns.

The view that frequency of linguistic events can constrain form choice is a radically different approach than structural approaches like that of Reinhart and Reuland (1993) and main stream linguistics in general. It also suggests that language theories need to be global because it means a target form (or from the hearer's perspective, interpretation choice) is influenced by competing possible forms (or interpretations). This means that such an effect would be difficult to model in a canonical constraint-based grammar if locality is required, and seems to suggest that a more global theory such as Optimality Theory (OT, Prince and Smolensky 1993) would be necessary.

But the first step is to determine if frequency does influence form choice in the way proposed. Fortunately, frequentist claims of this type can now be empirically tested by using recently available large scale parsed corpus data.

Haspelmath, Ariel and Smits et al. all substantiate their claims with corpus statistics. They differ in that Haspelmath and Ariel concentrate on typological work and use small scale, descriptive statistics to support their claims. Ariel (2008) outlines the role frequency based expectations may have played in the diachronic development of dedicated reflexive forms, while Haspelmath (2004) concentrates on identifying a number of linguistic universals related to the existence or choice of reflexive forms. Smits et al. (2007) on the other hand uses large scale statistics, questionnaire results and linear regression to come to similar conclusions. Let's consider each paper and their proposed models in more detail.

Ariel (2008)'s work develops the so-called 'functional proposal', the claim that expectations govern the use of marked or unmarked forms, and that verbs associated with a strong expectation of disjoint coarguments will use a marked reflexive, while verbs for which there is a strong expectation of coreference, a less marked reflexive or in some extreme cases even a pronoun should be possible. One example of this Ariel gives is 'buy', where 'I bought me a motorcycle.' is possible in colloquial spoken American English. Ariel's work also offers an account of the utility of having two coexisting reflexive forms: they can help mark degrees of unexpectedness. Ariel makes no suggestions that certain subsets of uses of the verb will be more predictive than others, so this work is consistent with the idea that all uses of the verb will contribute to it's choice of object argument.

Haspelmath (2004) makes a very different suggestion. He begins by listing seven universals related to the use of reflexives forms. Particularly relevant to the current work is Universal 1a:

Universal 1a. In all languages, verbs with higher frequency of reflexive use show shorter reflexive marking than verbs with low frequency of reflexive use. (Haspelmath 2004, p. 7)

Haspelmath (2004) suggests choice of reflexive form is related to how frequently a verb takes a reflexive object compared to how often it takes a pronominal notional object. An obvious problem with evaluating and testing Universal 1a. is that we have to determine when the frequency of reflexive use is common enough to be considered 'high'. Verbs that have traditionally been perceived as very self-directed or 'introverted' in Haspelmath's terms, will show a higher ratio of use of shorter reflexive forms than verbs that traditionally and semantically are perceived as other-directed, or 'extroverted'. The problem is that the introversion is treated as a category to which a verb belongs, or it does not, requiring some sort of cut-off point for membership. Haspelmath gives a number of examples of 'introverted' verbs, with a perceived tendency to often be self-directed activities, and a number of examples of 'extroverted' verbs, marking activities that are perceived as often other-directed. Introverted examples include 'wash', 'shave', 'dress' and 'defend' while examples of extroverted verbs include 'kill', 'hate', 'criticize' and 'attack'.

To check whether or not Universal 1a holds, Haspelmath (2004) did a small search for two English verbs, *kill* and *wash* in the British National Corpus (BNC, 100 million words). *Kill* is considered a typical extroverted verb while *wash* is considered to be a typical introverted verb. Looking at transitive uses of the verbs he found that for *kill*, 30% of the objects were pronouns and only 5% were reflexives, while for wash, 15 % of the objects were pronouns but 24 % of the objects were reflexives, a difference in the right direction but because only two examples are analyzed and because it is not clear at what frequency category membership is defined, much work needs to be done to verify if this particular ratio indeed distinguishes introverted and extroverted forms. Further, the idea of frequency influencing form choice suggests that form choice might be tendential, rather than categorical so a different type of model is needed. Finally, note that since English doesn't have two reflexive forms these results cannot actually be used to verify Universal 1a.

It is unclear is why the ratio of pronominal arguments to reflexive arguments would characterize introverted vs. extroverted verbs. Haspelmath says that the difference between other-directed and self-directed use is not distinguishing enough to be used as evidence of which verb is 'normally reflexive', but when only the transitive uses of the BNC data for 'kill' and 'wash' are considered, reflexives are used only 5% of the time with 'kill', but 24 % of the time with 'wash', a difference that seems just as substantial as the 30% vs. 15% difference reported for pronominal and reflexive objects respectively. Ideally an entire set of verbs should be examined, and statistical tests should be performed to see if the correlation between predictive factors really is significant.

This is in fact exactly what Smits, Hendriks, and Spenader (2007) do, working on predicting the choice between the Dutch unmarked reflexive pronoun *zich* and the marked pronoun *zichzelf*. First, they looked at verbs in an HPSG parsed version of the CLEF (Cross-Language Evaluation Forum) Corpus for Dutch made up of 72 million words and taken from the full content of the 1994 and 1995 Dutch daily newspapers of Algemeen Dagblad and the NRC Handelsblad. 60 verbs were studied. All uses of the verb with third person objects in the corpus were extracted and how often each verb occurred with a reflexive *zich*, *zichzelf* or with a non-reflexive object was noted. Unlike Haspelmath, Smits et al. used the percentage of third person reflexives among all third person objects in sentences with third person pronominal subjects as the predicting factor, rather than just the percentage of reflexive objects among reflexive and pronominal objects.

Linear regression analysis can show if there is a relationship between the tendency of an independent variable, here tendency to be used with self-directed actions, and a dependent variable, the choice of *zich*. They found such a relationship. The corpus results showed that 21% of the uses of *zich* can be predicted by the frequency with which the same verb is used with self-directed, or 'introverted' activities. ($R^2 = 0.21\%$, $t(63) = 3.924$, $p < .001$).

For a large number of verbs however, the number of occurrences was too low to achieve a reliable prediction of how natural it was to use *zich* or *zichzelf*. For this reason they combined these results with the results of an online questionnaire. Twenty-nine native adult speakers of Dutch were asked to make a forced choice between *zich* and *zichzelf* as the best argument for the same 60 verbs.[4] The questionnaire data was then used to derive the statistics for the preference for *zich* or *zichzelf* and the corpus data was used to provide information about

[4] Most subjects were students or staff of the Artificial Intelligence Department at the University of Groningen.

the frequency of self-directed events used with the verbs. When this information was combined with the corpus data and simple linear regression was again performed, Smits et al found that 83% of the distribution could be predicted (R^2 =0.83, t(61) = 16.9, p <001.) This shows that the tendency of verbs to be used with self-directed activities was a predictor of the tendency of reflexive instances to be marked with *zich*.

However, it is somewhat dissatisfying that even with such a large corpus (72 million words!) Smits, Hendriks, and Spenader (2007) could only predict 21% of the data. This could be explained in several ways. First, Smits, Hendriks, and Spenader (2007) only did automatic counts, assuming that the parser's analysis was correct. Noise in the data may have worsened the results. Second, they only looked at third person object forms with third person pronomial subjects. This restriction was to insure that the subjects in the majority of examples were agents acting on other agents.[5] But this restriction also excludes a number of actual animate agent examples. A further restriction was that only third person forms were considered because *zich* and *zichzelf* are the only reflexive forms used as third person objects. But may have underestimated the number of reflexive uses of the verb because first and second person reflexive uses were not counted. Considering first and second person reflexive uses might also improve the correlation. Third, for many of the verbs tested they had fewer than five reflexive uses in the entire corpus, e.g. *aai* (to pet) only had three cases of reflexive uses. There is clearly a sparse data problem and makes the counts for these verbs less reliable. To counteract this they used data from a questionnaire study to obtain more information about the tendency to choose *zich* or *zichzelf*, but ideally the analysis would be more attractive if a corpus large enough could be used.

Finally, instead of examining the results per verb in great detail, Smits, Hendriks, and Spenader (2007) chose to focus on the group of verbs as a system and to concentrate on general tendencies. It is possible that subgroups of verbs behave very differently.

3 Method

To determine what frequencies play a role (if any) in reflexive form choice we looked at the usage of 45 verbs (see Appendix). The particular verbs chosen were verbs frequently discussed in the literature as accidental reflexives. To solve the sparse data problem and improve upon the searches made in Smits, Hendriks, and Spenader 2007 we used the 2002 release of the Twente Nieuws Corpus (TwNC). The TwNC contains over 300 million words and is made up of texts from a number of Dutch daily newspapers from the 1990's until 2002, and a

[5] Only animate, agentive subjects could conceivably perform self-directed actions; since animacy isn't tagged in the corpus, excluding other NP subjects prevents cases of inanimate NPs acting on inanimate objects from incorrectly adding to the number of other-directed events, since non-agent subjects would be unlikely to occur with reflexive objects.

number of subtitle files. We searched through a version of the corpus parsed by the HPSG-based Alpino parser Bouma, van Noord, and Malouf (2000).

For each verb, all sentences where the verb occurred with an object were extracted from Alpino-parsed texts and saved according to subject type with the Alpino-identified object marked. These files were then used to obtain counts for the total number of cases of the verb used transitively, the number of times the verb was used with a self-directed meaning in all three persons (all cases with first person singular subject and first person singular object, all cases with second person subject and second person object, etc.), the number of times the verb was used with a pronominal subject, the frequency of pronominal objects for different sentence types and finally the total number of *zich*'s and *zichzelf* objects. After the final extraction we removed all verbs that lacked non-reflexive objects, which suggests the verb is an 'inherent reflexive', and all verbs that never occurred with *zichzelf* because this suggests that they are 'inherent reflexives', and that there is no actual choice between *zich* or *zichzelf* possible. Examples without *zichzelf* objects might also just be a consequence of the data being just too sparse to make good predictions possible.

32 verbs remained. For each of these verbs we looked at descriptive statistics of their usage and also performed linear regression on the data set, testing four different models of reflexive choice. For these models the dependent variable was always the proportion of *zich* used among *zich* and *zichzelf* uses and the independent predictor variable varied according to the model:[6]

1. Smits et al. (2007) : Does the percentage of reflexive objects in all third person uses of the verb with a pronominal subject correlate with the choice of *zich* among third person reflexive uses of the verb?
2. Smits et al. (2007) with 1st & 2nd person: Does the percentage of reflexive objects in all three persons for the verb with a pronominal subject correlate with the choice of *zich* among third person reflexive uses of the verb?
3. All verb uses : Does the percentage of reflexive objects for all uses of the verb correlate with the choice of *zich* among third person reflexive uses of the verb?
4. Haspelmath (2004): Does the percentage of reflexive objects within all reflexive or pronominal objects correlate with the choice of *zich* among third person reflexive uses of the verb?

[6] Note that because Smits et al. only looked at sentences with third person pronominal subjects, the percentage of *zich* is calculated only from examples with *zich* or *zichzelf* objects and third person pronominal subjects. Because this is a smaller number of examples, more verbs had to be excluded because they either had no examples with *zichzelf* or no examples with non-reflexive objects. The other two models include cases of *zich* or *zichzelf* objects with other NP subjects and used all 32 verbs mentioned above.

verb	English	expect?	*zich*	*zichzelf*	non-reflO	reflO	%zich	%self-dir
koop	'buy'	no	31	6	18379	48	0.838	0.003
verkoop	'sell'	yes	34	93	14583	147	0.268	0.010
schilder	'paint'	yes	4	25	2401	30	0.138	0.012
dood	'kill'	yes	1	19	1768	22	0.050	0.012
haat	'hate'	yes	0	7	2438	28	0.000	0.011
sla	'hit'	yes	298	38	8823	406	0.887	0.044
was	'wash'	yes	75	4	1244	107	0.949	0.079
straf	'punish'	no	2	30	393	42	0.063	0.097
scheer	'shave'	yes	40	4	94	71	0.909	0.430

Table 1. Sample of verbs and counts ordered according to percentage of self-directed events: 'expect' = Does the trend follow predictions?, 'zich': Number of *zich* objects, 'zichzelf': Number of *zichzelf* objects, non-reflO: Number of non-reflexive object, reflO: Number of reflexive objects, %zich: Percentage *zich*, %self-dir: Percentage self-directed events

4 Results

4.1 Descriptive statistics

From the group of 32 verbs meeting the initial criteria, the mean percentage of self-directed actions was about 11% (0.115), with values ranging from 0.00049 to 0.916 (for *graaf in*; 'bury'). The mean percentage of *zich* usage was 57%, with values ranging from no occurrances (e.g. *haat*, 'hate' and *ontdek*, 'discover') to 99% (e.g. *lach*, 'laugh').

In Table 1 statistics for a handful of verbs are given, ordered according to the percentage of self-directed events. One of the first things to notice is the great variation in frequency with which reflexives are used. We can see that *verkoop*, 'sell', *koop*, 'buy', *dood*, 'kill' and *haat*, 'hate', were overwhelmingly used with other-directed events.

How well do the data measure up in relation to the predictions made by Haspelmath, Ariel and Smits et al.? Haspelmath's universal says that if a verb is used frequently with a self-directed meaning, it will also use the unmarked reflexive. Remember that one of the problems with testing this type of claim is that we have to find a principled way to determine how to interpret 'frequently'. If we simply take values above the mean to be 'frequent', above 11 %, then the cut-off point seems too high and many verbs that intuitively are classified as highly introverted will be left out, e.g. all the verbs in Table 1 except for 'shave'. This is because the range of reflexive use is so wide that the mean is a poor characterization of the data. Since reflexives in the corpus seem to be very infrequent, a much lower number seems more appropriate. If we instead consider a rate of at least 2 % to be a 'high' percentage of reflexive usage then 'hit', 'wash', 'punish' and 'shave' are all verbs that could be considered 'introverted'. This seems more reasonable.

We also then have to determine what can be considered a high percentage of *zich*. Here taking the mean is again misleading, given the range of values. For the sake of discussion, let's let *zich* usage higher than 30 % be 'high'. This then make a substantial number of our set of verbs behave in the predicted way.[7]

Now we can examine how the predictions work out for individual verbs. Haspelmath predicts that if the frequency of self-directed events is high, the percentage of *zich* should be high. This is true for every verb except *straf*, 'punish' (including for verbs not listed in Table 1). Ariel and Smits et al. make a stronger prediction: they argue that the tendency for the verb to be used with self-directed events correlates linearly with the choice of *zich*. For this proposal we should also find that if the percentage of self-directed events is low (less than 2%), the percentage of *zich* will also be low (less than 30%). The results in the Table for *koop* do not follow this prediction, and neither do the results follow this prediction for the verbs *aai*, 'to pet', *bewonder*, 'to admire', *geef*, 'to give', *knip*, 'to cut', or *teken*, 'to draw'. For the entire set of 32 verbs, 6 go against the prediction, having a low level of self-directed activity but still a high percentage of *zich*.

So in general, all but one of the verbs follow Haspelmath's prediction, and for the broader prediction of Ariel (2008) and Smits et al. (2007), all but seven of the 32 verbs follow the prediction.

Look more closely at *koop*, 'buy' and *verkoop*, 'sell', which were discussed in Ariel(2008) as being good examples of a self-directed and an other-directed verb. 'Sell' unexpectedly has more self-directed events than 'buy'. Of course, both *verkoop* and *koop* occur very infrequently with self-directed events according to our definitions, and yet both use *zich* frequently as their reflexive argument. These results are difficult to account for in terms of expectation.

If we look at the pair *dood*, 'kill' and *was*, 'wash', two verbs discussed by Haspelmath (2004) in particular as an illustration of an introverted and an extroverted verb, we do see that they differ in their frequency of occurrence with a self-directed event, where 'wash' is much more likely to be self-directed, and this does correlate with the percentage of *zich* used with each verb. What we don't know however is if this correlation is perhaps due to chance, given the small number of verbs.

What this shows is that it is quite difficult to give evidence of a trend if only a handful of verbs are examined, because even cases that theoretically have been argued to be very different, e.g. 'sell' and 'buy', can be very similar in the extent to which they are used with self-directed or other-directed activities even in a very large corpus like TwNC. Descriptive statistics of individual verbs are only minimally informative. Testing the validity of the correlation for an entire group of verbs instead could give us more reliable estimations of the correctness of the basic claim that tendency to be used with self or other-directed meaning determines reflexive form choice.

[7] It's possible that there is a better system to determine cut-off points. We leave this for future work.

4.2 Linear Regression Analysis

Is there a statistically significant relationship between the frequency with which a verb occurs in a reflexive object versus a non-reflexive object, and the frequency with which the same verb in only reflexive events occurs with *zich* versus *zichzelf*? In other words, can we predict the frequency of the use of *zich* versus *zichzelf* based on the frequency of reflexive usage? We examined this question by making a simple linear regression analysis using the use of *zich* and the frequency of reflexive usages as regressors, using four different variations for counting the ratio of reflexive actions. What model predicts the choice between *zich* or *zichzelf* the best?

1. Smits et al. (2007) : This model did better than the original corpus study of Smits et al. (2007), accounting for 28 % of the data (R^2 =0.285, t(22) = 2.896, p <.009). It suggests that some of the sparse data problem is solved, even though more than half of the verbs had to be excluded from the analysis (23!) because of too little data
2. Smits et al. (2007) with 1st & 2nd person: Adding information about first and second person objects improved the model slightly so that 30% of the data is predicted (R^2 =0.30, t(22) = 2.998, p <.007). A plot of the estimated curve is shown in Figure 1.
3. All verb uses : Using all transitive uses of the verb, including all kinds of subjects, was a substantially poorer model only accounting for 23 % of the data. (R^2 =0.236, t(32) = 3.041, p <.005)
4. Haspelmath (2004):This was by far the most successful model, accounting for slightly over 45 % of the data. (R^2 =0.456, t(32) = 5.015, p <.000). A plot of the estimated curve is shown in Figure 2.

4.3 Shortcomings and Improvements

Beginning with the descriptive statistics for the individual verbs, what are some possible explanations for the outliers?

There were two types of outliers, seven verbs where a low frequency of self-directed actions in the corpus occurred with a high frequency of *zich*, and cases where a high frequency of self-directed actions correlated unexpectedly with a low frequency of *zich*.

For the first group there are several explanations. The first and most obvious is to follow Haspelmath and argue that frequency expectations only mandate that when self-directed actions are infrequent, there is nothing to prevent the shorter, less marked form from being used. However, this would be against the idea that the choice of reflexive form is governed by expectations. Another possible explanation is that in the examples some of the (infrequent) self-directed actions that did occur with *zich* were contextually habitual. This is related to an observation by Geurts (2004), who showed that the action 'inject oneself' (*toedienen* in Dutch) is not expected to be self-directed (intuitively) and this

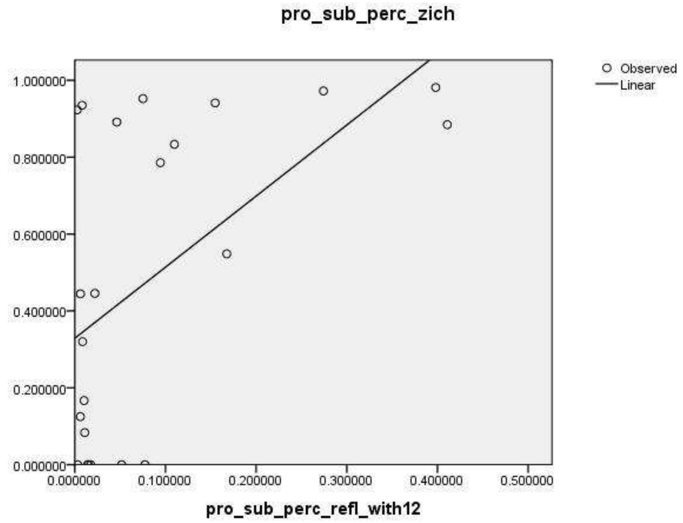


Fig. 1. Smits et al. (2007) model revised: Reflexive usage of each verb (x-axis) in first, second and third person in sentences with pronominal subjects compared with the percentage of the use of *zich* among *zich-zichzelf* reflexives (y-axis)

explains why it overwhelmingly occurs with *zichzelf*. However, if you make an example where the injections are habitual, for example the subject might be a heroin user, then *zich* becomes possible within the context.

The second possible reason might be that the low frequency of self-directed actions is actually incorrect. The corpus data contains only very simple statistics of objects and pronoun types. Reflexive objects are not the only means that language has to express reflexive events. Ariel (2008) discusses three ways in the diachronic development of a dedicated reflexive form that languages regularly use to mark reflexivity: 1) languages use a pronoun and rely on the hearer to use pragmatic inferencing to recognize the coreference, 2) languages use an emphatic marker with a pronoun to signal that the interpretation is marked, or 3), languages use a possessive pronoun + body part to signal the reflexive meaning.

While the first two strategies are unlikely to occur in languages with fully developed reflexive forms, the third strategy is quite prevalent. It can be used to literally describe reflexive physical actions, e.g. 'wash my hands' but can generalize to signal non-literal reflexive actions. For example, Ariel (2008) found that in Biblical Hebrew, which lacked a dedicated reflexive form, the word *libbo*, objective 'his heart' and *nafsho*, objective 'his soul', were used 68 out of 91 times (74.7%) with a reflexive meaning.

Ariel (2008) also found that for 31 of the 51 object possessor NP's, the possessor coreferred with the subject in the Santa Barbara Corpus (SBC) of Spoken

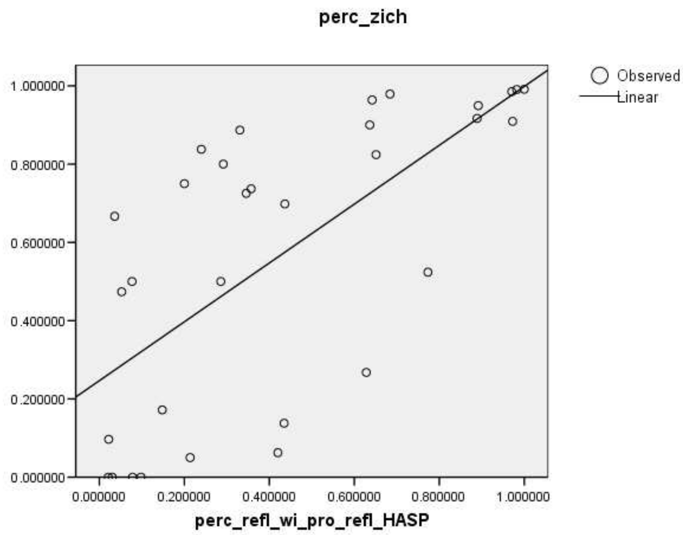


Fig. 2. Haspelmath 2004 model: Reflexive object usage of each verb (x-axis) within all reflexive and pronominal objects compared with the percentage of the use of *zich* within reflexive objects alone (y-axis)

English. This strongly suggests that to get an accurate picture of the general expectation for the tendency of a given verb to be used with a self-directed or other-directed action we should also look more closely at full NP objects with possessive pronouns and add the cases where the possessor is coreferential with the subject to our count of self-directed events.

In fact, the corpus contains numerous examples where the action is clearly reflexive but a possessed body part is used as the object.

(2) Volgens de politie had hij *zijn onderlichaam* groen geschilderd.
According to the police he had painted his lower body green.

(3) De aangesprokene mompelde iets , terwijl hij met een zakdoek *zijn lip* bedekte . (Algemene Dagblad, 1995-11-15)
The addressee mumbled something, while he covered his lip with a handkerchief.

To evaluate both of these proposals the examples and, in the case of habitual *zich*, the context in which they occur will have to be examined by hand. This is beyond the scope of the current work and will have to be relegated to future work.

Outliers, like the case of *straf*, 'punish', where the percentage of self-directed actions was high, yet the percentage of *zich* was low, can't be explained by the above. (In fact, there were no third person reflexive arguments for 'comb'). The only explanation put forth in the literature for cases where *zichzelf* has to be used but *zich* would be expected is when focus is involved. Only *zichzelf* can carry focus, so if the reflexive argument has to associate with focus because of some contrast, then even with a verb predicted to prefer *zich*, *zichzelf* will be required, such as e.g.

(4) Sinds 1981 schildert hij alleen maar [zichzelf] , maar dat hij daarmee echt veel over zichzelf te weten is gekomen , nou nee.
Since 1981 he only paints [himself], but that he by doing so truly came to know more about himself, no, not really.

5 Discussion

Haspelmath's model predicts the choice of *zich* or *zichzelf* the best. Remember, this model argues that the higher the rate of object reflexives compared to object pronouns, the more introverted the verb is, which then for our data predicts a correlation between degree of introversion and the frequency of the use of *zich* for reflexive marking.

This is still a model where the choice of *zich* can be said to be predicted by the tendency of the verb to be used with self-directed or other-directed actions, but where only other-directed actions that are sufficiently activated that they can be expressed with a pronoun count. But why does including other types of objects in the counts of other-directed actions lead to worse models? Haspelmath (2007) doesn't actually have much to say on this and indeed in the *wash* vs. *kill* example either ratio would have made the same prediction.

One possibility is that it has to do with the fact that pronouns and reflexive forms only truly compete when the object the speaker wants to refer to is activated. If a language has two reflexives it also generally has a dedicated pronoun as well, and the choice between pronoun and reflexive is controlled by so-called binding principles. Since reflexives are believed to develop from a pronominal form (see Ariel 2008), there may still be a close relationship here even in languages that have developed true pronoun and reflexive forms.

On the other hand, all models fail to account for the majority of the data. If frequency-based expections are a major (or 'the' major) factor motivating the choice between marked and unmarked forms, why isn't the correlation better?

There are several possible explanations. First, there is still a sparse data problem. For many of the verbs the reflexive forms were very infrequent. The fact that adding data on *zich* and *zichzelf* choices from a questionnaire in Smits et al. improved the correlation so much (over 80%) also points to sparse data still being a problem. 300 million words is already an extremely large parsed corpus. If more data is necessary we have to also start to question if learning expectations are really plausible. It's possible that genre had a major effect. Many actions, especially in a reflexive form, might be less likely to appear in a daily newspaper, e.g.
textitscheer, shave. But take even *dood*, 'kill', for example: Haspelmath found

many more examples in the balanced BNC corpus than we did, suggesting that genre does play a substantial role. If a frequency explanation is correct, we would expect that speakers have learned their expectation from speech data, so examining a spoken language corpus might be fruitful for overcoming the sparse data problem

Another factor that has been argued to play a major role in the choice of reflexive form is focus, and this was not examined at all in the study. It is well known that *zich* cannot take focus, so for focus positions, or in cases where the reflexive is contrasted, the speaker is forced to use *zichzelf*, and these examples cannot be clearly attributed to inherent characteristics of the verb. Note that another type of outlier that would be counter the Smits, Hendriks, and Spenader (2007) predictions would be cases where the percentage of reflexives is high yet the percentage of *zich* is low. However, there were no outliers in this direction (too many *zichzelf*'s for the percentage of self-directed events) so this is unlikely to be a factor that improves the correlation for the data studied here, but it could play a role if other genres are examined.

Actually, if the results of Smits, Hendriks, and Spenader (2007) and Ariel (2008) and Haspelmath (2004) are right, focus should play a rather minor role, since it is the tendency of a verb to be used to describe other-directed actions that should account for the frequent use of *zichzelf*. To study this we could examine the uses of *zichzelf* in the corpus for obvious contrast or focus positions, though this would have to be done manually. There could be something wrong with the verbs studied. Should they all be included, even the ones with very low (or non-existent) counts for *zich* and *zichzelf*? the verbs studied were not chosen in any principled way: verbs that had commonly been identified as accidental reflexives in the theoretical literature were used. But the typological and syntactic literature has long discussed the identification of several different groups of verbs with different tendencies to be used with self or other-directed events. Grooming verbs in particular are believed to have a relatively high frequency of self-directed uses. Since the semantic type of action coded by the verb is believed to strongly influence its co-occurrence with marked or unmarked forms, the verbs used to calculate the correlation should be chosen with care, and motivated. We leave this for future work.

Finally, what do the results tell us about what kind of information grammars will need to be able to take into account? The finding that frequency plays such a major role in form choice suggests that we need a global theory of grammar, where the grammaticality (or appropriateness, or preference) for a given form can be affected by not just the presence of other forms in the lexicon, but also by the way in which the two (or more) forms compete. It seems that speakers keep track of much more information than many theories would seem to predict. This suggests that a standard constraint based theory that requires locality would not be able to deal with this variation, though a more global theory like Optimality Theory seems to be a possibility.

References

Ariel, M. (2008). *Grammar and Pragmatics*. Cambridge Textbooks in Linguistics.

Bouma, G., G. van Noord, and R. Malouf (2000). *Computational Linguistics in the Netherlands*, Chapter Alpino: Wide-coverage computational analysis of Dutch. Rodopi.

Geurts, B. (2004). Weak and strong reflexives in Dutch. In *Proceeding of the ESSLLI workshop on semantic approaches to binding theory.*

Haspelmath, M. (2004). A frequentist explanation of some universals of reflexive marking. Draft of a paper presented at the Workshop on Reciprocals and Reflexives, Freie Universitaet Berlin, 1-2 October.

Prince, A. and P. Smolensky (1993). Optimality theory: Constraint interaction in generative grammar. Rutgers University and University of Colorado, Boulder.

Reinhart, T. and E. Reuland (1993). Reflexivity. *Linguistic Inquiry 24*, 656–720.

Smits, E.-J., P. Hendriks, and J. Spenader (2007). Using very large parsed corpora and judgement data to classify verb reflexivity. In A. Branco (Ed.), *DAARC 2007, 6th Discourse Anaphora and Anaphor Resolution Colloquium, March 29-30, 2007, Lagos (Algarve), LNAI 4410, 77?93. Berlin: Springer.*

Zipf, G. K. (1935). *The Psychobiology of Language*. Boston: Houghton Mifflin.

Appendix

aai (pet), achtervolg (follow), bedek (cover), bedrink (get drunk), bescherm (protect), bewonder (admire), bijt (bite), bind (tie up), borstel (brush), dood (kill), douche (shower), geef (give), graaf in (bury), haat (hate), help (help), hoor (hear), kam (comb), kietel (tickle), kleed aan (dress), knip (cut), koop (buy), kus (kiss), lach (laugh), maak op (make up), omhels (hug), ontdek (discover), pas op (watch), prik (poke, prick), ruik (smell), schaam (be ashamed), scheer (shave), schilder (paint), schmink (make up), schop (kick), sla (hit), snijd (cut), spuug (spit), straf (punish), til (lift), teken (draw), verkoop (sell), verstop (hide), vertel (tell), was (wash), zie (see),

Modelling Global Phenomena with Extended Local Constraints

Patrick McCrae Kilian A. Foth Wolfgang Menzel
{mccrae,foth,menzel}@informatik.uni-hamburg.de

Natural Language Systems Group
Department of Informatics
Hamburg University
D-22527 Hamburg
Germany

Abstract. Weighted constraint dependency systems (WCDG) perform in the antagonism between constraint expressivity and processing limitations. While for reasons of efficiency most implementations constrain themselves to the evaluation of localised unary and binary constraints, a whole range of modelling tasks require access to supra-local or global properties. In constraint dependency grammars, however, these can only be accessed via higher arity constraints. Typical examples from syntax include tests for verb valency or active/passive voice.
In this paper, we illustrate how two additional predicates (`is` and `has`) permit to model global and supra-local properties within the boundaries of a binary constraint syntax. With this extension, higher arity constraints can be cast as a single binary constraint calling unary ancillary constraints cascadingly or recursively. We demonstrate that our method is not limited to modelling strictly adjacent dependencies, but that it can be applied to an arbitrary set of dependencies connected in a dependency tree. In making global properties accessible within an otherwise localised formalism, this work provides significant expressivity enhancements in constraint-based dependency systems.

1 Global Phenomena in Dependency Grammar

Weighted constraint dependency grammar is a formalism which attains its global solution to a constraint satisfaction problem from searching a solution space containing ranked candidates obtained from the optimisation of local constraint evaluations. Given sufficient time, a complete search will find the global optimum as defined by the given set of constraints. Since the complexity of constraint evaluation grows exponentially with the highest constraint arity in the grammar, real-world applications are often restricted in their arity by performance considerations. Constraint-based systems also face a challenge arising from the very nature of their formalism: for unification-based parsing approaches such as HPSG or LFG, propagating feature information through the solution structure constitutes a fundamental principle, but for constraint-based dependency systems with

an inherently localised view, this process presents a considerable challenge. In the following we refer to any property which cannot be tested for by a single localised constraint as *supra-local*. Properties requiring knowledge of the entire solution structure we refer to as *global*. Both the original CDG [Maruyama, 1990] and the later extension WCDG originally supported only unary and binary constraints; in comparison, the similar XDG [Duchier, 1999] supports supra-local constraints by running on top of a general constraint programming system (Oz). In this paper we will show how the extension of the WCDG formalism by the two predicates `is` and `has` opens up new modelling pathways for constraint-based systems that—in combination with certain solution procedures—permit access to complex feature information as if it were propagated along dependency edges. Our work provides solid grounds for challenging the traditional view onto constraint-based formalisms as adopting a strictly localised view. Motivated by a range of German language modelling challenges, we present four applications of the new predicates that illustrate how they can be employed to re-formulate constraints whose complexity exceeds that of the classical weighted constraint dependency formalism based on unary and binary constraints.
This paper is structured as follows: In the following section we provide an introduction into the classical expressivity of the WCDG formalism. Section 3 outlines our extensions to that formalism based on modelling examples. The examples in Section 3 are ordered by increasing constraint expressivity as required by the respective modelling scenarios. Section 4 summarises our key findings, and Section 5 provides an outlook onto future directions for our research.

`X↑from`	linear position of the regent of edge X
`X↓word`	word form of the dependent of edge X
`X↑case`	'case' feature of the regent of edge X
`X.label`	dependency label of edge X
`& | -> <-> ~`	logical operators: *and*, *or*, *implication*, *biimplication*, *not*
`< > <= >= != =`	numerical logical operators: *less*, *greater*, *equal*, *unequal*

Table 1. Operators in WCDG

2 Expressivity of WCDG

The version of dependency grammar which we use in this discussion is WCDG [Menzel and Schröder, 1998], an implementation that allows the definition of weighted constraints on dependency structures across multiple levels. Taken together, all constraints define a grammar that assigns every possible dependency tree a score (essentially, the product of the scores of all violated constraints), thus defining a total ordering over all possible structures, where the highest-scoring structure is taken as the preferred one.
Grammar rules take the form of unary or binary all-quantified logical formulas describing properties that a dependency analysis should have. The most important operators available to the grammar writer are shown in Table 1;

full details can be found in [Schulz et al., 2005]. By combining these operators, logical conditions can be expressed that can be tested on one or two edges in a dependency structure at a time. For instance, the following real-world example penalises right extraposition of relative clauses in German main clauses:

```
{X:SYN,Y:SYN} : 'Extraposition über das Verb' : proj : 0.5 :
  X.label = SUBJ & Y.label = REL & X/ & Y\ & X↓from = Y↑from
  -> Y↓from < X↑from;
```

This constraint can be read as follows: When a relative clause (`Y.label=REL`) modifies a word (`Y↑from = X↓from`) that is a subject (`X.label=SUBJ`), and the relative clause is right-modifying (`Y\`) while the subject is left-modifying (`X/`), then the relative clause itself (`Y↓from`) must occur to the left (`<`) of the finite verb (`X↑from`). With constraint weights ranging between 0 (hard) and 1 (soft) this constraint's score of 0.5 indicates that it describes a dispreference rather than an outright prohibition. This accurately models the observation that while relative clauses can be extraposed out of their topological field, a position within the field of the antecedent is preferred:

"Ein Flugzeug, das drei Passagiere an Bord hatte, ist im Meer abgestürzt."
?"Ein Flugzeug ist im Meer abgestürzt, das drei Passagiere an Bord hatte."
(A plane carrying three passengers has crashed into the sea.)

While the formulation of grammar rules as defeasible, declarative constraints often seems odd to grammarians accustomed to constituent descriptions or generative rules, it is a flexible and highly expressive means of describing linguistic phenomena, particularly when a non-prescriptive grammar is intended. (For instance, a grammar of modern English might employ essentially the same constraint, but assign it a much stricter score.) In fact, as long as a configuration of not more than two dependency edges suffices to describe a phenomenon, WCDG can express any constraint on dependency structures that are formally definable.

3 Extending Local Constraints to Global Phenomena

3.1 Supra-local Constraints

The foremost linguistic phenomenon that cannot be expressed as a local condition is valency. The valency requirement of a word is a local phenomenon in the sense that a single dependency suffices to prove that it is satisfied. Still, it cannot be tested for by all-quantified binary constraints; because any of the other words in a sentence might be the required dependant, *all* dependency edges would have to be checked. Maruyama's solution [Maruyama, 1990] was to establish a second, sparse tree structure that mirrors syntactic valency dependencies as inverse dependencies on an auxiliary level. The mirror condition can be ensured with a binary constraint that couples both levels, and the valency itself requires only a unary constraint on the auxiliary level.

This is a viable solution, but leads to a profusion of auxiliary structures when more than one type of valency must be required, as is often the case with realistic verb forms. An alternative solution is to introduce new operators into the constraint language that can express the existence condition directly. This allows a more readable formulation of valency constraints and avoids auxiliary structures that exist for technical reasons only, but has an important consequence: a constraint which uses such a non-local operator cannot be evaluated in isolation anymore. Rather than just one or two at a time, the entire set of dependency edges in a solution candidate must be known to decide whether or not it satisfies such a supra-local constraint. This means that a supra-local constraint cannot safely be evaluated on partial structures, as is done during a propagation algorithm strategy or a best-first search. The best one can do is to delay the use of these constraints until a complete solution candidate is found and only then to apply the entire grammar. This method resembles the two-step re-ranking technique that is becoming common for other complete search algorithms [Charniak and Johnson, 2005].

Such an approach may be viable as long as only a few of the constraints in a grammar require supra-local conditions, but for a grammar that relies on them more heavily, the first step could become too unfocused to deliver good results. An alternative solution is to use algorithms that do have access to the entire solution candidate from the beginning. WCDG does provide such algorithms; in fact they have been proven to be the best solution strategy [Foth et al., 2000]. The basic strategy is to construct an initial analysis from the (unarily) best-ranked dependencies and then systematically to exchange those dependencies that violate constraints for others which satisfy them. This approach has proved to be workable despite its heuristic nature, and, in fact, is more successful than the (theoretically) complete but infeasible search, while preserving the anytime property of the filtering algorithms. This has prompted the implementation of several extended operators that rely on supra-local properties, and work well together with transformational search [Foth, 2007].

The foremost of these new operators in WCDG is called `has`. In its simplest form, it expresses the condition that a particular word is modified by at least one dependency edge that bears a specific label. This allows conditions like the following to be expressed:

```
X↓cat = VVFIN -> has(X↓id, SUBJ)
```
(Finite verbs need subjects.)

```
X↓cat = VVFIN & X↓transitive = yes -> has(X↓id, OBJA)
```
(Transitive verbs need objects.)

Although the obvious application of this operator is to enforce verb valency conditions, it lends itself to many related uses. For instance, German infinitive constructions can occur in the form of an infinitive with a particular marker word (of the category `PTKZU`), or as a special verb form that incorporates this marker (category `VVIZU`). Assuming that an external marker always bears the special-purpose label `ZU`, the `has` operator allows this condition to be expressed concisely:

```
X↓cat = VVIZU | (X↓cat = VVINF & has(X↓id, ZU))
```
(X↓ *is a valid infinitival construction.*)

The WCDG engine deduces the supra-local nature of a formula automatically by scanning its body and applies such constraints only if a complete solution candidate is available. Thus, conditions are still expressed over single edges or pairs of edges at a time, but during evaluation they can also examine additional neighbouring edges as required.

While the has operator expresses conditions on the dependents of a word, the similar is operator tests the label of the dependency edge above a given word. For instance, German main clauses generally place exactly one constituent in front of the final verb. This can be expressed by a constraint that forbids two dependencies modifying the same verb from the left. However, this condition only holds in main clauses (in which the verb itself is labelled as S), but not for subclauses or relative clauses (labelled e.g. NEB or REL). Therefore, three edges would have to be tested to detect such an illegal configuration: the two right-modifying depencies under the verb and the edge directly above. This would require a ternary constraint, which WCDG does not support. With the supra-local is operator, however, a single binary constraint suffices:

```
{X/SYN/\Y/SYN} : Vorfeld : 0.1 :
  X↑cat = VVFIN -> ~is(X↑id, S);
```

In fact, this constraint is considerably faster to evaluate than an all-quantified ternary constraint would be, because WCDG effectively has to check one additional edge (the one above the finite verb), and that only when the premise of the constraint actually holds. Thus, it allows for easier grammar development and more efficient evaluation than a grammar limited to strictly local constraints.

3.2 Recursive Tree Traversal

One limitation of the supra-local operators is and has described so far is that they operate only on direct neighbours of the dependency edges to which they are applied. This is often sufficient, but there are phenomena which require the presence of structurally more distant features. For instance, a subclause should be analysed as a relative clause (REL instead of NEB) exactly if it is marked by the presence of a relative pronoun, but this pronoun does not always modify the finite verb directly:

> "Es soll eine Art Frühwarnsystem eingerichtet werden, in dessen Zentrum der IWF steht."
> (*An early-warning system is planned whose center will be constituted by the IWF.*)

Similar cases of remote markers abound in German: the conjunction *sondern* is only used for phrases containing a negation somewhere, a genitive modifier must contain at least one overt genitive form, etc. To check such conditions, it is necessary to extend the semantics of the supra-local operators so that optionally they can also find indirect dependants or regents. In such cases it is useful to restrict the extended search in some way, both for operational and for linguistic reasons. For instance, when a subclause is modified by a nested relative clause, the subclause itself should not be labelled `REL`, even though the corresponding dependency subtree contains a relative pronoun further down. Similarly, in co-ordinated sentences the finite verb is labelled as `KON` (in asyndetic co-ordination) or `CJ` (in normal co-ordination) rather than `S` or `NEB`, so that even a lookup via `is` cannot determine whether main-clause or subclause ordering should be enforced; what counts is the label of the topmost finite verb in a co-ordination, which can be several edges apart.

Therefore, the notion of 'scope' has been implemented for the extended versions of the non-local operators: when used with four arguments, the search is extended across a specific set of labels, i.e. those which are subsumed by a particular pseudo-label in a special-purpose hierarchy. For instance, the actual test for sentence type in the Vorfeld constraint is closer to the following version:

`is(X↑id, S, Label, Konjunkt)` (X↑ *is eventually labelled* `S`)

where `Konjunkt` subsumes both `KON` and `CJ` in the hierarchy 'Label'.

This construct effectively ascends the tree from a finite verb until a label other than `KON` or `CJ` is found, and compares this label to the main-clause marker `S`. The `has` operator has been extended in the corresponding way; for instance, it can be programmed to descend into a sentence labelled `REL` to detect a relative pronoun, but only until another subclause indicator such as `REL`, `NEB` or `S` intervenes. This use of a label-delimited semi-global search resembles the notion of *barriers* in Government and Binding theory [Chomsky, 1986], but it does not claim to be a fundamental principle. Indeed, by varying the set of labels to traverse, it can be restricted more or less; for instance, it can operate only upon an NP, or upon the entire tree structure.

3.3 Localised Ancillary Constraints

The syntax for the `is` and `has` predicates introduced so far permits to test for static attributes of the edges above or below the dependency under consideration. A useful extension to the concepts of `is` and `has` therefore is to include a check for the most general edge property expressible: the satisfaction of an arbitrary constraint. Since `is` and `has` are evaluated in the context of a normal constraint, we refer to their argument constraint as *ancillary constraint.* To motivate this extension linguistically, consider thematic role assignment in German non-modal perfect tense active sentences:

Der Mann [AGENT] hat die Frau mit dem Fernrohr gesehen.
(*The man* [AGENT] *has seen the woman with the telescope.*)

In all of the following examples we assume the full verb to be agentive. `SUBJ` and `AGENT` dependencies then originate from the same node in the constraint

net. Non-modal perfect tense active in German is a composite tense formed by a finite auxillary in combination with a full verb's past participle. In constraint terms, this tense can be characterised by a dependency with the following properties: An `AUX` edge (`X.label=AUX`) links a finite auxilliary verb form (`X↑cat=VAFIN`) of *haben* or *sein* (`X↑base=haben | X↑base=sein`) as regent with a full verb's past participle (`X↓cat=VVPP`) as dependant.
Figure 1 illustrates that constraining the origin of the `AGENT` dependency (orange) to the origin of the `SUBJ` dependency (blue) in a non-modal perfect passive sentence requires a ternary constraint involving the `SUBJ`, `AGENT` and `AUX` (green) edges. Moreover, formulating this constraint requires to impose restrictions on the `AUX` edge as well as on the nodes linked by it.
In requiring satisfaction of an ancillary constraint via `is` or `has`, the origin of the `AGENT` dependency in German perfect tense active sentences can elegantly be formulated as follows: A `SUBJ` edge (`X.label=SUBJ`) meeting with an `AUX` edge that marks a perfect active sentence (`has(X↑id, 'Detect perfect tense active')`) must have an edge originating from its bottom node (`X↓id = Y@id`) which bears the label `AGENT` (`Y.label=AGENT`). It is this use of `has` in combination with the ancillary constraint that allows us to express a genuinely ternary supra-local relation as a WCDG-licensed binary constraint.
The ancillary constraint to be satisfied by the edge meeting the `SUBJ` dependency enforces exactly the set of properties previously identified for the detection of a perfect tense active sentence:

```
{X:SYN} : 'Detect perfect tense active' : ancillary : 1 :
  X.label = AUX
  & X↑cat = VAFIN
  & (X↑base = haben | X↑base = sein)
  & X↓cat = VVPP;
```

By employing the ancillary constraint as argument to `has`, we have effectively extended the scope of properties accessible on a neighbouring dependency—from access to a single static edge property to the full range of edge and node properties available. Constraint expressivity is enhanced because we can now create general custom predicates that neighbouring edges need to fulfill. Clearly, the conjunction of features `X↑cat=VAFIN & (X↑base=haben | X↑base=sein) & X↓cat=VVPP` was intractable with the static arguments to `is` or `has` presented in the previous sections.
The elegance of this approach lies in the fact that ancillary constraints of arbitrary complexity can now be employed as re-usable functional blocks to perform checks for linguistically intuitive, yet formally complex properties over and over again. Notable from a performance point of view is, that the WCDG implementation is such that, once an ancillary constraint has been evaluated for a given edge, its result will be cached and afterwards is available for repeated use at no extra cost computationally.

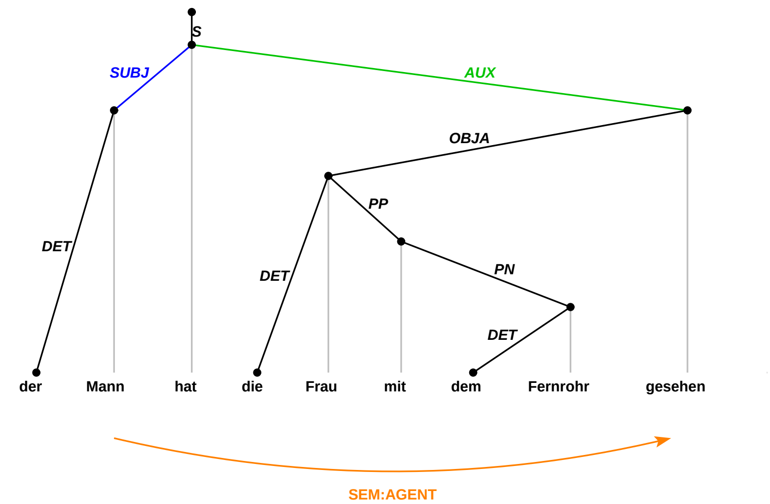


Fig. 1. AGENT assignment in German perfect tense active sentences

3.4 Cascading and Recursive Ancillary Constraints

The ancillary constraints presented so far are localised unary constraints—and as such provide full access to the properties of the next-neighbour edges above and below a given dependency in the syntax tree. As we will now illustrate, the syntax-semantic interface exhibits phenomena the modelling of which requires even higher expressivity than is provided by the extended localised unary ancillary constraints. We proceed to describe an additional expressivity enhancement that utilises cascading and recursive calls to ancillary constraints. This enables us to model properties spanning across arbitrarily large sections of the dependency tree, e.g. global properties, with just binary constraints.

As an example consider `AGENT` thematic role assignment in German passive sentences. The `AGENT` in a German passive sentence typically is embedded as the PP filler noun (`X.label=PN`) in a *von*-PP (`X.label=PP & X↓word=von`) which modifies the past participle of a full verb(`X↑cat=VVPP`) (see Figure 2).

Der Mann wird von der Frau [AGENT] mit dem Fernrohr gesehen.
(*The man is being seen by the woman* [AGENT] *with the telescope.*)

The full-verb past participle, in turn, must be correctly embedded in the lowest-lying `AUX` dependency in order for the sentence to be in passive voice. We can therefore formulate the constraint on the origin of the `AGENT` dependency in German passive sentences with the following cascade of ancillary constraints (colours refer to the highlights in Figure 2):

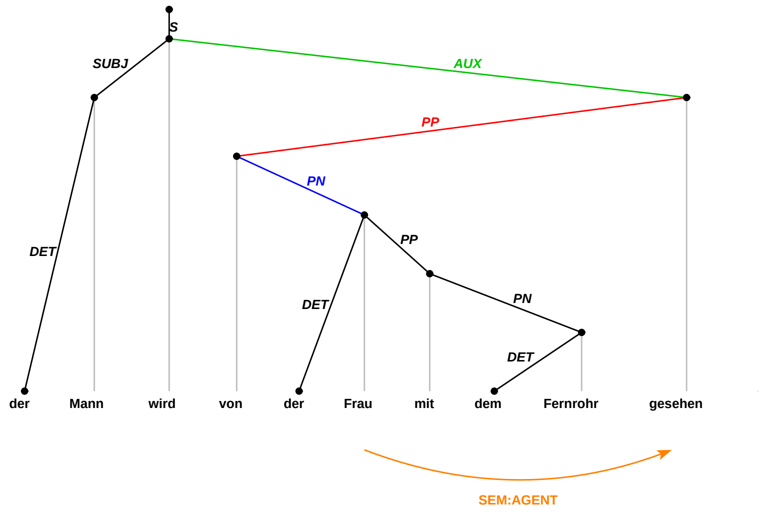


Fig. 2. AGENT assignment in German passives

Binary invocation constraint
For the pair of edges X (orange) and Y (blue) which share the same origin node (`X↓id = Y↓id`) we demand: If Y is a `PN` edge (blue) and the edge above it (red) satisfies the ancillary constraint `'Detect full-verb modifying von-PP in passive'`, then X (orange) must be an `AGENT` dependency.

```
{X:SEM,Y:SYN} :
  X↓from = Y↓from & Y.label = PN
  & is(Y↑id, 'Detect full-verb modifying von-PP in passive')
  -> X.label = AGENT;
```

Ancillary constraint #1: `'Detect full-verb modifying von-PP in passive'`.

The red edge above `PN` (blue) must be a full-verb modifying *von*-PP. This is tested for by ancillary constraint #3. The edge above the red edge must be the lowest-lying `AUX` edge in a passive construction (green), which is tested for in ancillary constraint #2.

```
  is(X↓id, 'Detect full-verb modifying von-PP')
  & is(X↑id, 'Detect passive bottom-up');
```

Ancillary constraint #2: `'Detect passive bottom-up'`.

The edge above the full-verb modifying `PP` must be a passive-marking `AUX` edge. Passive sentences are identified based on their lowest-lying `AUX` edge (green) which connects a past participle dependant with its auxilliary regent of base form *werden*. The regent's category depends on tense.

```
X.label = AUX
& X↓cat = VVPP
& ~has(X↓id, AUX)
& (X↑cat = VAFIN                            // Pres, Simple Past
   | (X↑cat = VAPP & is(X↑id, AUX) )        // Perfs, FutII, SubjII
   | (X↑cat = VAINF & is(X↑id, AUX) ) )     // FutI, SubjI
& X↑base = werden;
```

Ancillary constraint #3: `'Detect full-verb modifying von-PP'`.

A PP is of relevance to `AGENT`-assignment in a passive constructions if it contains the preposition *von* and attaches to the full verb's past participle.

```
X.label= PP & X↑cat = VVPP & X↓word = von;
```

Again, use of an ancillary constraint permits us to express in a binary constraint a condition which otherwise would have required a quarternary constraint construction relating the `PN`, `PP`, `AUX`, and `AGENT` dependencies.

A related, though again slightly more complex modelling task is to constrain the origin of the `AGENT` dependency in German active sentences. Due to the large number of structurally diverse active constructions in German it can be more convenient to model an active voice sentence as a sentence which is not in passive voice.[1] As mentioned above, German passives can be identified based on their lowest-lying `AUX` edge. Since the actual location of this edge depends on tense and mode, the constraint for its detection needs to be flexible. We employ a constraint which moves down the dependency tree by recursively invoking itself until it either finds an `AUX` edge satisfying the bottom-up criteria for passive detection or until it cannot descend further and fails altogether. The `AGENT` dependency in an active voice sentence may originate from the origin of the `SUBJ` dependency, while in a passive voice sentence it originates from the origin of the `PN` dependency contained in a full-verb modifying PP. Note that these conditions include the global properties *active* and *passive* voice. We can now conveniently formulate this complex requirement in the following recursive ancillary constraint invocation: Given an `AGENT` dependency, it originates either from a `SUBJ` edge in an active sentence or from a `PN` edge in a full-verb modifying *von*-PP in a passive construction.

```
X.label = AGENT ->
  (Y.label = SUBJ & ~has( Y↑id, 'Detect passive sentence top-down')) |
  (Y.label = PN
   & is(Y↑id, 'Detect full-verb modifying von-PP in passive'));
```

While the detection procedure for the full-verb modifying *von*-PP in a passive construction has been outlined above, the detection of the active construction merits further explanation. Starting from the `SUBJ` edge (blue) in Figure 3, the ancillary constraint first checks the green `OBJA` edge (green) for satisfaction of the ancillary

[1] This modelling decision may require justification beyond the scope of this paper. Suffice it here to say that replacing the indirect detection of active voice by a direct detection has no impact on our line of argument. The ancillary constraints merely need to be re-formulated to detect the structural features of an active voice sentence.

constraint `'Detect passive bottom-up'`. Since this is unsuccessful, it then continues to descend down the right hand-side of the dependency tree, progressing edge by edge, recursively re-invoking itself until it either finds an `AUX` edge satisfying the ancillary constraint and terminates, or until no further alternatives are available and it fails.

```
{X:SYN} : 'Detect passive top-down' : ancillary : 1 :
  X.label = AUX
  & (is( X↓id, 'Detect passive bottom-up')
     | has( X↓id, 'Detect passive top-down'));
```

This formulation is an expressive extension to the recursive tree traversal introduced in Section 3.2.

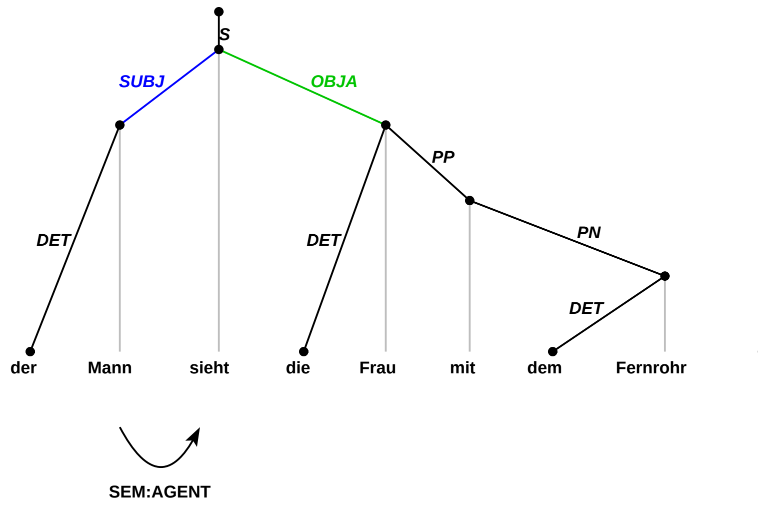


Fig. 3. AGENT assignment based on active/passive detection in German

The increase in constraint expressivity required in this modelling scenario arises from the fact that the dependencies constrained are farther apart in the dependency tree and thus are not contiguous anymore. So, although our approach for extending access to edge and node properties of neighbouring dependency edges is based on `is` and `has`, it is by no means limited in applicability to neighbouring edges. Before, supra-local properties would have needed to be tested for by a higher arity constraint, which was unavailable in WCDG's formalism. Now, such a higher arity constraint can be re-formulated as a suitably expressive binary constraint operating on a contiguous dependency structure that contains all edges we wish to predicate. Since an ancillary constraint can only extend access to one neighbouring dependency above or below, there is a linear relationship between the number of invocations to an ancillary constraint and the distance between the considered edges in the dependency tree.

4 Conclusions

In this paper we have outlined the gain in constraint expressivity achieved with the introduction of two additional predicates `is` and `has`. We showed how these predicates extend constraint access in the dependency tree and thus open up a path to the effective and efficient handling of higher arity constraints within the formal and operational limitations of the WCDG formalism. Motivated by examples from the syntax-semantics interface, we illustrated that the consecutive extension of the predicate syntax for `is` and `has` in combination with cascading and recursive invocations to ancillary constraints produces a significant increase in constraint expressivity. Most notably, we have demonstrated how global syntactic properties such as active or passive voice can be made accessible within a binary constraint dependency formulation.

5 Future work

Our work so far has focused on implementations involving unary ancillary constraints. With few changes the WCDG formalism can be extended to support the evaluation of binary ancillary constraints as well. A systematic investigation into the effects of this is pending.
From a theoretical point of view, a formal analysis of the expressivity enhancements achieved with `is` and `has` appears challenging and rewarding. While we have focused on the use of `is` and `has` to solve specific modelling tasks, we conjecture that the full expressive potential resulting from the use of these predicates in combination with ancillary constraints has not yet been exhausted.

6 Acknowledgements

Parts of this research were supported by the CINACS graduate research project funded by the German Research Foundation (DFG). We would also like to express our sincere thanks to the four anonymous reviewers for their detailed and constructive feedback on an earlier version of this paper.

References

[Charniak and Johnson, 2005] Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proc. ACL 2005*, pages 173–180, Ann Arbor, Michigan, June. Assoc. for Computational Linguistics.

[Chomsky, 1986] Noam Chomsky. 1986. *Barriers.* MIT Press.

[Duchier, 1999] Denys Duchier. 1999. Axiomatizing dependency parsing using set constraints. In *Proceedings of MOL6*, pages 115–126, Orlando/USA.

[Foth et al., 2000] Kilian A. Foth, Wolfgang Menzel, and Ingo Schröder. 2000. A Transformation-based Parsing Technique with Anytime Properties. In *4th Int. Workshop on Parsing Technologies, IWPT-2000*, pages 89–100.

[Foth, 2007] Kilian A. Foth. 2007. *Hybrid Methods of Natural Language Analysis.* Ph.D. thesis, Universität Hamburg.

[Maruyama, 1990] Hiroshi Maruyama. 1990. Structural disambiguation with constraint propagation. In *Proc. 28th Annual Meeting of the ACL (ACL-90)*, pages 31–38, Pittsburgh, PA.

[Menzel and Schröder, 1998] Wolfgang Menzel and Ingo Schröder. 1998. Decision procedures for dependency parsing using graded constraints. In Sylvain Kahane and Alain Polguère, editors, *Proc. Coling-ACL Workshop on Processing of Dependency-based Grammars*, pages 78–87, Montreal, Canada.

[Schulz et al., 2005] Michael Schulz, Stefan Hamerich, Ingo Schröder, Kilian Foth, and Tomas By, 2005. *[X]CDG User Guide.* Natural Language Systems Group, Hamburg University, Hamburg, Germany.

Partial Ordering Constraints for Representations of Context in Ambient Intelligence Applications*

Hedda R. Schmidtke and Woontack Woo

GIST U-VR Lab.
Gwangju 500-712, Korea
{schmidtk,wwoo}@gist.ac.kr

Abstract. We give an overview of a framework within which Ambient Intelligence (AmI) applications could be realised. We start from available sensors/actuators producing and consuming numerical data at high frequencies and available logic-based knowledge representation and reasoning systems using expressive logical languages. We argue that a constraint-based partial-order reasoning system for six aspects of context can be used as a central component that helps to bridge the gap between information from sensors and logic-based reasoning. Our work is motivated by, and oriented towards, cognitive representation and processing mechanisms.

1 Introduction

Ambient intelligence (AmI) has been suggested as a key application scenario for artificial intelligence techniques. In AmI scenarios (such as those compiled by Ducatel et al., 2001) smart devices – sensors, actuators, and information services – cooperate to provide services to users that make sense in a given situation and environment. Intelligent agent programs (see Wooldridge, 1999, for an overview) installed on such devices and connected by a network infrastructure can form coalitions, which together achieve (sub-)goals derived from user profiles or user-defined scripts.

A central component of such a spatially distributed intelligent system is a *context model*, i.e. a representation of the situation and environment that is at least partially shared among components. In particular, such a representation has to provide information of how the contexts of different components are related to each other and to the current physical context. This task can be formulated in terms of constraints on the relations between contexts.

Starting from the context model of Jang et al. (2005), we investigated a representation based on six parameters of context (for brevitiy called the *5W1H* parameters): the context of an interacting coalition of agents in this model is fully described if we know *who* interacts *when* and *where* with *what why* and *how*, that is, if we know the users, time, place, objects, events/actions, and the

* This research is supported by the UCN Project, the MIC 21C Frontier R&D Program in Korea.

states/conditions of an interaction. Following the analysis of Benerecetti et al. (2000), we assume that contexts correspond to portions of the world, which are perceived with a certain level of detail, and from a certain perspective. Accordingly, relations between contexts can be of three different types: mereologic, part-of relations (in a wide sense Bittner et al., 2004), approximation relations (particularly granularity, Schmidtke and Woo, 2007), and perspectival relations (such as cardinal directions in geographic space, Schmidtke, 2001).

In previous work (Hong et al., 2007), we introduced a logical language with mereologic relations for four of these six parameters, namely classes of users (*who*-domain) and classes of objects (*what*) with respective taxonomic relations, on the one hand, and temporal and spatial extents with mereologic relations, on the other hand. In Schmidtke et al. (2008), we extended the language to cover all six parameters, so that knowledge about states and events can also be represented. We characterised the six relations as partial orders. Consequently, a system for reasoning over such constraints on contexts can make use of partial order reasoning, which can be implemented in an efficient way using graph-based representations. In this paper, we show how such a reasoning system can be embedded into a cognitively motivated three-layered agent architecture (Lee et al., 2007) and we outline how AmI applications can be developed in this framework.

The paper has two main parts. In the first part (Sect. 2), we shortly introduce our cognitively motivated system architecture for AmI environments. We sketch how information from sensors is interpreted and translated into constraints on relations between sensor contexts and more abstract contexts; and we outline how this knowledge can be used to trigger or adapt behaviour of actuators and services, in order to reach sub-goals given from a planner or user-defined script. The second part (Sect. 3) gives details about the language and discusses how it can be used for representing knowledge about contexts. We sketch in how far our conceptualisation of context fulfils requirements from cognitive theories and qualitative reasoning. After discussing the architecture and language, we illustrate how AmI applications can be developed and integrated with a simple example (Sect. 4).

2 System Architecture

We developed a cognitively motivated layered multi-agent framework (Lee et al., 2007) to realise the requirements outlined above. We regard human cognition as the ideal model for AmI. In particular, we try to address the problem of how to bridge the gap between sensory data and logic-based representations with a cognitively motivated approach.

2.1 Processing Contextual Information

Agents in our model represent separate cognitive facilities of an intelligent system, which can be a single smart object or a whole smart environment composed of many individual sub-systems. We assume three types of agents corresponding to layers of representational abstraction found in human cognition. Following the analysis of Gärdenfors (2005), we distinguish between three types of information processing and action of an intelligent system in its environment:

- In the case of *transduction*, behaviour is generated directly in response to perceptual input. Internal representations are not necessary. An example for this type of behaviour in animals is phototaxis.
- *Cued* representations are mental representations of objects or events that are perceptually directly accessible in the current context, or triggered by some recent situation. Using inference, e.g., categorisation, over these representations, an intelligent system can react to the situation.
- *Detached* representations are independent from the current context. Imagination about an object that does not exist or situations that have not, or not yet, happened are examples of inference over these representations. Planning requires detached thinking.

We identify these layers of abstraction with critical time frames of tasks in a dynamically changing computing environment.

Corresponding to transduction, the layer of *continuous responsiveness* is bound to the time frame of visual continuity, for which we assumed a maximal delay of 40ms. Sensors and actuators should react without any perceivable delay.

The layer of *immediate reaction*, on the other hand, is bound by the time frame of learnt reactions to change in an environment. Inference over representations cued by perceptual input are necessary to react to a situation. We chose a threshold of one second maximal delay for this layer. In the case of an obstacle appearing suddenly on a road, for instance, a driver would within one second be able to generate a more or less adequate reaction, such as breaking, steering, or a learnt sequence of these actions. Likewise in verbal communication, human beings usually require some response from a dialogue partner within the time frame of a second, even if it is just a nod or 'hmm' for signalling demand for a larger time frame. Our efforts were concentrated on realising this layer. In our implementation, the current context as retrieved from the responsive layer is sorted into a hierarchy of abstract contexts according to the constraints that can be derived from sensor readings.

The third layer is the layer of *pro-activity*. Intelligent actions in a changing but predictable world require computational processes of higher complexity. For reasoning and planning about anticipated situations, detached representations are necessary. Typical AI problems, which require expressive logic formalisms or planning are computed on this layer. In our current system, we do not implement this layer. We consider plans to be given to the system in terms of condition-action pairs and a partial ordering, for which the second layer provides reasoning support.

Figure 1 illustrates the three types of agents. In this example, a group of five agents is organised as a coalition for solving a certain problem: three responsive agents (smart sensors and actuators) are connected to a reactive component that analyses sensor inputs and triggers accurate responses of actuator components; the analysed representation of the current context is also given on to a pro-active component, which generates a higher-level representation of the current context that can be used to modify the behaviour of the reactive component in case of unforeseen difficulties.

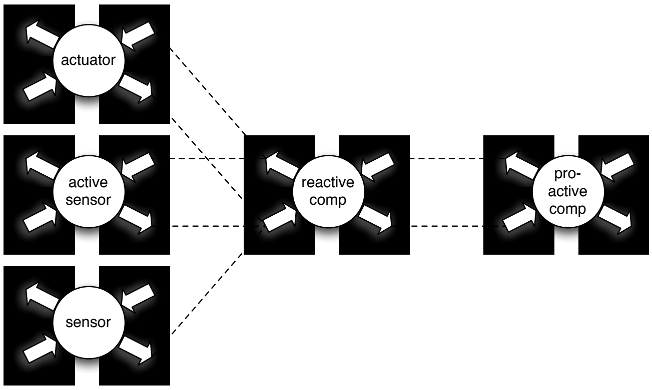


Fig. 1. Simple example for a coalition of five agents.

2.2 Representations on the Three Layers

The representations used on each layer have to differ so as to support the individual tasks within the required time frame and flexibility. Table 1 summarises the differences in perspective assumed for the three layers and illustrates the differences in representations for the example of spatial information.

Table 1. Representation and processing of contextual knowledge on the three layers. The example shows how spatial information could be handled on the three layers.

Layer	Representation	Processing	Example
Responsive	numeric, basic data types	procedural, non-symbolic	loc = $(35.226°N126.842°E)$, at precision $\pm 0.001°$
Reactive	context-oriented, qualitative relations	graph-based, constraint-based	[currC1 $\sqsubseteq_{\text{where}}$ Gwangju] $\wedge$ [Gwangju $\sqsubseteq_{\text{where}}$ SKorea]
Pro-active	logic-based	logic-based reasoning	$\forall x : \text{city}(x) \rightarrow \exists t : \text{trainCon}(t) \wedge \text{reach}(t, x)$

On the layer of responsiveness, computation is performed directly on the mainly numerical input from sensors. This allows for especially fast processing, as required for algorithms at the sensor interface. The result of analysis can be

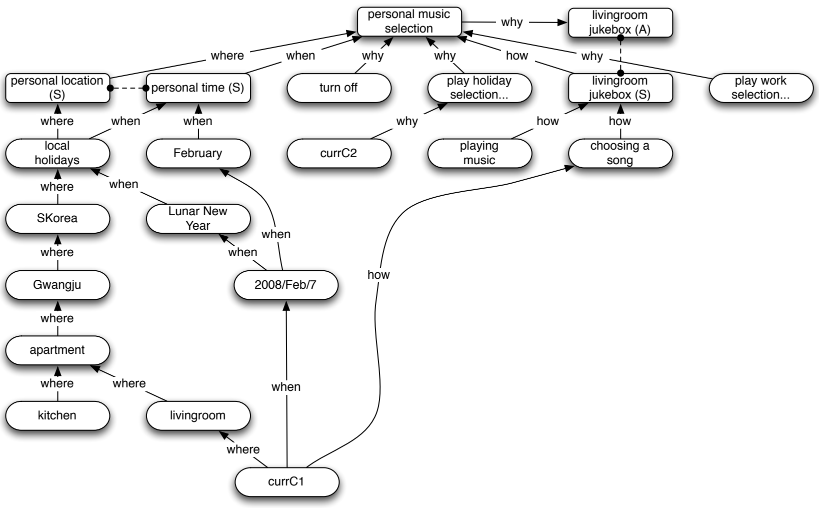


Fig. 2. The current context as retrieved from sensors can be sorted into a graph-based representation in which $\sqsubseteq_m$-constraints can be read from the edges. The statement [currC1 $\sqsubseteq_{\text{where}}$ Gwangju]$\wedge$[Gwangju $\sqsubseteq_{\text{where}}$ SKorea], from Tab. 1 can be retrieved from the above graph. Rounded rectangles represent agents, which are notified whenever their node in the graph is activated.

transferred directly to an actuator, or it can be transformed into a representation of constraints describing the *current context*. These constraints are distributed to recipient agents on the reactive layer.

On the reactive layer, information about parameters of the current context from different sources is integrated into a graph-based representation of the current state of the world around an agent (an example is shown in Fig. 2). Knowledge about the current context can be represented by activating nodes in the graph to which the current context is related. For long term storage, a new node representing the current context can be added to the graph by linking it to related nodes. Nodes in the graph represent previous contexts, knowledge about contexts given in application ontologies, and knowledge about contexts described in user profiles and scripts. In addition, nodes can be connected to agents. Activating such a node results in a description about the current context being sent to the agent.

Each edge of the graph corresponds to one of six specific relations, so that all information encoded in the graph can easily be translated into a logic-based format, which can be sent to recipients on the pro-active layer for further analysis. The identified situation can thus activate the next action to be executed from a given action plan, which in turn can automatically trigger or update actuators (see Sect. 4 for an example). A similar mechanism of spreading activation along edges with a specific semantics has been used by Kokinov (1999) to model a broad range of phenomena of context-dependency in human cognition. Our variant for AmI differs mainly in two respects: first, the types of relations are restricted to the six partial ordering relations, so that the resulting graph for each relation is a directed acyclic graph; and second, nodes can be connected to arbitrary types of agents, so that arbitrary types of external applications can be triggered.

For the pro-active layer, logic-based knowledge representation systems can be used. The input from the reactive layer is sufficiently abstract, so as to be accessible to higher-level reasoning. Reasoning tasks necessary for AmI systems, such as planning or diagnosis, can thus be decoupled from the reactive systems. Information about the current context can be de-contextualised, in order to receive a more objective and far-reaching representation about ongoing and past processes. The behaviour of a reactive component can be generated or modified by a planner on the pro-active layer. Agents on the reactive layer can execute plans, given as partially ordered sequences of actions in the *why*-graph. However, the planning task itself can have a high complexity and might in general require more than one second, that is, more than the time frame of the reactive layer permits. Planning itself therefore belongs to the pro-active layer.

3 Representations of Context on the Reactive Layer

The reactive layer has been the focus of our research for three reasons. First, the time frame of one second demands a specialised fast reasoning mechanism for all domains of parameters of context. Second, the symbol grounding problem has to be handled on this layer, between the sensory input and the logic-based representations on the one hand, and between underspecified logic-based plans and the concrete parameters required by actuators, on the other hand. The third reason for highlighting the special role of the reactive layer, is that the notion of context is established and dealt with mainly on this layer. Agents on the responsive layer produce/consume absolute values, and agents on the pro-active layer de-contextualise the information they obtain. The cued representations, in contrast, are representations activated by, and relative to, the current physical context.

Crucial to realising this layer is an appropriate representation and reasoning mechanism that can handle the non-symbolic contextual information format of sensors/actuators as well as logic-based representations that can be processed on the pro-active layer. As sketched above, we assume that numeric values retrieved from sensors can be interpreted as basic constraints on the current context of the device. The context representation is based on partial ordering relations between six specific aspects of a context. With this approach we follow a similar strategy as in logics of indexicals (Forbes, 1989), which use a fixed, finite number of parameters. However, we represent time and location not with point-like timestamps and coordinate locations but with a qualitative representation based on time intervals and spatial regions, respectively. In this manner, we can represent the relation between partial contexts as a containment relation and we can allow for coarse location and time, such as "this country" and "this week" in combination with fine grained notions, such as "this room" and "this moment."

The relation of containment has been studied in theories of mereology. Properties stated in mereologic theories form the basic ontology of our framework, without implying that we only deal with part-whole structures in the narrow sense of spatial or temporal part-whole relations. We specify six partial ordering relations ($\sqsubseteq_{\text{who}}, \sqsubseteq_{\text{what}}, \sqsubseteq_{\text{when}}, \sqsubseteq_{\text{where}}, \sqsubseteq_{\text{why}}, \sqsubseteq_{\text{how}}$) relating between the six parameters that fully describe a context c in this approach.

The characterisation follows ideas in other approaches that combine knowledge about spatial relations with temporal knowledge or knowledge about concepts (Eschenbach, 2004; Donnelly, 2004; Bittner et al., 2004). However, we added reasoning about two other partial ordering relations for the domains of states and tasks: reasoning about conditions and states which we consider to be constrained by an implication relation (*how*-domain), and reasoning about tasks, actions and events in partially ordered causal structures (*why*-domain).

3.1 Logical Formalism

The logical language consists of the recursively defined context terms, representing contexts, constraints over *context terms* for relating contexts, and formulae for combining such descriptions. The set of context terms is defined as the smallest set over a set of atomic context terms that fulfils:

1. All atomic context terms and the special symbols $\top$ (called: the *maximal context*) and $\bot$ (the *impossible context*) are context terms.
2. If c and d are context terms then $-c$ (*complement*), $(c \sqcup d)$ (*summation*), and $(c \sqcap d)$ (*intersection*) are also context terms.

A context term is called *context literal* if it is an atomic context term or the complement of an atomic context term. A *context formula* is formed from two context terms with one of six sub-context relations:

1. If c and d are context terms, then $[c \sqsubseteq_{\text{who}} d]$, $[c \sqsubseteq_{\text{what}} d]$, $[c \sqsubseteq_{\text{when}} d]$, $[c \sqsubseteq_{\text{where}} d]$, $[c \sqsubseteq_{\text{why}} d]$, and $[c \sqsubseteq_{\text{how}} d]$ are atomic formulae.
2. If ϕ is a formula, then $\neg\phi$ also is a formula.
3. If ϕ and ψ are formulae, then $(\phi \vee \psi)$ and $(\phi \wedge \psi)$ are formulae.

Table 2 summarises the intended interpretations for the atomic formulae.

We introduce further relations as abbreviations of formulae. Here, and in the following we make use of schemata to abbreviate definitions: m denotes one of

Table 2. Syntax and reading of sub-context relations.

Syntax	Reading
$[c \sqsubseteq_{\text{who}} d]$	c is a social sub-context of d
$[c \sqsubseteq_{\text{what}} d]$	c is a conceptual sub-context of d
$[c \sqsubseteq_{\text{when}} d]$	c is a temporal sub-context of d
$[c \sqsubseteq_{\text{where}} d]$	c is a spatial sub-context of d
$[c \sqsubseteq_{\text{how}} d]$	c is a conditional sub-context of d
$[c \sqsubseteq_{\text{why}} d]$	c is a task sub-context of d

the six parameters of context, i.e. *who*, *what*, *when*, *where*, *why*, or *how*.[1]

$$[c \sqsubseteq d] \overset{def}{\Leftrightarrow} [c \sqsubseteq_{\text{who}} d] \wedge [c \sqsubseteq_{\text{what}} d] \wedge [c \sqsubseteq_{\text{when}} d] \wedge [c \sqsubseteq_{\text{where}} d] \wedge [c \sqsubseteq_{\text{why}} d] \wedge [c \sqsubseteq_{\text{how}} d] \tag{D1}$$

$$[c \bigcirc_m d] \overset{def}{\Leftrightarrow} \neg[c \sqcap d \sqsubseteq_m \bot] \tag{D2}$$

$$[c \bigcirc d] \overset{def}{\Leftrightarrow} \neg[c \sqcap d \sqsubseteq \bot] \tag{D3}$$

$$[c =_m d] \overset{def}{\Leftrightarrow} [c \sqsubseteq_m d] \wedge [d \sqsubseteq_m c] \tag{D4}$$

For two arbitrary context terms, c is a called a *sub-context* of d ($\sqsubseteq$) if it is socially, conceptually, temporally, spatially, and with respect to conditions, and tasks a sub-context of d (D1). Two contexts overlap with respect to the parameter m if their intersection is an m-sub-context (D2). The contexts c and d overlap if they have a common sub-context in any domain (D3), that is, if they overlap with respect to any one domain. Finally, (D4) defines that c and d are equal with respect to domain m if they are m-sub-contexts of each other.

Before we illustrate particular aspects of the six domains and illustrate the intended meaning for the 21 relations (the general $\sqsubseteq$, $\bigcirc$, together with identity $=$, and the six variants of $\sqsubseteq_m$, $\bigcirc_m$, and $=_m$), we shortly list some basic properties of partial ordering relations, which form the foundation for the semantics of the logical language (see Schmidtke et al., 2008, for a specification of the semantics based on Kripke frames). The below statements hold for arbitrary context terms x, x', x_1 independently from their meaning. We do not attempt an axiomatisation in this paper but refer to the wealth of known results from research on the properties of mereologic relations (cf. particularly Donnelly, 2004; Link, 1983).

For each of the relations $\sqsubseteq_m$, we state that $\sqsubseteq_m$ be reflexive (1) and transitive (2). Antisymmetry does not hold for $\sqsubseteq_m$ since two contexts that are identical with respect to one parameter may disagree with respect to another parameter. However, the relations $=_m$, which hold between contexts that agree on the parameter m, are equivalence relations, i.e reflexive (3), transitive (4), and

[1] For brevity, we also consider the additional logical conjunctives $\rightarrow$ and $\leftrightarrow$ to be defined as usual, and introduce rules for saving brackets. The following precedence of logical connectives is assumed $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.

symmetric (5).

$$[x \sqsubseteq_m x] \tag{1}$$

$$[x_1 \sqsubseteq_m x_2] \wedge [x_2 \sqsubseteq_m x_3] \rightarrow [x_1 \sqsubseteq_m x_3] \tag{2}$$

$$[x =_m x] \tag{3}$$

$$[x_1 =_m x_2] \wedge [x_2 =_m x_3] \rightarrow [x_1 =_m x_3] \tag{4}$$

$$[x_1 =_m x_2] \rightarrow [x_2 =_m x_1] \tag{5}$$

The overlap relations $\bigcirc_m$ are reflexive for non-empty contexts (6) and symmetric (7) for any context term.

$$\neg[x \sqsubseteq \bot] \rightarrow [x \bigcirc_m x] \tag{6}$$

$$[x_1 \bigcirc_m x_2] \rightarrow [x_2 \bigcirc_m x_1] \tag{7}$$

With these properties stated, we can now illustrate their intended meaning with respect to the six domains with examples of statements in the logical language.

3.2 Modelling Knowledge about Individual Domains

Similar to the relation of containment between collections axiomatised by Bittner et al. (2004), the intended meaning for $\sqsubseteq_{\text{who}}, \sqsubseteq_{\text{what}}$ is *group inclusion* on groups of agents and groups of objects, respectively. With this interpretation, the properties stated above are intuitively plausible. An example for transitivity of $\sqsubseteq_{\text{who}}$, for instance, is given in (8): a group of users *admin* included in the group of users *staff* is also included in any group that includes the latter, such as *notificationRecipient*. Knowing that Bob is in the group of administrators, we know that he will receive a notification (9).

$$[\mathit{admin} \sqsubseteq_{\text{who}} \mathit{staff}] \wedge [\mathit{staff} \sqsubseteq_{\text{who}} \mathit{notificationRecipient}] \\ \rightarrow [\mathit{admin} \sqsubseteq_{\text{who}} \mathit{notificationRecipient}] \tag{8}$$

$$[\mathit{bob} \sqsubseteq_{\text{who}} \mathit{admin}] \rightarrow [\mathit{bob} \sqsubseteq_{\text{who}} \mathit{notificationRecipient}] \tag{9}$$

In our mereologic framework, a single agent, such as *bob*, or a single object in an interaction is always interpreted as a group of one agent or object. That is, *bob* is not interpreted by a token corresponding to the user Bob but by the singleton set containing this token. This may seem to be a counter-intuitive by-effect of the mereologic axiomatisation. With respect to the cognitive motivation of our approach however, we might remark that this property has been shown by Link (1983) to have distinct advantages for the formal specification of nouns in natural language semantics: singular and plural meanings of a noun can be represented as having the same type with the mereologic, but not with a set-theoretic characterisation.

For time and space, we also use a mereologic interpretation for the two relations $\sqsubseteq_{\text{when}}, \sqsubseteq_{\text{where}}$. A mereotopologic characterisation of spatial entities, which starts from a mereologic basis, is discussed in detail by Casati and Varzi (1999).

For the domain of time, the discussion on interval-based calculi started by Allen (1984) serves as a reference. However, our notion is not restricted to convex intervals but covers arbitrary sums and intersections of intervals, such as generalised intervals (Ligozat, 1998), and moments, i.e. intervals with no extension.

Using this concept we could for instance represent a context c_{21} in which Allen and Beth are at Incheon Airport at January 1st in 2007.

$$[allen \bigcirc_{\text{who}} c_{21}] \wedge [beth \bigcirc_{\text{who}} c_{21}] \\ \wedge [c_{21} \sqsubseteq_{\text{when}} [\text{Jan 1, 2007}]] \wedge [IncheonAirport \bigcirc_{\text{where}} c_{21}]$$

Allen and Beth are users of the system in the context of c_{21}; c_{21} is at some time during January 1st and overlaps the region of Incheon airport.

In the case of agents and objects, taxonomies can be created by summation and intersection. It is important to note here, that we cannot construct arbitrary taxonomies and partonomies of locations and times with the operation of summing locations and times, respectively. For instance, it makes sense to construct a time *morningGMT* as the sum of all intervals between, e.g. 4 and 12 GMT. Likewise, we can define times *afternoonGMT* (12–18), *eveningGMT* (18–22), and *nightGMT* (22–4). The sum of these temporal entities, however, would be the trivial interval covering the whole of time, but not the set of days.

From an AI point of view, the parameter of *how* corresponds to states that hold in a certain context, whereas the parameter of *why* corresponds to events that occur, and actions that are executed in the context. Causal dependencies can be expressed by combining knowledge about states, events, and time. The context term operators, sum, intersection, and complement, can be used to combine states or events. The logical formalism thus provides in-built reasoning about such combinations, for which classical AI approaches based on first-order logic have to describe specific reification mechanisms (Galton, 2006). Our logical language, being designed for representing knowledge on the reactive layer, thus can support generating a description of the current context and triggering actions according to expected situations. However, it does not have the expressiveness and reasoning capabilities necessary to support reasoning about possible time lines, for instance. For this task more expressive formalisms, and thus reasoning on the pro-active layer, would be needed.

In concrete AmI applications, developers might use the *how*-part to represent status information, such as “on vacation,” or information about states of a component, such as “playing music” in Fig. 2. Another type of states particularly relevant to AmI applications is qualitative information about environment parameters derived from quantitative sensory data. For instance, the state “cold weather” might be defined as holding whenever a temperature sensor yields a temperature below some threshold, such as “$0°C$.” Agents that yield such qualitative statements should themselves take the context of the measurement in account, so that the context-dependency of adjectives, such as “cold,” can be reflected: A summer day in Rome, for instance, would have a higher threshold for being called “cold” than a winter night in Moskau. The partial order structure underlying the *why*-component represents causal dependencies of events in a context as given, for instance, in an action description (Brézillon, 2005) or from a planner. User preferences could also be encoded in such task structures.

In the example of Fig. 2, the time and location sensors activated all nodes that are ancestors of the node *currC1* along the *when* and *where* edges. Also the jukebox works as a sensor, notifying all related nodes about its current state along the *how*-edge. The music selection reacts to the context *currC1* by activating all nodes above *currC2*. In the example of Fig. 2, tasks of the music selection agent are thus handed on to the jukebox actuator along the edges of the *why*-graph.

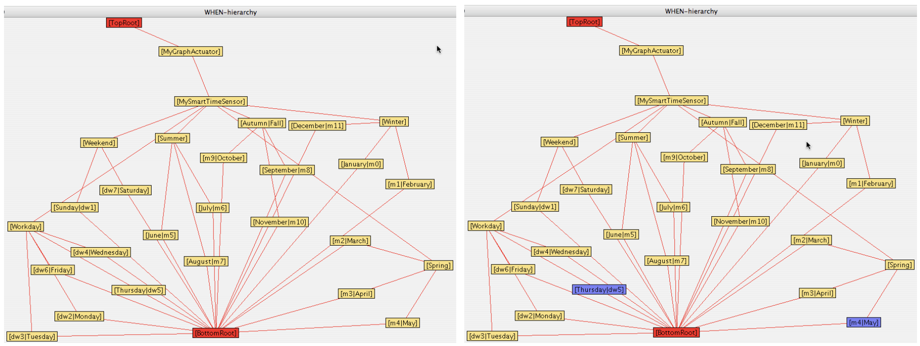
4 Example: Developing an AmI Application

The core components of the framework have been implemented. Classes for implementing the reactive components of the architecture, described in Sect. 2.2, and a knowledge base with a simple reasoning mechanism for the logical language have been realised in Java. Several test applications using this core framework are currently under development. The reasoning mechanism is implemented as a tableau prover supported by six directed acyclic graphs, which represent the partial ordering constraints between atomic context terms. The knowledge base receives input in the form of *profiles*, in which components describe their basic vocabulary and their own relation with respect to this vocabulary and other components.

When a sensor produces a value, a reactive component S wrapped around this sensor translates this value into a possibly complex context logic term c using its vocabulary. It then notifies the knowledge base about c. The intended meaning is that S activates c as a description of the current context as perceived and analysed by S. The reasoning mechanism then determines where the sent context term would be positioned in the directed acyclic graphs of the knowledge base. All related nodes, that is, all nodes that would be ancestors or descendants of a node corresponding to c are then notified about c by the knowledge base. Using this procedure, the description of the current context as generated by S is sent to all components to which it is relevant.

Figure 3 shows a screen shot of a graph drawing application being notified by a time sensor. The application displays the graph of the $\sqsubseteq_{\text{when}}$-relation, as it is represented in the knowledge base. It reacts to notifications by simply highlighting the most specific nodes activated. We developed this application as a tool for debugging. For being notified about terms related to the vocabulary of the time sensor the graph actuator's profile needs only a single line: `[MySmartTimeSensor pwhen MyGraphActuator]`. Figure 4 shows the context knowledge loaded with the time sensor.

Two classes were implemented for realising the time sensor and graph actuator, respectively. The complete code for the time sensor is shown in Fig. 5. It simply constructs a context term from integer values read from a time stamp. The notification sent by the time sensor results in statements, which identifiy the current time as being a time interval that can be described as the intersection



(a) (b)

Fig. 3. The graph actuator at two times: a) in idle state, b) after being notified of the context term `(y2008 sqand (dw6 sqand (dm15 sqand (h2 sqand (ampm1 sqand (hd19 sqand (min49 sqand (sec50 sqand m4))))))))` (here `sqand` is the ASCII-encoding for $\sqcap$).

```
[dw1 eqwhen Sunday]
[dw2 eqwhen Monday]
[dw3 eqwhen Tuesday]
[dw4 eqwhen Wednesday]
[dw5 eqwhen Thursday]
[dw6 eqwhen Friday]
[dw7 eqwhen Saturday]
[m0 eqwhen January]
[m1 eqwhen February]
[m2 eqwhen March]
[m3 eqwhen April]
[m4 eqwhen May]
[m5 eqwhen June]
[m6 eqwhen July]
[m7 eqwhen August]
[m8 eqwhen September]
[m9 eqwhen October]
[m10 eqwhen November]
[m11 eqwhen December]
[Workday pwhen MySmartTimeSensor]
[Weekend pwhen MySmartTimeSensor]
[Spring pwhen MySmartTimeSensor]
[Fall pwhen MySmartTimeSensor]
[Winter pwhen MySmartTimeSensor]
[Summer pwhen MySmartTimeSensor]
[Monday pwhen Workday]
[Tuesday pwhen Workday]
[Wednesday pwhen Workday]
[Thursday pwhen Workday]
[Friday pwhen Workday]
[Saturday pwhen Weekend]
[Sunday pwhen Weekend]
[March pwhen Spring]
[April pwhen Spring]
[May pwhen Spring]
[June pwhen Summer]
[July pwhen Summer]
[August pwhen Summer]
[September pwhen Fall]
[October pwhen Fall]
[November pwhen Fall]
[December pwhen Winter]
[January pwhen Winter]
[February pwhen Winter]
[Fall eqwhen Autumn]
```

Fig. 4. The profile file for the time sensor `MySmartTimeSensor.clf` (`pwhen` is the ASCII encoding for $\sqsubseteq_{\text{when}}$, `eqwhen` stands for $=_{\text{when}}$).

```
/** A sensor that notifies of the current time using Java's
Calendar class. **/
public class SmartTimeSensor extends SmartSensor {
Calendar cal;
   public SmartTimeSensor() {
      super(ContextProcess.TimeFrame.REACTIVE);
      this.cal = Calendar.getInstance();
   }
   /** Notifies about the time using Java's Calendar class. **/
   protected void doStep() {// called every second
      cal.setTime(new Date());
      if (cal.get(Calendar.SECOND)%10 == 0) {
         String year = "(y"+cal.get(Calendar.YEAR)+" sqand ";
         String dayofweek =
            "(dw"+cal.get(Calendar.DAY_OF_WEEK)+" sqand ";
         String dayofmonth =
            "(dm"+cal.get(Calendar.DAY_OF_MONTH)+" sqand ";
         String hour1 = "(h"+cal.get(Calendar.HOUR)+" sqand ";
         String ampm = "(ampm"+cal.get(Calendar.AM_PM)+" sqand ";
         String hourofday =
            "(hd"+cal.get(Calendar.HOUR_OF_DAY)+" sqand ";
         String minute = "(min"+cal.get(Calendar.MINUTE)+" sqand ";
         String seconds = "(sec"+cal.get(Calendar.SECOND)+" sqand ";
         String month = "m"+cal.get(Calendar.MONTH);
         mem.notify(year+dayofweek+dayofmonth+hour1
         +ampm+hourofday+minute+seconds+month+"))))))))");
      }
   }
   /** This sensor does not react to any input. **/
   public void onNotify(String str) {}
}
```

Fig. 5. The implementation of the time sensor class.

```
public static void main (String[] args) {
   SimpleCKB ckb = new SimpleCKB();
   SmartTimeSensor sim = new SmartTimeSensor();
   sim.init(ckb,"profiles/MySmartTimeSensor.clf","MySmartTimeSensor");
   xmpl.caedit.GraphActuator graph =
   new xmpl.caedit.GraphActuator(T5W1H.WHEN);
   graph.init(ckb,"profiles/MyGraphActuator2.clf","MyGraphActuator");
}
```

Fig. 6. Loading and starting the test application.

of several intervals represented in the *when*-hierachy in the knowledge base. The graph actuator class correctly identified the most specific active nodes shown to be `dw5` (temporally equivalent to `Thursday`) and `m4` (temporally equivalent to `May`). It was notified because it is itself an ancestor node of activated nodes.

The main method of the test application (Fig. 6) initialises the knowledge base (class `SimpleCKB`) and creates an instance of the time sensor. The profile of the time sensor is loaded into the knowledge base and the sensor is associated with the node `MySmartTimeSensor`, the top-node of all terms mentioned in the profile. The graph actuator is initialised to show the *when*-domain of the knowledge base. Its simple profile places it above the time sensor's node in the *when*-hierarchy. It can visualise any other class if another profile is loaded.

5 Outlook and Conclusions

We gave an overview of a framework within which AmI applications can be realised. We started from available sensors/actuators producing and consuming numerical data at high frequencies and available logic-based knowledge representation and reasoning systems using expressive logical languages. We argued that a constrained-based partial-order reasoning system is a core component that helps to bridge the gap between sensors and logic-based reasoning. Our work is motivated by, and oriented towards, cognitively adequate representations and processing mechanisms. The current framework is limited in that only partial ordering relations are represented. In future works, we will focus on integrating further types of relations, in particular, granularity relations and perspectival relations.

References

J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual reasoning distilled. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(3):279–305, 2000.

T. Bittner, M. Donnelly, and B. Smith. Individuals, universals, collections: On the foundational relations of ontology. In A. Varzi and L. Vieu, editors, *Third Conference on Formal Ontology in Information Systems*. IOS Press, 2004.

P. Brézillon. Task-realization models in contextual graphs. In A. K. Dey, B. N. Kokinov, D. B. Leake, and R. M. Turner, editors, *International Conference on Modeling and Using Context*, volume 3554 of *LNCS*, pages 55–68. Springer, 2005.

R. Casati and A. C. Varzi. *Parts and Places: the Structure of Spatial Representations.* MIT Press, 1999.

M. Donnelly. A formal theory for reasoning about parthood, connection, and location. *Artifcial Intelligence*, 160(1-2):145–172, 2004.

K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman. Scenarios for ambient intelligence in 2010. Technical report, ISTAG, 2001.

C. Eschenbach. How to interweave knowledge about object structure and concepts. In D. Dubois, C. A. Welty, and M.-A. Williams, editors, *Principles of Knowledge Representation and Reasoning*, pages 300–310. AAAI Press, 2004.

G. Forbes. Indexicals. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume IV, pages 463–490. D. Reidel, 1989.

A. Galton. Operators vs. arguments: the ins and outs of reification. *Synthese*, 150:415–441, 2006.

P. Gärdenfors. The detachment of thought. In C. Erneling and D. Johnson, editors, *The mind as a scientific subject: between brain and culture*, pages 323–341. Oxford University Press, 2005.

D. Hong, H. R. Schmidtke, and W. Woo. Linking context modelling and contextual reasoning. In A. Kofod-Petersen, J. Cassens, D. B. Leake, and S. Schulz, editors, *4th International Workshop on Modeling and Reasoning in Context (MRC)*, pages 37–48. Roskilde University, 2007.

S. Jang, E.-J. Ko, and W. Woo. Unified user-centric context: Who, where, when, what, how and why. In H. Ko, A. Krüger, S.-G. Lee, and W. Woo, editors, *Personalized Context Modeling and Management for UbiComp Applications*, volume 149 of *CEUR-WS*, pages 26–34, 2005.

B. N. Kokinov. Dynamics and automaticity of context: A cognitive modeling approach. In P. Bouquet, L. Serafini, P. Brézillon, M. Benerecetti, and F. Castellani, editors, *International Conference on Modeling and Using Context*, pages 200–213, 1999.

Y. Lee, H. R. Schmidtke, Y. Suh, and W. Woo. Realizing seamless interaction: a cognitive agent architecture for virtual and smart environments. In D. Hong and S. Jeon, editors, *International Symposium on Ubiquitous Virtual Reality*, volume 260 of *CEUR-WS*, pages 5,6, 2007.

G. Ligozat. Generalized intervals: A guided tour. In *Proceedings of the ECAI-98 Workshop on Spatial and Temporal Reasoning*, Brighton, UK, 1998.

G. Link. The logical analysis of plurals and mass terms: A lattice-theoretical approach. In R. Bäuerle, C. Schwarze, and A. von Stechow, editors, *Meaning, Use and Interpretation of Language*, pages 302–323. De Gruyter, 1983.

H. R. Schmidtke. The house is north of the river: Relative localization of extended objects. In D. Montello, editor, *International Conference on Spatial Information Theory*, volume 2205 of *LNCS*, pages 414–430, Berlin, 2001. Springer.

H. R. Schmidtke and W. Woo. A size-based qualitative approach to the representation of spatial granularity. In M. M. Veloso, editor, *Twentieth International Joint Conference on Artificial Intelligence*, pages 563–568, 2007.

H. R. Schmidtke, D. Hong, and W. Woo. Reasoning about models of context: A context-oriented logical language for knowledge-based context-aware applications. *Revue d'Intelligence Artificielle*, 2008.

M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multiagent Systems*. MIT Press, 1999.

Constructive Optimality Theoretic Syntax

Henk Zeevat

ILLC, Nieuwe Doelenstraat 15, 1012 CP Amsterdam
henk.zeevat@uva.nl

Abstract. This paper documents efforts at developing a constructive optimality theory. Several years ago, two 3rd year AI-students, Chris Regenboog and Jacco van Willegen, came up with an idea to implement optimality theory, in a special case —a system partially describing Dutch word order—. Me and some students have been trying to work out their ideas further in the period since, but many issues still remain open. This paper reports on a reconstruction of Bresnan's Optimal Syntax and a new integrated approach to word order, in three Germanic languages: Dutch, English and German, as a direct illustration of the concept. The idea is the following: interpret every constraint as a procedure that tries to add more structure to an underspecified specification and call the procedures in the order of the strength of the constraints they implement. The main consequence is that the notion of error disappears altogether from optimality theory. Errors are failed attempts to overwrite structure that is already filled in. The process is monotone increase of information and fully deterministic. There should be enough constraints to make sure that the process delivers a fully instantiated ground structure. Two other things that disappear from sight are GEN and various economy constraints: what can be constructed gives an upper bound on what should be in GEN, economy violations are just not generated.

The idea[1] also provides a serious alternative in thinking about the cognitive interpretation of optimality theory. The neural nets from which optimality theory emerged give rise to the idea of harmonic grammar in which the violations of constraints determine an integrated total violation value with respect to which the optimal candidate should win. The empirical discovery that gave rise to optimality theory as we know it is however the discovery that for the purpose of phonological description the possibility of an integrated violation value for several constraints in combination can be ignored: the strongest constraint always takes the decision. While it is possible to formalise optimality theory in harmonic grammar and to implement it in the corresponding nets[9], optimality theory does not arise naturally out of this setup.

The alternative model for OT is to think of a single object that is being built by a group of workers each with a particular task with respect to the structure. They all try to add details to it and conflicts are solved by a strict

[1] Some work on phonology has been omitted from this paper for reasons of space. Exploratory implementations have been produced in C, Prolog and Perl.

power hierarchy between them: the more powerful worker always gets his way, but even the most powerful worker is specialised and only decides some of the details, so that the activities of the lower ranked workers also show up in the finished structure.

A worker can be thought of as some simple map from an input to an output that is an informational increment of the input. While it is possible to conceive of this as a parallel process, in this paper, a serial composition approach will be adopted.

1 Optimality Theory

In optimality theory, structures are characterised as the best candidate for given input given a linearly ordered set of constraints $C_0 \ldots C_k$. The set of candidates and the input give an optimisation problem, the constraints determine how it should be solved. In linguistics, the task of selecting a verbal message for a given semantics, selecting a pronunciation for a form, and the task of finding an interpretation for a pronunciation can all be usefully considered to be optimisation problems. But there is no need for a limitation to linguistics.

In standard OT, by definition, $output_1$ is better than $output_2$ iff they are equally good or bad with respect to the first i constraints but $output_2$ violates C_{i+1} more often than $output_1$. An optimisation problem is an input, a constraint system and a set of possible outputs. A solution to the optimisation problem is a candidate that is best: any other candidate is worse or equally good.

The famous applications are in linguistics, particularly in production phonology (mapping an abstract phonological structure to a concrete one) and in production syntax (mapping semantics to a string of words). Less famous are applications to the semantics and pragmatics of natural language.

The application to phonology and syntax also comes with an important typological interpretation: the constraints are the same in all languages and typological variation is just changing the ordering of the contraints.

This offers a way of arguing for or against particular constraints: they should lead to correct typological predictions. But it also imposes a restriction on cognitively plausible implementations: constraint systems and their implementation should allow reordering. The typological interpretation turns OT phonology and syntax into a formal reconstruction of Jakobson's markedness theory [5].

The abstract model of optimality theory allows a direct computational interpretation if one can keep the number of candidates finite. But this would still be a bad strategy, comparable to generate-and-test in classical parsing. The neural basis has also been taken as a starting point and has led to experimental implementations of neural net-based phonology (Smolensky, Oren Schwarz). This is promising, but still experimental.

[4] and [6] have proposed an almost correct implementation for OT phonology using lenient composition over finite state transducers corresponding to the constraints. This is only an approximation since it requires that the number of errors is bounded by a maximum and the OT formalism does not allow such a

maximum number. In addition, this method is limited to phonology. [2] offers a less restricted model of phonology.

[7] proposes OT as a specification mechanism for LFG based processing. This paper agrees more with this strategy than is maybe apparent, but nonetheless as an implementation strategy, it does not qualify as being purely OT. In sum, direct implementations of OT syntax that can be useful in NL processing are missing. This paper hopes to make a contribution to this area, but the main idea is also applicable outside syntax, to phonology and maybe to semantics.

The implementation proposed in this paper can be given by a metaphor based on unification grammar. Inputs are underspecified feature structures only containing the input. Outputs are fully instantiated ground feature structures. The constraints are also feature structures. Outputs are constructed by default unifications of the constraints applied in the order of their strength to the input structure.

The effect is that constraint violations correspond to vacuous default unification where the partially specified feature structure would get inconsistent information. As a result of the input and the stronger constraints that apply first errors are the cases where information in the constraint conflicts with information in the feature structure.

Perhaps a literal implementation of this metaphor is possible in phonology, by coding the constraints as recursive feature structures.

Less metaphorically, constraints are procedures that add information to underspecified structures if the new information is consistent with the given underspecified structure. Inputs are underspecified structures only containing the input specification. Outputs are complete structures. Outputs are obtained by applying the constraint procedures in order of their strength to the input structure.

An abstract implementation strategy can be based on constraint logic programming. The structure is then a consistent theory satisfied by a class of finite models. Constraints are universal statements and can be represented by the set of their instances (with the constants taken from the objects of the theory so far). Default application is then a question of adding all those instances that are consistent with the theory to it. The output structure is a or the minimal model of the resulting theory. This strategy is be always available, but it is not necessarily efficient.[2]

In the abstract case, the input is a theory, GEN is the set of all its models and optimal elements of GEN are the minimal ground models of the theory after optimisation.

Efficiency can be increased if the theory or class of models can be efficiently represented by a finite structure that can be updated by the constraints. Let's call this "concrete constructive". The rest of the paper presents two instances of concrete constructive OT syntax.

[2] Let T be a finitely witnessed theory and C a universal statement. The incrementation of T by C can be given as $T(C) = T \cup \{C(a) : a \text{ is a witness of } T \text{ and } T \not\models \neg C(a)\}$

GEN is then the set of all ground instances of the input, and the winners those elements of GEN that are ground instances of the optimised input.

2 Bresnan's Optimal Syntax

This following is reconstruction[3] of [1]. The constraints appear in order of strength.

MAX(SPEC,PRED,TENSE,MODE, ...)
specifiers, predicates, tense and modal values are expressed by lexical and morphological means.

HEADS
f-structure heads are c-structure heads

LP COMPLEMENTS
lexical phrase complements are f-structure complements

FP COMPLEMENTS
functional phrase complements are not f-structure complements

PROM
the functional hierarchy of arguments corresponds to c-structure order

CC
the f-structure is complete and coherent

OP-SPEC
the operator appears as the specifier of C

* LEX-F
no lexical heads in functional projections

OB-HEAD
every category has an extended head

AGR
the subject and its predicate agrees

FULL INTERPRETATION
the output f-structure does not add to the content of the input structure

STAY
categories dominate their extended heads

NEG-TO-I
negation adjoins to I

LEX
structural inventory items are phonologically realised

Inputs are abstract f-structures, candidates associations of a c-structure with an f-structure.

Example:
$[_{cp}$*what do* $[_{ip}$*they* $[_{vp}$*read*$]]]$
$[_{cp}$*what read* $[_{ip}$*they*$[_{vp}]]]$ (*LEX-F)
$[_{ip}$*they*$[_{vp}$*read what*$]]$ (OP-SPEC)

[3] Bresnan's presentation is not formal enough for the purposes of this paper, so there are quite a number of decisions hidden both in my version of the constraint system and in the procedural translation. [1] provides empirical and theoretical motivation for the approach

$[_{ip}$*they do* $[_{vp}$*readwhat*$]]]$ (OP-SPEC)
$[_{cp}$*what* $[_{ip}$*they* $[_{vp}$*read*$]]]$ (OB-HEAD(2))
$[_{cp}$*what* $[_{ip}$*they do* $[_{vp}$*read*$]]]$ (OB-HEAD)

The following assigns a procedural meaning to the constraints, in some cases to several constraints at once.

The underspecified data structures are partially specified f-structures with some extra features and lexical items assigned to the nodes. Underspecification show up in missing position features and lexical annotations. Other underspecification are abstract governable functions (gf1, gf2 etc) that can be further specified to be subj, obj, obj1 etc and the possibility of inserting "dummy" features. The non-standard position features express where the node has to be realised in c-structure or in the surface string and the "dummy" feature allows the insertion of dummy elements, like auxiliary "do".

Procedural meaning

The starting point are abstract f-structures as in [1] to position annotated and lexicalised f-structures. (1) should be read as: the root's subject's predicate is *tom* (a semantic predicate), annotated with lexical item "Tom" (a string of phonemes) and the position feature SCP (the specifier of CP).

(1) subj:pred:tom+Tom+SCP

The abstract feature structure that is given as the input is also a representation of the set of all candidates: these are given by any way of assigning position features and lexical and morphological annotations to nodes in the f-structure and adding "dummy branches".

1. CC combined with MAX(SPEC,PRED,MODE,TENSE,...)

This is the lexicalisation step: the PRED from the input structure needs to be realised as a lexical item which governs the governable functions given in the input. The instantition of PRED also assigns concrete grammatical functions and case features to the GFs in the input structure and assigns morphology or lexical items to the features mentioned in MAX(SPEC,PRED,MODE,TENSE,...). This step can be seen as the definition of "legal lexicalised f-structure corresponding to the abstract input".

2. OP-SPEC

This constraint assigns the syntactic position feature SCP (specifier of CP) to the operator (long distance dependencies are captured in the f-structure by having a discourse function DF in the outermost f-structure unified with the operator. The value of DF is the operator.)

3. A combination of HEADS and *LEX-F

Together they force lexical heads of category X to occupy the structural position HXP (head of an XP). Without *LEX-F, the position can be classified as an HYP with Y a functional projection of category Y. Especially HVP could also be HIP or HCP. The combination instantiates the syntactic position parameter of an V, N, P or A to be HVP, HNP, HPP or HAP respectively.

4. OB-HEAD

a. It creates heads for CP (if needed: CP must be non-empty to have a HCP) or for IP (if CP does not have a head of category I and I is needed because of a SIP or IAdjunct). "Do" is the dummy HCP or HIP and inserted when there is no lexical element of category I in the f-structure.

b. auxiliaries are moved to CP: assign HCP to the auxiliary if there is a SCP but no C.

5. FP COMPLEMENTS

If TENSE, MODE or SPEC is realised by auxiliaries, complementisers or determiners these will get syntactic position features HIP or HCP (auxiliaries can be both the head of C and the head of I) and HDP.

6. LP COMPLEMENTS + PROM

They sort the LP complements in the order given by the functional hierarchy and can be used to assign syntactic position features XARG1 to XARGn to these complements.

7. AGR

1. assign SIP (specifier of IP) to the subject if there is an IP, SSU (sentential subject) otherwise.

2. assign agreement features and morphology to the predicate of the subject (the HIP or HCP, the HVP if there is no IP or CP.

8. NEG-TO-I

assign the IADJUNCT label to NEG

FULL INTERPRETATION has no good interpretation in this setup, STAY does not need to be implemented because it is emergent: the procedure does not produce unnecessary violations of it.

The annotated f-structures determine a string of words. Take the lexical word and morphology annotations and order them (cyclically) by their position annotations in the order:
SCP HCP SIP HIP Iadjunct SSU HVP VPARG1 VPARG2 VPARG3.

Example:

q=gf2
gf1:pred:pro
gf1:agr:pl
gf2:pred:pro
gf2:agr:nsg
pred:read(gf1,gf2)

CC + MAX(SPEC,PRED,MODE,TENSE,...)

q=obj
subj:pred:pro + they
subj:agr:pl
obj:pred:pro + what
obj:agr:nsg
pred:read(subj,obj)+ read

OP-SPEC

q=obj
subj:pred:pro + they
subj:agr:pl
obj:pred:pro + what+SCP

obj:agr:nsg
pred:read(subj,obj)+ read
HEADS + *LEX-F
q=obj
subj:pred:pro + they
subj:agr:pl
obj:pred:pro + what+SCP
obj:agr:nsg
pred:read(subj,obj)+ read+HVP
OB-HEAD q=obj
subj:pred:pro + they
subj:agr:pl
dummy:do+HCP
obj:pred:pro + what+SCP
obj:agr:nsg
pred:read(subj,obj)+ read+HVP
LP COMPLEMENTS + PROM
q=obj
subj:pred:pro + they +VARG1
subj:agr:pl
dummy:do+HCP
obj:pred:pro + what+SCP
obj:agr:nsg
pred:read(subj,obj)+ read+HVP
AGR
q=obj subj:pred:pro + they +VARG1
subj:agr:pl
dummy:do-0+HCP
obj:pred:pro + what+SCP
obj:agr:nsg
pred:read(subj,obj)+ read+HVP The last f-structure can be read out as the c-structure:

$[_{cp} what\ do\ [_{ip} they\ [_{vp} read]]]$

3 Dutch OT Syntax for Word Order

The constraint system of Bresnan does not seem to be applicable to Dutch or German, except for the formation of questions: German and Dutch allow lexical elements to be the heads of functional categories. The agenda in this section and the next two ones will be to develop some ideas about Dutch, then German and finally English. It is not the aim to be complete or even correct: its aim is to demonstrate the feasibility of an interesting and typologically valid account within concrete constructive OT. In the treatment, the problems in arriving at suitable f-structures from semantic representations are ignored as well as morphological issues. These aspects could be integrated, possibly along the lines of the reconstruction of [1] in the last section.

Constraints (ordered from strong to weak)

CLOSE(REL,SQ)
relative clauses and indirect questions are closed off for long distance movement

ONE = WH
the element ONE is the WH-phrase (finite verbs in main yes-no questions are also wh-elements in this view)

CLOSE(DP)
DPs are closed off for long distance movement

ONE = CT
the element ONE is the contrastive topic

CLOSE(SCOMP)
S structures are closed off for long distance movement

ONE = SUBJ
the element ONE is the subject

ONE < X
ONE comes first

V[FIN,MAIN] < X
the main finite verb comes first

SUBJ < X
the subject comes first

TOP(PP) < X
topical PPs come first

IO < X
the indirect object comes first

OBJ < X
the clause's object comes first

S > X
SCOMPS and Relative clauses come last

HV < X (headverbs come last) pending with VCOMP > X (VCOMPS come last)

input:
hij^{su} *leert* Jan^{obj} $[zwemmen]^{vcomp}$

	11	v1	su1	ob1	hv1	vc>
zwemmen Jan leert hij	***	**	***	*		***
zwemmen Jan hij leert	**	***	**	*		***
zwemmen leert Jan hij	***	*	*	**		***
zwemmen leert hij Jan	**	*	***	***		***
zwemmen hij Jan leert	*	***	**	**		***
zwemmen hij leert Jan	*	**	***	***		***
Jan zwemmen leert hij	***	**	***		*	**
Jan zwemmen hij leert	**	***	**		*	**
Jan leert zwemmen hij	***	*	***		**	*
Jan leert hij zwemmen	**	*	**		***	
Jan hij zwemmen leert	*	***	*		**	*
Jan hij leert zwemmen	*	**	*		***	
leert Jan zwemmen hij	***		***	*	**	*
leert Jan hij zwemmen	**		**	*	***	
leert zwemmen Jan hij	***		***	**	*	**
leert zwemmen Jan hij	***		***	**	*	**
leert hij Jan zwemmen	*		*	**	***	
leert hij zwemmen Jan	*		*	***	**	*
⇒hij leert Jan zwemmen		*		**	***	
hij leert zwemmen Jan		*		***	**	*
hij zwemmen leert Jan		**		***	*	**
hij zwemmen Jan leert		***		**	*	**
hij Jan zwemmen leert		***		*	**	*
hij Jan leert zwemmen		**		*	***	*

Procedural interpretation

1. Initialise by putting f-structure nodes between open and closing brackets $<$ and $>$. (These are really sets, linear order emerges by taking elements out of these sets and is coded by the string).

2. Apply the constraints topdown to all constituent lists $< \ldots >$ of the input, going freely into closed constituent lists. Put αs immediately before left bracket (after the left bracket) for ordering constraints $\alpha < X$ and $\alpha > X$. α's can be dragged out from anywhere except from a *close*(...) structure.

3. CLOSE(α): replace anything $Y^{..\alpha..}$ by close($Y^{..\alpha..}$)

4. ONE = X: assign a feature ONE to X if ONE has not been assigned yet

Example

$< hij^{subj}\ heeft^{main}\ Maria^{obj}\ < laten\ Jan^{obj}\ < leren\ zwemmen^{vcomp}\ >^{vcomp}>^{vcomp}>$

ONE=SUBJ

$< hij^{subj,one}\ heeft^{main}\ Maria^{obj}\ < laten\ Jan^{obj}\ < leren\ zwemmen^{vcomp}\ >^{vcomp}>^{vcomp}>$

ONE < X

$hij^{subj,one}\ < heeft^{main}\ Maria^{obj}\ < laten\ Jan^{obj}\ < leren\ zwemmen^{vcomp}\ >^{vcomp}>^{vcomp}>$

MAIN < X

$hij^{subj,one}\ heeft^{main}\ < Maria^{obj}\ < laten\ Jan^{obj}\ < leren\ zwemmen^{vcomp}\ >^{vcomp}>^{vcomp}>$

$hij^{subj,one}\ heeft^{main}\ < Maria^{obj}\ < laten\ Jan^{obj}\ < leren\ zwemmen^{vcompvcompvcomp}$

OBJ < X

$hij^{subj,one}\ heeft^{main}\ Maria^{obj}\ < Jan^{obj}\ < laten\ < leren\ zwemmen^{vcomp}\ >^{vcomp}>^{vcomp}>$

VCOMP < X

$hij^{subj,one}\ heeft^{main}\ Maria^{obj}\ <>< Jan^{obj}\ < laten\ < leren\ zwemmen^{vcomp}\ >^{vcomp}>$

$hij^{subj,one}heeft^{main}\ Maria^{obj}\ <>< Jan^{obj} >^{vcomp} < laten\ < leren\ zwemmen^{vcomp} >^{vcomp}>$
$hij^{subj,one}heeft^{main}\ Maria^{obj}\ <>< Jan^{obj} >^{vcomp} < laten >^{vcomp} < lerenzwemmen^{vcomp} >$
$hij^{subj,one}heeft^{main}\ Maria^{obj}\ <>< Jan^{obj} >^{vcomp}< laten >^{vcomp}< leren >^{vcomp}$
zwemmen Omitting brackets and features this gives: *hij heeft Maria Jan laten leren zwemmen.*

Calling HV $>$ X instead on:
$hij^{subj,one}heeft^{main}\ Maria^{obj}\ << Jan^{obj}\ < laten\ < leren\ zwemmen^{vcomp} >^{vcomp}>^{vcomp}>$
will give
HV $>$ X
$hij^{subj,one}heeft^{main}\ Maria^{obj} << Jan^{obj} << leren\ zwemmen^{vcomp} >^{vcomp}>^{vcomp}>$
laten
$hij^{subj,one}\ heeft^{main}\ Maria^{obj}\ << Jan^{obj}\ << zwemmen^{vcomp} >^{vcomp}\ leren >^{vcomp}>$
laten i.e. the alternative Dutch order (the only German one):
hij heeft Maria Jan zwemmen leren laten.

4 Provisional German

The following gives a rough version of German. German allows slightly more long distance dependencies and considerably more freedom in word order. After the main verb, priorities are better given by a constraint PROM which allows fronting any constituent that more prominent in the dimensions of grammatical obliqueness, animacy, topicality, referentiality. PROM would be doing the work of the Dutch constraints: SUBJ $<$ X, TOP $<$ X, IO $<$ X or OBJ $<$ X. German has also chosen against the cross-serial dependency order in the verbal complex.

CLOSE(REL,SQ)
ONE = WH
ONE = CT
CLOSE(DP, SCOMP)
ONE = SUBJ
ONE $<$ X
V[FIN,MAIN] $<$ X
PROM
S $>$ X
HV $<$ X
VCOMP $>$ X

5 English

Is it possible to see English, Dutch and German as typological variants of each other? It seems to work, if another approach is chosen for the auxiliary verb syntax than the one adopted by Bresnan. Remember that Dutch and German put finite main verbs in the second position. And that Dutch and German put subjects first. In English, the restriction to main clauses has disappeared: it is a general property of finite verbs that they come first. The hypothesis is that the Dutch/German subject constraint and verb second in English (finite verbs come first after the occupants of ONE) have become merged: they are ranked equally, but instead of choosing the verb or the subject to come first they must both be obeyed[4] This goes together with free insertion of finite dummy *do*.

This gives the complex constraint (2):

(2) V[FIN] < X >=< SUBJ < X[-AUX]

(2) can be implemented by the following procedure.

a. Put the finite verb out to the left if there is no subject in the constituent list.

b. Put finite aux out and then the subject. If there is no aux, put a version of *do*.

With this complex constraint, the English system is quite close to Dutch and German. The German PROM < X has to be split up even further than for Dutch.

CLOSE(REL,SQ)
ONE = WH
ONE = CT
CLOSE(DP,PP,SCOMP.VCOMP)
ONE = SUBJ
ONE < X
V[FIN] < X >=< SUBJ < X[-AUX]
V < X
IO < X
OBJ < X
PP < X
S > X
VCOMP > X

6 Analysis

The above gives a simple and fast model of generation and it should be possible to use it in a model of parsing. The idea would be to use a statistical dependency parser like [8]. These are able to generate the kind of analyses that can be used as input by the generator defined above[5]. There are two things the combination can do: one is a check on parsing. If the generator cannot generate the parse, use the next parse until one is found that can be generated. This increases the quality of the parsing and allows the parser to be imperfect. Second, it is the ideal setting for improving the generator.

[4] The constructive approach forces a distinction between the situation that two constraints are ranked equal and a choice has to be made for the first constraint to apply and the merging of two constraints: they together determine a procedure which is called as a unit.

[5] This is the reverse to what is proposed in [3] where LFG parsing is improved by a kind of interpretational OT

But there is a third application. An OT description of the kind that I have been considering above cannot deal with word order freezing (presumably the reason why English and Dutch syntax have cut up PROM into a small more specific system of constraints. All the older versions of Germanic are more like German, including Anglosaxon. A direct precursor of Dutch is not known.

What sets English apart from German is the almost complete disappearance of the case system which in German prevents accidents with the liberal rules of word order in the middle field of German. It prevents accidents in German only to some extent since the German case system is not nearly as robust as the Latin or the Russian case system: many times the case morphology does not tell subjects and objects apart. In those situations, the thematic dimension of PROM becomes all important and the dominant order is canonical: subjects precede indirect and direct objects. (Freezing is only one of many phenomena of this kind.)

This can be solved by looking at the most probable parse: if it is not the generator's input, other generations and variations of the input must be considered until the parser can return the input. The consideration of other generations will deal with word order freezing in the middle field. For similar freezing effects around the first position, a possibility of switching off the CT-feature is an option. This pragmatic feature can be marked by word order, but it can also be clear from the context or marked by intonation. An option could be to have a variant UCT of the feature CT in the input which does not force the constituent to become ONE.

So these are the claims of this paper:

1. Concrete and constructive OT is possible in syntax. It is much less different from LFG or HPSG than one would think. It is essentially generation oriented and this restriction seems hard to remove.

2. Unification grammar has always been concrete and this never was an obstacle to description. At some level of abstraction, concrete constructive OT is constraint-based unification grammar with unification replaced by default unification.

3. Pending constraints cannot be reduced to considering both orders in constructive OT.

And there is one more claim that needs to be proved in future work:

Constructive OT syntax is a learnable system and can be learnt with the help of a state of the art probabilistic parser. This parser can also help in acquiring a capacity to monitor generation for understanding and so reach proper syntactic correctness (including freezing) and improve the quality of its output for a human user. Inversely, a fully developed generator can increase the correctness of the probabilistic parser. If this works, OT syntax can have technological impact after all.

References

1. Joan Bresnan. Optimal syntax. In J. Dekkers, F. van der Leeuw, and J. van de Weijer, editors, *Optimality Theory: Phonology, Syntax, and Acquisition*, pages 334–

385. Oxford University Press, Oxford, 2000.
2. Jason Eisner. Directional constraint evaluation in optimality theory. In *COLING 2000*, pages 257–263. ACL, 2000.
3. Martin Forst, Jonas Kuhn, and Christian Rohrer. Corpus-based learning of ot constraint rankings for large-scale lfg grammars. In *Proceedings of the LFG05 Conference*. CoNLL, Bergen, Norway, 2005.
4. Robert Frank and Giorgio Satta. Optimality theory and the generative complexity of constraint violability. computational linguistics. *Computational Linguistics*, 24(3):307–315, 1998.
5. R.O. Jacobson. Morphological observations on Slavic declension (the structure of Russian case forms). In Linda R. Waugh and Morris Halle, editors, *Roman Jakobson. Russian and Slavic grammar: Studies 1931-1981*, pages 105–133. Mouton de Gruyter, Berlin, 1958/1984.
6. Lauri Karttunen. The proper treatment of optimality in computational phonology. In Lauri Karttunen, editor, *FSMNLP'98: International Workshop on Finite State Methods in Natural Language Processing*, pages 1–12. Association for Computational Linguistics, Somerset, New Jersey, 1998.
7. Jonas Kuhn. *Optimality-Theoretic Syntax: A Declarative Approach*. CSLI Publications, Stanford, 2003.
8. J. Nivre, J. Hall, and J. Nilsson. Memory-based dependency parsing. In H. T. Ng and E. Riloff, editors, *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, pages 49–56. CoNLL, Boston, Massachusetts, 2004.
9. Paul Smolensky and Geraldine Legendre. *The Harmonic Mind From Neural Computation to Optimality-Theoretic Grammar*. MIT Press, Cambrdige (MA), 2006.

RECENT RESEARCH REPORTS

#121 Ben Schouten and Niels Christian Juul, editors. *Proceedings of the First European Workshop on Biometrics and Identity Management (BIOID 2008)*, Roskilde, Denmark, April 2008.

#120 Peter Danholt. *Interacting Bodies: Posthuman Enactments of the Problem of Diabetes Relating Science, Technology and Society-studies, User-Centered Design and Diabetes Practices*. PhD thesis, Roskilde, Denmark, February 2008.

#119 Alexandre Alapetite. *On speech recognition during anaesthesia*. PhD thesis, Roskilde, Denmark, November 2007.

#118 Paolo Bouquet, editor. *CONTEXT'07 Doctoral Consortium Proceedings*, Roskilde, Denmark, October 2007.

#117 Kim S. Henriksen. *A Logic Programming Based Approach to Applying Abstract Interpretation to Embedded Software*. PhD thesis, Roskilde, Denmark, October 2007.

#116 Marco Baroni, Alessandro Lenci, and Magnus Sahlgren, editors. *Proceedings of the 2007 Workshop on Contextual Information in Semantic Space Models: Beyond Words and Documents*, Roskilde, Denmark, August 2007.

#115 Paolo Bouquet, Jérôme Euzenat, Chiara Ghidini, Deborah L. McGuinness, Valeria de Paiva, Luciano Serafini, Pavel Shvaiko, and Holger Wache, editors. *Proceedings of the 2007 workshop on Contexts and Ontologies Representation and Reasoning (C&O:RR-2007)*, Roskilde, Denmark, August 2007.

#114 Bich-Liên Doan, Joemon Jose, and Massimo Melucci, editors. *Proceedings of the 2nd International Workshop on Context-Based Information Retrieval*, Roskilde, Denmark, August 2007.

#113 Henning Christiansen and Jørgen Villadsen, editors. *Proceedings of the 4th International Workshop on Constraints and Language Processing (CSLP 2007)*, Roskilde, Denmark, August 2007.

#112 Anders Kofod-Petersen, Jörg Cassens, David B. Leake, and Stefan Schulz, editors. *Proceedings of the 4th International Workshop on Modeling and Reasoning in Context (MRC 2007) with Special Session on the Role of Contextualization in Human Tasks (CHUT)*, Roskilde, Denmark, August 2007.

#111 Ioannis Hatzilygeroudis, Alvaro Ortigosa, and Maria D. Rodriguez-Moreno, editors. *Proceedings of the 2007 workshop on REpresentation models and Techniques for Improving e-Learning: Bringing Context into the Web-based Education (ReTIeL'07)*, Roskilde, Denmark, August 2007.