

A Uniform Approach to Underspecification and Parallelism

Joachim Niehren
Programming Systems Lab
Universität des Saarlandes
Saarbrücken, Germany
niehren@ps.uni-sb.de

Manfred Pinkal
Department of
Computational Linguistics
Universität des Saarlandes
Saarbrücken, Germany
pinkal@coli.uni-sb.de

Peter Ruhrberg
Department of
Computational Linguistics
Universität des Saarlandes
Saarbrücken, Germany
peru@coli.uni-sb.de

Abstract

We propose a unified framework in which to treat semantic underspecification and parallelism phenomena in discourse. The framework employs a constraint language that can express equality and subtree relations between finite trees. In addition, our constraint language can express the equality up-to relation over trees which captures parallelism between them. The constraints are solved by context unification. We demonstrate the use of our framework at the examples of quantifier scope, ellipsis, and their interaction.¹

1 Introduction

Traditional model-theoretic semantics of natural languages (Montague, 1974) has assumed that semantic information, processed by composition and reasoning processes, is available in a completely specified form. During the last few years, the phenomenon of semantic underspecification, i.e. the incomplete availability of semantic information in processing, has received increasing attention. Several aspects of underspecification have been focused upon, motivated mainly by computational considerations: the ambiguity and openness of lexical meaning (Pustejovsky, 1995; Copestake and Briscoe, 1995), referential underspecification (Asher, 1993), structural semantic underspecification caused by syntactic ambiguities (Egg and Lebeth, 1995), and by the underdetermination of scope relations (Alshawi and Crouch, 1992; Reyle, 1993). In addition, external factors such as insufficient coverage

of the grammar, time-constraints for parsing, and most importantly the kind of incompleteness, uncertainty, and inconsistency, coming with spoken input are coming more into the focus of semantic processing (Bos et al., 1996; Pinkal, 1995).

The aim of semantic underspecification is to produce compact representations of the set of possible readings of a discourse. While the readings of a discourse may be only partially known, the interpretations of its components are often strongly correlated. In this paper, we are concerned with a uniform treatment of underspecification and of phenomena of discourse-semantic parallelism. Some typical parallelism phenomena are ellipsis, corrections, and variations. We illustrate them here by some examples (focus-bearing phrases are underlined):

- (1) John speaks Chinese. Bill too.
- (2) John speaks Japanese. - No, he speaks Chinese.
- (3) ??? - Bill speaks Chinese, too.

Parallelism guides the interpretation process for the above discourses. This is most obvious in the case of ellipsis interpretation (1), but is also evident for the resolution of the anaphor in the correction in (2), and in the variation case (3) where the context is unknown and has to be inferred.

The challenge is to integrate a treatment of parallelism with underspecification, such as in cases of the interaction of scope and ellipsis. Problematic examples like (4) have been brought to attention by (Hirschbuehler, 1982). The example demonstrated that earlier treatments of ellipsis based on copying of the content of constituents are insufficient for such kinds of parallelism.

- (4) Two European languages are spoken by many linguists, and two Asian ones (are spoken by many linguists), too.

¹The research reported in this paper has been supported by the SFB 378 at the Universität des Saarlandes and the Esprit Working Group CCL II (EP 22457).

The first clause of (4) is scope-ambiguous between two readings. The second, elliptic one, is too. Its interpretation is indicated by the part in parentheses. The parallelism imposed by ellipsis requires the scope of the quantifiers in the elliptical clause to be analogous to the scope of the quantifiers in the antecedent clause. Thus, the conjunction of both clauses has only two readings: Either the interpretation is the wide scope existential one in both cases (two specific European languages as well as two specific Asian languages are widely known among linguists), or it is the narrow scope existential one (many linguists speak two European languages, and many linguists speak two Asian languages).

A natural approach for describing underspecified semantic information is to use an appropriate constraint language. We use constraints interpreted over finite trees. A tree itself represents a formula of some semantic representation language. This approach is very flexible in allowing various choices for the particular semantic representation language, such as first-order logic, intensional logic (Dowty, Wall, and Peters, 1981), or Discourse Representation Theory, DRT, (Kamp and Reyle, 1993). The constraint approach contrasts with theories such as Reyles UDRT (1993) which stresses the integration of the levels of semantic representation language and underspecified descriptions.

For a description language we propose the use of context constraints over finite trees which have been investigated in (Niehren, Pinkal, and Ruhrberg, 1997). This constraint language can express equality and subtree relations between finite trees. More generally it can express the “equality up-to” relation over trees, which captures (non-local) parallelism between trees. The general case of equality up-to constraints cannot be handled by a system using subtree plus equality constraints only. The problem of solving context constraints is known as context unification, which is a subcase of linear second-order unification (Lévy, 1996; Pinkal, 1995). There is a complete and correct semi-decision procedure for solving context constraints.

Context unification allows to treat the interaction of scope and ellipsis. Note that in example (4) the trees representing the semantics of the source and target clause must be equal up to the positions corresponding to the contrasting elements (two European languages / two Asian languages). Thus, this is a case where the additional expressive power of context constraints is crucial. In this paper, we elaborate on the example of scope and ellipsis interaction. The framework appears to extend, however, to

all kinds of cases where structural underspecification and discourse-semantic parallelism interact.

In Section 2, we will describe context unification, and present some results about its formal properties and its relation to other formalisms. Section 3 demonstrates the application to scope underspecification, to ellipsis, and to the combined cases. In Section 4, the proposed treatment is compared to related approaches in computational semantics. Section 5 gives an outlook on future work.

2 Context Unification

Context unification is the problem of solving context constraints over finite trees. The notion of context unification stems from (Lévy, 1996) whereas the problem originates from (Comon, 1992) and (Schmidt-Schauß, 1994). Context unification has been formally defined and investigated by the authors in (Niehren, Pinkal, and Ruhrberg, 1997). Here, we select and summarize relevant results on context unification from the latter.

Context unification subsumes string unification (see (Baader and Siekmann, 1993) for an overview) and is subsumed by linear second-order unification which has been independently proposed by (Lévy, 1996) and (Pinkal, 1995). The decidability of context unification is an open problem. String unification has been proved decidable by (Makanin, 1977). The decidability of linear second-order unification is an open problem too whereas second-order unification is known to be undecidable (Goldfarb, 1981).

The syntax and semantics of context constraints are defined as follows. We assume an infinite set of *first-order variables* ranged over by X, Y, Z , an infinite set of *second-order variables* ranged over by C , and a set of *function symbols* ranged over by f , that are equipped with an arity $n \geq 0$. Nullary function symbols are called *constants*. *Context constraints* φ are defined by the following abstract syntax:

$$\begin{aligned} t &::= X \mid f(t_1, \dots, t_n) \mid C(t) \\ \varphi &::= t = t' \mid \varphi \wedge \varphi' \end{aligned}$$

A (second-order) *term* t is either a first-order variable X , a construction $f(t_1, \dots, t_n)$ where the arity of f is n , or an application $C(t)$. A *context constraint* φ is a conjunction of equations between second-order terms.

Semantically, we interpret first-order variables X as *finite constructor trees*, which are first-order terms without variables, and second-order variables C as context functions that we define next. A *context with*

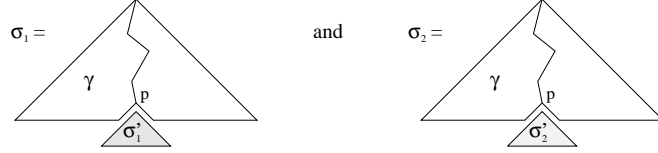


Figure 1: The equality up-to relation

hole X is a term t that does not contain any other variable than X and has exactly one occurrence of X . A *context function* γ is a function from trees to trees such that there exists a variable X and a context t with hole X satisfying the equation:

$$\gamma(\sigma) = t[\sigma/X] \quad \text{for all trees } \sigma.$$

Note that context functions can be described by linear second-order lambda terms of the form $\lambda X.t$ where X occurs exactly once in the second-order term t . Let α be a variable assignment that maps first-order variables to finite trees and second-order variables to context functions. The interpretation $\alpha(t)$ of a term t under α is the finite tree defined as follows:

$$\begin{aligned} \alpha(a(t_1, \dots, t_n)) &= a(\alpha(t_1), \dots, \alpha(t_n)) \\ \alpha(C(t)) &= \alpha(C)(\alpha(t)) \end{aligned}$$

A *solution* of a context constraint φ is a variable assignment α such that $\alpha(t) = \alpha(t')$ for all equations $t = t'$ in φ . A context constraint is called *satisfiable* if it has a solution. *Context unification* is the satisfiability problem of context constraints.

Context constraints (plus existential quantification) can express subtree constraints over finite trees. A subtree constraint has the form $X \ll X'$ and is interpreted with respect to the subtree relation on finite trees. A subtree relation $\sigma \ll \sigma'$ holds if σ is a subtree of σ' , i.e. if there exists a context function γ such that $\sigma' = \gamma(\sigma)$. Thus, the following equivalence is valid over finite trees:

$$X \ll X' \leftrightarrow \exists C(X' = C(X))$$

Context constraints are also more general than equality up-to constraints over finite trees, which allow to describe parallel tree structures. An equality up-to constraint has the form $X_1/X'_1 = Y_1/Y'_1$ and is interpreted with respect to the equality up-to relation on finite trees. Given finite trees $\sigma_1, \sigma'_1, \sigma_2, \sigma'_2$, the equality up-to relation $\sigma_1/\sigma'_1 = \sigma_2/\sigma'_2$ holds if σ_1 is equal to σ_2 up-to one position p where σ_1 has the subtree σ'_1 and σ_2 the subtree σ'_2 . This is depicted in Figure 1. In this case, there exists a context function

γ such that $\sigma_1 = \gamma(\sigma'_1)$ and $\sigma_2 = \gamma(\sigma'_2)$. In other words, the following equivalence holds:

$$X/X' = Y/Y' \leftrightarrow \exists C(X = C(X') \wedge Y = C(Y'))$$

Indeed, the satisfiability problems of context constraints and equality up-to constraints over finite trees are equivalent. In other words, context unification can be considered as the problem of solving equality up-to constraints over finite trees.

2.1 Solving Context Constraints

There exists a correct and complete semi-decision procedure for context unification. This algorithm computes a representation of all solutions of a context constraint, in case there are any. We illustrate the algorithm in figure 2. There, we consider the constraint

$$X_s = @(@(s, c), j) \wedge X_s = C(X_{cs}) \wedge X_{cs} = j$$

which is also discussed in example (11)(i) as part of an elliptical construction.

Our algorithm proceeds on pairs consisting of a constraint and a set of variable bindings. At the beginning the set of variable bindings is empty. In case of termination with an empty constraint, the set of variable bindings describes a set of solutions of the initial constraint.

Consider the run of our algorithm in figure 2. In the first step, $X_s = @(@(s, c), j)$ is removed from the constraint and the variable binding $X_s \mapsto @(@(s, c), j)$ is added. This variable binding is applied to the remaining constraint where X_s is substituted by $@(@(s, c), j)$. The second computation step is similar. It replace the to constraint $X_{cs} = j$ by a variable binding $X_{cs} \mapsto j$ and eliminates X_{cs} in the remaining constraint.

The resulting constraint $@(@(s, c), j) = C(j)$ presents an equation between a term with a constant $@$ as its (“rigid”) head symbol and a term with a context variable C as its (“flexible”) head symbol. In such a case one can either apply a projection rule that binds C to the identity context $\lambda Y.Y$ or an

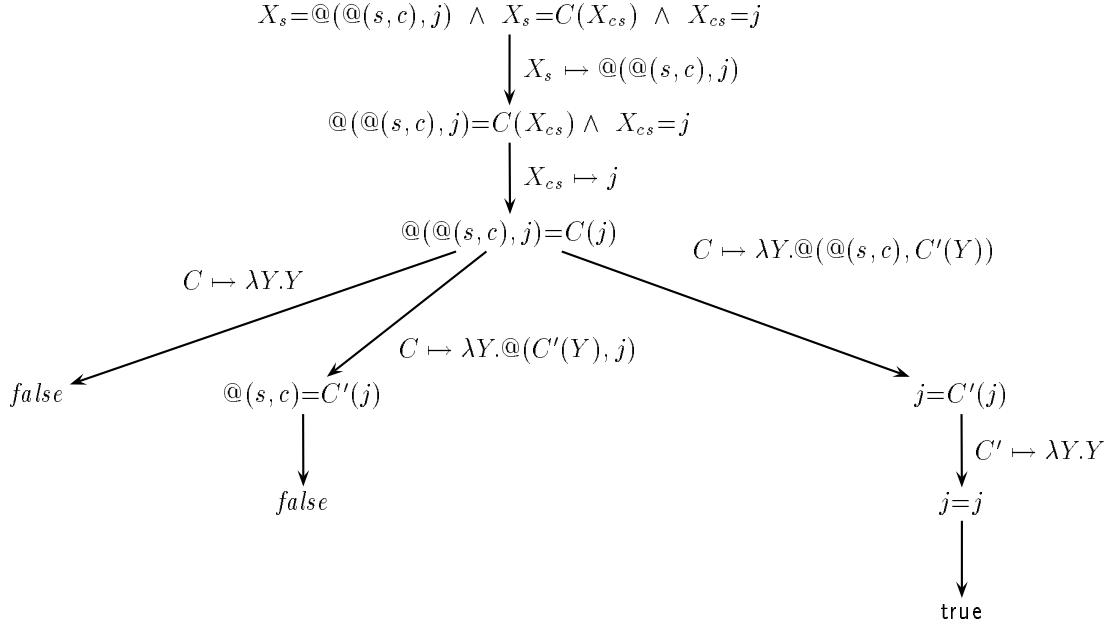


Figure 2: Solving the context constraints of example (11)(i)

imitation rule. Projection produces a clash of two rigid head symbols @ and j . Imitation presents two possibilities for locating the argument j of the context variable C as a subtree of the two arguments of the rigid head symbol @. Both alternatives lead to new rigid-flexible situations. The first alternative leads to failure (via further projection or imitation) as $@(s, c)$ does not contain j as a subtree. The second leads to success by another projection step.

The unique solution of the constraint in figure 2 can be described as follows:

$$\begin{array}{l}
X_s \mapsto @(@(s, c), j), \\
X_{cs} \mapsto j, \\
C \mapsto \lambda Y. @(@(s, c), Y)
\end{array}$$

The full version of (Niehren, Pinkal, and Ruhrberg, 1997) contains discussions of two algorithms for context unification. For a discussion on decidable fragments of context constraints, we also refer to this paper.

3 Underspecification and Parallelism

In this section, we discuss the use of context unification for treating underspecification and parallelism by some concrete examples. The set of solutions of a context constraint represents the set of possible readings of a given discourse. The trees assigned by

the solutions represent expressions of some semantic representation language. Here, we choose (extensional) typed higher-order logic, HOL, (Dowty, Wall, and Peters, 1981). However, any other logical language can be used in principle, so long as we can represent its syntax in terms of finite trees.

It is important to keep our semantic representation language (HOL) clearly separate from our description language (context constraints over finite trees). We assume an infinite set of HOL-variables ranged over by \mathbf{x} and \mathbf{y} . The signature of context constraints contains a unary function symbol $\text{lam}_{\mathbf{x}}$ and a constant $\text{var}_{\mathbf{x}}$ per HOL-variable \mathbf{x} . Furthermore, we assume a binary function symbol @ that we write in left associative infix notation and constants like `john`, `language`, etc. For example the tree

$$(\text{many}@(\text{language})@(\text{lam}_{\mathbf{x}}(\text{spoken_by}@(\text{john})@\text{var}_{\mathbf{x}})))$$

represents the HOL formula

$$(\text{many}(\text{language}))(\lambda \mathbf{x}. (\text{spokenby}(\text{john})(\mathbf{x}))).$$

Note that the function symbol @ represents the application in HOL and the function symbols $\text{lam}_{\mathbf{x}}$ the abstraction over \mathbf{x} in HOL.

3.1 Scope

Scope underspecification for a sentence like (5) is expressed by the equations in (6):

- (5) Two languages are spoken by many linguists.
- (6) $X_S = C_1((\text{two@language})@\text{lam}_x(C_3(X'_S))) \wedge$
 $X_S = C_2((\text{many@linguist})@\text{lam}_y(C_4(X'_S))) \wedge$
 $X'_S = \text{spoken_by}@\text{var}_y@\text{var}_x$

The algorithm for context unification leads to a disjunction of two solved constraints given in (7) (i) and (ii).

- (7) (i) $X_S =$
 $C_1((\text{two@language})@\text{lam}_x($
 $C_5((\text{many@linguist})@\text{lam}_y($
 $C_4(\text{spoken_by}@\text{var}_y@\text{var}_x))))$
- (ii) $X_S =$
 $C_2((\text{many@linguist})@\text{lam}_y($
 $C_6((\text{two@language})@\text{lam}_x($
 $C_3(\text{spoken_by}@\text{var}_y@\text{var}_x))))$

The algorithm does in fact compute a third kind of solved constraint for (6), where none of the quantifiers `two@language` and `many@linguist` are required to be within the scope of each other. This possibility can be excluded within the given framework by using a stronger set of equations between second-order terms as in (6'). Such equations can be reduced to context constraints via Skolemisation.

- (6') $C_S = \lambda X.C_1(\text{two@language}@\text{lam}_x(C_3(X))) \wedge$
 $C_S = \lambda X.C_2(\text{many@linguist}@\text{lam}_y(C_4(X))) \wedge$
 $X_S = C_S(\text{spoken_by}@\text{var}_y@\text{var}_x)$

Both solved constraints in (7) describe infinite sets of solutions which arise from freely instantiating the remaining context variables by arbitrary contexts. We need to apply a *closure operation* consisting in projecting the remaining free context variables to the identity context $\lambda X.X$. This gives us in some sense the minimal solutions to the original constraint. It is clear that performing the closure operation must be based on the information that the semantic material assembled so far is complete. Phenomena of incomplete input, or coercion, require a withholding, or at least a delaying of the closure operation. The closure operation on (7) (i) and (ii) leads to the two possible scope readings of (5) given in (8) (i) and (ii) respectively.

- (8) (i) $X_S \mapsto$
 $(\text{two@language})@\text{lam}_x($
 $(\text{many@linguist})@\text{lam}_y($
 $\text{spoken_by}@\text{var}_y@\text{var}_x))$
- (ii) $X_S \mapsto$
 $(\text{many@linguist})@\text{lam}_y($
 $(\text{two@language})@\text{lam}_x($
 $\text{spoken_by}@\text{var}_y@\text{var}_x))$

A constraint set specifying the scope-neutral meaning information as in (6') can be obtained in a rather simple compositional fashion. Let each node P in the syntactic structure be associated with three semantic meta-variables X_P , X'_P , and C_P , and let $I(P)$ be the scope boundary for each node P . Rules for obtaining semantic constraints from binary syntax trees are:

- (9) (i) For every S-node P add $X_P = C_P(X'_P)$, for any other node add $X_P = X'_P$.
- (ii) If $[_P Q R]$, Q and R are not NP nodes, add $X'_P = X_Q@X_R$ or $X'_P = X_R@X_Q$, according to HOL type.
- (iii) If $[_P Q R]$ or $[_P R Q]$, and R is an NP node, then add $X'_P = X_Q@var_x$ and $C_{I(P)} = \lambda X.C_0(X_R@\text{lam}_x(C_1(X)))$.

For example, the first two constraints in example (6') result from applying rule (iii), where the values for the quantifiers `two@language` and `many@linguist` are already substituted in for the variables X_R in both cases. The quantifiers themselves are put together by rule (ii). The third constraint results from rule (i) when the semantics of X'_S is filled in. The latter is a byproduct of the applications of rule (iii) to the two NPs.

3.2 Ellipsis

We now look into the interpretation of examples (1) to (4), which exhibit forms of parallelism. Let us take X_s and X_t to represent the semantics of the source and the target clause (i.e., the first and the second clause of a parallel construction; the terminology is taken over from the ellipsis literature), and X_{cs} and X_{ct} to refer to the semantic values of the contrast pair. The constraint set of the whole construction is the union of the constraint sets obtained by interpreting source and target clause independent of each other plus the pair of constraints given in (10).

- (10) $X_s = C(X_{cs}) \wedge X_t = C(X_{ct})$

The equations in (10) determine that the semantics of the source clause and the semantics of the target clause are obtained by embedding the representations of the respective contrasting elements into the same context. In other words: Source semantics and target semantics must be identical up to the positions of the contrasting elements.

As an example, consider the ellipsis construction of Sentence (1), where for simplicity we assume that proper names are interpreted by constants and not as quantifiers. It makes no difference for our treatment of parallelism.

$$(11) \quad (i) \quad X_s = \text{spea}k@chinese@john \quad \wedge \\ X_{cs} = john \quad \wedge \quad X_s = C(X_{cs}) \quad \wedge \\ (ii) \quad X_{ct} = bill \quad \wedge \quad X_t = C(X_{ct})$$

By applying the algorithm for context unification to this constraint, in particular to part (i) as demonstrated in figure 2, we can compute the context C to be $\lambda Y. (\text{spea}k@chinese@Y)$. This yields the interpretation of the elliptical clause, which is given by $X_t \mapsto \text{spea}k@chinese@bill$.

Note that the treatment of parallelism refers to contrasted and non-contrasted portions of the clause pairs rather than to overt and phonetically unrealized elements. Thus it is not specific for the treatment of ellipsis, but can be applied to other kinds of parallel constructions, as well. In the correction pair of Sentence (2), it provides a certain unambiguous reading for the pronoun, in (3), it gives $X_s = \text{spea}k@chinese@X_{cs}$ as a partial description of the (overheard or unuttered) source clause.

3.3 Scope and Ellipsis

Finally, let us look at the problem case of parallelism constraints for structurally underspecified clause pairs. We get a combination of constraints for a scope underspecified source clause (12) and parallelism constraints between source and target (13).

$$(12) \quad C_s = \lambda X. C_1((\text{two}@e_language)@lam_x(C_3(X))) \\ \wedge \\ C_s = \lambda X. C_2((\text{many}@linguist)@lam_y(C_4(X))) \\ \wedge \\ X_s = C_s(\text{spoken_by}@var_y@var_x)$$

$$(13) \quad X_s = C(\text{two}@e_language) \quad \wedge \\ X_t = C(\text{two}@a_language)$$

The conjunction of the constraints in (12) and (13) correctly allows for the two solutions (14) and (15),

with corresponding scopings in X_s and X_t after closure.²

$$(14) \quad X_s \mapsto \\ (\text{two}@e_language)@lam_x(\\ (\text{many}@linguist)@lam_y(\\ \text{spoken_by}@var_y@var_x)) \quad \wedge$$

$$X_t \mapsto \\ (\text{two}@a_language)@lam_x(\\ (\text{many}@linguist)@lam_y(\\ \text{spoken_by}@var_y@var_x)) \quad \wedge$$

$$C \mapsto \\ \lambda Y. Y@lam_x(\\ (\text{many}@linguist)@lam_y(\\ \text{spoken_by}@var_y@var_x))$$

$$(15) \quad X_s \mapsto \\ (\text{many}@linguist)@lam_y(\\ (\text{two}@e_language)@lam_x(\\ \text{spoken_by}@var_y@var_x)) \quad \wedge$$

$$X_t \mapsto \\ (\text{many}@linguist)@lam_y(\\ (\text{two}@a_language)@lam_x(\\ \text{spoken_by}@var_y@var_x)) \quad \wedge$$

$$C \mapsto \\ \lambda Y. (\text{many}@linguist)@lam_y(\\ Y@lam_x(\\ \text{spoken_by}@var_y@var_x))$$

Mixed solutions, where the two quantifiers take different relative scope in the source and target clause are not permitted by our constraints. For example, (16) provides no solution to the above constraints.

$$(16) \quad X_s \mapsto \\ (\text{two}@e_language)@lam_x(\\ (\text{many}@linguist)@lam_y(\\ \text{spoken_by}@var_y@var_x))$$

$$X_t \mapsto \\ (\text{many}@linguist)@lam_y(\\ (\text{two}@a_language)@lam_x(\\ \text{spoken_by}@var_y@var_x))$$

²Notice that closure is applied to the solved form of the combined constraints (i.e. (14) and (15) respectively) of the two sentences here, rather than to solved forms of (12) and (13) separately. This reflects the dependency of the interpretation of the second sentence on material in the first one.

From the trees in (16) one cannot construct a context function to be assigned to C which solves the parallelism constraints in (13).

4 Comparison to other Theories

Standard theories for scope underspecification make use of subtree relations and equality relations only. Such relationships may be expressed on a level of a separate constraint language, as in our case, or be incorporated into the semantic formalism itself, as it is done for DRT by the system of UDRT (Reyle, 1993). In UDRT one introduces “labels” that behave very much like variables for DRsEs. These labels figure in equations as well as subordination constraints to express scope relations between quantifiers. Equations and subordination constraints alone do not provide us with a treatment of parallelism. An idea that seems to come close to our notion of equality up-to constraints is the co-indexing technique in (Reyle, 1995), where non-local forms of parallelism are treated by dependency marking on labels. We believe that our use of a separate constraint language is more transparent.

A treatment for ellipsis interpretation which uses a form of higher-order unification has been proposed in (Dalrymple, Shieber, and Pereira, 1991) and extended to other kinds of parallel constructions by (Gardent, Kohlhase, and van Leusen, 1996; Gardent and Kohlhase, 1996). Though related in some respects, there are formal differences and differences in coverage between this approach and the one we propose. They use an algorithm for higher-order matching rather than context unification and they do not distinguish an object and meta language level. As a consequence they need to resort to additional machinery for the treatment of scope relations, such as Pereira’s scoping calculus, described in (Shieber, Pereira, and Dalrymple, 1996).

On the other hand, their approach treats a large number of problems of the interaction of anaphora and ellipsis, especially strict/sloppy ambiguities. Our use of context unification does not allow us to adopt their strategy of capturing such ambiguities by admitting non-linear solutions to parallelism constraints.

5 Outlook

Extensions of context unification may be useful for our applications. For gapping constructions, contexts with multiple holes need to be considered. The algorithm for context unification described in the

complete version of (Niehren, Pinkal, and Ruhrberg, 1997) makes use of contexts with multiple holes in any case.

So far our treatment of ellipsis does not capture strict-sloppy ambiguities if that ambiguity is not postulated for the source clause of the ellipsis construction. We believe that the ambiguity can be integrated into the framework of context unification without making such a problematic assumption. This requires modifying the parallelism requirements in an appropriate way. We hope that while sticking to linear solutions only, one may be able to introduce such ambiguities in a very controlled way, thus avoiding the overgeneration problems that come from freely abstracting multiple variable occurrences. This work is currently in progress, and a deeper comparison between the approaches has yet to be carried out.

An implementation of a semi-decision procedure for context unification has been carried out by Jordi Lévy, and we applied it successfully to some simple ellipsis examples. Further experimentation is needed. Hopefully there are decidable fragments of the context unification problem that are empirically adequate for the phenomena we wish to model.

References

- Alshawi, H. and D. Crouch. 1992. Monotonic semantic interpretation. In *30th Annual Meeting of the Association of Computational Linguistics*, pages 32–38.
- Asher, Nick. 1993. *Reference to abstract objects in discourse*. Kluwer, Dordrecht.
- Baader, F. and J. Siekmann. 1993. Unification theory. In D. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford University Press.
- Bos, Johan, Björn Gambäck, Christian Lieske, Yoshiki Mori, Manfred Pinkal, and Karsten Worm. 1996. Compositional semantics in Verbmobil. In *Proceedings of the 16th International Conference on Computational Linguistics*, volume 1, pages 131–136, København, Denmark, August. ACL.
- Comon, Hubert. 1992. Completion of rewrite systems with membership constraints. In W. Kuich, editor, *Proc. 19th Int. Coll. on Automata, Languages and Programming, LNCS 623*, Vienna. Springer-Verlag.
- Copestake, A. and E. J. Briscoe. 1995. Semi-productive polysemy and sense extension. *Journal of Semantics*, 12:15–67.

- Dalrymple, Mary, Stuart Shieber, and Fernando Pereira. 1991. Ellipsis and higher order unification. *Linguistics and Philosophy*, 14:399–452.
- Dowty, D., R. Wall, and S. Peters. 1981. *Introduction to Montague semantics*. Reidel, Dordrecht.
- Egg, M. and K. Lebeth. 1995. Semantic underspecification and modifier attachment ambiguities. In J. Kilbury and R. Wiese, editors, *Integrative Ansätze in der Computerlinguistik*. Duesseldorf, pages 19–24.
- Gardent, Claire and Michael Kohlhase. 1996. Focus and higher-order unification. In *Proceedings of COLING-96*, Copenhagen.
- Gardent, Claire, Michael Kohlhase, and Noor van Leusen. 1996. Corrections and Higher-Order Unification. In *Proceedings of KONVENS-96*. De Gruyter, Bielefeld, Germany, pages 268–279.
- Goldfarb, W. D. 1981. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230.
- Hirschbuehler, Paul. 1982. Vp deletion and across the board quantifier scope. In J. Pustejovsky and P. Sells, editors, *NELS 12*, University of Massachusetts, Amherst.
- Kamp, H. and U. Reyle. 1993. *From Discourse to Logic*. Kluwer, Dordrecht.
- Lévy, Jordi. 1996. Linear second order unification. In *Proceedings of the Conference on Rewriting Techniques and Applications*. Springer-Verlag.
- Makanin, G.S. 1977. The problem of solvability of equations in a free semigroup. *Soviet Akad. Nauk SSSR*, 223(2).
- Montague, R. 1974. The proper treatment of quantification in ordinary english. In R. Thomason, editor, *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, New Haven and London, pages 247–271.
- Niehren, Joachim, Manfred Pinkal, and Peter Ruhrberg. 1997. On equality up-to constraints over finite trees, context unification and one-step rewriting. In *Proceedings of the 14th International Conference on Automated Deduction*. A complete version is available from <http://www.ps.uni-sb.de/~niehren>. In Press.
- Pinkal, Manfred. 1995. Radical underspecification. In Paul Dekker and Martin Stokhof, editors, *Proceedings of the 10th Amsterdam Colloquium*, University of Amsterdam.
- Pustejovsky, J. 1995. *The Generative Lexicon*. MIT press, Mambridge MA.
- Reyle, Uwe. 1993. Dealing with ambiguities by underspecification: construction, representation, and deduction. *Journal of Semantics*, 10:123–179.
- Reyle, Uwe. 1995. Co-indexing labelled DRSs to represent and reason with ambiguities. In S. Peters and K. van Deemter, editors, *Semantic Ambiguity and Underspecification*. CSLI Publications, Stanford.
- Schmidt-Schauß, Manfred. 1994. Unification of stratified second-order terms. Technical Report internal report 12/94, J. W. Goethe Universität, Frankfurt, Germany.
- Shieber, Stuart, Fernando Pereira, and Mary Dalrymple. 1996. Interactions of scope and ellipsis. *Linguistics and Philosophy*, 19:527–552.