Constraints over Lambda-Structures in Semantic Underspecification

Markus Egg and **Joachim Niehren*** and **Peter Ruhrberg** and **Feiyu Xu**
Department of Computational Linguistics / *Programming Systems Lab
Universität des Saarlandes, Saarbrücken, Germany
{egg,peru,feiyu}@coli.uni-sb.de
niehren@ps.uni-sb.de

Abstract

We introduce a first-order language for semantic underspecification that we call Constraint Language for Lambda-Structures (CLLS). A λ-structure can be considered as a λ-term up to consistent renaming of bound variables (α-equality); a constraint of CLLS is an underspecified description of a λ-structure. CLLS solves a capturing problem omnipresent in underspecified scope representations. CLLS features constraints for dominance, lambda binding, parallelism, and anaphoric links. Based on CLLS we present a simple, integrated, and underspecified treatment of scope, parallelism, and anaphora.

1 Introduction

A central concern of semantic underspecification (van Deemter and Peters, 1996) is the underspecification of the scope of variable binding operators such as quantifiers (Hobbs and Shieber, 1987; Alshawi, 1990; Reyle, 1993). This immediately raises the conceptual problem of how to avoid variable-capturing when instantiating underspecified scope representations. In principle, capturing may occur in all formalisms for structural underspecification which represent binding relations by the coordination of variables (Reyle, 1995; Pinkal, 1996; Bos, 1996; Niehren et al., 1997a). Consider for instance the verb phrase in

(1) Manfred [$_{VP}$ knows every student]

An underspecified description of the compositional semantics of the VP in (1) might be given along the lines of (2):

(2) $X{=}C_1(\forall x(\text{student}(x) \rightarrow C_2(\text{know}(Z,x))))$

The meta-variable X in (2) denotes some tree representing a predicate logic formula which is underspecified for quantifier scope by means of two place holders C_1 and C_2 where a subject-quantifier can be filled in, and a place holder Z for the subject-variable. The binding of the object-variable x by the object-quantifier $\forall x$ is coordinated through the *name* of the object-variable, namely 'x'. Capturing occurs when a new quantifier like $\exists x$ is filled in C_2 whereby the binding between x and $\forall x$ is accidentally undone, and is replaced with a binding of x by $\exists x$.

Capturing problems raised by variable coordination may be circumvented in simple cases where all quantifiers in underspecified descriptions can be assumed to be named by distinct variables. However, this assumption becomes problematic in the light of *parallelism* between the interpretations of two clauses. Consider for instance the *correction* of (1) in (3):

(3) No, Hans [$_{VP}$ knows every student]

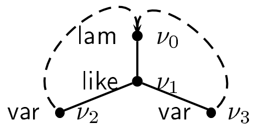
The description of the semantics of the VP in (3) is given in (4):

(4) $Y{=}C_3(\forall y(\text{student}(y) \rightarrow C_4(\text{know}(Z',y))))$

But a full understanding of the combined clauses (1) and (3) requires a grasp of the semantic identity of the two VP interpretations. Now, the VP interpretations (2) and (4) look very much alike but for the different object-variable, namely 'y' instead of 'x'. This illustrates that in cases of parallelism, like in corrections, different variables in parallel quantified structures have to be matched against each other, which requires some form of renaming to be done on them. While this is unproblematic for fully specified structures, it presents serious problems with underspecified structures like (2) and (4), as there the names of the variables are crucial for insuring the right bindings. Any attempt to integrate parallelism with scope underspecification thus has to cope with conflicting requirements on the choice of variable names. Avoiding capturing requires variables to be renamed apart but parallelism needs parallel bound variables to be named alike.

We avoid all capturing and renaming problems by introducing the notion of λ-*structures*, which represent binding relations without naming variables. A λ-*structure* is a standard predicate logic tree structure which can be considered as a λ-term or some other logical formula up-to consistent renaming of bound variables (α-equality). Instead of variable names, a λ-structure provides a partial function on tree nodes for expressing variable binding. A graphical illustration of the λ-structure corresponding to the λ-term λx.like(x,x) is given (5).

(5)

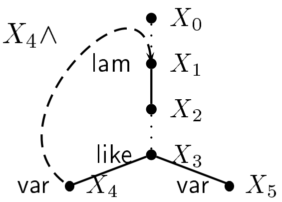


λx.like(x,x)

Formally, the binding relation of the λ-structure in (5) is expressed through the partial function $\lambda^{(5)}$ defined by $\lambda^{(5)}(\nu_2) = \nu_0$ and $\lambda^{(5)}(\nu_3) = \nu_0$.

We propose a first-order constraint language for λ-structures called CLLS which solves the capturing problem of underspecified scope representations in a simple and elegant way. CLLS subsumes dominance constraints (Backofen et al., 1995) as known from syntactic processing (Marcus et al., 1983) with tree-adjoining grammars (Vijay-Shanker, 1992; Rogers and Vijay-Shanker, 1994). Most importantly, CLLS constraints can describe the binding relation of a λ-structure in an underspecified manner (in contrast to λ-structures like (5), which are always fully specified). The idea is that λ-binding behaves like a kind of *rubber band* that can be arbitraryly enlarged but never broken. E.g., (6) is an underspecified CLLS-description of the λ-structure (5).

(6)
$X_0 \lhd^* X_1 \wedge \lambda(X_1)=X_4 \wedge$
$X_1{:}\mathsf{lam}(X_2) \wedge$
$X_2 \lhd^* X_3 \wedge$
$X_3{:}\mathsf{like}(X_4, X_5) \wedge$
$X_4{:}\mathsf{var} \wedge X_5{:}\mathsf{var}$



The constraint (6) does not determine a unique λ-structure since it leaves e.g. the space between the nodes X_2 and X_3 underspecified. Thus, (6) may eventually be extended, say, to a constraint that fully specifies the λ-structure for the λ-term in (7).

(7) $\lambda y.\lambda z.\mathsf{and}(\mathsf{person}(y), \mathsf{like}(y, z))$

λz intervenes between λy and an occurrence of y when extending (6) to a representation of (7) without the danger of undoing their binding.

CLLS is sufficiently expressive for an integrated treatment of semantic underspecification, parallelism, and anaphora. To this purpose it provides parallelism constraints (Niehren and Koller, 1998) of the form $X/X' {\sim} Y/Y'$ reminiscent to equality up-to constraints (Niehren et al., 1997a), and anaphoric bindings constraints of the form $\mathsf{ante}(X){=}X'$.

As proved in (Niehren and Koller, 1998), CLLS extends the expressiveness of context unification (Niehren et al., 1997a). It also extends its linguistic coverage (Niehren et al., 1997b) by integrating an analysis of VP ellipses with anaphora as in (Kehler, 1995). Thus, the coverage of CLLS is comparable to Crouch (1995) and Shieber et al. (1996). We illustrate CLLS at a benchmark case for the interaction of scope, anaphora, and ellipsis (8).

(8) Mary read a book she liked before Sue did.

The paper is organized as follows. First, we introduce CLLS in detail and define its syntax and semantics. We illustrate CLLS in sec. 3 by applying it to the example (8) and compare it to related work in the last section.

2 A Constraint Language for λ-Structures (CLLS)

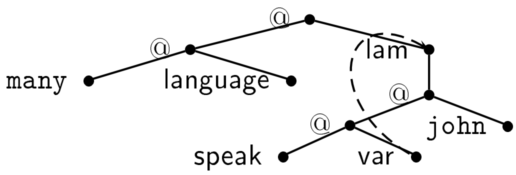
CLLS is an ordinary first-order language interpreted over λ-structures. λ-structures are particular predicate logic tree structures we will introduce. We first exemplify the expressiveness of CLLS.

2.1 Elements of CLLS

A λ-structure is a tree structure extended by two additional relations (the binding and the linking relation). We represent λ-structures as graphs. Every λ-structure characterizes a unique λ-term or a logical formula up to consistent renaming of bound variables (α-equality). E.g., the λ-structure (10) characterizes the higher-order logic (HOL) formula (9).

(9) (many(language))(λx.speak(x)(john))

(10)



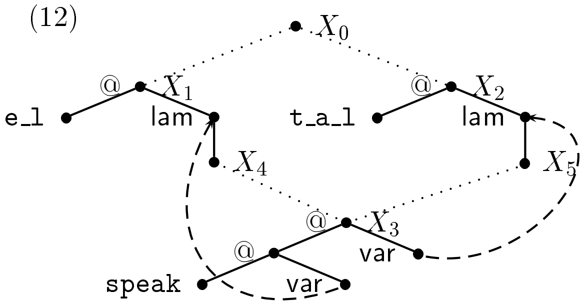
Two things are important here: the label '@' represents explicitly the operation of function application, and the binding of the variable x by the λ-operator λx is represented by an explicit *binding relation* λ between two nodes, labelled as var and lam. As the binding relation is explicit, the variable and the binder need not be given a name or index such as x.

We can fully describe the above λ-structure by means of the constraints for immediate dominance and labeling $X{:}f(X_1, \ldots, X_n)$, (e.g. $X_1{:}@(X_2, X_3)$ and $X_3{:}\mathsf{lam}(X_4)$ etc.) and binding constraints $\lambda(X){=}Y$. It is convenient to display such constraints graphically, in the style of (6). The difference of graphs as constraints and graphs as λ-structures is important since underspecified structures are always seen as descriptions of the λ-structures that satisfy them.

Dominance. As a means to underspecify λ-structures, CLLS employs constraints for *dominance* $X{\triangleleft^*}Y$. Dominance is defined as the transitive and reflexive closure of immediate dominance. We represent dominance constraints graphically as dotted lines. E.g., in (11) we have the typical case of undetermined scope. It is analysed by constraint (12), where two nodes X_1 and X_2, lie between an upper bound X_0 and a lower bound X_3. The graph can be linearized by adding either a constraint $X_1{\triangleleft^*}X_2$ or $X_2{\triangleleft^*}X_1$, resulting in the two possible scoping readings for the sentence (11).

(11) Every linguist speaks two Asian languages.

(12)



Parallelism. (11) may be continued by an elliptical sentence, as in (13).

(13) Two European ones too.

We analyse elliptical constructions by means of a *parallelism constraint* of the form

(14) $X_s/X_p{\sim}Y_t/Y_p$

which has the intuitive meaning that the semantics X_s of the *source clause* (12) is *parallel* to the semantics Y_t of the elliptical *target clause*, *up-to* the exceptions X_p and Y_p, which are the semantic representations of the so called *parallel elements* in source and target clause. In this case the parallel elements are the two subject NPs.

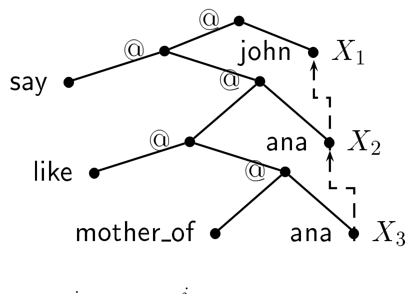
(11) and (13) together give us a 'Hirschbühler sentence' (Hirschbühler, 1982), and our treatment in this case is descriptively equivalent to that of (Niehren et al., 1997b). Our parallelism constraints and their *equality up-to* constraints have been shown to be (non-trivially) intertranslatable (Niehren and Koller, 1998) if binding and linking relations in λ-structures are ignored.

For the interaction of binding with parallelism we follow the basic idea that binding relations should be isomorphic between two similar substructures. The cases where anaphora interact with ellipsis are discussed below.

Anaphoric links. We represent anaphoric dependencies in λ-structures by another explicit relation between nodes, the *linking* relation. An anaphor (i.e. a node labelled as ana) may be linked to an antecedent node, which may be labelled by a name or var, or even be another anaphor. Thus, links can form chains as in (15), where a constraint such as $\mathsf{ante}(X_3){=}X_2$ is represented by a dashed line from X_3 to X_2.

The constraint (15) analyzes (16), where the second pronoun is regarded as to be linked to the first, rather than linked to the proper name:

(15)



(16) John^i said he^j_i liked his$_j$ mother

In a semantic interpretation of λ-structures, analoguously to a semantics for lambda terms,[1] linked nodes get identical denotations. Intuitively, this means they are interpreted as if names, or variables with their binding relations, would be copied down the link chain. It is crucial though not to use such copied structures right away: the link relation gives precise control over strict and sloppy interpretations when anaphors interact with parallelism.

E.g., (16) is the source clause of the many-pronouns-puzzle, a problematic case of interaction of ellipsis and anaphora. (Xu, 1998), where our treatment of ellipsis and anaphora was developed, argues that link chains yield the best explanation for the distribution of strict/sloppy readings involving many pronouns.

The basic idea is that an elided pronoun can either be linked to its parallel pronoun in the source clause (referential parallelism) or be linked in a structurally parallel way (structural parallelism). This analysis agrees with the proposal made in (Kehler, 1993; Kehler, 1995). It covers a series of problematic cases in the literature such as the many-pronouns-puzzle, cascaded ellipsis, or the five-reading sentence (17):

(17) John revised his paper before the teacher did, and so did Bill

The precise interaction of parallelism with binding and linking relations is spelled out in sec. 2.2.

2.2 Syntax and Semantics of CLLS

We start with a set of labels $\Sigma = \{@^2, \mathsf{lam}^1, \mathsf{var}^0, \mathsf{ana}^0, \mathsf{before}^2, \mathsf{mary}^0, \mathsf{read}^0, \ldots\}$, ranged over by f^i, with arity i which may be omitted. The syntax of CLLS is given by:

$$\begin{array}{lll} \varphi ::= & X{:}f(X_1,\ldots,X_n) & (f^n \in \Sigma) \\ \mid & X \triangleleft^* Y & \\ \mid & \lambda(X)=Y & \\ \mid & \mathsf{ante}(X)=Y & \\ \mid & X/X' \sim Y/Y' & \\ \mid & \varphi \wedge \varphi' & \end{array}$$

The semantics of CLLS is given in terms of first order structures L, obtained from underlying tree structures, by adding relations ζ_L for each CLLS relation symbol $\zeta \in \{\triangleleft^*, \lambda(\cdot)=\cdot, \mathsf{ante}(\cdot)=\cdot, \cdot/\cdot\sim\cdot/\cdot, :@, :\mathsf{lam}, :\mathsf{var}, \ldots\}$.

[1] We abstain from giving such a semantics here, as we would have to introduce types, which are of no concern here, to keep the semantics simple.

A (finite) tree structure, underlying L, is given by a set of *nodes* $\nu, \nu', \ldots$ connected by *paths* $\pi, \pi', \ldots$ (possibly empty words over positive integers), and a *labelling function* l from nodes to labels. The number of daughters of a node matches the arity of its label. The relationship $\nu{:}f_L(\nu_1, \ldots, \nu_n)$ holds iff $l(\nu)=f$ and $\nu.i = \nu_i$ for $i = 1..n$, where $\nu.\pi$ stands for the node that is reached from ν by following the path π (if defined). To express that a path π is defined on a node ν in L we write $\nu.\pi{\downarrow}_L$. We write $\pi \leq \pi'$ for π being an initial segment of π'. The *dominance* relation $\nu \triangleleft^*_L \nu'$ holds if $\exists \pi\ \nu.\pi = \nu'$. If π is non-empty we have *proper dominance* $\nu \triangleleft^+_L \nu'$.

A *λ-structure* L is a tree structure with two (partially functional) binary relations $\lambda_L(\cdot)=\cdot$, for *binding*, and $\mathsf{ante}_L(\cdot)=\cdot$, for anaphor-to-antecedent *linking*. We assume that the following conditions hold: (1) binding only holds between variables (nodes labelled var) to λ-binders (nodes labelled lam); (2) every variable has exactly one binder; (3) variables are dominated by their binders; (4) only anaphors (nodel labelled ana) are linked to antecendents; (2) every anaphor has exactly one antecendent; (5) antecedents are terminal nodes; (6) there are no cyclic link chains; (7) if a link chain ends at a variable then each anaphor in the chain must be dominated by the binder of that variable.

The not so straight forward part of the semantics of CLLS is the notion of *parallelism*, which we define for any given λ-structure L as follows:

$$\nu_1/\nu_1' \sim_L \nu_2/\nu_2'$$

iff there is a path π_0 such that:

1. π_0 is the "exception path" from the top node of the parallel structures the the two exception positions: $\nu_1' = \nu_1.\pi_0 \wedge \nu_2' = \nu_2.\pi_0$
2. the two *contexts*, which are the trees below ν_1 and ν_2 up-to the trees below the exception positions ν_1' and ν_2', must have the same structure and labels: $\forall \pi\ \ \neg \pi_0 \leq \pi \ \Rightarrow\ ((\nu_1.\pi{\downarrow}_L \ \Leftrightarrow\ \nu_2.\pi{\downarrow}_L) \wedge (\nu_1.\pi{\downarrow}_L \Rightarrow l(\nu_1.\pi) = l(\nu_2.\pi))))$
3. there are no 'hanging' binders from the contexts to variables outside them: $\forall \nu \forall \nu'\ \neg(\nu_1 \triangleleft^*_L \nu \triangleleft^+_L \nu_1' \triangleleft^*_L \nu' \wedge \lambda_L(\nu')=\nu)$
4. binding is structurally isomorphic within the two contexts:

$\forall\pi\forall\pi' \; \neg\pi_0{\leq}\pi \wedge \nu_1.\pi{\downarrow}_L \wedge \neg\pi_0{\leq}\pi' \wedge \nu_1.\pi'{\downarrow}_L \Rightarrow$
$(\lambda_L(\nu_1.\pi){=}\nu_1.\pi' \Leftrightarrow \lambda_L(\nu_2.\pi){=}\nu_2.\pi')$

5. two variables in identical positions within their context and bound outside their context must be bound by the same binder:
$\forall\nu\forall\pi \; (\nu \triangleleft_L^+ \nu_1 \vee \nu \triangleleft_L^+ \nu_2) \wedge \neg\pi{\leq}\pi_0 \wedge \nu_1.\pi{\downarrow}_L \Rightarrow$
$(\lambda_L(\nu_1.\pi){=}\nu \Leftrightarrow \lambda_L(\nu_2.\pi){=}\nu)$

6. two anaphors in identical positions within their context must have isomorphic links within their context, or the target sentence anaphor is linked to the source sentence anaphor:
$\forall\nu\forall\pi \quad \neg\pi_0{\leq}\pi \wedge \nu_1.\pi{\downarrow}_L \wedge \mathsf{ante}_L(\nu_1.\pi){=}\nu \Rightarrow$
$(\exists\pi'(\nu{=}\nu_1.\pi' \wedge \neg\pi_0{<}\pi' \wedge \mathsf{ante}_L(\nu_2.\pi){=}\nu_2.\pi')$
$\vee \; \mathsf{ante}_L(\nu_2.\pi){=}\nu_1.\pi)$

3 Interaction of quantifiers, anaphora, and ellipsis

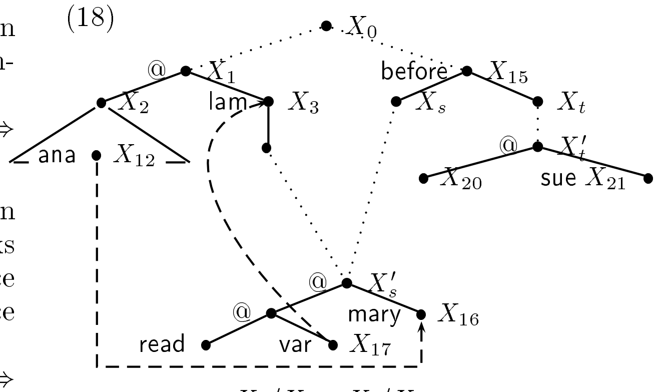
In this section, we will illustrate our analysis of a complex case of the interaction of scope, anaphora, and ellipsis. In the case (8), both anaphora and quantification interact with ellipsis.

(8) Mary read a book she liked before Sue did.

(8) has three readings (see (Crouch, 1995) for a discussion of a similar example). In the first, the indefinite NP a book she liked takes wide scope over both clauses (a particular book liked by Mary is read by both Mary and Sue). In the two others, the operator before outscopes the indefinite NP. The two options result from the two possibilities of reconstructing the pronoun she in the ellipsis interpretation, viz., 'strict' (both read some book that Mary liked) and 'sloppy' (each read some book she liked herself).

The constraint for (8), displayed in (18), is an underspecified representation of the above three readings. It can be derived in a compositional fashion along the lines described in (Niehren et al., 1997b). X_s and X_t represent the semantics of the source and the target clause, while X_{16} and X_{21} stand for the semantics of the parallel elements (Mary and Sue) respectively. For readability, we represent the semantics of the complex NP a book she liked by a triangle dominated by X_2, which only makes the anaphoric content X_{12} of the pronoun she within the NP explicit. The anaphoric relationship between the pronoun she and Mary is represented by the linking relation between X_{12} and X_{16}. (X_{20} represents the semantics of the elided part of the target clause.)

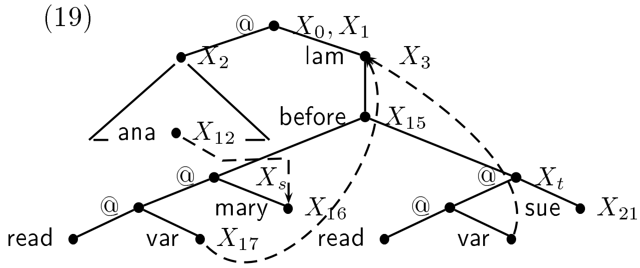
(18)



$X_s/X_{16}{\sim}X_t/X_{21}$

The first reading, with the NP taking wide scope, results when the relative scope between X_1 and X_{15} is resolved such that X_1 dominates X_{15}. The corresponding solution of the constraint is visualized in (19).

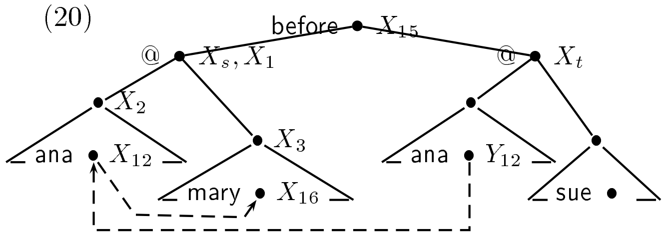
(19)



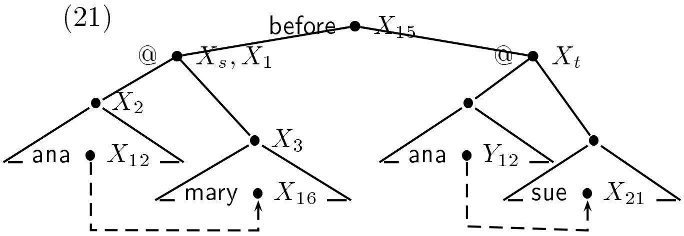
The parallelism constraint $X_s/X_{16}{\sim}X_t/X_{21}$ is satisfied in the solution because the node X_t dominates a tree that is a copy of the tree dominated by X_s. In particular, it contains a node labelled by var, which has to be parallel to X_{17}, and therefore must be λ-linked to X_3 too.

The other possible scoping is for X_{15} to dominate X_1. The two solutions this gives rise to are drawn in (20) and (21). Here X_1 and the interpretation of the indefinite NP directly below enter into the parallelism as a whole, as these nodes lie below the source node X_s. Thus, there are two anaphoric nodes: X_{12} in the source and its 'copy' Y_{12} in the target semantics. For the copy to be parallel to X_{12} it can either have a link to X_{12} to have a same referential value (strict reading, see (20)) or a link to X_{21} that is structurally parallel to the link from X_{12} to X_{16}, and hence leads to the node of the parallel element Sue (sloppy reading, see (21)).

(20)



(21)



4 Related Work

CLLS allows a uniform and yet internally structured approach to semantic ambiguity. We use a single constraint formalism in which to describe different kinds of information about the meaning of an utterance. This avoids the problems of order dependence of processing that for example Shieber et al. (1996) get by interleaving two formalisms (for scope and for ellipsis resolution). Our approach follows Crouch (1995) in this respect, who also includes parallelism constraints in the form of substitution expressions directly into an underspecified semantic formalism (in his case the formalism of Quasi Logical Forms QLF). We believe that the two approaches are roughly equivalent empirically. But in contrast to CLLS, QLF is not formalised as a general constraint language over tree-like representations of meaning. QLF has the advantage of giving a more direct handle on meanings themselves - at the price of its relatively complicated model theoretic semantics. It seems harder though to come up with solutions within QLF that have an easy portability across different semantic frameworks.

We believe that the ideas from CLLS tie in quite easily with various other semantic formalisms, such as UDRT (Reyle, 1993) and MRS (Copestake et al., 1997), which use dominance relations similar to ours, and also with theories of Logical Form associated with GB style grammars, such as (May, 1977). In all these frameworks one tends to use variable-coordination (or coindexing) rather than the explicit binding and linking relations we have presented here. We hope that these approaches can potentially benefit from the presented idea of rubber bands for binding and linking, without having to make any dramatic changes.

Our definition of parallelism implements some ideas from Hobbs and Kehler (1997) on the behavior of anaphoric links. In contrast to their proposal, our definition of parallelism is not based on an abstract notion of *similarity*. Furthermore, CLLS is not integrated into a general theory of abduction. We pursue a more modest aim at this stage, as CLLS needs to be connected to "material" deduction calculi for reasoning with such underspecified semantic representation in order to make progress on this front. We hope that some of the more ad hoc features of our definition of parallelism (e.g. axiom 5) may receive a justification or improvement in the light of such a deeper understanding.

Context Unification. CLLS extends the expressiveness of context unification (CU) (Niehren et al., 1997a), but it leads to a more direct and more structured encoding of semantic constraints than CU could offer. There are three main differences between CU and CLLS. 1) In CLLS variables are interpreted over *nodes* rather than whole trees. This gives us a direct handle on *occurrences* of semantic material, where CU could handle occurrences only indirectly and less efficiently. 2) CLLS avoids the capturing problem. 3) CLLS provides explicit anaphoric links, which could not be adequately modeled in CU.

The insights of the CU-analysis in (Niehren et al., 1997b) carry over to CLLS, but the awkward second-order equations for expressing dominance in CU can be omitted (Niehren and Koller, 1998). This omission yields an enormous simplification and efficiency gain for processing.

Tractability. The distinguishing feature of our approach is that we aim to develop efficiently treatable constraint languages rather than to apply maximally general but intractable formalisms. We are confident that CLLS can be implemented in a simple and efficient manner. First experiments which are based on high-level concurrent constraint programming have shown promising results.

5 Conclusion

In this paper, we presented CLLS, a first-order language for semantic underspecification. It represents ambiguities in simple underspecified structures that are transparent and suitable for processing. The application of CLLS to some difficult cases of ambiguity has shown that it is well suited for the task of representing ambiguous expressions in terms of underspecification.

Acknowledgements

This work was supported by the SFB 378 (project CHORUS) at the Universität des Saarlandes. The authors wish to thank Manfred Pinkal, Gert Smolka, the commentators and participants at the Bad Teinach workshop on underspecification, and our anonymous reviewers.

References

Hiyan Alshawi. 1990. Resolving quasi logical form. *Computational Linguistics*, 16:133–144.

R. Backofen, J. Rogers, and K. Vijay-Shanker. 1995. A first-order axiomatization of the theory of finite trees. *J. Logic, Language, and Information*, 4:5–39.

Johan Bos. 1996. Predicate logic unplugged. In *Proceedings 10th Amsterdam Colloquium*, pages 133–143.

Ann Copestake, Dan Flickinger, and Ivan Sag. 1997. Minimal Recursion Semantics. An Introduction. Manuscript, available at `ftp://csli-ftp.stanford.edu/linguistics/sag/mrs.ps.gz`.

Richard Crouch. 1995. Ellipsis and quantification: A substitutional approach. In *Proceedings EACL'95*, pages 229–236, Dublin.

Paul Hirschbühler. 1982. VP deletion and across the board quantifier scope. In J. Pustejovsky and P. Sells, editors, *NELS 12*, Univ. of Massachusetts.

Jerry R Hobbs and Andrew Kehler. 1997. A theory of parallelism and the case of VP-ellipsis. In *Proceedings ACL'97*, pages 394–401, Madrid.

J.R. Hobbs and S. Shieber. 1987. An algorithm for generating quantifier scoping. *Computational Linguistics*, 13:47–63.

Andrew Kehler. 1993. A discourse copying algorithm for ellipsis and anaphora resolution. In *Proceedings of EACL*.

Andrew Kehler. 1995. *Interpreting Cohesive Forms in the Context of Discourse Inference.* Ph.D. thesis, Harvard University.

M. Marcus, D. Hindle, and M. Fleck. 1983. D-theory: Talking about talking about trees. In *Proceedings of the 21st ACL*, pages 129–136.

Robert May. 1977. *The Grammar of Quantification.* Doctoral dissertation, MIT, Cambridge Mass.

Joachim Niehren and Alexander Koller. 1998. Dominance Constraints in Context Unification, January. `http://www.ps.uni-sb.de/Papers/abstracts/Dominance.html`.

J. Niehren, M. Pinkal, and P. Ruhrberg. 1997a. On equality up-to constraints over finite trees, context unification, and one-step rewriting. In *Proceedings 14th CADE.* Springer-Verlag, Townsville.

J. Niehren, M. Pinkal, and P. Ruhrberg. 1997b. A uniform approach to underspecification and parallelism. In *Proceedings ACL'97*, pages 410–417, Madrid.

Manfred Pinkal. 1996. Radical underspecification. In *Proceed. 10th Amsterdam Colloquium*, pages 587–606.

Uwe Reyle. 1993. Dealing with ambiguities by underspecification: construction, representation, and deduction. *Journal of Semantics*, 10:123–179.

Uwe Reyle. 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.

J. Rogers and K. Vijay-Shanker. 1994. Extracting trees from their descriptions: an application to tree-adjoining grammars. *Computational Intelligence*, 10:401–421.

Stuart Shieber, Fernando Pereira, and Mary Dalrymple. 1996. Interaction of scope and ellipsis. *Linguistics and Philosophy*, 19:527–552.

Kees van Deemter and Stanley Peters. 1996. *Semantic Ambiguity and Underspecification.* CSLI, Stanford.

K. Vijay-Shanker. 1992. Using description of trees in tree adjoining grammar framework. *Computational Linguistics*, 18.

Feiyu Xu. 1998. Underspecified representation and resolution of ellipsis. Master's thesis, Universität des Saarlandes. `http://www.coli.uni-sb.de/~feiyu/thesis.html`.