Talking about Trees and Truth-conditions *

Reinhard Muskens

Abstract

An attractive way to model the relation between an underspecified syntactic representation and its completions is to let the underspecified representation correspond to a *logical description* and the completions to the *models* of that description. This approach, which underlies the *Description Theory* of (Marcus et al. 1983) was integrated with a pure unification approach to *Lexicalized Tree-Adjoining Grammars* (Joshi et al. 1975, Schabes 1990) in (Vijay-Shanker 1992) and was further developed in the 'D-Tree Grammars' (DTG) of (Rambow et al. 1995). We generalize Description Theory by integrating semantic information, that is, we propose to tackle both syntactic *and* semantic underspecification using descriptions. Our focus will be on underspecification of *scope*. We use a generalized and completely declarative version of the D-Tree formalism. Although trees in our set-up have surface strings at their leaves and are in fact very close to ordinary surface trees, there is also a strong connection with the *Logical Forms* (LFs) of (May 1977). We associate logical interpretations with these LFs using a technique of *internalising the logical binding mechanism* (Muskens 1996). The net result is that we obtain a Description Theory-like grammar in which the descriptions underspecify semantics. Since everything is framed in classical logic it is easily possible to *reason* with these descriptions.

Contents

*This work is part of an ongoing collaboration with Emiel Krahmer. I wish to thank Kurt Eberle, Barbara Partee, Stanley Peters and all other participants of the Bad Teinach Workshop on Models of Underspecification and the Representation of Meaning (May 1998) for their comments and criticisms on an earlier version of this paper. Stimulating comments were also obtained from the participants of the Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks in Philadelphia (August 1998).

An attractive way to model the relation between an underspecified syntactic representation and its completions is to let the underspecified representation correspond to a *logical description* and the completions to the *models* of that description. This approach, which underlies the *Description Theory* of (Marcus et al. 1983) was integrated with a pure unification approach to *Lexicalized Tree-Adjoining Grammars* (Joshi et al. 1975, Schabes 1990) in (Vijay-Shanker 1992) and was further developed in the 'D-Tree Grammars' (DTG) of (Rambow et al. 1995). We generalize Description Theory by integrating semantic information, that is, we propose to tackle both syntactic *and* semantic underspecification using descriptions.[1] Our focus will be on underspecification of *scope*. We use a generalized and completely declarative version of the D-Tree formalism. Although trees in our set-up have surface strings at their leaves and are in fact very close to ordinary surface trees, there is also a strong connection with the *Logical Forms* (LFs) of (May 1977). We associate logical interpretations with these LFs using a technique of *internalising the logical binding mechanism* (Muskens 1996). The net result is that we obtain a Description Theory-like grammar in which the descriptions underspecify semantics. Since everything is framed in classical logic it is easily possible to *reason* with these descriptions.

1 Syntactic Composition

Descriptions in our theory model three kinds of information. First, there are *input descriptions*, which vary per sentence. For example, for sentence (1) we have (2) as an input description. It says that there are two lexical nodes,[2] labeled *John* and *walks* respectively; that the first of these precedes the second; and that these two lexical nodes are all that were encountered. Secondly, there is a *lexicon* which includes semantic information. The entries for *John* and *walks* are given in (3) and (4).

(1) John walks.

(2) $\exists n_1 n_2 (n_1 \prec n_2 \wedge lab(n_1, john) \wedge lab(n_2, walks) \wedge \forall n(lex(n) \leftrightarrow (n = n_1 \vee n = n_2)))$

(3) $\forall n_1(lab(n_1, john) \rightarrow$
$\exists n_3(lab(n_3, np) \wedge n_3 \lhd n_1 \wedge \sigma(n_3) = John \wedge$
$\forall n(\alpha^+(n) = n_1 \leftrightarrow (n = n_3 \vee n = n_1)) \wedge \forall n(\alpha^-(n) = n_1 \leftrightarrow n = n_1)))$

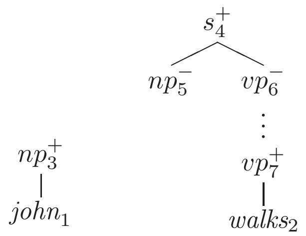
(4) $\forall n_2(lab(n_2, walks) \rightarrow$
$\exists n_4 n_5 n_6 n_7 (lab(n_4, s) \wedge lab(n_5, np) \wedge lab(n_6, vp) \wedge lab(n_7, vp) \wedge n_4 \lhd n_5 \wedge$
$n_4 \lhd n_6 \wedge$
$n_6 \lhd^* n_7 \wedge n_7 \lhd n_2 \wedge n_5 \prec n_6 \wedge \sigma(n_4) = \sigma(n_6)(\sigma(n_5)) \wedge \sigma(n_7) = \lambda v.walk\, v$

[1]The approach to underspecified semantics taken in (Muskens 1995) was very much inspired by Description Theory and the work of Vijay-Shanker in (Vijay-Shanker 1992) but did not offer an actual integration with Tree-Adjoining Grammars. In this paper we endeavour to set this right.

[2]With *lexical nodes* we mean those leaves in a tree which carry a lexeme.

$\wedge \forall n(\alpha^+(n) = n_2 \leftrightarrow (n = n_4 \vee n = n_7 \vee n = n_2))$
$\wedge \forall n(\alpha^-(n) = n_2 \leftrightarrow (n = n_5 \vee n = n_6 \vee n = n_2))))$

The function symbol α^+ used in these descriptions *positively anchors* nodes to lexical nodes, α^- *negatively anchors* nodes and σ gives a node its semantic value. Since descriptions are unwieldy we partially abbreviate them with the help of pictures:



Here uninterrupted lines represent immediate dominance ($\lhd$) and dotted lines represent dominance ($\lhd^*$), as usual. Additionally we mark positive and negative anchoring in the following way. If a description contains the information that a certain node is positively (negatively) anchored, the term referring to that node gets a plus (minus) sign. But pluses and minuses cancel and terms that would get a $\pm$ by the previous rule will be left unmarked. Terms marked with a plus (minus) sign are to be compared with the bottom (top) parts of Vijay-Shanker's 'quasi-nodes' in (Vijay-Shankar 1992). There is also an obvious close connection with positive (negative) occurrences of types in complex types in Categorial Grammar (CG).[3]

To the third and final kind of descriptions belong axioms which say that $\lhd$, $\lhd^*$ and $\prec$ behave like immediate dominance, dominance and precedence in trees ($\mathcal{A}1$ - $\mathcal{A}10$, see also e.g., Cornell 1994, Backofen et al. 1995:9)[4] combined with other general information, such as the statements that labeling is functional ($\mathcal{A}11$), and that different label names denote different labels ($\mathcal{A}12$). $\mathcal{A}13$ and $\mathcal{A}14$ say that all nodes must be positively anchored to lexical nodes and that all nodes except the root r are also negatively anchored to some lexical node. The root is negatively anchored to itself ($\mathcal{A}15$).

$\mathcal{A}1$ $\quad \forall k\,[r \lhd^+ k \vee r = k]$

$\mathcal{A}2$ $\quad \forall k \neg\, k \lhd^+ k$

$\mathcal{A}3$ $\quad \forall k_1 k_2 k_3\,[[k_1 \lhd^+ k_2 \wedge k_2 \lhd^+ k_3] \rightarrow k_1 \lhd^+ k_3]$

$\mathcal{A}4$ $\quad \forall k \neg\, k \prec k$

$\mathcal{A}5$ $\quad \forall k_1 k_2 k_3\,[[k_1 \prec k_2 \wedge k_2 \prec k_3] \rightarrow k_1 \prec k_3]$

$\mathcal{A}6$ $\quad \forall k_1 k_2\,[k_1 \prec k_2 \vee k_2 \prec k_1 \vee k_1 \lhd^+ k_2 \vee k_2 \lhd^+ k_1 \vee k_1 = k_2]$

[3]Interestingly, Prof. Guy Perrier of Nancy University independently arrives at a syntactic model strikingly similar to ours by translating proof nets from Lambek Categorial Grammar / Linear Logic in a certain way (Perrier 1998). The similarity between D-Trees and CG is also the subject of (Hepple 1998).

[4]Note that $\mathcal{A}9$ and $\mathcal{A}10$ in themselves do not suffice to exclude that some nodes are connected by a dominance relation without there being a (finite) path of immediate dominances between them. In fact the nature of our input descriptions and the form of our lexicon exclude this.

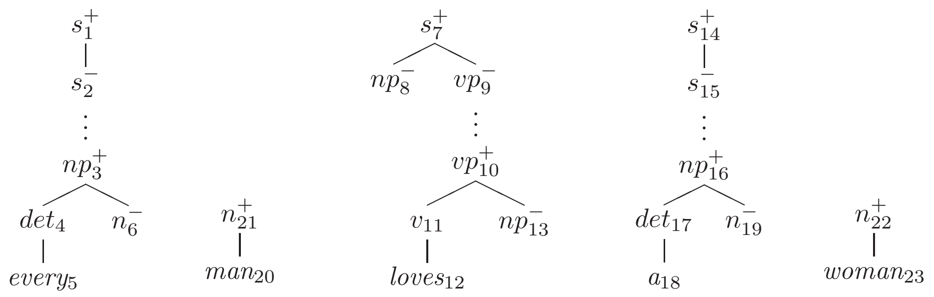


Figure 1: Elementary descriptions for *every man loves a woman*

$\mathcal{A}7$ $\forall k_1 k_2 k_3\, [[k_1 \lhd^+ k_2 \wedge k_1 \prec k_3] \rightarrow k_2 \prec k_3]$

$\mathcal{A}8$ $\forall k_1 k_2 k_3\, [[k_1 \lhd^+ k_2 \wedge k_3 \prec k_1] \rightarrow k_3 \prec k_2]$

$\mathcal{A}9$ $\forall k_1 k_2\, [k_1 \lhd k_2 \rightarrow k_1 \lhd^+ k_2]$

$\mathcal{A}10$ $\forall k_1 k_2 k_3\, \neg[k_1 \lhd k_3 \wedge k_1 \lhd^+ k_2 \wedge k_2 \lhd^+ k_3]$

$\mathcal{A}11$ $\forall k \forall \ell_1 \ell_2\, [[lab(k, \ell_1) \wedge lab(k, \ell_2)] \rightarrow \ell_1 = \ell_2]$

$\mathcal{A}12$ $l_1 \neq l_2$, if l_1 and l_2 are distinct label names

$\mathcal{A}13$ $\forall k\, lex(\alpha^+(k))$

$\mathcal{A}14$ $\forall k\, [k = r \vee lex(\alpha^-(k))]$

$\mathcal{A}15$ $\alpha^-(r) = r$

Together with this extra information (2), (3) and (4) conspire to determine a single model. Only n_1 and n_2 are lexical nodes. All nodes must be positively anchored to a lexical node. The set of nodes positively anchored to n_1 is $\{n_1, n_3\}$ and the set positively anchored to n_2 is $\{n_2, n_4, n_7\}$. So the remaining n_5 and n_6 must corefer with one of the constants mentioned, the only possibility being that $n_5 = n_3$ and that $n_6 = n_7$.[5] The reader will note that in the resulting model $\sigma(n_4) = walk\ John$. The general procedure for finding out which models satisfy a given description is to identify positively marked terms with negatively marked ones in a one-to-one fashion. The root counts as negatively marked. The descriptions in our version of D-Tree Grammars (we may perhaps call the syntactic part of the theory *Polarised D-Tree Grammars (PDTG)*) have only a finite number of finite models and, unlike in standard DTG, no circumscription operation such as minimalisation is required to arrive at the intended structures.

In the given example only one tree was described, but this is indeed an exceptional situation. It is far more common that a multiplicity of trees satisfy a given description. This kind of underspecification enabled (Marcus et al. 1983) to define a parser which does not only work in a strict left-right fashion but is also incremental in the sense that at no point during a parse information

[5] In general, the form of our input descriptions and the form of our lexicon will ensure that input descriptions (+ the lexicon + our axioms) will only have models with a finite number of nodes (other domains may be infinite). Universal (existential) quantification over nodes will therefore amount to taking certain finite conjunctions (disjunctions) and, as long as semantics is disregarded (as long as expressions $\sigma(n)$ are treated as unanalizable wholes), entailment is decidable.

need be destroyed. A necessary condition for this form of underspecification is that there are *structures* which can be described. In the context of semantic scope differences it therefore is natural to turn to (May 1977)'s Logical Forms, as these are the kind of models required. In fact we use a variant of May's trees which is very close to ordinary surface structure: although we will allow NPs to be raised, the syntactic material of such NPs will in fact remain *in situ*. But while the only syntactic effect of raising will be the creation of an extra S node and Logical Forms will have their corresponding surface structures as subtrees, the 'movement' has an important effect on semantic interpretation. Consider example (5).

(5) Every man loves a woman.

We have depicted its five lexical items in fig. 1. With two exceptions they pretty much conform to expectation. The exceptions are that each determiner comes with a pair of S nodes dominating its NP. The basic idea here is that the long-distance phenomenon of quantifying-in is treated within the domain of extended locality of a determiner. In each case the semantics of the higher S will be composed out of the semantics of the lower S and the semantics of the NP, the semantic composition rule being quantifying-in.[6] The two Ss are to be compared to the two Ss at the adjunction site of a raised NP in May's theory. There is also an obvious connection with the (single) S where 'NP-retrieval' occurs in Cooper's theory of Quantifier Storage (see Cooper 1983).

It is easily seen that in any model of the descriptions in fig. 1 (+ the input description for (5) + our axioms) certain identities must hold: $n_6 = n_{21}$, $n_{19} = n_{22}$, $n_9 = n_{10}$, $n_8 = n_3$, and $n_{13} = n_{16}$ are derivable. But there is a choice between two further possibilities, as it can be the case that $n_2 = n_{14}$ and $n_{15} = n_7$, or, alternatively, that $n_{15} = n_1$ and $n_2 = n_7$. These two possibilities will correspond to the two different readings of the sentence.

2 Internalising Binding

How can we assign a semantics to the lexical descriptions in fig. 1? We must e.g. be able to express the semantics of n_1 in terms of the semantics of n_2, *whatever the latter turns out to be*, i.e. we must be able to express the result of quantification into an arbitrary context. In mathematical English we can say that, for any φ, the value of $\forall x\varphi$ is the set of assignments a such that for all b differing from a at most in x, b is an element of the value of φ. We need to be able to say something similar in our logical language. The language must talk *about* meaning; it must talk about things that function like variables and constants, things that function like assignments, etc. The first will be called *registers*, the second *states*. Two primitive types are added to the logic: π and s, for registers and states respectively. We shall have *variable registers*, which stand proxy for variables and *constant registers* for constants. However, since registers are simply objects in our models, both variable registers and constant

[6] In this paper only quantification into S is considered, but in a fuller version we shall generalise this to quantification into arbitrary phrasal categories.

$$\sigma(r) = \Pi u_{n_5}[man\ u_{n_5} \Rightarrow \Sigma u_{n_{18}}[woman\ u_{n_{18}}\ \&\ u_{n_5}\ loves\ u_{n_{18}}]]\vee$$
$$\sigma(r) = \Sigma u_{n_{18}}[woman\ u_{n_{18}}\ \&\ \Pi u_{n_5}[man\ u_{n_5} \Rightarrow\ u_{n_5}\ loves\ u_{n_{18}}]]$$

Figure 2: A Derivable Disjunction

registers can be denoted with variables as well as with constants. Here are some axioms:

$\mathcal{A}16$ $\forall i_s \forall v_\pi \forall x_e\, [VAR(v) \to \exists j_s\, [i[v]j \wedge V(v)(j) = x]]$

$\mathcal{A}17$ $\forall k\, VAR(u(k))$

$\mathcal{A}18$ $\forall k_1 k_2\, [u(k_1) = u(k_2) \to k_1 = k_2]$

$\mathcal{A}19$ $\forall i.V(John_\pi)(i) = \text{john}_e$,
$\forall i.V(Mary)(i) = \text{mary},\ldots$

Here VAR is a predicate which singles out variable registers, V assigns a value to each register v in each state j, and $i[\delta]j$ is an abbreviation of $\forall w[w \neq \delta \to V(w)(i) = V(w)(j)]$. $\mathcal{A}16$ forces states to behave like assignments in an essential way. The function u assigns variable registers to nodes ($\mathcal{A}17$). Each node is assigned a fresh register ($\mathcal{A}18$). Constant registers have a fixed value ($\mathcal{A}19$). For more information on a strongly related set of axioms see (Muskens 1996).

These axioms essentially allow our logical language to talk about binding and we can now use this expressivity to embed predicate logic into (the first-order part of) type theory, with the side-effect that binding can take place on the level of registers. Write

$$\begin{aligned}
R\delta_1 \ldots \delta_n \quad &\text{for} \quad \lambda i.R(V(\delta_1)(i), \ldots, V(\delta_n)(i)),\\
\sim \varphi \quad &\text{for} \quad \lambda i \neg\varphi(i),\\
\varphi\ \&\ \psi \quad &\text{for} \quad \lambda i[\varphi(i) \wedge \psi(i)],\\
\varphi \Rightarrow \psi \quad &\text{for} \quad \lambda i[\varphi(i) \to \psi(i)],\\
\Sigma\delta\, \varphi \quad &\text{for} \quad \lambda i \exists j[i[\delta]j \wedge \varphi(j)],\\
\Pi\delta\, \varphi \quad &\text{for} \quad \lambda i \forall j[i[\delta]j \to \varphi(j)].
\end{aligned}$$

We have essentially mimicked the Tarski truth conditions for predicate logic in our object language and in fact it can be proved that, under certain conditions,[7] we can reason with terms generated in this way as if they were the predicate logical formulas they stand proxy for (see Muskens 1998).

It should be stressed that the technique discussed here can be used to embed any logic with a decent interpretation into classical logic. For example, (Muskens 1996) shows that we can use the same mechanism to embed Discourse Representation Theory (DRT, Kamp & Reyle 1993). In a full version of this paper we shall also present a version of our theory based on DRT.

[7]The relevant condition is that in each term φ we are using in this way, and each pair $u(n), u(n')$ occurring in φ, with n and n' syntactically different, we must be justified to assume $n \neq n'$. In the application discussed below this condition is met automatically.

3 Semantic Composition

We can now integrate semantic equations with the lexical items occurring in fig. 1.

$$
\begin{array}{rcl}
\sigma(n_3) & = & u_{n_5} \\
\sigma(n_1) & = & \Pi u_{n_5}[\sigma(n_6)(u_{n_5}) \Rightarrow \sigma(n_2)] \\
\sigma(n_{10}) & = & \lambda v.v \ \mathit{loves} \ \sigma(n_{13}) \\
\sigma(n_7) & = & \sigma(n_9)(\sigma(n_8)) \\
\sigma(n_{16}) & = & u_{n_{18}} \\
\sigma(n_{14}) & = & \Sigma u_{n_{18}}[\sigma(n_{19})(u_{n_{18}}) \ \& \ \sigma(n_{15})] \\
\sigma(n_{21}) & = & \lambda v.\mathit{man} \ v \\
\sigma(n_{22}) & = & \lambda v.\mathit{woman} \ v
\end{array}
$$

The first two equations derive from the lexical item for *every*, the third and fourth from *loves*, the fifth and sixth from *a*, and the last two from the common nouns. Note that in the translation of *every*, n_3 only gets a referent as its translation (namely $u(n_5)$, which for readability we write as u_{n_5}), while the real action is taking place upstairs. A similar remark holds for the other determiner.

The equations given here, together with the other constraints depicted in fig. 1, *describe* the meaning of the input sentence. They do not give its meaning directly, as it is done in more traditional approaches. As we shall see, they also *underspecify* the meaning of (5) in the sense that they do not suffice to enforce that the root r has the same value for σ in all models.

As we have seen earlier, in any model of the relevant descriptions $n_6 = n_{21}$, $n_{19} = n_{22}$, $n_9 = n_{10}$, $n_8 = n_3$, and $n_{13} = n_{16}$ hold. From this it follows that

$$
\begin{array}{rcl}
\sigma(n_7) & = & u_{n_5} \ \mathit{loves} \ u_{n_{18}} \\
\sigma(n_1) & = & \Pi u_{n_5}[\mathit{man} \ u_{n_5} \Rightarrow \sigma(n_2)] \\
\sigma(n_{14}) & = & \Sigma u_{n_{18}}[\mathit{woman} \ u_{n_{18}} \ \& \ \sigma(n_{15})]
\end{array}
$$

The relevant constraints further imply that either $n_2 = n_{14}$ and $n_{15} = n_7$, or, alternatively, that $n_{15} = n_1$ and $n_2 = n_7$. For the moment let us assume the second possibility. Since $u_{n_5} \ \mathit{loves} \ u_{n_{18}}$ is a *closed term* (u is a function constant and n_5 and n_{18} are constants that witness existential quantifiers in the input description of (5)), the assumption that $n_2 = n_7$ allows us to conclude

$$\sigma(n_1) = \Pi u_{n_5}[\mathit{man} \ u_{n_5} \Rightarrow u_{n_5} \ \mathit{loves} \ u_{n_{18}}]$$

Note that this is the point where we have made essential use of our internalisation of binding: had we used ordinary variables instead of our register-denoting terms, the substitution would not have been possible.

Continuing our reasoning, we see that under the given assumption the root node r ($=n_{14}$ in this case) will be assigned the $\exists\forall$ reading of the sentence. Without assumptions the disjunction in fig. 7 is derivable.

We conclude that the leading idea behind Marcus' Description Theory allows us to underspecify semantic information much in the same way as syntactic information is underspecified in his theory. This is partly because we use the technique of 'Talking about Trees' to talk about our variant of LFs, partly because we talk about meaning as well as about trees. The price for using the method is that we must accept that different semantic readings correspond to different structures, as the theory only allows underspecification of the latter.

References

[1] Backofen, R., J. Rogers and K. Vijay-Shanker. 1995. A First-Order Axiomatization of the Theory of Finite Trees. *Journal of Logic, Language and Information* **4**:5-39.

[2] Cooper, R. 1983. *Quantification and Syntactic Theory*. Reidel. Dordrecht.

[3] Cornell, T. 1994. On Determining the Consistency of Partial Descriptions of Trees. *Proceedings of ACL 94*. 163-170.

[4] Hepple, M. 1998. On Some Similarities between D-Tree Grammars and Type-Logical Grammars. *Proc. Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4)*. Institute for Research in Cognitive Science. Philadelphia, PA.

[5] Joshi, A., L. Levy and M. Takahashi. 1975. Tree Adjunct Grammars. *Journal of the Computer and System Sciences* **10**: 136-163.

[6] Kamp H. and U. Reyle. 1993. *From Discourse to Logic*. Kluwer, Dordrecht.

[7] Marcus, M., D. Hindle and M. Fleck. 1983. D-theory: Talking about Talking about Trees. *Proceedings of the 21st ACL*. 129–136.

[8] May, R. 1977. *The Grammar of Quantification*, PhD thesis, MIT, Cambridge.

[9] Muskens, R. 1995. Order-Independence and Underspecification. In J. Groenendijk, editor, *Ellipsis, Underspecification, Events and More in Dynamic Semantics*. DYANA Deliverable R.2.2.C, 1995.

[10] Muskens, R. 1996. Combining Montague Semantics and Discourse Representation. *Linguistics and Philosophy* **19**: 143–186.

[11] Muskens, R. 1998. Underspecified Semantics. Manuscript. Tilburg University.

[12] Muskens, R. and E. Krahmer. 1998. Description Theory, LTAGs, and Underspecified Semantics. *Proc. Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4)*. Institute for Research in Cognitive Science. Philadelphia, PA.

[13] Perrier, G. 1998. Precedence Trees in Proof Nets and Grammatical Formalisms. Talk held at the Workshop on Non-Commutative Linear Logic and Linguistic Applications. Nancy, October 29-30, 1998.

[14] Rambow, O., Vijay-Shanker, K. and D. Weir. 1995. D-Tree Grammars. *Proc. ACL-95*.

[15] Schabes, Y. 1990. *Mathematical and Computational Aspects of Lexicalized Grammars*, Ph.D. thesis, University of Pennsylvania.

[16] Vijay-Shanker, K. 1992. Using Descriptions of Trees in a Tree Adjoining Grammar. *Computational Linguistics* **18**:481–518.