Underspecified Semantics*

Reinhard Muskens

1 Introduction

Ambiguities in natural language can multiply so fast that no person or machine can be expected to process a text of even moderate length by enumerating all possible disambiguations. A sentence containing n scope bearing elements which are freely permutable will have $n!$ readings, if there are no other, say lexical or syntactic, sources of ambiguity. A series of m such sentences would lead to $(n!)^m$ possibilities. Some alternative scopings may boil down to the same reading. The relative order in which we scope two existentially quantified noun phrases, for example, will not matter if no other material intervenes. But all in all the growth of possibilities will be so fast that generating readings first and testing their acceptability afterwards will not be feasible.

This insight has led a series of researchers (e.g. [35, 12, 13, 25, 2, 31, 29, 30, 22, 27, 28]) to adopt a level of representation at which ambiguities remain unresolved. The idea here is not to generate and test many possible interpretations but to first generate one 'underspecified' representation which in a sense represents all its complete specifications and then use whatever information is available to further specify the result. Some mechanism of reasoning, presumably a mechanism with a monotonic core but also admitting of 'jumping to conclusions', must strengthen the original underspecified representation of a given sentence until it is either equivalent to a unique reading for that sentence or until its possible further specifications are within some otherwise acceptable range.

What is the nature of underspecified semantic representations? The program sketched here seems to impose two irreconcilable requirements on them. On the one hand underspecified representations must be ambiguous, on the other hand it should be possible to reason with them. In fact there are two kinds of reasoning that should be possible. The language user must of course be able to reason about the *form* of the representations involved, as understanding form is a necessary precondition for understanding content. But reasoning about the form of a representation may very well have to involve reasoning on the basis of the *content* of some of its partial specifications. Given some representation R

*This paper was written while I was staying at the Universität des Saarlandes (SFB 378–Ressourcenadaptive Kognitive Prozesse). I wish to thank Manfred Pinkal for inviting me to the stimulating environment of Saarbrücken's Computational Linguistics group.

which has to be evaluated in a context C, for example, the hearer may conclude that quantifier Q_1 cannot scope over quantifier Q_2 because the underspecified representation R' which is equivalent to 'R + Q_1 scopes over Q_2' already follows from C or is inconsistent with C. If R' follows from C, or is inconsistent with C, then general Gricean considerations will allow the hearer to conclude that R' was not what was meant and in that case R can be strengthened to an underspecified representation 'R + Q_1 does not scope over Q_2'. The reasoning crucially involves inference on the basis of C and R' and since R' may still contain ambiguities, even if it is more specified than R is, the language user must be able to reason on the basis of ambiguous information.

From this it would seem that we need a logic that can reason with ambiguous expressions[1] and indeed some progress in the direction of defining such logics was made in [32, 33, 37, 5]. In this paper I shall go another way, however, and show that it is possible to use a *standard* logic for underspecified representation provided that we use it for reasoning about the forms of expressions as well as about their contents. Language users typically do not only reason about what is the case and whether one sentence follows from another; they also reason about what was said and which reading was meant. If the previous reflections are correct, both forms of reasoning are intertwined. From a theoretical point of view the simplest hypothesis to adopt is that there is one logical language in which both reasoning processes are carried out. We shall try to model part of this reasoning process with the help of classical logic.

One central hypothesis in the paper will be that the relation between an underspecified representation and its full representations is not so much the relation between one structure and a set of other structures but is in fact the relation between a *description* (a set of logical sentences) and its models.[2] This picture, which comes quite natural to logicians, in fact also underlies the notion of underspecification in syntax, as many researchers have noticed. Section 2 will rehearse some of the reasons why syntacticians have found it attractive to think of syntactic representations in terms of logical descriptions and not directly in terms of the structures satisfying those descriptions. Section 3 will then give a series of axioms which, together with any description of one or more surface structures, will admit the possible *Logical Forms* connected with those structures. It will typically be possible that many Logical Forms correspond to one surface structure and surface structures will be substructures of the corresponding Logical Forms. In sections 4 and 5, then, we shall give more axioms which in effect *interpret* the Logical Forms admitted by our theory and the input surface description. The axioms in section 5 in particular axiomatise

[1] It is of course possible to define a consequence relation on ambiguous expressions in terms of the standard consequence relation and all possible disambiguations of the expressions involved: A follows from B iff any disambiguation of A follows from any disambiguation of B. However, such a definition cannot serve as the basis of a practical inference system in view of the number of disambiguations which may be involved. One criterion a logic of ambiguous expressions has to satisfy is that inference should not require disambiguation.

[2] Since the finite models which we are interested in can be described uniquely, the relation between an underspecified representation and one of its full specifications also corresponds to the relation between descriptions D and D' such that $D \models D'$ and D' has exactly one model.

the usual interpretation procedure in Montague Semantics (see [21]). In our last section we shall show how this set-up allows us to perform inferences in some cases, even where the Logical Forms of some of the premises or the conclusion are not uniquely described.

2 Descriptions in Syntax

In linguistics, as elsewhere, it is often useful to distinguish between *structures* and their *descriptions*.[3] Let us consider the description in (1), a set of atomic sentences.[4] It talks about a certain collection of nodes, referred to with the constants $n_1, \ldots, n_{14}$. The nodes are labeled s, np, $vp, \ldots$ and stand in certain relations, such as *proper dominance* ($\lhd^+$) and *precedence* ($\prec$) and (1) says, among many other things, that the S node n_1 properly dominates the NP node n_2 and the VP node n_3. The constants *every*, *boy*, *adores* etc. refer to the lexical items which label certain nodes.

(1)

$n_1 \lhd^+ n_2$	$n_5 \lhd^+ n_{11}$	$lab(n_1, s)$	$lab(n_{10}, every)$
$n_1 \lhd^+ n_3$	$n_6 \lhd^+ n_{12}$	$lab(n_2, np)$	$lab(n_{11}, boy)$
$n_2 \lhd^+ n_4$	$n_8 \lhd^+ n_{13}$	$lab(n_3, vp)$	$lab(n_{12}, adores)$
$n_2 \lhd^+ n_5$	$n_9 \lhd^+ n_{14}$	$lab(n_4, det)$	$lab(n_{13}, a)$
$n_3 \lhd^+ n_6$	$n_2 \prec n_3$	$lab(n_5, n)$	$lab(n_{14}, girl)$
$n_3 \lhd^+ n_7$	$n_4 \prec n_5$	$lab(n_6, v)$	
$n_7 \lhd^+ n_8$	$n_6 \prec n_7$	$lab(n_7, np)$	
$n_7 \lhd^+ n_9$	$n_8 \prec n_9$	$lab(n_8, det)$	
$n_4 \lhd^+ n_{10}$		$lab(n_9, n)$	

We are only interested in *trees* satisfying the description in (1) and for the sake of definiteness we shall axiomatise collections of trees (or rather *forests*, because we shall not impose the condition that our structures be rooted) by $\mathcal{A}1$–$\mathcal{A}9$ below (see also [9, 3]). Axioms $\mathcal{A}1$ and $\mathcal{A}2$ say that proper dominance is a strict partial order. Axioms $\mathcal{A}3$ and $\mathcal{A}4$ say the same about linear precedence. $\mathcal{A}5$ (*Exhaustiveness*) says that nonidentical nodes must either be in a proper dominance or in a precedence relation. Axioms $\mathcal{A}6$ and $\mathcal{A}7$, whose conjunction is called *Inheritance*, state that nodes inherit the precedence properties of their ancestors in the proper dominance relation. $\mathcal{A}8$ requires the labeling relation to be functional and the last axiom demands that distinct label names denote distinct labels. This excludes the perverse situation that *np* and *vp*, or *det* and *every*, denote the same label.

$\mathcal{A}1$ $\forall k \neg k \lhd^+ k$

$\mathcal{A}2$ $\forall k_1 k_2 k_3 [[k_1 \lhd^+ k_2 \wedge k_2 \lhd^+ k_3] \rightarrow k_1 \lhd^+ k_3]$

[3] This distinction was emphasized with particular clarity in [15, 18, 38, 4].

[4] We use a many-sorted language which distinguishes between the sort of *nodes* and the sort of *labels*. The variable k will be used to range over nodes; ℓ will range over labels. In later sections more sorts will be introduced. Consult Table 1 for an overview of the typographical conventions I have used.

$\mathcal{A}3$ $\forall k \neg k \prec k$

$\mathcal{A}4$ $\forall k_1 k_2 k_3 [[k_1 \prec k_2 \wedge k_2 \prec k_3] \rightarrow k_1 \prec k_3]$

$\mathcal{A}5$ $\forall k_1 k_2 [k_1 \prec k_2 \vee k_2 \prec k_1 \vee k_1 \lhd^+ k_2 \vee k_2 \lhd^+ k_1 \vee k_1 = k_2]$

$\mathcal{A}6$ $\forall k_1 k_2 k_3 [[k_1 \lhd^+ k_2 \wedge k_1 \prec k_3] \rightarrow k_2 \prec k_3]$

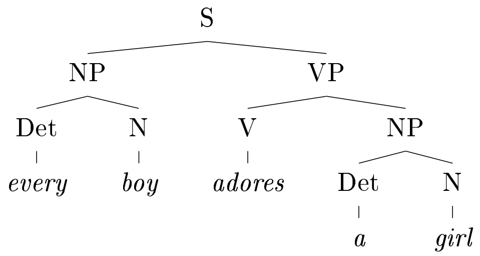
$\mathcal{A}7$ $\forall k_1 k_2 k_3 [[k_1 \lhd^+ k_2 \wedge k_3 \prec k_1] \rightarrow k_3 \prec k_2]$

$\mathcal{A}8$ $\forall k \forall \ell_1 \ell_2 [[lab(k, \ell_1) \wedge lab(k, \ell_2)] \rightarrow \ell_1 = \ell_2]$

$\mathcal{A}9$ $l_1 \neq l_2$, if l_1 and l_2 are distinct label names.

Other axioms may be natural too, such as the requirement that nodes labeled with a lexical element are terminal (have no successor in the proper dominance relation), but we shall refrain from formalising such extra conditions here. Two abbreviations will prove useful. We use $\lhd^*$ for the *dominance* relation, i.e. we write $n \lhd^* n'$ for $n \lhd^+ n' \vee n = n'$. The relation of *immediate dominance*, $\lhd$, is obtained by writing $n \lhd n'$ as an abbreviation for $n \lhd^+ n' \wedge \forall k \neg [n \lhd^+ k \wedge k \lhd^+ n']$. Any forest which satisfies the extra condition of *Rootedness*, $\exists k_1 \forall k_2 \, k_1 \lhd^* k_2$, is called a *tree*.

(2)



If we are to write down a model for the description in (1) plus $\mathcal{A}1$–$\mathcal{A}9$, we soon come up with the structure in (2), which satisfies all requirements. It is clear, of course, that there are many trees and forests besides (2) which satisfy (1), but (2) has a special position among these, as it is the forest with fewest nodes satisfying the description. In fact, it is the unique forest satisfying (1) plus (3), the statement that $n_1, \ldots, n_{14}$ are all nodes available.

(3) $\forall k \, (k = n_1 \vee \ldots \vee k = n_{14})$

Descriptions such as the one in (1) are not very user-friendly, as we cannot see at one glance what the domination and precedence relations are. Fortunately, there is an alternative way of interpreting (2), not as the graphical representation of a certain model, but as a graphically appealing way to write down the essence of (1). In the next section, we shall make use of this possibility to interpret graphical representations either way, as descriptions, or as the kind of structures that can satisfy them.

Type	Variables	Constants
$node$	k	n
$label$	ℓ	$s, np, vp, \ldots, e, every, boy, \ldots$
π	v	w, h, TIM, TOM, MARY,...
e	x, y, z	TIM, TOM, MARY,...
s	i, j	
$node \to t$		$surface$, $quant$, $island$, $root$
$node \times node \to t$		$link$, $\lhd^+$, $\lhd$, $\lhd^*$, $\prec$, par
$node \times label \to t$		lab
$\pi \to t$		VAR
$\pi \times s \to e$		V
$node \to \pi$		ρ
$s \to t$	p	
$\pi \to (s \to t)$	P	
$e \times e \to t$		RUN, MAN
$e \times e \times e \to t$		FIND, B
$node \times \alpha \to t$		σ^α, σ_0^α

Table 1: An overview of variables and constants that will be used

There are many reasons to think about linguistic representations in terms of descriptions rather than in terms of structures. Some of these reasons have to do with the processing of language, in particular the parsing process, other reasons have to do with syntactic ambiguities, others again concern syntactic theory itself, the task to characterise which structures are possible. Let us turn to the reasons that come from parsing theory first. A locus classicus here is [18], where (4) is considered and where it is noticed that a straightforward left-to-right deterministic parse of this sentence is impossible if structures are built in a conventional way. A left-to-right parser generating phrase structure trees will necessarily hypothesise at some point that *drove my aunt* is a VP constituent. This means that backtracking will be necessary.

(4) I drove my aunt from Peoria's car

The solution proposed by [18] is to let the parser create descriptions, not structures, and to let these descriptions be stated in terms of the relation of dominance, not immediate dominance.[5] After having read *I drove my aunt* the 'Description Theory' parser discussed in [18] will have created a partial description containing, among other material, the statements in (5).

(5)
$$\begin{array}{lll} n_1 \lhd^* n_2 & n_3 \lhd^* n_5 & lab(n_3, np) \\ n_1 \lhd^* n_3 & lab(n_1, vp) & lab(n_4, det) \\ n_3 \lhd^* n_4 & lab(n_2, v) & lab(n_5, n) \end{array}$$

[5]The relation of proper dominance, which we are taking as primitive, would work as well.

A conventional parser at this point would have created a structure which would not be embeddable in the desired final result, but in the Description Theory approach it is possible to continue the analysis without destroying the intermediate description. Reading the rest of the sentence, the parser will create more statements, such as the ones in (6) for example. These say that there is material between the VP and NP mentioned in (5), but the original description is retained. The new statements are consistent with the previous ones and can simply be added. Unlike in structural approaches, where part of the structure that has been built up will need to be destroyed at this point, no earlier analysis need be revoked.

(6) $n_1 \lhd^* n_6$ $n_7 \lhd^* n_3$ $lab(n_7, det)$
$n_6 \lhd^* n_7$ $lab(n_6, np)$

While this example shows how representation by means of descriptions helps to avoid backtracking during the parsing process, there are also examples which show how descriptions can provide us with a compact and efficient way of representing syntactic ambiguity. [18] and [17] consider the sentence in (7a), which can either be analysed as (7b) or as (7c).[6] Parsing (7), one may well want to leave this ambiguity unresolved, perhaps until extra information which resolves it becomes available. One certainly does not want to sum up all attachment possibilities, as the number of these in general is not polynomial in the size of the input so that any kind of generate-and-test procedure would make parsing practically impossible.

(7) a. They sell green apples, pears and bananas from Erie
b. They sell [$_{NP}$ green [$_N$ apples], [$_N$ pears], [$_N$ and bananas] from Erie]
c. They sell [$_{NP}$ green apples], [$_{NP}$ pears], [$_{NP}$ and bananas from Erie]

The Description Theory parser does not generate either of the structures (7b) or (7c). Instead it provides us with a description which is true of both. The statements in (8) are part of this description. Note that the constants n_3, n_4 and n_5 are interpreted as the *same* NP node in (7b) while they denote *different* NP nodes in (7c).

(8)
$n_1 \lhd^* n_2$	$n_5 \lhd^* n_9$	$lab(n_7, n)$
$n_1 \lhd^* n_3$	$n_5 \lhd^* n_{10}$	$lab(n_4, np)$
$n_1 \lhd^* n_4$	$lab(n_1, vp)$	$lab(n_8, n)$
$n_1 \lhd^* n_5$	$lab(n_2, v)$	$lab(n_5, np)$
$n_3 \lhd^* n_6$	$lab(n_3, np)$	$lab(n_9, n)$
$n_3 \lhd^* n_7$	$lab(n_6, ap)$	$lab(n_{10}, pp)$
$n_4 \lhd^* n_8$		

The Description Theory approach to syntactic ambiguity is based on the fact that a description can have any number of models. Even when only models

[6] [18] and [17] follow the set-up of [7] in this analysis and allow coordinated constituents to be 'stacked'.

which are are minimal in some sense are considered[7] no unique structure need result. The idea allows one to give a single representation for a multitude of syntactic readings.

The reasons for replacing structures with descriptions of structures that were adduced so far had to do with linguistic processing, not with the task of characterising all and only those structures which are linguistically possible. But here the method has advantages too. Sometimes a description is capable of capturing a generalisation that is missed otherwise. Consider the sentences in (9), which were taken from [39].

(9) a. I opened up Michelle a new bank account

b. I opened Michelle up a new bank account

c. I opened Michelle a new bank account up

We could analyse these sentences by enumerating their three possible syntactic structures, as in (10) (we show only the relevant VP and her daughters; PL is the verb particle).

(10)

VP	VP	VP
V PL NP NP	V NP PL NP	V NP NP PL

But by taking the trees in (10) as the relevant representations of the sentences in (9), i.e. by just summing up the various possibilities, we miss the generalisation that the grammar in fact prescribes no ordering among the daughter nodes here, except that the lexicalised head (the V) must precede her sisters. Giving a description which underspecifies the ordering, such as the one in (11), is a better option.

(11)
$$
\begin{array}{lll}
n_1 \lhd^+ n_2 & lab(n_1, vp) & n_2 \prec n_3 \\
n_1 \lhd^+ n_3 & lab(n_2, v) & n_2 \prec n_4 \\
n_1 \lhd^+ n_4 & lab(n_3, np) & n_2 \prec n_5 \\
n_1 \lhd^+ n_5 & lab(n_4, np) & n_3 \neq n_4 \\
 & lab(n_5, pl) &
\end{array}
$$

Here we have a single representation characterising the three relevant possibilities: The structures in (10) are the minimal trees satisfying the description in (11).

An even better solution would result if the three linear precedence statements in (11) would be derivable as consequences of a global (language-specific) rule, say *"lexicalised heads precede their complements"*.[8] Once such rules can be expressed by the description language we are within the ID / LP (immediate dominance / linear precedence) format of Generalised Phrase Structure Grammar (GPSG). For an explanation of this format and for more motivation see

[7] As we did when we added (3) as an extra condition to (1), or as it is done in Description Theory when 'standard referents' are taken.

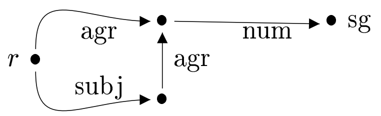
[8] Depending on one's theory of features, the inequality $n_3 \neq n_4$ could probably follow from incompatible feature assignments to n_3 and n_4 in a full-fledged approach.

[11]. In the present context the main point is that ID / LP rules are an example of the use of underspecification to obtain greater generality and that the formulation of these rules essentially rests upon a distinction between structures and the descriptions satisfied by those structures.

Trees are by no means the only syntactic structures which can, and should, be underspecified. *Feature structures* are another example. Consider the attribute value matrix in (12). One possible perspective on attribute value matrices is that they are a convenient way to denote *labeled graphs.* The graph corresponding to (12) is given in (13).

(12) $$\begin{bmatrix} \textsc{agr} & \boxed{1}\begin{bmatrix}\textsc{num} & \textsc{sg}\end{bmatrix} \\ \textsc{subj} & \begin{bmatrix}\textsc{agr} & \boxed{1}\end{bmatrix} \end{bmatrix}$$

(13)



In (14) below we use the first-order language of [14] to write down a description of graph (13). Johnson's language essentially consists of a three-place relation symbol *arc*—with $arc(f_1, a, f_2)$ saying that f_1 and f_2 are connected by an arc labeled a—plus a finite number of individual constants for attributes (AGR, SUBJ, NUM, PERS, ...) and values (SG, PL, 1, 2, 3, +, −, ...). We shall assume that there is also a constant r denoting the *root* of the graph.

(14) $\exists f f'(arc(r,\textsc{agr}, f) \wedge arc(r,\textsc{subj}, f') \wedge arc(f',\textsc{agr}, f) \wedge arc(f,\textsc{num},\textsc{sg}))$

Graph (13) and sentence (14) exemplify a duality between finite labeled graphs that are models for Johnson's language and the sentences of this language which are formed from atomic formulas with $\wedge$ and $\exists$ only (call these $\{\wedge, \exists\}$-sentences). Given any finite graph G which is a model for the language, first name all G's unnamed nodes with distinct variables and then take the existential closure of the conjunction of all atomic formulas which are true in G and contain only variables that were used for naming nodes in G. Call this description $\mathcal{D}(G)$. Conversely, given any $\{\wedge, \exists\}$-sentence D, strip off its existential quantifiers to get a conjunction of atomic formulas D' and consider the graph with has as its domain all constants of the language and all free variables in D' and interprets *arc* as just those triples which D' says are in the *arc* relation. Call this graph $\mathcal{G}(D)$. It is clear that $\mathcal{G}(\mathcal{D}(G))$ is isomorphic to G and that $\mathcal{D}(\mathcal{G}(D))$ is logically equivalent with D. Moreover, for any D and G, $G \models D$ if and only if $\mathcal{G}(D) \sqsubseteq G$, where $\sqsubseteq$ is the *subsumption* relation.[9] From this it follows by elementary

[9]A model M' *subsumes* a model M, $M' \sqsubseteq M$, if and only if M' is homomorphic to a submodel of M, i.e. if and only if there is a structure-preserving mapping (a homomorphism) from the domain of M' to the domain of M. For a precise definition of the *homomorphism* relation, see [6], for examples of the subsumption relation in linguistics and a discussion of its importance in unification theory see [36]. The present statement is true in view of Proposition 2.1.12 of [6].

reasoning that $G \sqsubseteq G'$ if and only if $\mathcal{D}(G') \models \mathcal{D}(G)$, for all G and G'.[10]

This nice duality between feature graphs and certain first-order descriptions with a very limited syntax allows underspecification of one structure by means of another. The traditional interpretation of feature structures is that they may give only partial information about a linguistic object. Supposing that a feature structure G models *all* relevant information about some linguistic expression E, we may approximate this information by using feature structures G' such that $G' \sqsubseteq G$. Such feature structures can be combined by means of unification and if $G'' \sqsubseteq G' \sqsubseteq G$ then G' is as least as good an approximation to the information about E as G'' is. The reason that this works is that the feature structures G' approximating G are in a one-to-one relationship with certain descriptions $\mathcal{D}(G')$ of G, that a description of a finite structure in a sense is an approximation of that structure and that the inverse of entailment can function as the relation 'is no worse approximation than'.

But the price of such an underspecification of structures by structures is in fact a limitation to the $\{\wedge, \exists\}$-fragment of the language and there is evidence that this limited expressivity is not adequate for saying the things we would like to be able to say in linguistics. Consider uninflected English verbs such as *work* or *kiss*, for example. Such verbs can either occur in infinitival form (as in *John wanted to kiss Mary*), or can be finite (as in *I work in the factory there*), but in the latter case they may not be third person singular (**John work in the factory there*). Within the framework of Lexical-Functional Grammar (LFG, see [15]) [19] propose to capture these facts by letting the functional structure f of uninflected verbs be constrained by the description in (15). A translation of (15) into Johnson's perspicuous first-order feature language is given in (16).

(15) $((f \text{ INF}) = - \wedge (f \text{ TENSE }) = \text{PRES} \wedge$
$\neg[(f \text{ SUBJ NUM}) = \text{SG} \wedge (f \text{ SUBJ PERS}) = 3]) \vee (f \text{ INF}) = +$

(16) $(arc(f,\text{INF}, -) \wedge arc(f,\text{TENSE},\text{PRES}) \wedge$
$\neg\exists g(arc(f,\text{SUBJ}, g) \wedge arc(g,\text{NUM},\text{SG}) \wedge arc(g,\text{PERS}, 3))) \vee arc(f,\text{INF}, +)$

This description clearly does not fall within the $\{\wedge, \exists\}$-fragment of the language. In (17) we give three structures in which (16) is true. The first cannot be unified with either of the other two and although the second graph subsumes the third it also subsumes structures which do not satisfy (16). It is therefore evident that there is no single structure G such that, for any G', $G' \models (16)$ if and only if $G \sqsubseteq G'$, as was the case before. The trick to let one graph represent all graphs that are subsumed by it breaks down.

[10] In [14] Johnson considers four axiom schemes, which require, among other things, the *arc* relation to be functional. If these axioms are used to restrict the graphs under consideration the duality between graphs and $\{\wedge, \exists\}$-sentences turns into a duality between graphs and those $\{\wedge, \exists\}$-sentences which are consistent with the axioms. Moreover, we have that, for any two graphs G_1 and G_2, if $\mathcal{D}(G_1) \wedge \mathcal{D}(G_2)$ is consistent with Johnson's axioms, then $\mathcal{G}(\mathcal{D}(G_1) \wedge \mathcal{D}(G_2))$ is just the *unification* of G_1 and G_2.

(17)

$$\begin{bmatrix}\text{INF} & +\end{bmatrix} \qquad \begin{bmatrix}\text{INF} & - \\ \text{TENSE} & \text{PRES}\end{bmatrix} \qquad \begin{bmatrix}\text{INF} & - \\ \text{TENSE} & \text{PRES} \\ \text{SUBJ} & \begin{bmatrix}\text{NUM} & \text{SG} \\ \text{PERS} & 2\end{bmatrix}\end{bmatrix}$$

This means that using feature graphs as a vehicle for linguistic representation does not give us enough expressivity but that we can get the extra expressivity by turning to feature descriptions instead of feature graphs.[11] Anything we can do with the graphs we can do with the descriptions, but we can do more, as the graphs only correspond to a very simple part of the description language.

3 Constraining Logical Form

In the previous section we have seen examples of the use of descriptions in syntax and the possibility of underspecification that results from this use. The question now poses itself whether the method can also be used in semantics. There is a tradition within constraint-based linguistics which uses constraints to *describe logical sentences*. [10], for example, propose to represent the formula $\forall x\, kick(j, x)$ as the attribute value matrix in (18). I.e. the formula is represented as a certain graph, which can then be underspecified, a possibility which is used in [25] and also lies at the heart of the 'Minimal Recursion Semantics' of [8].

(18)

$$\begin{bmatrix}\text{FORMULA} & \begin{bmatrix}\text{QP} & \begin{bmatrix}\text{QUANT} & \forall \\ \text{VAR} & x\end{bmatrix} \\ \text{FORMULA} & \begin{bmatrix}\text{REL} & kick \\ \text{ARG.1} & j \\ \text{ARG.2} & x\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Although some interesting results have been obtained with the help of such descriptions of logical formulas, I think that, apart from the unwieldiness of (18) as compared with the simple representation $\forall x\, kick(j, x)$, there are at least two reasons to find the procedure unsatisfactory. The first objection is that linguistic representations should be used to characterise linguistic objects and

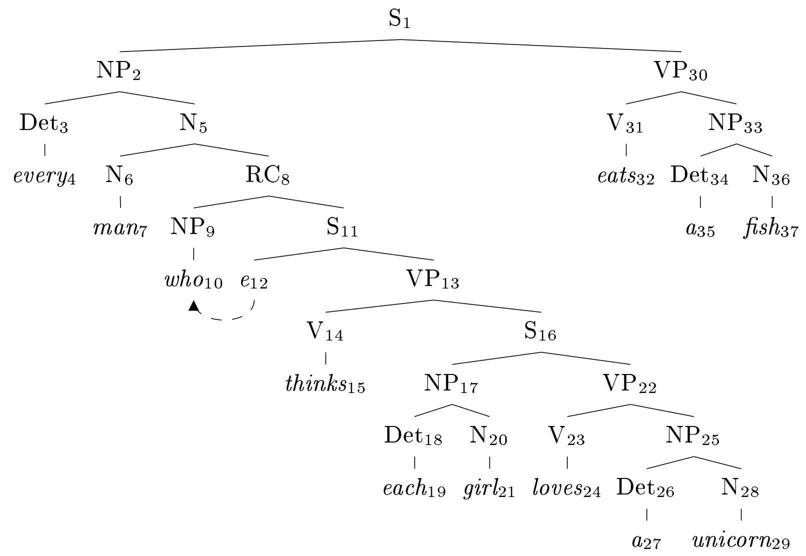
[11] From a computational point of view there are good reasons to restrict the expressivity that is gained in this way. [14] proposes to restrict descriptions to those first-order sentences which are equivalent to a sentence of the form $\exists x_1 \ldots x_n \forall y_1 \ldots y_m \varphi$, where φ does not contain function symbols. (16) is of this form. This fragment has nice computational properties and may function as an alternative to the feature logic of [16]. Blackburn (see [4]) limits expressivity by proposing to use a modal logic for describing linguistic structures. Since most linguistic structures are graphs and since modal logics are tailored to reason about graphs (in the guise of Kripke models), this proposal is very natural. The LFG description in (15) could be rendered as

$$(\langle\text{INF}\rangle - \wedge \langle\text{TENSE}\rangle\text{PRES} \wedge \neg\langle\text{SUBJ}\rangle(\langle\text{NUM}\rangle\text{SG} \wedge \langle\text{PERS}\rangle 3)) \vee \langle\text{INF}\rangle +$$

in Blackburn's poly-modal language. Here INF etc. stand for certain accessibility relations and $-$, PRES, etc. are propositional constants. Blackburn also proposes to reinterpret attribute-value matrices as expressions in this poly-modal language, i.e. as descriptions, not as structures.

(19) Every man who thinks each girl loves a unicorn eats a fish

(20)



that logical formulas do not qualify as such. It is one thing to assume that a logical formula is adequate for describing the meaning of a natural language expression but quite another to assume that all aspects of the particular form of such a formula are linguistically relevant. Representations like the one in (18) bring historically contingent aspects of our way to denote logical formulas into linguistic theory. The second objection is that if we represent formulas with the help of graphs and then describe graphs with the help of formulas we have something that looks very much like an epicycle. It would be much simpler to use formulas directly for describing semantic values.

In this paper I want to show that this can be done if a two-stage procedure is followed. The first stage consists of describing all *Logical Forms* which are connected with a certain surface description and the second stage will be an interpretation of those Logical Forms. The structures that I have in mind are close to, but not isomorphic with, the Logical Forms which are to be found in Chomsky's theory (see [20]). The main difference between our structures and the more standard ones arises from the fact that in our set-up surface structures are always submodels of the Logical Forms connected with them. Given any Logical Form, we can always arrive at the surface structure it is connected with by blotting out material.

In order to explain how Logical Forms are characterised by surface descriptions we need an example of the latter. Consider the multiply ambiguous sentence (19). Parsing this sentence may give (20), interpreted as a description. As was mentioned in the previous section there are two ways to interpret graphical representations: they can be used to denote models in a user-friendly way, or they can be used as an equally user-friendly way to write down descriptions. We shall interpret (20) as a description, with every subscript representing a constant (e.g. the subscript 9 in NP_9 refers to n_9), every arc representing a proper domination statement (e.g. we have that $n_{11} \lhd^+ n_{13}$), every left-right ordering of sisters corresponding to a precedence statement (e.g. $n_{31} \prec n_{33}$), and every category label, trace or lexical element representing a lab statement (e.g. $lab(n_{22}, vp)$, $lab(n_{19}, each)$, $lab(n_{12}, e)$). The dashed arrow between the trace and the element *who*, which represents coindexing, is formalised as $link(n_{12}, n_{10})$. In this way (20) is really just a convenient representation of a description which would otherwise be unwieldy and unreadable.

(21) $\forall k\,(surface(k) \leftrightarrow (k = n_1 \vee \ldots \vee k = n_{37}))$

In analogy with (3) we circumscribe the domain of surface nodes in (21).[12] This time, however, we are not assuming that $n_1, \ldots, n_{37}$ are all nodes whatsoever; we only stipulate that these are exactly all *surface* nodes. A limited set of extra nodes will be allowed and different possibilities of relating these to the surface tree will result in different Logical Forms. Although only one tree will be possible on the 37 nodes for which the predicate $surface$ holds,[13] these extra nodes and the various possible ways of placing them lead to many more trees. In order to restrict the number of extra nodes and to restrict the possibilities for building trees, we impose some more axioms.

$\mathcal{A}10$ $\forall k_1 k_2 k_3\,[[link(k_1, k_2) \wedge link(k_1, k_3)] \rightarrow k_2 = k_3]$

$\mathcal{A}11$ $\forall k_1\,[surface(k_1) \vee \exists k_2\,[surface(k_2) \wedge link(k_2, k_1)]]$

$\mathcal{A}12$ $\forall k_1\,[quant(k_1) \leftrightarrow [lab(k_1, np) \wedge \exists k_2[k_1 \lhd k_2 \wedge lab(k_2, det)]]]$

$\mathcal{A}13$ $\forall k_1 k_2\,[link(k_1, k_2) \rightarrow [surface(k_1) \wedge [quant(k_1) \vee lab(k_1, e)]]]$

$\mathcal{A}14$ $\forall k_1\,[quant(k_1) \rightarrow \exists k_2\,[link(k_1, k_2) \wedge lab(k_2, s) \wedge \neg surface(k_2) \wedge$
$\forall k_3\,[link(k_3, k_2) \rightarrow k_3 = k_1] \wedge \forall k_3[[island(k_3) \wedge k_3 \lhd^+ k_1] \rightarrow k_3 \lhd^* k_2] \wedge$
$\exists k_3[lab(k_3, s) \wedge k_2 \lhd k_3 \lhd^+ k_1 \wedge \forall k_4[k_2 \lhd k_4 \rightarrow k_3 = k_4]]]]$

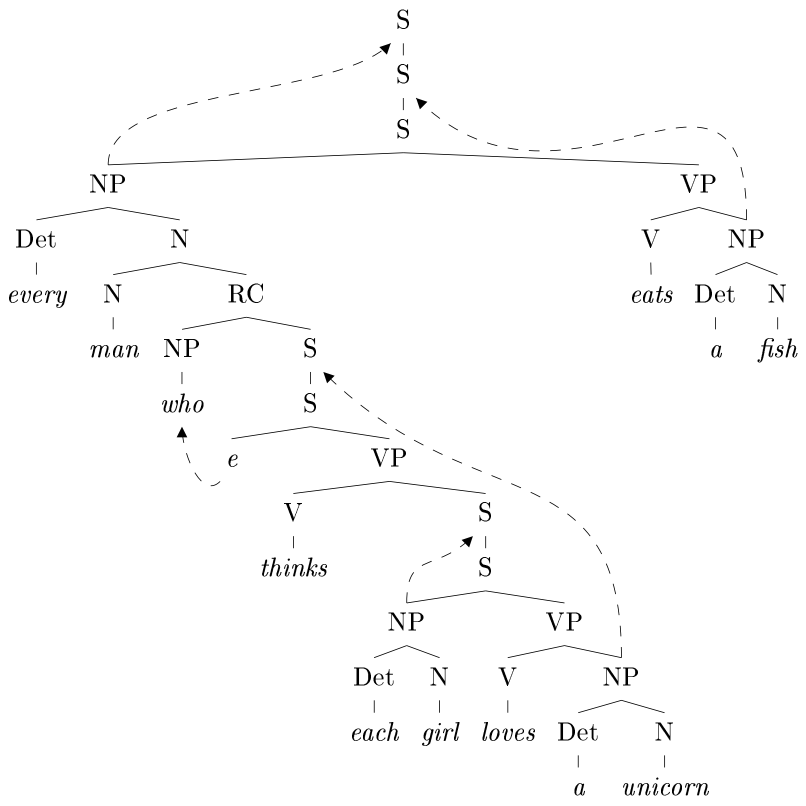
$\mathcal{A}15$ $\forall k_1\,[island(k_1) \leftrightarrow [lab(k, np) \vee \exists k_2\,[k_1 \lhd k_2 \wedge lab(k_2, conj)]]]$

Axiom $\mathcal{A}10$ says that the linking relation is functional and $\mathcal{A}11$ states that each non-surface node is an image of a surface node under the linking relation. This already severely restricts the set of non-surface nodes. Axiom $\mathcal{A}12$ defines quantified NPs to be NPs which immediately dominate a determiner and $\mathcal{A}13$ postulates that only surface nodes which are quantified NPs or traces can be in the domain of the linking relation. Axiom $\mathcal{A}14$ is our equivalent of Quantifier Raising (see [20]). It says that each quantified NP is linked to a non-surface

[12] This extra statement can be thought of as an additional result of the parsing process.

[13] That the description (20) + (21) admits of only one surface tree is an inessential feature of our example. We may allow syntactic underspecification and use descriptions which are true on more than one surface tree.

(22)



S node k_2; that no other node is linked to k_2; that any scope island properly dominating the NP also dominates k_2; that k_2 immediately dominates an S node which properly dominates the NP and that there are no other nodes which it dominates immediately. As the reader will have guessed, the extra S node which is created by each quantified NP will correspond to the place where the NP is quantified-in.[14] This leaves us with the question exactly which nodes are scope islands. Since there is some evidence that Quantifier Raising obeys Ross's ([34]) Complex NP Constraint (CNPC) and Coordinate Structure Constraint (CSC),[15] we implement these in $\mathcal{A}15$.

---

[14] The extra S node may be compared to the extra S node that is created as a result of Chomsky-adjoining NP to S (Quantifier Raising) in Generative Grammar.

[15] Unfortunately the theory that Quantifier Raising obeys the CNPC also meets with counterexamples. E.g. [26] gives (1) below, whose only reading requires *every minister* to be raised out of the subject NP, while [1] offers (2), which has as a natural reading that, for each race, the dogs that won it were hungry. In view of such counterexamples we like to interpret our axiom $\mathcal{A}15$ merely as an example of how, given a theory of scope islands, such a theory could be formalised within our set-up, not as a definite proposal regarding restrictions on semantic scope.

1. The slush fund that every minister needs is kept by his private secretary

Let us consider what models (20) + (21) admits, given our axioms. It is not difficult to see that, apart from the original 37 surface nodes, only four extra nodes are allowed. Each quantified NP will license an extra node, which will be labeled S and must be placed just above another S node. The reader will have no difficulty in recognising that there are in total twelve possibilities of positioning these four new nodes in the tree. This means that the theory which consists of our axioms $\mathcal{A}1$–$\mathcal{A}15$ plus the input description (20) + (21) has twelve models. Each of these twelve models has the same surface tree as a submodel.

The models of our surface descriptions plus our axioms will be our Logical Forms. In (22) we have depicted one Logical Form for (20) + (21), the one where the NP *a unicorn* scopes over *thinks* but *each girl* does not, and where the subject scopes over the object. The NPs remain in situ syntactically, but the linking arrows tell where quantifying-in takes place semantically. The arrows can also be interpreted as telling what movements take place in more standard theories of Logical Form, but since their existence already provides enough information for semantic interpretation, we prefer not to let these movements actually be carried out.

That surface descriptions underspecify Logical Form means that language users must sometimes use other information in order to narrow down the range of possibilities. If you need to know exactly what was said, or if you want a better approximation to what was said than what could be inferred from surface information alone, you need to draw in other sources. Intonational information readily comes to mind, semantic and contextual information are obviously very important, and there may also be all kinds of default assumptions, rules of thumb like '*each* likes wide scope', 'NPs are scoped in surface order if no other information makes that unlikely' etc.

We assume that this extra information—information which is not derived from the parsing process proper but from other linguistic sources—comes in exactly the same form as the information derived from parsing, i.e. as a series of additional statements in the same classical language as the one we have been using thus far. We suggest, moreover, that the way such information is derived from those additional sources is essentially classical deduction, possibly extended with some mechanism for default reasoning. Suppose, for example, that a hearer employs contextual reasoning to derive that the subject of (19) scopes over its object. This can be expressed as $\exists k_1 k_2\, [link(n_2, k_1) \wedge link(n_{33}, k_2) \wedge k_1 \lhd^+ k_2]$. Adding this statement to the surface description (20) + (21) would not remove all ambiguity, but it would remove some, as the number of readings would decrease from twelve to six.

4 Internalising Variable Binding

In the next section we shall provide our descriptions with interpretations, but before we can do so we must remove a technical difficulty. Consider (20) and

2. The dogs that won each race were hungry

the Logical Forms that it admits. If we want to interpret this description and proceed in a Montagovian way, many of its node names can be provided with a meaning without any problems. The constant n_{17}, for example, can be associated with a certain term by finding translations of *each* and *girl* in the lexicon and then using function application to form, say, *each'(girl')*. But not all nodes can be treated in this way. Consider the (nonsurface) S node n that will be linked to n_{17} in every Logical Form connected with (20) in view of $\mathcal{A}14$. It is reasonable to assume that the semantic value associated with n is the result of a quantification: *each'(girl')* must be applied to the result of $\lambda x.A$, where A is the translation of some node n' immediately dominated by n. There will be a free x in A, which is the *direct* contribution of n_{17} to the semantic value of n'. But n will dominate different nodes n' in different Logical Forms and hence A will have different values in those different structures. This means that the value of n' cannot enter a description of the value of n directly but that it must be left indeterminate. The usual technique for letting things remain indeterminate is to represent them as variables, but this cannot be done here. Suppose we would describe the value of the node in question as *each'(girl')*$(\lambda x.Y)$, depending on another description to instantiate the value of Y as A, then the abstraction λx would be vacuous and substitution of A for Y would not be possible.

One way to circumvent such difficulties would be the introduction of *metavariables*, ranging over formulas, as it is done in [27, 28]. If Y were such a metavariable we could instantiate it as A no doubt, but the move would take us away from classical logic. Instead of making such a step we describe a simple technique which will allow us to interpret trees by means of *closed* terms only. Since there are no conditions on substitution of equals for equals as far as closed terms are concerned, the difficulty described above will vanish.

The idea is to give, at the object level of the language, a simple axiomatisation of the usual machinery that is connected with the binding of variables. When this is done we no longer need to depend on the standard binding mechanism for describing quantification in natural language, but may replace it with the internalised version. Since I have described the internalisation of binding in some detail elsewhere (see e.g. [24]), I shall be relatively brief about it here.

In order to mimick the binding machinery we need objects that stand proxy for variables as well as objects that stand for assignments. Things of the first kind will be called *registers* and those of the second will be called *states*. States and registers are primitive objects in our models and are of types s and π respectively. The letters i, j will typically be used to range over states and v will be used as a variable over registers. We shall also have constants $w, w_1, w_2, \ldots, h, h_1, h_2, \ldots$,TOM, MARY,... for registers, and a function ρ of type *node* $\rightarrow \pi$ will assign a register to each node.[16] Closed terms of type π will be called *referents*.

We can think of our registers as the registers in a computer, which may contain a value in any given state. These values will be objects of type e in our

[16]The reason why we assign registers to nodes is that this provides us with a mechanism which will give us a 'fresh' register for every quantified noun phrase. The next section will make this clearer.

set-up, *entities*, for which we use variables x, y, z. A function V of type $\pi \times s \to e$ will assign a value to each register in each state. To make life simpler we do not make a type distinction between entities proper and *worlds*. In particular we think of the value of register w in any state i as being the current world of evaluation.

If i and j are terms of type s and δ is a term of type π we may write $i[\delta]j$ for $\forall v\,[v \neq \delta \to V(v,i) = V(v,j)]$, which expresses that i and j can differ in value only in register δ. The following axioms[17] embody our internalisation of binding. $\mathcal{A}16$ requires that there be enough states to allow selective updating of 'varying' registers (those registers that correspond to variables, not to constants); $\mathcal{A}17$ and $\mathcal{A}18$ say that all w's and h's and also the values of ρ are 'varying' and hence updatable; $\mathcal{A}19$ and $\mathcal{A}20$ stipulate that all these registers are pairwise distinct and that ρ is an injection; and $\mathcal{A}21$ makes the obvious connection between 'constant' registers and type e constants. Referents denoting such 'constant' registers will be called *specific* referents. We distinguish them from type e constants by using initial capitals for the former, just small caps for the latter.

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

$\mathcal{A}17$ $VAR(u)$, where $u \in \{w, w_1, w_2, \ldots, h, h_1, h_2, \ldots\}$

$\mathcal{A}18$ $\forall k\,VAR(\rho(k))$

$\mathcal{A}19$ $u \neq u' \wedge \forall k\,u \neq \rho(k)$, where u and u' are syntactically different constants $\in \{w, w_1, w_2, \ldots, h, h_1, h_2, \ldots\}$

$\mathcal{A}20$ $\forall k_1 k_2\,[\rho(k_1) = \rho(k_2) \to k_1 = k_2]$

$\mathcal{A}21$ $\forall i.V(\text{Tom}, i) = \textsc{tom}$,
$\forall i.V(\text{Mary}, i) = \textsc{mary}$,
$\forall i.V(\text{Tim}, i) = \textsc{tim}$, etc., for all names.

These axioms will allow us to embed predicate logic[18] into (the first-order part of) type theory in a very special way: all binding will take place by means of registers. We write

$$
\begin{array}{rcl}
R\{\delta_1, \ldots, \delta_n\} & \text{for} & \lambda i.R(V(\delta_1, i), \ldots, V(\delta_n, i)) \\
\delta_1 \textbf{ is } \delta_2 & \text{for} & \lambda i.V(\delta_1, i) = V(\delta_2, i) \\
\textbf{not}\ \gamma & \text{for} & \lambda i\,\neg\gamma(i) \\
\gamma \Rightarrow \gamma' & \text{for} & \lambda i\,[\gamma(i) \to \gamma'(i)] \\
\textbf{all}(\delta, \gamma) & \text{for} & \lambda i \forall j\,[i[\delta]j \to \gamma(j)]
\end{array}
$$

[17]These axioms are slightly different from the ones in [24], the main difference resulting from our introduction of the function ρ here.

[18]The technique sketched here is by no means confined to predicate logic. In [24] the very same trick is used to embed the core part of Discourse Representation Theory into type logic and it is remarked there that in fact any logic with a decent semantics can be treated in this way.

In fact we have transcribed the usual Tarski truth definition in our logic here, using states in lieu of assignments and referents in lieu of variables. Some more abbreviations, in terms of the previous ones, will turn out handy. Write

$$
\begin{array}{rll}
\gamma \mathbin{\&} \gamma' & \text{for} & \mathbf{not}[\gamma \Rightarrow \mathbf{not}\ \gamma'] \\
\gamma \Leftrightarrow \gamma' & \text{for} & [\gamma \Rightarrow \gamma'] \mathbin{\&} [\gamma' \Rightarrow \gamma] \\
\gamma\ \mathbf{or}\ \gamma' & \text{for} & \mathbf{not}\ \gamma \Rightarrow \gamma' \\
\mathbf{some}(\delta, \gamma) & \text{for} & \mathbf{not}\ \mathbf{all}(\delta, \mathbf{not}\ \gamma) \\
\mathbf{no}(\delta, \gamma) & \text{for} & \mathbf{all}(\delta, \mathbf{not}\ \gamma)
\end{array}
$$

We need to show that the definitions here really mimick the usual ones for first-order logic in the sense that they really express what they are obviously intended to express. To this end, let us consider the fragment of our logic which is generated with the help of these definitions. More precisely, let $n_1, n_2, \ldots, n_m, \ldots$ be some countable enumeration of constants of type *node* and, taking referents u from the set $\mathcal{U} = \{w, w_1, \ldots, h, h_1, \ldots, \rho(n_1), \rho(n_2), \ldots\}$, and referents δ from $\Delta = \mathcal{U} \cup \{\text{Tim}, \text{Tom}, \text{Mary}, \ldots\}$, consider the set of $s \to t$ terms which is generated by the following Backus-Naur Form.

$$
\gamma \quad ::= \quad R\{\delta_1, \ldots, \delta_n\} \ \mid\ \delta_1\ \mathbf{is}\ \delta_2 \ \mid\ \mathbf{not}\ \gamma \ \mid\ \gamma \Rightarrow \gamma' \ \mid\ \mathbf{all}(u, \gamma)
$$

This fragment can be translated into the more usual notation for predicate logic in the following obvious way. Let $x_1, \ldots, x_m, \ldots$ and $y, y_1, \ldots, y_m, \ldots$ and $z, z_1, \ldots, z_m, \ldots$ be enumerations of type e variables (all variables pairwise distinct). Define $w^\dagger = y$, $h^\dagger = z$ and $\rho(n_m)^\dagger = x_m$, $w_m^\dagger = y_m$, $h_m^\dagger = z_m$, for all m, while $\text{Tom}^\dagger = \text{tom}$, $\text{Mary}^\dagger = \text{mary}$, etc.. Let

$$
\begin{array}{rcl}
\mathbf{tr}(R\{\delta_1, \ldots, \delta_n\}) & = & R(\delta_1^\dagger, \ldots, \delta_n^\dagger) \\
\mathbf{tr}(\delta_1\ \mathbf{is}\ \delta_2) & = & \delta_1^\dagger = \delta_2^\dagger \\
\mathbf{tr}(\mathbf{not}\ \gamma) & = & \neg \mathbf{tr}(\gamma) \\
\mathbf{tr}(\gamma \Rightarrow \gamma') & = & \mathbf{tr}(\gamma) \to \mathbf{tr}(\gamma') \\
\mathbf{tr}(\mathbf{all}(u, \gamma)) & = & \forall u^\dagger\, \mathbf{tr}(\gamma)
\end{array}
$$

For any formula φ and state variable i, let φ^i be the result of substituting $V(u, i)$ for each free $u^\dagger$ in φ. That our fragment is really just another incarnation of predicate logic (provided that different node names refer to different nodes) is the content of the following theorem.

Theorem 1 *Let γ be a term of type $s \to t$ as defined above. Assume $\mathcal{A}16 - \mathcal{A}20$ and assume that $n \neq n'$ for every pair n, n' of syntactically different node names in γ. Then $\mathbf{tr}(\gamma)^i$ is equivalent with $\gamma(i)$, for any state variable i.*

Proof. Let us write $[t/x]\varphi$ for the result of substituting t for each free x in φ. It follows from $\mathcal{A}16$–$\mathcal{A}18$ that, for each $u \in \mathcal{U}$,

$$
\forall i\, [\forall x\, \varphi \leftrightarrow \forall j\, [i[u]j \to [V(u, j)/x]\varphi]].
$$

Moreover, by $\mathcal{A}19$, $\mathcal{A}20$ and our assumption of noncoreference of different node names, we have that $\forall j\,[i[u]j \rightarrow V(u', j) = V(u', i)]$ if u and u' are syntactically different referents $\in \mathcal{U}$. Hence

$$\forall j\,[i[u]j \rightarrow [\varphi^j \leftrightarrow ([V(u,j)/u^\dagger]\varphi])^i]].$$

Using these observations, the theorem can easily be proved with the help of an induction on the construction of γ. We do two cases here, leaving the other three to the reader. In the following '$\approx$' stands for 'is equivalent with'.

- $(\mathbf{tr}(R\{\delta_1, \ldots, \delta_n\}))^i \approx R(\delta_1^\dagger, \ldots, \delta_n^\dagger)^i$
  $\approx$ (by $\mathcal{A}21$ and the definition of $(.)^i$)
  $R(V(\delta_1, i), \ldots, V(\delta_n, i)) \approx R\{\delta_1, \ldots, \delta_n\}(i)$

- $\mathbf{tr}(\mathbf{all}(u, \gamma))^i \approx (\forall u^\dagger\, \mathbf{tr}(\gamma))^i$
  $\approx$ (by the first observation above)
  $(\forall j\,[i[u]j \rightarrow [V(u,j)/u^\dagger]\mathbf{tr}(\gamma)])^i \approx \forall j\,[i[u]j \rightarrow ([V(u,j)/u^\dagger]\mathbf{tr}(\gamma))^i]$
  $\approx$ (by the second observation)
  $\forall j\,[i[u]j \rightarrow (\mathbf{tr}(\gamma))^j] \approx$ (by induction)
  $\forall j\,[i[u]j \rightarrow \gamma(j)] \approx \mathbf{all}(u, \gamma)(i)$

$\square$

The theorem's requirement on different node names will easily be seen to be fulfilled in the next section.

(23) a. $\mathbf{all}(h_1, \textsc{girl}\{h_1\} \Rightarrow p)$
b. $\textsc{dance}\{h_1\}$
c. $\mathbf{all}(h_1, \textsc{girl}\{h_1\} \Rightarrow \textsc{dance}\{h_1\})$
d. $\forall z_1\,[\textsc{girl}(z_1) \rightarrow \textsc{dance}(z_1)]$

Our $s \rightarrow t$ terms just express what can be expressed using predicate logic in a less roundabout way and the theorem shows that we can reason with them exactly as we reason with first-order formulas. But the theorem is restricted to a special subset of terms and as soon as we leave this class there are some surprises. Since binding is obtained with the help of closed terms, we can do substitutions which have no direct analogue in the more usual set-up. For example, since (23b) is a closed term, substitution of (23b) for p (a variable of type $s \rightarrow t$) in (23a) is possible, even though this seems to result in a form of binding: while h_1 does not occur within in some context $\mathbf{all}(h_1, \gamma)$ in (23b), the substituted occurrence is within such a context in (23c), the result of substitution. Nothing impermissable has occurred however, as the reader may perhaps want to verify by expanding definitions. Obviously, the translation of (23c) is (23d). Had we tried to obtain this result by performing a similar substitution directly on more conventional forms, a clash of variables would have prevented us.

(24) $\lambda v.\mathbf{some}(v, \textsc{man}\{v\})$

Another possibility that we certainly do not have with the standard forms is that we can abstract over 'binder' positions. In (24) the variable v occurs three times. The first occurrence is in an ordinary binder which binds the other two. But the second occurrence is in a position where binding can be mimicked. If we apply (24) to some referent, $\rho(n_1)$ say, we obtain $\mathbf{some}(\rho(n_1), \textsc{man}\{\rho(n_1)\})$, which mimicks $\exists x_1\ \textsc{man}(x_1)$. An expansion of the definitions involved will show that the 'binding of a binder' in (24) is correct.

5 Constraining Meanings

We can now interpret Logical Forms in the usual way, by telling what the values of their lexical nodes are and describing values of nonlexical nodes in terms of the values of their daughters and the nodes linked to them. Here is a very simple lexicon that will serve for purposes of exposition.

$$\begin{array}{rcl}
\textit{John} & \leadsto & \textsc{John} \\
\textit{runs} & \leadsto & \lambda v.\textsc{run}\{v, w\} \\
\textit{man} & \leadsto & \lambda v.\textsc{man}\{v, w\} \\
\textit{find} & \leadsto & \lambda v' \lambda v.\textsc{find}\{v, v', w\} \\
\textit{is} & \leadsto & \lambda v' \lambda v.v \textbf{ is } v' \\
\textit{believe} & \leadsto & \lambda v \lambda p.\mathbf{all}(w_1, w_1 \textbf{ is } w \Rightarrow \mathbf{all}(w, B\{v, w, w_1\} \Rightarrow p)) \\
\textit{who} & \leadsto & \lambda v' \lambda p \lambda P \lambda v.[v \textbf{ is } v' \ \& \ P(v') \ \& \ p] \\
\textit{every}, \textit{each} & \leadsto & \lambda P \lambda v \lambda p.\mathbf{all}(v, P(v) \Rightarrow p) \\
\textit{a}, \textit{some} & \leadsto & \lambda P \lambda v \lambda p.\mathbf{some}(v, P(v) \ \& \ p) \\
\textit{no} & \leadsto & \lambda P \lambda v \lambda p.\mathbf{no}(v, P(v) \ \& \ p) \\
\textit{the} & \leadsto & \lambda P \lambda v \lambda p.\mathbf{some}(v, \mathbf{all}(h_1, P(h_1) \Leftrightarrow h_1 \textbf{ is } v) \ \& \ p)
\end{array}$$

Note that constants such as $\textsc{run}$, $\textsc{find}$, etc. come with an argument w, which refers to the world of evaluation. A sentence 'John runs' will be interpreted as $\textsc{run}\{\textsc{John}, w\}$, which is equivalent with $\lambda i.\textsc{run}(\textsc{john}, V(w, i))$, the proposition which is true at an index i if John runs in the world component of i. In the present set-up translations of complete sentences are only dependent on the world component of indices, but the technique easily allows for introducing more components (such as reference time, utterance time, time of evaluation, situation of utterance, etc.) without a change in type assignment (see [23] for such multicomponent translation). The referent w will get 'bound' in intensional contexts. In (25b), the translation of (25a), for example, the term $\textsc{run}\{\textsc{John}, w\}$ is brought into a context where binding of w is mimicked. Since, in view of Theorem 1, we can reason with our terms just as if they were ordinary predicate logical formulas, the resulting term can be simplified to (25c).

(25) a. Mary believes John runs

b. $\mathbf{all}(w_1, w_1 \textbf{ is } w \Rightarrow \mathbf{all}(w, B\{\textsc{Mary}, w, w_1\} \Rightarrow \textsc{run}\{\textsc{John}, w\}))$

c. $\mathbf{all}(w_1, B\{\text{Mary}, w_1, w\} \Rightarrow \text{run}\{\text{John}, w_1\}))$

We interpret the relation B as the relation of being a doxastic alternative in Hintikka's sense, i.e. $B(x, y, y')$ stands for 'world y is a doxastic alternative of x in world y'', so that (25c) states that 'John runs' is true in all Mary's doxastic alternatives.

The next seven axioms associate Logical Forms with meanings. In order to express that a node has a certain meaning we have introduced families of constants σ^α and σ_0^α of type $node \times \alpha \to t$. $\sigma^\alpha(n, A)$ says that node n has meaning A (of type α) and σ_0^α is used to store intermediate values. Axiom $\mathcal{A}22$ assigns lexical meanings to lexical nodes. $\mathcal{A}23$ and $\mathcal{A}24$ give the standard way in which meanings are inherited from below: nodes with a single daughter inherit their meaning from that daughter and nodes with two daughters get their meaning by means of function application. The meaning that is computed is stored with the help of the relevant σ_0^α, but $\mathcal{A}25$ ascertains that $\sigma^\alpha(n, A)$ can be concluded from $\sigma_0^\alpha(n, A)$ in the default case where there is no link from n or to n. If there is such a link, however, $\mathcal{A}26$ makes sure that its source node will be interpreted as a certain register (the register which ρ connects with the target node) while $\mathcal{A}27$ and $\mathcal{A}28$ demand that this very same register will also enter the interpretation of the target. In the case of 'moved' constituents which have left a trace e, the preliminary interpretation of the moved constituent must be applied to the value of the trace to obtain a final interpretation and in the case of quantification the final interpretation of the target S node can be computed from its preliminary interpretation (the interpretation of its daughter), the preliminary interpretation of the NP and the register in question.

$\mathcal{A}22$ $\forall k\, [lab(k, a) \to \sigma_0^\alpha(k, A)]$, where $a \rightsquigarrow A$ is in the lexicon and the type of A is α

$\mathcal{A}23$ $\forall k_1 k_2 \forall X_\alpha\, [[k_1 \lhd k_2 \wedge \forall k_3\, [k_1 \lhd k_3 \to k_3 = k_2] \wedge \sigma^\alpha(k_2, X_\alpha)] \to \sigma_0^\alpha(k_1, X_\alpha)]$

$\mathcal{A}24$ $\forall k_1 k_2 k_3 \forall F_{\alpha\beta} \forall X_\alpha\, [[k_1 \lhd k_2 \wedge k_1 \lhd k_3 \wedge \forall k_4\, [k_1 \lhd k_4 \to [k_4 = k_2 \vee k_4 = k_3]]$
$\wedge k_2 \neq k_3 \wedge \sigma^{\alpha\beta}(k_2, F) \wedge \sigma^\alpha(k_3, X)] \to \sigma_0^\beta(k_1, F(X))]$

$\mathcal{A}25$ $\forall k_1\, [\neg[\exists k_2 link(k_1, k_2) \vee \exists k_2 link(k_2, k_1)] \to \forall X_\alpha\, [\sigma_0^\alpha(k_1, X) \to \sigma^\alpha(k_1, X)]]$

$\mathcal{A}26$ $\forall k_1 k_2\, [link(k_1, k_2) \to \sigma^\pi(k_1, \rho(k_2))]$

$\mathcal{A}27$ $\forall k_1 k_2 \forall f_{\pi\alpha}\, [[link(k_1, k_2) \wedge lab(k_1, e) \wedge \sigma_0^{\pi\alpha}(k_2, f)] \to \sigma^\alpha(k_2, f(\rho(k_2)))]$

$\mathcal{A}28$ $\forall k_1 k_2 \forall Q_\mu \forall p\, [[link(k_1, k_2) \wedge quant(k_1) \wedge \sigma_0^\mu(k_1, Q) \wedge \sigma_0^{s \to t}(k_2, p)] \to$
$\sigma^{s \to t}(k_2, Q(\rho(k_2))(p))]$, where $\mu = \pi \to ((s \to t) \to (s \to t))$

This axiomatises the process of assigning meanings to trees and, from any description of a tree we may now draw conclusions about the meaning that should be associated with its top node, even if no unique meaning can be isolated. For an example, let us return to the description (20) + (21), which has twelve Logical Forms according to our axioms. Given this description, many statements about the meanings of nodes can be inferred. For example, $\mathcal{A}22$ tells us that

(26a) holds, and since it can be shown that n_{19} is neither the source nor the target of a link, (26b) holds as well on the basis of $\mathcal{A}25$. Since n_{19} is the sole daughter of n_{18} in any of the Logical Forms admitted by our theory, it follows from $\mathcal{A}23$ that n_{19} can be replaced with n_{18} in (26a). Continuing this kind of reasoning we may derive (26c).

(26) a. $\sigma_0^{(\pi\to(s\to t))\to\mu}(n_{19}, \lambda P\lambda v\lambda p.\mathbf{all}(v, P(v) \Rightarrow p))$
b. $\sigma^{(\pi\to(s\to t))\to\mu}(n_{19}, \lambda P\lambda v\lambda p.\mathbf{all}(v, P(v) \Rightarrow p))$
c. $\sigma_0^{\mu}(n_{17}, \lambda v\lambda p.\mathbf{all}(v, \text{GIRL}\{v, w\} \Rightarrow p))$

What is the σ value of the NP node n_{17}? Since the *quant* predicate holds of this node in all models satisfying our theory, the existential statement (27a) is derivable with the help of $\mathcal{A}14$. In (27b) a witness n_{38} is taken and we may conclude, using $\mathcal{A}26$ that $\sigma^{\pi}(n_{17}, \rho(n_{38}))$. In a similar way we may introduce constants n_{39}, n_{40} and n_{41} such that $link(n_{25}, n_{39})$, $link(n_2, n_{40})$ and $link(n_{33}, n_{41})$ and $\sigma^{\pi}(n_{25}, \rho(n_{39}))$, $\sigma^{\pi}(n_2, \rho(n_{40}))$ and $\sigma^{\pi}(n_{33}, \rho(n_{41}))$.

(27) a. $\exists k_2\,[link(n_{17}, k_2) \wedge \exists k_3[k_2 \lhd k_3 \wedge \forall k_4[k_2 \lhd k_4 \to k_3 = k_4]]]$
b. $link(n_{17}, n_{38}) \wedge \exists k_3[n_{38} \lhd k_3 \wedge \forall k_4[n_{38} \lhd k_4 \to k_3 = k_4]]$
c. $n_{38} \lhd n_{42} \wedge \forall k_4[n_{38} \lhd k_4 \to n_{42} = k_4]]$
d. $\forall p\,[\sigma^{s\to t}(n_{42}, p) \to \sigma_0^{s\to t}(n_{38}, p)]$
e. $\forall p\,[\sigma^{s\to t}(n_{42}, p) \to \sigma^{s\to t}(n_{38}, \mathbf{all}(\rho(n_{38}), \text{GIRL}\{\rho(n_{38}), w\} \Rightarrow p))]$

We can also obtain information about the semantic value of n_{38}, the node that is linked to n_{17}, but this information must necessarily be hypothetical. Let us first take a witness n_{42} for the node which according to (27b) is immediately dominated by n_{38}, i.e. let us derive (27c). Then we can use $\mathcal{A}23$ to derive (27d) and, using this last statement, (26c) and $\mathcal{A}28$, derive (27e). In a similar way we can take witnesses n_{42}, n_{43} and n_{44} for the S nodes immediately dominated by n_{39}, n_{40} and n_{41} respectively. The first three items in Table 2 express the values of n_{38}, n_{39} and n_{41} in terms of those of n_{42}, n_{43} and n_{45}. The value of n_{40} can also be expressed, but not in terms of the value of n_{44} alone. It is clear that in every model of the theory, the RC node n_8 will immediately dominate some node n_{46} which is different from the NP node n_9 (and that there will be no daughters of n_6 other than these two nodes). The value of n_{40} can be computed from the values of this node and n_{44} with a result as in the Table. The meaning which is attached to the S node n_{11} depends in a similar way on the value of the node n_{47} which is immediately dominated by the VP n_{13} but not equal to the V. The Table gives the relevant statement and lists two other sentences which are also derivable.

We have taken witnesses n_{42}–n_{47} for various existential statements; but we also know from $\mathcal{A}10$–$\mathcal{A}13$ that n_1–n_{41} are all the nodes. This means that n_{42}–n_{47} must corefer with node names in this last set and, due to various restrictions on dominance, incompatibility of categorial information, island constraints etc. there are exactly twelve ways of matching node names. In (22) the following matching holds.

$\forall p\,[\sigma(n_{42}, p) \rightarrow \sigma(n_{38}, \mathbf{all}(\rho(n_{38}), \text{GIRL}\{\rho(n_{38}), w\} \Rightarrow p))]$
$\forall p\,[\sigma(n_{43}, p) \rightarrow \sigma(n_{39}, \mathbf{some}(\rho(n_{39}), \text{UNICORN}\{\rho(n_{39}), w\} \ \&\ p))]$
$\forall p\,[\sigma(n_{45}, p) \rightarrow \sigma(n_{41}, \mathbf{some}(\rho(n_{41}), \text{FISH}\{\rho(n_{41}), w\} \ \&\ p))]$
$\forall pp'\,[\sigma(n_{46}, p) \rightarrow [\sigma(n_{44}, p') \rightarrow$ $\sigma(n_{40}, \mathbf{all}(\rho(n_{40}), [\rho(n_{40}) \ \mathbf{is}\ \rho(n_{10}) \ \&\ \text{MAN}\{\rho(n_{10}), w\} \ \&\ p] \Rightarrow p')]]$
$\sigma(n_{16}, \text{LOVE}\{\rho(n_{38}), \rho(n_{39}), w\})$
$\sigma(n_1, \text{EAT}\{\rho(n_{40}), \rho(n_{41}), w\})$
$\forall p\,[\sigma(n_{47}, p) \rightarrow \sigma(n_{11}, \mathbf{all}(w_1, w_1 \ \mathbf{is}\ w \Rightarrow \mathbf{all}(w, B\{\rho(n_{10}), w, w_1\} \Rightarrow p))]$

Table 2: Some sentences derivable from (20) + (21)

(28) $n_{42} = n_{16},\ n_{43} = n_{11},\ n_{44} = n_{41},\ n_{45} = n_1,\ n_{46} = n_{39},\ n_{47} = n_{38}$

Combining this information with the statements in Table 2 reveals that in (22) the sentence (29), which can be simplified to (30), holds.

(29) $\sigma(n_{40}, \mathbf{all}(\rho(n_{40}), [\rho(n_{40}) \ \mathbf{is}\ \rho(n_{10}) \ \&\ \text{MAN}\{\rho(n_{10}), w\} \ \&\ \mathbf{some}(\rho(n_{39}), \text{UNICORN}\{\rho(n_{39}), w\} \ \&\ \mathbf{all}(w_1, w_1 \ \mathbf{is}\ w \Rightarrow \mathbf{all}(w, B\{\rho(n_{10}), w, w_1\} \Rightarrow \mathbf{all}(\rho(n_{38}), \text{GIRL}\{\rho(n_{38}), w\} \Rightarrow \text{LOVE}\{\rho(n_{38}), \rho(n_{39}), w\}))))] \Rightarrow \mathbf{some}(\rho(n_{41}), \text{FISH}\{\rho(n_{41}), w\} \ \&\ \text{EAT}\{\rho(n_{40}), \rho(n_{41}), w\})))$

(30) $\sigma(n_{40}, \mathbf{all}(h_{40}, [\text{MAN}\{h_{40}, w\} \ \&\ \mathbf{some}(h_{39}, \text{UNICORN}\{h_{39}, w\} \ \&\ \mathbf{all}(w_1, B\{h_{40}, w_1, w\} \Rightarrow \mathbf{all}(h_{38}, \text{GIRL}\{h_{38}, w_1\} \Rightarrow \text{LOVE}\{h_{38}, h_{39}, w_1\})))] \Rightarrow \mathbf{some}(h_{41}, \text{FISH}\{h_{41}, w\} \ \&\ \text{EAT}\{h_{40}, h_{41}, w\})))$

Can we say something in general about the meaning that is connected with (20) + (21)? Let us write $root(n)$ if n has no predecessors in the proper dominance relation, i.e. define $root(n)$ to be an abbreviation of $\neg\exists k\, k \lhd^+ n$. In every possible Logical Form the root will have a certain value, so a sentence of the form (31) will be derivable from our description. One of the disjuncts here is given by (30) the result of the matching in (28), the others by the results of the remaining possible matchings.

(31) $\exists k\,[root(k) \wedge [\sigma(k, \gamma_1) \vee \ldots \vee \sigma(k, \gamma_{12})]]$

Our theory should be contrasted with attempts to describe ambiguous meanings in terms of disjunctions in a more direct way. In particular, it should be noted that (32) is not derivable from our input description. Theories which explain ambiguity by means of disjunction in this way have difficulties with explaining entailment as the meanings assigned to ambiguous statements generally are too weak. Suppose we have established that there is no man who does not eat a fish. Then we should not be entitled to derive (19) in its full ambiguity, as some of its readings follow, others do not. Additional information about (19) may reveal that an existential reading was meant and—as far as the monotonic core of our reasoning system is involved[19]—we do not wish to have to revise judgements about entailment. Since we explain ambiguity along the lines of (31) such dubious entailments will not be predicted to be valid and we will not have to retract any judgments if more information about what was meant becomes available.

(32) $\exists k\,[root(k) \wedge \sigma(k, \gamma_1 \textbf{ or} \ldots \textbf{or } \gamma_{12})]$

The statement (31) is deriv*able* from (20) + (21) in our system, but it should not be concluded that a hearer who has inferred from the parsing process that (20) + (21) holds will go on to actually derive it, as his reasoning would involve an inspection of all possibilities of disambiguation, the very kind of inspection we have assumed is highly implausible on grounds of efficiency. A more rational course of action for the hearer is to try to strengthen the description (draw in conclusions from other sources, 'jump to conclusions' on the basis of rules of thumb) until it is strong enough to provide the information she desires to extract.

6 Reasoning with Underspecified Meanings

Let us suppose that our hearer hears the two sentences in (33a). What can she do in order to establish that the argument is valid? Let us assume that she parses the sentences and arrives at a simultaneous description of them. Each model of this description will be a forest consisting of two trees, with any node of the tree for 'Every man dances' ordered before any node of the other tree. We may assume that our hearer circumscribes the domain of the forest, so that no model will contain nodes for which there is no evidence. The first sentence is unambiguous, but the second is sixfold ambiguous, so all in all there will be six possible Logical Forms. The root node n of the first tree will get a value along the lines of (33b). For the root node n_{40} of the second tree there is no single definite value. Nevertheless, there is information about the value of this node, for among the statements derivable from the description that was obtained will be one of the form (33c), which gives the value of n_{40} in terms of the value of the embedded sentence.

(33) a. Every man dances. (Therefore) every man who thinks each girl loves a unicorn dances.

b. $\sigma(n, \mathbf{all}(h_1, \text{MAN}\{h_1, w\} \Rightarrow \text{DANCE}\{h_1, w\}))$

c. $\forall p\,[\sigma(n_{46}, p) \rightarrow \sigma(n_{40}, \mathbf{all}(\rho(n_{40}),$
$[\rho(n_{40}) \textbf{ is } \rho(n_{10}) \ \&\ \text{MAN}\{\rho(n_{10}), w\} \ \&\ p] \Rightarrow \text{DANCE}\{\rho(n_{40}), w\}))]$

[19] We do not mean to preclude nonmonotonic forms of reasoning but it seems reasonable to demand that the *only* reason for retracting judgements should be that we have jumped to a conclusion on the basis of a reliable default rule and that drawing that conclusion has been somewhat rash. There should be no revisions without default assumptions.

Clearly, whatever value is chosen for the embedded sentence node n_{46}, the σ value assigned to n_{40} will follow from the value assigned to n. The hearer can conclude to the validity of the argument without having to do any disambiguation.

(34) $$\mathcal{D} \models_{AX} \exists k_1 [root(k_1) \wedge \forall k_2 [root(k_2) \rightarrow k_2 \prec k_1] \wedge \\ \forall p [\sigma(k_1, p) \rightarrow \forall i [\forall k_2 \forall p' [[root(k_2) \wedge k_2 \prec k_1 \wedge \sigma(k_2, p')] \rightarrow p'(i)] \rightarrow p(i)]]$$

Formally we can characterise validity in the following way. Given any description $\mathcal{D}$ of a forest, $\mathcal{D}$ describes a valid argument if (34) holds (we write $\models_{AX}$ for entailment given our axioms). This means that, for every possible choice of Logical Forms for premises and conclusion consistent with $\mathcal{D}$, the value of the conclusion follows from the values of the premises in the sense that the conclusion holds for every state for which all premises hold.

Our notion of consequence is rather strict in a sense. For example, it will not in general support the validity of 'A, therefore A' if A is an ambiguous sentence. An argument is valid if the conclusion follows from the premises whatever readings we assign to conclusion and premises and in the 'A, therefore A' case we may choose different readings for the two occurrences of A. The fallacy is one of Equivocation.

(35) Every man loves a woman. (Therefore,) every man loves a woman.

However, there remains a feeling that (35) should be valid since anyone who would utter it (for whatever reason) is likely to intend premise and conclusion to have the same reading. Reyle (see [32, 33]) has argued that such arguments are indeed valid and has accounted for the alleged validity in terms of correlations between premise and conclusion. The idea is that ambiguous sentences of the same type have to be disambiguated simultaneously.

I do not wish to deny that there may be correlations between the premises and the conclusions of an argument which may force disambiguations to take place in tandem. On the contrary, I think that the correlations in question are not limited to the expressions we find in arguments but are omnipresent in natural language and should be headed under the general phenomenon of *parallelism*. It does not follow, however, that such correlations should be made a part of the *logic* of ambiguous expressions; they are part of the linguistic data and can be used as extra premises. Let us suppose that our hearer has some means of recognising parallelism and let us model the fact that nodes n and n' are recognised to be parallel as the atomic statement $par(n, n')$. The following seem reasonable requirements on the parallelism relation.

(36) a. $\forall k_1 k_2 k_3 k_4 [[par(k_1, k_3) \wedge par(k_2, k_4) \wedge k_1 \lhd^+ k_2] \rightarrow k_3 \lhd^+ k_4]$

b. $\forall k_1 k_2 k_3 k_4 [[par(k_1, k_3) \wedge link(k_1, k_2) \wedge link(k_3, k_4)] \rightarrow par(k_2, k_4)]$

We may add these requirements to our list of axioms. The first demands parallelism to preserve proper dominance while the second says that nodes linked to parallel nodes are themselves parallel. Now consider a description $\mathcal{D}$ of the argument in (35). We may suppose that the subject of the first occurrence of

'Every man loves a woman' is associated with a node n_1 in $\mathcal{D}$, the object with a node n_2, the subject of the second occurence with n_3 and the object with n_4. A hearer who recognises that $par(n_1, n_3)$ and $par(n_2, n_4)$ may add these expressions to the description $\mathcal{D}$ and get a strengthened description $\mathcal{D}'$. But in doing so, the hearer has removed some ambiguity. For since the NPs are parallel, the places where they are linked to are parallel too, and if one NP outscopes the other in one structure its parallel element must outscope the other's parallel element in the other structure. The number of readings reduces from four to two and $\mathcal{D}'$, unlike $\mathcal{D}$, describes a valid argument.

7 Conclusion

In this paper we have sketched the outlines of a theory of semantic underspecification which is based on ordinary logic and we have given a characterisation of the notion of consequence for ambiguous expressions. This characterisation was also formulated in terms of classical logic and in order to prove an argument valid we may have to reason about the forms of the expressions involved as well as about their contents. This, we feel, is exactly what happens when language is processed and it is wrong to separate the two kinds of reasoning. As soon as it is recognised that reasoning about form and reasoning about content go hand in hand, the mysterious aspects of underspecified semantic representation disappear. For then it becomes clear that we do not have to find a completely new logic which uses underspecified forms as its vehicle of reasoning, but that we can use a well-understood one to reason in tandem about what was said and what is the case.

References

[1] J. Allen. *Natural Language Understanding.* Benjamin/Cummings, Redwood City, CA, 1995.

[2] H. Alshawi and R. Crouch. Monotonic Semantic Interpretation. In *Proceedings of the ACL*, pages 32–39, 1992.

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

[4] P. Blackburn. Modal Logic and Attribute-Value Structures. In M. de Rijke, editor, *Diamonds and Defaults.* Kluwer, Dordrecht, 1993.

[5] J. Bos. Predicate Logic Unplugged. In P. Dekker and M. Stokhof, editors, *Proceedings of the 10th Amsterdam Colloquium*, pages 133–142, Amsterdam, 1995.

[6] C.C. Chang and H.J. Keisler. *Model Theory.* North-Holland, Amsterdam, 1973.

[7] N. Chomsky. *Some Concepts and Consequences of the Theory of Government and Binding.* MIT Press, Harvard, 1982.

[8] A. Copestake, D. Flickinger, R. Malouf, S. Riehemann, and I. Sag. Minimal Recursion Semantics, 1995. Manuscript.

[9] T.L. Cornell. On Determining the Consistency of Partial Descriptions of Trees. In *Proceedings of ACL-94*, 1994.

[10] J.E. Fenstad, P.-K. Halvorsen, T. Langholm, and J. van Benthem. *Situations, Language and Logic.* Reidel, Dordrecht, 1987.

[11] G. Gazdar, E. Klein, G. Pullum, and I. Sag. *Generalized Phrase Structure Grammar.* Harvard University Press, Cambridge MA, 1985.

[12] J. Hobbs. Representing Ambiguity. In *Proceedings of the First WCCFL*, pages 15–28, 1982.

[13] J. Hobbs. An Improper Treatment of Quantification in Ordinary English. In *Proceedings of the ACL*, pages 57–63, 1983.

[14] M. Johnson. Logic and Feature Structures. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.

[15] R. Kaplan and J. Bresnan. Lexical-Functional Grammar: a Formal System for Grammatical Representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA, 1982.

[16] R. Kasper and W. Rounds. A Logical Semantics for Feature Structures. In *Proc. 24th meeting of the ACL*, pages 101–115, 1986.

[17] M.P. Marcus. Deterministic Parsing and Description Theory. In P. Whitelock, M.M. Wood, H.L. Somers, R. Johnson, and P. Bennett, editors, *Linguistic Theory and Computer Applications*, pages 69–112. Academic Press, London, 1987.

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

[19] J.T. Maxwell III and R.M. Kaplan. A Method for Disjunctive Constraint Satisfaction. In M. Tomita, editor, *Current Issues in Parsing Technology*, pages 173–190. Kluwer, Dordrecht, 1991.

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

[21] R. Montague. The Proper Treatment of Quantification in Ordinary English. In *Formal Philosophy*, pages 247–270. Yale University Press, New Haven, 1973.

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

[23] R.A. Muskens. Tense and the Logic of Change. In U. Egli, E.P. Pause, C. Schwarze, A. Von Stechow, and G. Wienold, editors, *Lexical Knowledge in the Organization of Language*, pages 147–183. John Benjamins, Amsterdam, 1995.

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

[25] J. Nerbonne. Constraint-based Semantics. In P. Dekker and M. Stokhof, editors, *Proceedings of the Eighth Amsterdam Colloquium*. ILLC, Amsterdam, 1992.

[26] F.C.N. Pereira. Categorial Semantics and Scoping. *Computational Linguistics*, 16:1–9, 1990.

[27] M. Pinkal. Constraints for Semantic Underspecification. CLAUS report 73, Universität des Saarlandes, 1996.

[28] M. Pinkal. Radical Underspecification. In *Proceedings of the Tenth Amsterdam Colloquium*, pages 587–606, 1996.

[29] M. Poesio. Ambiguity, Underspecification and Discourse Interpretation. In H. Bunt, R.A. Muskens, and G. Rentier, editors, *Proceedings of the First International Workshop on Computational Semantics*, pages 151–160, Tilburg, 1994. ITK, Tilburg University.

[30] S.G. Pulman. A Computational Theory of Context Dependence. In H. Bunt, R.A. Muskens, and G. Rentier, editors, *Proceedings of the First International Workshop on Computational Semantics*, pages 161–170, Tilburg, 1994. ITK, Tilburg University.

[31] U. Reyle. Dealing with Ambiguities by Underspecification: Construction, representation and deduction. *Journal of Semantics*, 10:123–179, 1993.

[32] U. Reyle. On Reasoning with Ambiguities. In *Proceedings of the EACL-95*, Dublin, 1995.

[33] U. Reyle. Co-indexing Labelled DRSs to Represent and Reason with Ambiguities. In S. Peters and K. van Deemter, editors, *Semantic Ambiguity and Underspecification*. CSLI, Stanford CA, 1996.

[34] J. Ross. *Constraints on Variables in Syntax*. PhD thesis, MIT, 1967.

[35] L.K. Schubert and F.J. Pelletier. From English to Logic: Context-free computation of conventional logic translation. *American Journal of Computational Linguistics*, 8:165–176, 1982.

[36] S.M. Shieber. *An Introduction to Unification-based Approaches to Grammar*. CSLI Lecture Notes. Chicago University Press, Chicago, 1986.

[37] J. van Eijck and J. Jaspars. Ambiguity and Reasoning. Technical Report CS-R9616, Centrum voor Wiskunde en Informatica, 1996.

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

[39] XTAG Research Group. A Lexicalized Tree Adjoining Grammar for English. IRCS Report 95-03, University of Pennsylvania, 1995.