A Modal Perspective on the Computational Complexity of Attribute Value Grammar

PATRICK BLACKBURN
Department of Philosophy, University of Utrecht, Heidelberglaan 8,
3584 CS Utrecht, The Netherlands.
Email: patrick@phil.ruu.nl

and

EDITH SPAAN
Department of Mathematics and Computer Science, University of Amsterdam,
Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
Email: edith@fwi.uva.nl

(Received 5 May, 1992; in final form 6 November, 1992)

Abstract. Many of the formalisms used in Attribute Value grammar are notational variants of languages of propositional modal logic, and testing whether two Attribute Value Structures unify amounts to testing for modal satisfiability. In this paper we put this observation to work. We study the complexity of the satisfiability problem for nine modal languages which mirror different aspects of AVS description formalisms, including the ability to express re-entrancy, the ability to express generalisations, and the ability to express recursive constraints. Two main techniques are used: either Kripke models with desirable properties are constructed, or modalities are used to simulate fragments of Propositional Dynamic Logic. Further possibilities for the application of modal logic in computational linguistics are noted.

Key words: Modal logic, computational complexity, unification formalisms

1. INTRODUCTION

Attribute Value Structures (AVSs) are probably the most widely used means of representing linguistic structure in current computational linguistics, and the process of unifying descriptions of AVSs lies at the heart of many parsers. As a number of people have recently observed (see for example Kracht (1989), Blackburn (1991), Moss (1991), Reape (1991) and Schild (1990)) the most common formalisms for describing AVSs are notational variants of propositional modal languages, AVSs themselves are Kripke models, and unification amounts to looking for a satisfying model for $\phi \wedge \psi$ given two (modal) wffs ϕ and ψ. The purpose of this paper is to make use of this connection with modal logic to investigate the complexity of various unification tasks of interest in computational linguistics.

The paper is structured as follows. The second section begins with an introduction to such topics as 'attributes', 'values', and 'unification' and why they are of interest in computational linguistics. It then goes on to explain the

Journal of Logic, Language, and Information **2**: 129–169, 1993.

link with modal logic, and gives the syntax and semantics of three modal languages — L, L^N and L^{KR} — which correspond to three common unification formalisms. In the third section we examine the satisfiability problems for these languages and show, using a very simple 'small model' argument, that all three are NP complete. In the fourth section we introduce three stronger languages, $L^{\Box}$, $L^{N\Box}$ and $L^{KR\Box}$. These are L, L^N and L^{KR} respectively augmented by the universal modality $\Box$. Adding this modality allows general constraints on linguistic structure to be expressed. As we will show, however, there is a price to pay: the satisfiability problem for $L^{KR\Box}$ is Π^0_1 complete. We then go on to show that dropping the ability to enforce generalisations involving re-entrancy results in decidable systems. In fact we show that the satisfiability problems for both $L^{\Box}$ and $L^{N\Box}$ are EXPTIME complete. In the fifth section we examine modal languages in which recursive constraints on linguistic structure can be expressed, namely systems built using the master modality $[*]$ of Gazdar *et al.* (1988) and Kracht (1989). We augment our base languages L, L^N and L^{KR} with $[*]$, forming $L^{[*]}$, $L^{N[*]}$ and $L^{KR[*]}$ respectively, and investigate the complexity of their satisfiability problems. We show that many of the proof methods and results from our discussion of the the universal modality transfer to the new setting, though in the case of most interest the satisfiability problem for $L^{KR[*]}$ turns out to be highly undecidable, in fact, Σ^1_1 complete. We conclude the paper with a table summarising our results and a discussion of more general issues arising from this work.

The paper is relatively self contained; in particular, all the necessary concepts from unification based grammar and modal logic are presented. However we do assume that the reader understands what is meant by such complexity classes as NP, EXPTIME and so on; such definitions may be found in Balcázar *et al.* (1988), for example. Further, later in the paper some ideas from Propositional Dynamic Logic (PDL) are used. While these are explained, some readers may find the additional background provided by Harel (1984) helpful. For further information on modal logic the reader is referred to Hughes and Cresswell (1984); and for more on unification based grammar, Shieber (1986) and Carpenter (1992) are useful. Finally, it's worth remarking that there is a hidden agenda: although we emphasize the use of modal logic as tool for grammar specification, it is our belief that modal techniques have a wider role to play in computational linguistics and some possibilities are noted in the course of the paper.

2. ATTRIBUTE VALUE LOGIC

Even the most cursory examination of recent proceedings of computational linguistics conferences reveals that there is a substantial level of interest in such topics as ‘attributes’, ‘values’, and ‘unification’. This section presents a brief introduction to these topics, and explains what they have to do with modal logic. The basic point it makes is that the most common machinery underlying Attribute Value grammar formalisms is simply that of propositional modal logic, and that testing whether unification is possible amounts to testing for modal satisfiability. This correspondence provides the *raison d'être* of the paper: by examining the complexity of the satisfiability problem for the modal languages involved, we learn — often very straightforwardly — about the complexity of various tasks of interest to computational linguistics.

Perhaps the best way of approaching these topics is via Attribute Value Matrices (AVMs), or Feature Value Matrices as they are sometimes called. A (rather simple) AVM might look something like this:

$$\begin{bmatrix} \text{CASE} & \textit{nominative} \\ \text{AGREEMENT} & [\text{PERSON } \textit{1st}] \end{bmatrix}$$

Such an AVM is taken to be a partial description of some piece of linguistic structure. In this case we are describing a piece of linguistic structure that has two attributes, namely CASE and AGREEMENT. The CASE attribute takes as value the atomic value *nominative*, while the AGREEMENT attribute takes as value the complex entity [PERSON *1st*]. This complex entity consists of an attribute PERSON that takes as value the atomic value *1st*. The particular atomic values (or constants) and attributes (or features) that may occur in AVMs varies widely from theory to theory, but typical choices of atomic entities a syntactician might make are *singular, plural, 3rd, 2nd, 1st, genitive* and *accusative*; and when it comes to a choice of attributes the selection might include TENSE, NUMBER, PERSON, AGREEMENT, and CASE. But although the different theories differ on the particular choices made, and indeed in the uses they put this machinery to, they are united in agreeing that at least a part of our descriptions of linguistic structure should embody the idea of attributes taking (possibly complex) values.

The information expressed by AVMs can be considerably more complex than in the above example. The above AVM is purely conjunctive, but many linguists feel it is necessary to be able to express both disjunctive and negative information in their Attribute Value grammars. To give two well known examples due to Karttunen (1984), one might write

$$\begin{bmatrix} \text{NUMBER} & \textit{plural} \\ \text{CASE} & \{\textit{nominative, genitive, accusative}\} \end{bmatrix}$$

an AVM which states that the attribute CASE takes one of the values *nominative*, *genitive*, or *accusative*, but doesn't say which; or one might write

$$\begin{bmatrix}\text{NUMBER} & \textit{plural}\\ \text{CASE} & [\neg\ \textit{dative}]\end{bmatrix}$$

an AVM which specifies that CASE doesn't take the value *dative*.

It's worth making a short historical remark here. We'll shortly be introducing Attribute Value Structures (AVSs) and treating them as semantic structures for AVMs. That is, we're going to be adopting the now standard distinction between description languages (for example AVMs) and linguistic structure (the AVSs). Historically, the impetus for making this distinction was motivated by the difficulties involved in giving a precise account of AVMs that employed disjunction or negation. The distinction was introduced in Pereira and Shieber (1984), and it underpins the influential work of Kasper and Rounds (1986) and Rounds and Kasper (1986). Thus the move towards full Boolean expressivity marked an important turning point in the development of Attribute Value formalisms.

What do computational linguists do with AVMs? The answer is, they try to unify them. Intuitively, unifying two AVMs means forming another AVM that combines all the information about Attribute Value dependencies contained in the two constituent AVMs. For example, writing ⊔ to indicate unification, we have:

$$\begin{bmatrix}\text{AGR} & [\text{PER}\ \textit{1st}]\\ \text{CASE} & \textit{nominative}\end{bmatrix} \sqcup [\text{AGR}\ [\text{NUM}\ \textit{plural}]] = \begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{NUM} & \textit{plural}\\ \text{PER} & \textit{1st}\end{bmatrix}\\ \text{CASE} & \textit{nominative}\end{bmatrix}$$

There is a clear sense in which the AVM on the right hand side embodies all the information in the two constituent structures; it is the result of unifying these structures.

But this is rather vague. *Precisely* when is unification possible? Answering this question will lead us first to AVSs, the semantics of AVMs, and then, quite naturally, to the link with modal languages.

AVSs are certain kinds of *decorated labeled graphs*. Such graphs play the central role in unification based linguistics: they are the mathematical model of linguistic structure underlying these frameworks. A number of definitions of AVSs exist in the literature. We shall work with a particularly simple one:

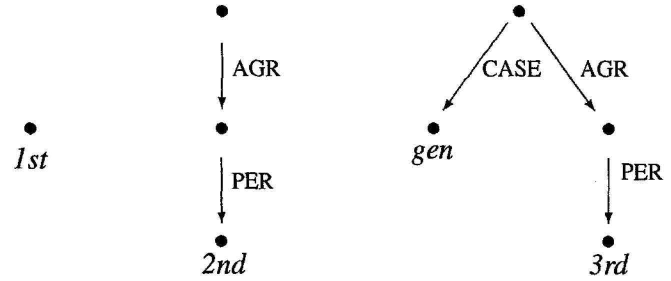
DEFINITION 2.1 (Attribute Value Structures). *Let $\mathcal{L}$ and $\mathcal{A}$ be non-empty finite or denumerably infinite sets, the set of* labels *and the set of* atomic information *respectively. An Attribute Value Structure (AVS) of signature $\langle \mathcal{L}, \mathcal{A} \rangle$*

is a triple $\langle W, \{R_l\}_{l \in \mathcal{L}}, \{Q_\alpha\}_{\alpha \in \mathcal{A}}\rangle$, *where* W *is a non-empty set, the set of nodes; for all* $l \in \mathcal{L}$, R_l *is a binary relation on* W *that is a partial function; and for all* $\alpha \in \mathcal{A}$, Q_α *is unary relation on* W. □

The most important thing to note about this definition is the requirement that all the binary relations be partial functions. As we shall see, this demand plays a crucial role in establishing some of our complexity results.

The definition covers all the well known definitions of Attribute Value Structures, and in particular those of Gazdar *et al.* (1988) and Kasper and Rounds (1986). Moreover it's not too loose: there are only two reasonably common further restrictions on the binary relations that it doesn't insist on. The first is that AVSs must be *point generated.* In point generated AVSs there is always a starting node $w_0 \in W$ such that all other nodes $w \in W$ are reachable via transition sequences from w_0. The second is that AVSs must be *acyclic*, which means that it is never possible to return to a node w by following some sequence of R_l transitions from w. As neither of these restrictions plays a prominent role in the linguistics literature anymore, we ignore them here. This definition also ignores three constraints computational linguists used to routinely place on node decoration. The constraints in question are these. First, for all $w \in W$ and all $\alpha, \beta \in \mathcal{A}$, if $w \in Q_\alpha$ and $\alpha \neq \beta$ then $w \notin Q_\beta$. That is, the constraint forbids what linguists call 'constant-constant clashes'. Second, for all $w \in W$, w is in Q_α for some $\alpha \in \mathcal{A}$ iff w is a terminal node. This constraint rules out 'constant-compound clashes'. Third, for all $w, w' \in W$, if $w \in Q_\alpha$ and $w' \in Q_\alpha$ then $w = w'$. Once again, the main reason for ignoring these demands is that they no longer play the prominent role they once did. Indeed in more recent work in computational linguistics, particularly work in the Head Driven Phrase Structure Grammar (HPSG) framework (see Pollard *et al.* (1987)) much use is made of *sorts* (or *types*); and sorts are essentially pieces of atomic information that don't obey these three restrictions.

Let's consider some concrete examples of AVSs. Suppose we are working with some linguistic theory which contains among its theoretical apparatus the attributes PERSON, CASE and AGREEMENT, and the atomic information *3rd*, *2nd*, *1st* and *genitive*. That is, our linguistic theorising has specified a signature $\langle \mathcal{L}, \mathcal{A}\rangle$ such that {PERSON, CASE, AGREEMENT} $\subseteq \mathcal{L}$, and {*3rd*, *2nd*, *1st*, *genitive*} $\subseteq \mathcal{A}$. Then the following graphs are all examples of AVSs of this signature, as (modulo some obvious abbreviations) nodes are decorated only with items drawn from $\mathcal{A}$ and transitions are labelled only with items drawn from $\mathcal{L}$:



What do AVSs have to do with AVMs? As has already been remarked, AVMs are partial descriptions of linguistic structure, and in fact the structure they describe is the structure embodied in the definition of AVSs. That is, AVMs are a formal language for describing linguistic structure, AVSs provide the interpretation for AVMs, and thus the relationship is that which always exists between semantic and syntactic entities: we talk of AVSs satisfying (or failing to satisfy) the AVMs. To return to our examples, the first graph, consisting of a single node decorated with the atomic information *1st*, satisfies the atomic AVM *1st*. Why? Because this atomic AVM demands a node decorated with the atomic information *1st*, and the first graph is such a node. The second graph satisfies the AVM [AGREEMENT]][PERSON at its root *2nd*
node. Why? Because this AVM demands a node in some piece of linguistic structure that has the following property: a transition along an $R_{\text{AGREEMENT}}$ relation takes one to a node from which it is possible to make an R_{PERSON} transition to a node decorated with the information $2nd$. The root node of the second graph has this property. Finally, consider the third graph. This satisfies the AVM

$$\begin{bmatrix} \text{AGREEMENT} & [\text{PERSON } \mathit{3rd}] \\ \text{CASE} & genitive \end{bmatrix}$$

at its root node.

Now, we could give a precise definition of what it means for an AVS to satisfy an AVM, but in fact this would be a waste of energy, for, as we'll now see, the satisfaction relation between AVSs and AVMs is just a disguised version of something very familiar: the satisfaction relation between Kripke models and modal wffs. There are two facets to this correspondence, the semantical and the syntactical. We'll treat each in turn, beginning with the semantical.

Consider once more the definition of AVSs as triples $\langle W, \{R_l\}_{l \in \mathcal{L}}, \{Q_\alpha\}_{\alpha \in \mathcal{A}} \rangle$. Such triples are just (multimodal) Kripke models: each R_l interprets a modal operator $\langle l \rangle$, and each unary relation Q_α interprets the propositional symbol p_α. To be sure, multimodal Kripke models are usually presented as triples $\langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$, where V is a valuation function from a collection of propositional symbols VAR to $Pow(W)$. (In such presentations the pair $\langle W, \{R_l\}_{l\in\mathcal{L}}\rangle$ is usually given a special name, namely *multiframe*.) But obviously there is no mathematical substance to this difference. Given a traditionally presented Kripke model $\langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$, we have that $\langle W, \{R_l\}_{l\in\mathcal{L}}, \{V(p) : p \in \text{VAR}\}\rangle$ is an AVS of signature $\langle\mathcal{L}, \text{VAR}\rangle$; and conversely, given any AVS $\langle W, \{R_l\}_{l\in\mathcal{L}}, \{Q_\alpha\}_{\alpha\in\mathcal{A}}\rangle$, we have that $\langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$ is a Kripke model, where V is the function from the set of (α-indexed) propositional variables VAR to $Pow(W)$ defined by $V(p_\alpha) = Q_\alpha$. In short, every AVS is a Kripke model, and vice versa.

Now for the syntactical correspondence. Consider the following AVM.

$$\begin{bmatrix} \text{AGREEMENT} & [\text{PERSON } \mathit{1st}] \\ \text{CASE} & \mathit{nominative} \end{bmatrix}$$

This corresponds to

$$\begin{array}{ll} & \langle\text{AGREEMENT}\rangle\langle\text{PERSON}\rangle \mathit{1st} \\ \wedge & \langle\text{CASE}\rangle \mathit{nominative} \end{array}$$

The key point to grasp is that the function of the attributes AGREEMENT, PERSON and CASE in the AVM is precisely analogous to the function of the existential modalities $\langle$AGREEMENT$\rangle$, $\langle$PERSON$\rangle$ and $\langle$CASE$\rangle$ in the modal wff. The function of the attributes is to demand the existence of certain transitions in AVSs, the function of the modalities is to demand the existence of certain transitions in Kripke models. But AVSs are just Kripke models, and thus the equivalence of the description languages is clear. The rest of the correspondence is straightforward: atomic values correspond to propositional symbols, and the modal wff is in effect just a linearisation of the AVM. To put it more generally, AVMs are just modal wffs written in a particularly perspicuous manner.

This correspondence extends in the obvious manner to AVMs with full Boolean expressivity. For example corresponding to the following AVM:

$$\begin{bmatrix} \text{NUMBER} & \neg \mathit{plural} \\ \text{CASE} & \{\mathit{nominative, genitive, accusative}\} \end{bmatrix}$$

we have the wff

$$\langle\text{NUMBER}\rangle\neg \mathit{plural} \wedge \langle\text{CASE}\rangle(\mathit{nominative} \vee \mathit{genitive} \vee \mathit{accusative}).$$

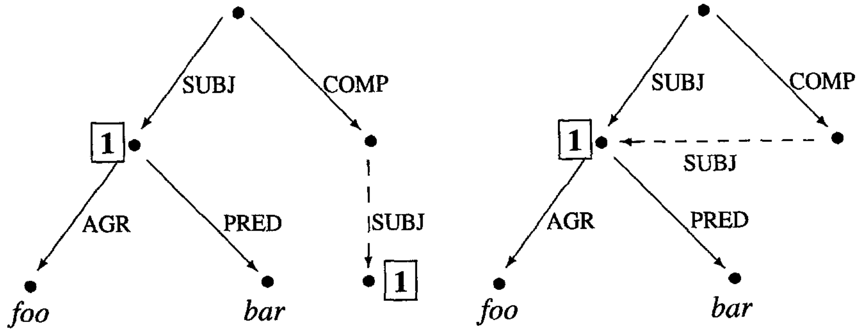
The most important aspect of the link between modal languages and AV formalisms is what it tells us about unification. Recall that unification is the attempt to coherently merge two AVMs. But what does 'coherent' mean? It means that the demands that the two AVMs make can be simultaneously satisfied at some node in some AVS. Now, both AVMs correspond to a modal wff. Call these two wffs ϕ and ψ respectively. Then we have that unification succeeds iff $\phi \wedge \psi$ is satisfiable at some node in some Kripke model. That is, *testing whether unification is possible amounts to testing for modal satisfiability.* This observation (familiar from the work of Kasper and Rounds (1990) and Kracht (1989)) lies at the heart of the paper.

The correspondence we have noted extends to richer unification formalisms than the rather simple AVMs so far considered. In particular, it extends to formalisms that have the ability to encode *re-entrancy*. Re-entrancy is a very influential idea in unification based approaches to grammar, and we need to discuss it, and how it can be dealt with in modal languages.

One of the best known notations for forcing re-entrancy is to use AVMs with 'boxlabels' (or 'tags'). Consider the following AVM:

$$\begin{bmatrix} \text{SUBJ} & \boxed{1}\begin{bmatrix}\text{AGR} & \textit{foo}\\ \text{PRED} & \textit{bar}\end{bmatrix}\\ \text{COMP} & [\text{SUBJ}\ \boxed{1}] \end{bmatrix}$$

The boxlabels are the $\boxed{1}$s. What is intended by this notation is explained by the following graphs:



(i) (ii)

The first graph does *not* satisfy the AVM at its root node. This is because $\boxed{1}$ is a name: it labels a *unique* node. The second graph *does* satisfy the AVM. The crucial difference is that in this graph the SUBJ *re-enters* the graph at the named node. Thus all the conditions demanded by the AVM are satisfied, including the demand that $\boxed{1}$ picks out a unique node.

How can we make modal languages referential in this way? The key idea needed can be traced back to early work by Arthur Prior (see Prior (1967)), and Robert Bull (see Bull (1970)): it is to introduce a second sort of atomic symbol constrained to be true at exactly one node. These new symbols 'name' the unique node they are true at. In this paper these symbols are called *nominals*, and they are usually written as i, j, k and m.

AVM boxlabels correspond straightforwardly to nominals. Consider once more the following AVM:

$$\begin{bmatrix} \text{SUBJ} & \boxed{\mathbf{1}}\begin{bmatrix}\text{AGR} & \textit{foo}\\ \text{PRED} & \textit{bar}\end{bmatrix} \\ \text{COMP} & [\text{SUBJ}\ \ \boxed{\mathbf{1}}] \end{bmatrix}$$

This corresponds to the following wff:

$$\begin{aligned} &\langle\text{SUBJ}\rangle(i \wedge \langle\text{AGR}\rangle \textit{foo} \wedge \langle\text{PRED}\rangle \textit{bar}) \\ \wedge\ &\langle\text{COMP}\rangle\langle\text{SUBJ}\rangle i \end{aligned}$$

Note that the nominal i is doing the same work in the modal wff that $\boxed{\mathbf{1}}$ does in the AVM. More generally, the use of nominals permits a transparent linearisation of those AVMs that utilise boxlabels.

Although AVM notation is widely used, it is certainly not the only notation computational linguists use to describe AVSs. Another influential notation arose from the command language of the PATR-II system (see Shieber (1986)). PATR-II is an 'implemented grammar formalism', a program which provides a high level interface language geared towards the needs of the linguist, together with a parser. The linguist writes grammars in the interface language and tests them using the parser. The use of *path equations* for specifying re-entrancy arose from this source. A user of PATR-II might write:

$$\langle \text{VP HEAD}\rangle = \langle\text{VP VERB HEAD}\rangle.$$

This path equation means that the sequence of transitions encoded by the list of attributes on the left takes one to the same node as the sequence of transitions encoded by the list of attributes on the right. That is, both transition sequences lead to the same node. Note that although this mechanism permits re-entrancy to be specified, it does so in a very different way from the 'boxlabels' approach: no node labelling is involved.

To capture the effect of this in a modal language, we're going to extend the basic language in such a way as to permit 'modal path equations' to be formed. In particular, we'll add a new primitive symbol $\approx$ to allow us to equate strings of modalities. This will permit wffs such as

$$\langle \text{VP} \rangle \langle \text{HEAD} \rangle \approx \langle \text{VP} \rangle \langle \text{VERB} \rangle \langle \text{HEAD} \rangle,$$

to be formed, and we will define the semantics of these new wffs so that they capture the meaning of the PATR-II path equations. Actually, we'll also add a second new primitive symbol, 0. This will be a name for the null transition, and with its help we will be able to write such path equations as $\langle b \rangle \langle a \rangle \approx 0$. This wff, for example, will mean that making an R_b transition followed by an R_a transition is the same as making the null transition. That is, the path $R_b R_a$ terminates at its starting point.

It should now be clear that various AV formalisms correspond straightforwardly to propositional modal languages. To conclude this section let's make our discussion of these modal languages more precise. Syntactically, the language L (of signature $\langle \mathcal{L}, \mathcal{A} \rangle$) is a language of propositional modal logic with an $\mathcal{L}$ indexed collection of distinct (existential) modalities and an $\mathcal{A}$ indexed collection of propositional symbols. As primitive Boolean symbols we choose $\neg$ and $\vee$. The wffs of the language are defined by saying that: (a) All propositional symbols p_α are wffs, for all $\alpha \in \mathcal{A}$; (b) If ϕ and ψ are wffs then so are $\neg\phi$, $\phi \vee \psi$, and $\langle l \rangle \phi$, for all $l \in \mathcal{L}$; (c) Nothing else is a wff. We define the other Boolean connectives $\rightarrow$, $\wedge$, $\leftrightarrow$, $\bot$, and $\top$ in the usual way. We also define $[l]\phi$ to be $\neg\langle l \rangle \neg\phi$, for all $l \in \mathcal{L}$ and all wffs ϕ. The following syntactic notions will be useful. The *degree* of a formula is the number of (primitive) connectives it contains. The *length* of a wff ϕ (denoted by $|\phi|$) is the number of (primitive) symbols it contains. (We will also use the '$|\,.\,|$' notation to indicate cardinality, but this double use should cause no confusion.)

To interpret L we use Kripke models M of signature $\langle \mathcal{L}, \mathcal{A} \rangle$. Such a Kripke model is a triple $\langle W, \{R_l\}_{l \in \mathcal{L}}, V \rangle$, where W is a non-empty set (the set of nodes); each R_l is a binary relation on W that is also a partial function, that is, for every node w there exists at most one w' such that wR_lw'; and V (the valuation) is a function which assigns each propositional symbol p_α a subset of W. We interpret wffs of L on models M in the familiar fashion:

$$\begin{array}{lll} \mathrm{M} \models p_\alpha[w] & \text{iff} & w \in V(p_\alpha) \\ \mathrm{M} \models \neg\phi[w] & \text{iff} & \mathrm{M} \not\models \phi[w] \\ \mathrm{M} \models \phi \vee \psi[w] & \text{iff} & \mathrm{M} \models \phi[w] \text{ or } \mathrm{M} \models \psi[w] \\ \mathrm{M} \models \langle l \rangle \phi[w] & \text{iff} & \exists w'(wR_lw' \mathbin{\&} \mathrm{M} \models \phi[w']) \end{array}$$

If $\mathrm{M} \models \phi[w]$ then we say that M *satisfies* ϕ at w, or ϕ is *true* in M at w. To sum up, the language L corresponds to the 'core' AVM notation used by computational linguists. Its models are just AVSs, and the way L formulas are evaluated in a model is just the way AVMs are checked against AVSs.

L lacks any mechanism for enforcing re-entrancy. This lack is made good in its extensions, L^N and L^{KR}. The language L^N (of signature $\langle \mathcal{L}, \mathcal{A}, \mathcal{B} \rangle$) is the language L (of signature $\langle \mathcal{L}, \mathcal{A} \rangle$) augmented by a $\mathcal{B}$ indexed collection of distinct new propositional symbols called *nominals*. These symbols are typically written as i, j, k and m and can be freely combined with the other symbols in the usual fashion to make wffs. We assume that $\mathcal{B}$ is at most countably infinite. To interpret nominals we insist that any valuation must assign a singleton subset to each nominal. That is, an L^N model is just an L model whose valuation has been extended to assign singletons to nominals. Because each nominal is thus true at exactly one node in any model, it acts as a 'name' identifying that node. L^N corresponds to AVMs augmented with 'boxlabels' for indicating re-entrancy. There have been a number of logical investigations of intensional languages containing nominals. In addition to the early work by Prior and Bull already mentioned, see Passy and Tinchev (1985), Gargov and Passy (1988) and Passy and Tinchev (1991) for an examination of nominals in the setting of Propositional Dynamic Logic (PDL); see Gargov, Passy and Tinchev (1986) and Gargov and Goranko (1989) for nominals in the setting of modal logic; and finally see Blackburn (1989) for nominals in tense logic.

The language L^{KR} is L augmented by two new symbols, 0 and $\approx$. The symbol 0 acts as a name for the null transition. In what follows we shall assume without loss of generality that $0 \notin \mathcal{L}$, and denote the identity relation on any set of nodes W by R_0. (This convention simplifies the statement of the following truth definition.) We use $\approx$ to make *path equations*: given any nonempty sequences A and B made up of modalities and $\langle 0 \rangle$, then $A \approx B$ is a path equation. Path equations are wffs and can be combined with other wffs in the usual way to make more complex wffs. L^{KR} models are just L models, and we interpret the path equations as follows. For all $l_1, \ldots, l_k, l'_1, \ldots, l'_m \in \mathcal{L} \cup \{0\}$:

$$\mathrm{M} \models \langle l_1 \rangle \cdots \langle l_k \rangle \approx \langle l'_1 \rangle \cdots \langle l'_m \rangle [w] \text{ iff } \exists w' (w R_{l_1} \ldots R_{l_k} w' \mathbin{\&} w R_{l'_1} \ldots R_{l'_m} w').$$

L^{KR} models the path equation mechanism of PATR-II. The negation free fragment of this language was first defined and studied by Kasper and Rounds (1986) and Rounds and Kasper (1986); a more detailed presentation of their work may be found in Kasper and Rounds (1990). Further logical investigations of L^{KR} may be found in Moss (1991) and Blackburn (1991).

It is instructive (and will later prove technically useful) to examine L, L^N and L^{KR} from the more general perspective provided by *modal correspondence theory*. This subject is the systematic study and exploitation of the relationships that exist between modal languages and various classical languages; an excellent overview is van Benthem (1984). The correspondence between L, L^N and L^{KR} and first order logic arises as follows. Note that AVSs (that is, Kripke models) can equally well be regarded as models for a certain first order language, namely the first order language (with equality) that contains a binary relation symbol F_l for each R_l, and a unary relation symbol P_α for each Q_α; we will call this language L^1. There is an obvious translation from our modal languages to L^1, the *standard translation.* These are the clauses for L:

$$\begin{array}{lcl} ST(p_\alpha) & = & P_\alpha x \\ ST(\neg\phi) & = & \neg ST(\phi) \\ ST(\phi \vee \psi) & = & ST(\phi) \vee ST(\psi) \\ ST(\langle l \rangle \phi) & = & \exists y(xR_l y \wedge [y/x]ST(\phi)) \end{array}$$

Here x is the first order variable that represents the evaluation node, and the $[y/x]$ in the final clause means substitute y for all free occurrences of x, where y is some fresh first order variable. Note that the standard translation is essentially another way of looking at the satisfiability definition for L, thus it is clear that the standard translation is truth preserving: that is, M $\models \phi[w]$ iff M $\models ST(\phi)[w]$. Note that on the left hand side of this equivalence $\models$ and $[w]$ are read modally (that is, in accordance with the satisfiability definition for L given above) whereas on the right hand side these symbols have their standard first order meaning. The standard translation shows that L can be regarded as a very simple fragment of L^1, namely a one-free-variable fragment in which only bounded quantification is used.

L^1 is also the first order correspondence language for both L^N and L^{KR}. To see this note that we can extend the standard translation to L^N by adding the following clause:

$$ST(i) = (x = x_i).$$

Again x is the first order variable that picks out the point of evaluation, and x_i is the first order variable that we have chosen to correspond to the nominal i. Similarly, we can extend the standard translation L^{KR} by adding the clause:

$$ST(\langle l_1 \rangle \cdots \langle l_k \rangle \approx \langle l'_1 \rangle \cdots \langle l'_m \rangle) = \exists y(xR_{l_1} \ldots R_{l_k} y \wedge xR_{l'_1} \ldots R_{l'_m} y).$$

Both extensions are truth preserving, thus the use of nominals can be seen as the use of certain extra equalities, while the use of $\approx$ is essentially the use of an additional form of bounded quantification. Thus all three of our base languages are rather small fragments of L^1.

These observations immediately link the modal approach of this paper with other approaches to Attribute Value logic which may more familiar to the reader. Note in particular that the standard translation links our approach with that of Smolka (1989). Smolka was perhaps the first person to make explicit the connection between AVSs and first order models, and he has proved a number of results concerning a certain first order language of AVSs, namely the language we have here called L^1. Thus, via correspondence theory, many of the results of the present paper can be seen as an investigation of the complexity of certain fragments of Smolka's language; this includes the results concerning the yet to be introduced universal modality. However the word 'many' is important. Modal operators *aren't* restricted to having first order correspondences, and when we later consider the master modality we will in effect be working with a small fragment of infinitary logic.

This completes our discussion of the theoretical background of the paper. Let's now turn to the issue of most immediate relevance to computational linguistics: the complexity of various satisfiability problems. As most AV grammar formalisms assume a finite collection of both attributes and atomic symbols, the key problem is the satisfiability problem for languages of signature $\langle \mathcal{L}, \mathcal{A} \rangle$ where both $\mathcal{L}$ and $\mathcal{A}$ are *finite*. Actually, with one interesting exception, our results are insensitive to the cardinality of $\mathcal{L}$ for $|\mathcal{L}| \geq 2$, however when we treat the richer languages involving the universal or master modalities extra work is required to show that our results go through for the case of $\mathcal{A}$ finite. In order to minimise the work involved we shall proceed as follows. We will first prove results which hold for languages with $|\mathcal{L}| \geq 2$ and $\mathcal{A}$ countably infinite; this allows natural proofs to be given. Later on a very general result is proved *(the Single Variable Reduction Theorem)* which allows all these results to be sharpened to cover languages containing only one propositional variable p. (In fact, in order to give a complete classification of the problem we're even going to show that our results hold for $|\mathcal{L}| \geq 2$ when no propositional variables at all are used; all one needs is a primitive truth symbol $\top$. We will call languages with a primitive $\top$ symbol and no propositional variables languages of signature $\langle \mathcal{L}, \emptyset \rangle$.) Finally, we know of no linguistic theory which puts a fixed finite upper bound on the number of boxlabels that may be used, thus for languages with nominals the complexity of the satisfiability problem when $\mathcal{B}$ is countably infinite is the most important.

3. COMPLEXITY RESULTS FOR L, L^N and L^{KR}

In this section we show that the satisfiability problems for L, L^N and L^{KR} are all NP complete. The fundamental result is that for L, for it turns out that the method used for this language generalises straightforwardly to its two extensions. The key to the NP completeness result for L is to show that given a formula ϕ which is satisfiable at a node w in some model M, we can always find a suitably small model $\mathrm{M}|\mathit{nodes}(\phi, w)$ which also satisfies ϕ. Once we have defined $\mathrm{M}|\mathit{nodes}(\phi, w)$ and determined its size the NP completeness result is immediate.

The definition of $\mathrm{M}|\mathit{nodes}(\phi, w)$ follows from the following general property of modal languages: when evaluating a wff in a model, only a certain selection of the model's nodes are actually relevant to the truth or falsity of the wff; all other nodes can be discarded. The nodes that are relevant when evaluating a wff ϕ at a node w in a model M are the nodes picked out by the function *nodes* : WFF × $W \longrightarrow Pow(W)$ that satisfies the following conditions:

$$\begin{aligned}
\mathit{nodes}(p, w) &= \{w\} \\
\mathit{nodes}(\neg\phi, w) &= \mathit{nodes}(\phi, w) \\
\mathit{nodes}(\phi \vee \psi, w) &= \mathit{nodes}(\phi, w) \cup \mathit{nodes}(\psi, w) \\
\mathit{nodes}(\langle l \rangle\phi, w) &= \{w\} \cup \textstyle\bigcup_{w' : wR_l w'} \mathit{nodes}(\phi, w')
\end{aligned}$$

Given a model M, a wff ϕ and a node w we form $\mathrm{M}|\mathit{nodes}(\phi, w)$ in the obvious way: the nodes of this model are $\mathit{nodes}(\phi, w)$, and the relations and valuation are the restriction of those of M to this set. The following lemma shows that *nodes* selects the correct nodes:

LEMMA 3.1 (Selection Lemma). *For all models* **M**, *all nodes w of* **M** *and all wffs ϕ,*

$$\mathbf{M} \models \phi[w] \ \textit{iff}\ \mathbf{M}|\mathit{nodes}(\phi, w) \models \phi[w].$$

Proof: By induction on the degree of ϕ. Note that it follows from the definition of *nodes* that $w \in \mathit{nodes}(\phi, w)$, which is all that is needed to drive the induction through. □

The selection lemma is a completely general fact about modal languages. It *doesn't* depend on any assumptions we have made in this paper; in particular we haven't yet made use of the fact that we're only concerned with models in which each of the R_l is a partial function. However when we take this into account we notice that $\mathrm{M}|\mathit{nodes}(\phi, w)$ has a pleasant property: it is very small. There can only be one more node in $\mathrm{M}|\mathit{nodes}(\phi, w)$ than there are occurrences of modalities in ϕ.

LEMMA 3.2 (Size Lemma). *Let $\mathit{mod}(\phi)$ be the number of occurrences of modalities in ϕ. Then for all models* **M** *and all nodes w in* **M**, $|\mathit{nodes}(\phi, w) \backslash \{w\}| \leq \mathit{mod}(\phi)$.

Proof: By induction on the degree of ϕ. For the base case note that for all atomic formulas p we have that $|nodes(p, w)\backslash\{w\}| = \emptyset$, thus the result holds. So assume the result for all wffs of degree less than k. Now if ϕ is a wff of degree k of the form $\psi \vee \theta$ then we have:

$$\begin{aligned} |nodes(\psi \vee \theta, w)\backslash\{w\}| &\leq |nodes(\psi, w)\backslash\{w\}| + |nodes(\theta, w)\backslash\{w\}| \\ &\leq mod(\psi) + mod(\theta) \quad \text{(by Inductive Hypothesis)} \\ &= mod(\psi \vee \theta). \end{aligned}$$

Thus the required result holds for disjunctions. The case for negations is similar.

There only remains the case for modalities, so suppose that ϕ is a wff of degree k of the form $\langle l \rangle\psi$. We wish to show that $|nodes(\langle l \rangle\psi, w)\backslash\{w\}| \leq mod(\langle l \rangle\psi)$. There are two cases to consider. The first is that there there are no nodes w' such that wR_lw'. But then $|nodes(\langle l \rangle\psi, w)\backslash\{w\}| = \emptyset$ and the result is immediate. So next consider the case when there is a node w' such that wR_lw'. *Note that as we are working with partial functional relations this w' must be unique.* Thus we have the following:

$$\begin{aligned} |nodes(\langle l \rangle\psi, w)\backslash\{w\}| &\leq |nodes(\psi, w')| \\ &\leq |(nodes(\psi, w')\backslash\{w'\}) \cup \{w'\}| \\ &\leq |nodes(\psi, w')\backslash\{w'\}| + 1 \\ &\leq mod(\psi) + 1 \quad \text{(by Inductive Hypothesis)} \\ &= mod(\langle l \rangle\psi) \end{aligned}$$

Thus the required result also holds for modalities, and hence the truth of the lemma follows by induction. □

Together the selection lemma and the size lemma lead directly to the main result:

THEOREM 3.1. *Let L be a language of signature $\langle \mathcal{L}, \mathcal{A} \rangle$ where $\mathcal{A}$ is countably infinite. Then the satisfiability problem for L is NP complete.*

Proof: That this satisfiability problem is NP hard is clear, for as we have a countably infinite collection of propositional variables at our disposal the problem contains the satisfiability problem for propositional calculus as a special case. That the problem is in NP follows directly from the fact that any satisfiable L wff ϕ can be satisfied in a model containing at most $mod(\phi) + 1$ nodes; this we know from the selection and size lemmas. Thus, given ϕ we can non-deterministically choose a suitable model of at most this size, and evaluate ϕ in this model in polynomial time. □

Let's turn to the complexity of the satisfiability problem for the language L^N. Recall that this language is L augmented by a distinct new set of atomic symbols called nominals which are constrained to be true at exactly one node in any model. It is easy to use the machinery developed above to prove that the satisfiability problem for L^N is also NP complete, in fact there is almost nothing new to be done. Given a L^N model M, a node w in M, and an L^N wff ϕ we define M|*nodes*(ϕ, w) exactly as described above. Both the selection and size lemmas hold, thus we are almost through. There is only one snag: M|*nodes*(ϕ, w) is not guaranteed to be an L^N model as some nominals may be not denote any node at all. But this problem is more apparent than real. By adjoining a brand new node (say $*$) to M|*nodes*(ϕ, w) and insisting that all 'unassigned nominals' denote $*$ we convert M|*nodes*(ϕ, w) into an L^N model [M|*nodes*(ϕ, w)]*. Of course to maintain the truth of the selection lemma we have to be careful where we place $*$, but there are two obvious 'safe' choices. The simplest choice is to insist that $*$ is unrelated (by any of the relations) to any of the points in M|*nodes*(ϕ, w). The second, which is slightly more elegant, is to insist that $*$ is related to w by some relation, but that none of the points in S is related to $*$; choosing this second option means that $*$ point generates [M|*nodes*(ϕ, w)]*. Either way it it clear that the addition of $*$ is harmless: we still have that that [M|*nodes*(ϕ, w)]* $\models \phi[w]$. And [M|*nodes*(ϕ, w)]* is still small, having at most $mod(\phi) + 2$ nodes. Thus by precisely the same argument as for L we have:

THEOREM 3.2. *Let L^N be a language with nominals of signature $\langle \mathcal{L}, \mathcal{A}, \mathcal{B} \rangle$, where both $\mathcal{A}$ and $\mathcal{B}$ are countably infinite. Then the satisfiability problem for L^N is NP complete.* □

Finally we turn to L^{KR}. The satisfiability problem for this language is also NP complete, but how are we to show this? Our definition of *nodes* says nothing about occurrences of path equations. Actually the easiest way to proceed is not to extend the definition of *nodes*, but rather to first transform L^{KR} wffs into a certain special form. The following example shows what is involved.

Suppose we have a model M which verifies $\langle a \rangle \approx \langle b \rangle$ at a node w. This means there is a node w' such that $wR_a w'$ and $wR_b w'$. But as *nodes*$(\langle a \rangle \approx \langle b \rangle, w)$ is undefined, in general we will not have that w' is a part of the small model we build. However if we first rewrite $\langle a \rangle \approx \langle b \rangle$ into a logically equivalent form that makes explicit the existential demands of the path equations, everything proceeds smoothly. Rewrite $\langle a \rangle \approx \langle b \rangle$ as $\langle a \rangle \approx \langle b \rangle \wedge \langle a \rangle \top \wedge \langle b \rangle \top$. Clearly this formula is logically equivalent to the original, however the new syntactic form is very useful: the two new conjuncts make the the modalities $\langle a \rangle$ and $\langle b \rangle$ available to *nodes*. Consider what happens when we apply *nodes* to this new formula at w. As *nodes* commutes over $\wedge$, we must calculate *nodes*$(\langle a\rangle \approx \langle b\rangle, w)$, *nodes*$(\langle a\rangle\top, w)$ and *nodes*$(\langle b\rangle\top, w)$. As before, we can't do anything further with *nodes*$(\langle a\rangle \approx \langle b\rangle, w)$, but we can evaluate both *nodes*$(\langle a\rangle\top, w)$ and *nodes*$(\langle b\rangle\top, w)$, as *nodes* is defined for such expressions. Evaluating these formulas will produce the point w' that we need to build an equivalent small model.

Let's make this precise. Any path equation $\langle A\rangle \approx \langle B\rangle$ is logically equivalent to $\langle A\rangle \approx \langle B\rangle \wedge \langle A\rangle\top \wedge \langle B\rangle\top$. For any path equation $\langle A\rangle \approx \langle B\rangle$ we'll call $\langle A\rangle \approx \langle B\rangle \wedge \langle A\rangle\top \wedge \langle B\rangle\top$ its *explicit form.* Given an L^{KR} wff ϕ which we seek to satisfy, we'll first form a new L^{KR} wff ϕ^* by simultaneously substituting, for each occurrence of a path equation in ϕ, its explicit form. Note that ϕ^* is logically equivalent to ϕ, and that the length of ϕ^* is linear in the length of ϕ. The effect of this rewriting of ϕ means that our existing definition of *nodes* suffices to produce all the points needed for the small model: precisely as illustrated in the above example, when we apply *nodes* the occurrences of the new subformulas of the form $\langle A\rangle\top$ and $\langle B\rangle\top$ ensure that all the needed evaluation points are selected. Thus we can make M$|$*nodes*(ϕ, w) as before and both the selection and size lemmas hold. So, by exactly the same argument we have that:

THEOREM 3.3. *Let L^{KR} be a Kasper Rounds language that has a signature $\langle \mathcal{L}, \mathcal{A}\rangle$ where $\mathcal{A}$ is countably infinite. Then the satisfiability problem for L^{KR} is NP complete.* □

In the above proofs was assumed that we had a countably infinite supply of atomic symbols at our disposal. However most Attribute Value formalism use a finite number of atomic symbols. Given that the number of atomic symbols is some fixed finite number, might this not permit us to evade the NP hardness result? (As is well known, for both propositional logic and for S5, such a restriction lowers the complexity of the satisfiability problem to P.) However this is *not* the case here: the satisfiability problem for L (and thus for L^N and L^{KR}) remains NP hard, even if we use only one propositional variable, and one modal operator. This can be seen as follows. Consider the following set of L formulas: $\{p, \langle a\rangle p, \langle a\rangle\langle a\rangle p, \ldots, \langle a\rangle^k p\}$. The values of these formulas are all independent, that is, for any sequence of truth values $b_0, \ldots, b_k$, there exists a model such that M $\models \langle a\rangle^i p$ iff b_i is true. Now define function f from propositional formulas to L-formulas as follows:

$$f(\phi(p_0, \ldots, p_k)) = \phi(p, \langle a\rangle p, \langle a\rangle\langle a\rangle p, \ldots, \langle a\rangle^k p).$$

Obviously, f is polynomial time computable, and ϕ is satisfiable iff $f(\phi)$ is L satisfiable. Thus, we can summarise the complexity results of this section as follows:

THEOREM 3.4. *If* $|\mathcal{L}| \geq 1$ *and* $|\mathcal{A}| \geq 1$, *the satisfiability problems for* L, L^N, *and* L^{KR} *are NP complete.* □

Actually, if we look at the previous encoding carefully, we can see that if our language contains at least two modalities, we don't need any propositional variables to encode propositional satisfiability in an L formula; all we need is a primitive constant truth symbol $\top$. Define:

$$f(\phi(p_0, \ldots, p_k)) = \phi(\langle b\rangle\top, \langle a\rangle\langle b\rangle\top, \langle a\rangle\langle a\rangle\langle b\rangle\top, \ldots, \langle a\rangle^k\langle b\rangle\top).$$

Obviously, f is polynomial time computable, and ϕ is satisfiable iff $f(\phi)$ is L satisfiable, which leads to the following theorem:

THEOREM 3.5. *If* $|\mathcal{L}| \geq 2$ *and* $|\mathcal{A}| \geq 0$, *the satisfiability problems for* L, L^N, *and* L^{KR} *are NP complete.* □

Let's summarise our results so far. The satisfiability problem for the core AV language L is NP complete. Adding either of two re-entrancy forcing mechanisms — nominals or the Kasper Rounds path equality — does not increase the complexity: satisfiability remains NP complete. These results hold even if we have only one modal operator and one atomic symbol at our disposal. There is a result from the literature worth noting here: Kasper and Rounds (1990) show, using a disjunctive normal form argument, that when attention is confined to those models in which a) each atom is true at at most one node, b) no two atoms are true at the same node, and c) atoms are true only at terminal nodes, then the satisfiability problem for the negation free fragment of L^{KR} is NP hard (and in fact NP complete). The interesting part of their result is the NP hardness part, for as their language lacks negation this is not obvious. The non-trivial part of our result, on the other hand, is our model theoretic proof that an NP time algorithm exists even if full Boolean expressivity is allowed.

What can be said at a more general level about these results? From the point of view of modal logic they're somewhat unexpected: with the exception of S5 most familiar modal logics are PSPACE complete. To put it loosely, usually adding modalities to a language of propositional logic makes matters worse, but here it hasn't. The reason, of course, is due to the fundamental constraint on our models, namely that all the relations be partial functional. It's this requirement which enabled us to build small models and thus kept the complexity to that of propositional logic. It's worth adding that this constraint seems to be peculiar to the representational formalisms used in computational linguistics. Various representation formalisms used in AI, such as KL-ONE, can be viewed from a modal perspective, and as Schild (1990) has recently observed, terminological logics are also modal logics. But from the point of view of complexity there is a difference: the modal logics inspired by AI typically don't usually obey the partial functionality constraint. Usually they are multimodal versions of K, the modal logic which puts no constraints on accessibility relations. As is well known, the satisfiability problem for this logic is PSPACE complete (see Ladner (1977)).

4. THE UNIVERSAL MODALITY

In this section we are going to examine the complexity of the satisfiability problems for three stronger modal languages, $L^{\Box}$, $L^{N\Box}$ and $L^{KR\Box}$. These languages are, respectively, L, L^{N} and L^{KR} augmented by the *universal modality*. The universal modality is a modal operator written as $\Box$ which has the following semantics: for all models $\mathbf{M}$, all nodes w, and all wffs ϕ

$$\mathbf{M} \models \Box\phi[w] \text{ iff } \mathbf{M} \models \phi[w'], \text{ for all nodes } w' \text{ in } \mathbf{M}.$$

That is, $\Box\phi$ holds iff ϕ is true at all nodes. Note that all three enriched languages are fragments of L^1, the first order language of AVSs, as adding the following (truth preserving) clause to the standard translation correctly deals with occurrences of the universal modality:

$$ST(\Box\phi) = \forall y([y/x]ST(\phi)).$$

For a detailed discussion of the logical consequences of enriching modal languages with the universal modality, see Goranko *et al.* (1992). The authors know of only one explicit application of the universal modality to linguistic theorising, namely Evan's (1987) analysis of the feature specification defaults of GPSG, which we shall consider shortly. However, as we shall see, the universal modality seems to have been implicitly used on other occasions.

But why should linguists be interested in $L^{\Box}$, $L^{N\Box}$ and $L^{KR\Box}$? One answer is as follows. Underlying much work in Attribute Value grammar is an idea that can loosely be described as 'grammar equals feature logic'. Somewhat more precisely, the use of the apparatus of unification formalisms is attractive to many linguists because it enables them to view grammars of natural languages as theories in some sort of calculus of attributes and values. According to such a view, linguistic structure can be adequately modelled by Attribute Value Structures (possibly augmented by the notion of phrase structure), and the linguists' business is to state general constraints about which AVSs are admissible. Such views are discernible in some of the earliest work in attribute value grammar, namely Lexical Functional Grammar (LFG) (see Kaplan and Bresnan (1982)). Generalised Phrase Structure Grammar (GPSG) (see Gazdar *et al.* (1985)), explicitly espouses such views, and its work on feature co-occurrence restrictions remains one of the best examples of the approach in action. More recently, Head Driven Phrase Structure Grammar (HPSG) (see Pollard and Sag (1987)), has taken this approach even further. Whereas in both GPSG and LFG the idea of unification was only one component (albeit an important one) of the systems, in HPSG the unificational apparatus completely dominates.

It is these ideas that motivate the work of the present section. As we have seen the most common unificational formalisms are nothing but modal languages. However as they stand these languages aren't strong enough to express generalisations, and indeed as the 'grammar equals feature logic' equation has become more widely accepted, work in Attribute Value grammar has tended to abandon the simple languages we have considered so far in favour of increasingly powerful formalisms. The work of this section is an exploration of the computational consequences of adding just enough power to the base languages to enable generalisations to be expressed.

Let's consider matters more concretely. Suppose we strengthen our languages by adding the universal modality: what linguistic principles can we now express? Consider a typical GPSG feature co-occurrence restriction, for example:

$$[\text{VFORM FIN}] \Rightarrow [\text{-N}, \text{+V}].$$

This states that if a node has the value FIN for the attribute VFORM, then that node has the properties of being -N and +V. In other words, only a verb can have tense.

The important thing about this constraint is its generality. It's not something which happens to hold of this or that piece of linguistic structure, it's a pervasive fact of English. Any AVS which doesn't satisfy this generalisation can't represent an English sentence. We can express this generalisation in $L^{\Box}$ as follows:

$$\Box(\langle \text{VFORM} \rangle fin \rightarrow -n \wedge +v).$$

(Here *fin*, $-n$ and $+v$ are propositional symbols and $\langle \text{VFORM} \rangle$ is a modality.) In short we can view the $\Rightarrow$ notation of GPSG as what modal logicians have traditionally called *strict implication*. Viewing $\phi \Rightarrow \psi$ in this way accounts for the generality of feature co-occurrence restrictions.

Evans (1987) also makes use of the universal modality in connection with GPSG, but to express defaults, not generalisations. Evans uses $L^{\Box}$ and mostly works with the dual of the universal modality ($\Diamond\phi = \neg\Box\neg\phi$), which he gives an autoepistemic reading: $\Diamond\phi$ means that ϕ is consistent with all known information. For example he uses the wff $\Diamond\langle \text{CASE} \rangle dat \rightarrow \langle \text{CASE} \rangle dat$

to express the feature specification default: 'If it is consistent with all known information that case is dative, then case is dative'. The idea of using a modal operator to express linguistic defaults is interesting, though we would argue that such an operator would need to be added in addition to the generalisation expressing universal modality. But this is to argue over details. There are many ideas worth pursuing in Evans work, and the underlying philosophy is in harmony with that of the present paper: indeed in a footnote Evans raises the possibility of formalising all of GPSG in a modal language.

Let's consider the use of $L^{KR\Box}$. This language is powerful enough to capture the content of the Head Feature Convention of GPSG (or indeed HPSG). The essence of the GPSG version is that for any phrasal constituent, the value of its head attribute is shared with the value of the head attribute of its head daughter. For a discussion of what this terminology means, and why it's linguistically useful the reader is referred to Gazdar (1985); here we'll be content to indicate how the constraint can be expressed:

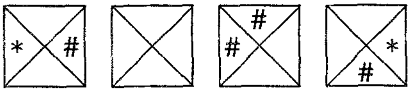
$$\Box(\mathit{phrasal} \rightarrow \langle \text{HEAD} \rangle \approx \langle \text{HEAD-DTR} \rangle \langle \text{HEAD} \rangle).$$

Once again note that this is a strict implication; we could rewrite it as:

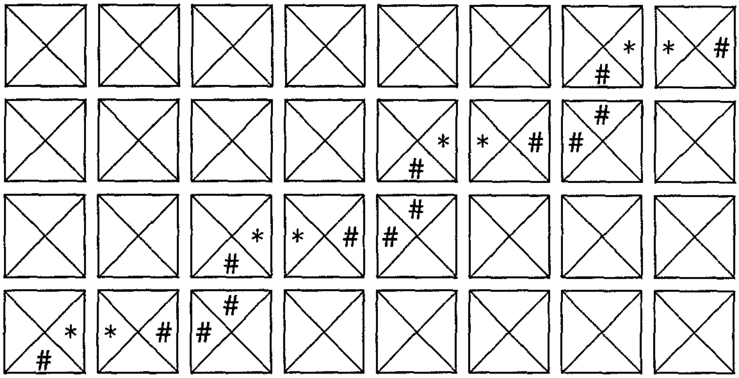
$$\mathit{phrasal} \Rightarrow \langle \text{HEAD} \rangle \approx \langle \text{HEAD-DTR} \rangle \langle \text{HEAD} \rangle.$$

Further experimentation will convince the reader that $L^{KR\Box}$ is a language capable of expressing interesting linguistic constraints. However it has also crossed an important complexity boundary; as we shall now show its satisfiability problem is undecidable. To prove the undecidability result it suffices to give a reduction from a Π^0_1 hard problem to $L^{KR\Box}$ satisfiability. As is shown in Harel (1983), *tiling problems* provide a particularly elegant method of proving lower bounds for modal logics, so we'll use such an approach here.

A tile T is a 1×1 square fixed in orientation with coloured edges *right*(T), *left*(T), *up*(T), and *down*(T) taken from some denumerable set. A tiling problem takes the following form: given a finite set of $\mathcal{T}$ of tile types, can we cover a certain part of $\mathbf{Z} \times \mathbf{Z}$, using only tiles of this type, in such a way that adjacent tiles have the same colour on the common edge, and such that the tiling obeys certain constraints? One of the attractive features of tiling problems is that they are very easy to visualise. As an example, consider the following puzzle. Suppose $\mathcal{T}$ consists of the following four types of tile:



Can an 8 by 4 rectangle be tiled with the fourth type placed in the left hand corner? Indeed it can:



There exist complete tiling problems for many complexity classes. In the proof that follows we make use of a certain Π^0_1 complete tiling problem.

THEOREM 4.1. *If* $|\mathcal{L}| \geq 2$, *and* $\mathcal{A}$ *is countably infinite then the satisfiability problem for* $L^{KR\Box}$ *is* Π^0_1 *hard.*

Proof: As shown in Berger (1966) and Robinson (1971), the following problem is Π^0_1 complete:

N × N tiling: Given a finite set $\mathcal{T}$ of tiles, can $\mathcal{T}$ tile **N** × **N**?

That is, does there exist a function t from **N** × **N** to $\mathcal{T}$ such that:

$$\begin{aligned} \mathit{right}(t(n,m)) &= \mathit{left}(t(n+1,m)), \text{ and} \\ \mathit{up}(t(n,m)) &= \mathit{down}(t(n,m+1))? \end{aligned}$$

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a set of tiles. We construct a formula ϕ such that:

$\mathcal{T}$ tiles **N** × **N** iff ϕ is satisfiable.

First of all we will ensure that, if ϕ is satisfiable in M, then M contains a gridlike structure, the nodes of M (henceforth W) play the role of points in a grid, R_r is the right successor relation, and R_u is the upward successor relation. Define:

$$\phi_{grid} = \Box(\langle r\rangle\langle u\rangle \approx \langle u\rangle\langle r\rangle).$$

Suppose M $\models \phi_{grid}[w_0]$. Then there exists a function f from $\mathbf{N} \times \mathbf{N}$ to W such that: $f(0,0) = w_0$, $f(n,m)R_r f(n+1,m)$, and $f(n,m)R_u f(n,m+1)$.

Next we must tile the model. To do this we use propositional variables $t_1, \ldots, t_k$, such that t_i is true at some node w, iff tile T_i is placed at w. To force a proper tiling, we need to satisfy the following three requirements:

1. There is exactly one tile placed at each node.
$$\phi_1 = \Box(\bigvee_{i=1}^{k} t_i \wedge \bigwedge_{1 \leq i < j \leq k} \neg(t_i \wedge t_j))$$
2. If T is the tile at w, and T' the tile at the right successor of w, then *right*(T) = *left*(T').
$$\phi_2 = \Box(\bigvee_{\mathit{right}(T_i) = \mathit{left}(T_j)} (t_i \wedge \langle r \rangle t_j))$$
3. If T is the tile at w, and T' the tile at the up successor of w, then *up*(T) = *down*(T').
$$\phi_3 = \Box(\bigvee_{\mathit{up}(T_i) = \mathit{down}(T_j)} (t_i \wedge \langle u \rangle t_j))$$

Putting this all together, we define ϕ to be $\phi_{grid} \wedge \phi_1 \wedge \phi_2 \wedge \phi_3$. We will prove that $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N}$ iff ϕ is satisfiable.

First suppose $t : \mathbf{N} \times \mathbf{N} \rightarrow \mathcal{T}$ is a tiling of $\mathbf{N} \times \mathbf{N}$. We construct the satisfying model for ϕ as follows: M $= \langle W, R_r, R_u, V \rangle$ such that:

$$\begin{array}{lcl} W & = & \{w_{n,m} : n, m \in \mathbf{N}\} \\ R_r & = & \{\langle w_{n,m}, w_{n,m+1} \rangle : n, m \in \mathbf{N}\} \\ R_u & = & \{\langle w_{n,m}, w_{n+1,m} \rangle : n, m \in \mathbf{N}\} \\ V(t_i) & = & \{w_{n,m} : n, m \in \mathbf{N} \text{ and } t(n,m) = T_i\} \end{array}$$

Clearly, ϕ holds at any node w of M. To see that the converse also holds, suppose that M $\models \phi[w_0]$. Let f from $\mathbf{N} \times \mathbf{N}$ to W be such that $f(0,0) = w_0$, $f(n,m)R_r f(n+1,m)$ and $f(n,m)R_u f(n,m+1)$. Define the tiling $t : \mathbf{N} \times \mathbf{N} \rightarrow \mathcal{T}$ by $t(n,m) = T_i$ iff M $\models t_i[f(n,m)]$. Note that t is well-defined and total by ϕ_1. Furthermore, if $t(n,m) = T_i$ and $t(n+1,m) = T_j$, then $f(n,m)R_r f(n,m+1)$, M $\models t_i[f(n,m)]$, and M $\models t_j[f(n,m+1)]$. Since M satisfies ϕ_2, we can conclude that *right*(T_i) = *left*(T_j). Similarly, if $t(n,m) = T_i$ and $t(n,m+1) = T_j$, then ϕ_3 ensures that *up*(T_i) = *down*(T_j). Thus, $\mathcal{T}$ tiles $\mathbf{N} \times \mathbf{N}$. □

Thus the satisfiability problem for $L^{KR\Box}$ is undecidable. Note, however, that the above proof depends on having access to an unlimited supply of propositional variables. (The above argument shows how any tiling problem can be reduced to $L^{KR\Box}$ satisfiability by representing tiles as propositional symbols. But there is no pre-determined size limit on the set of tiles $\mathcal{T}$ that we may be given.) This problem will be dealt with later.

The satisfiability problem for $L^{KR\Box}$ is in fact Π^0_1 complete. Given the previous result, all we need to to show is that the $L^{KR\Box}$ validities can be recursively enumerated. One way of doing this is to give a recursive axiomatisation of $L^{KR\Box}$. This can be done by building on the completeness proof for L^{KR} given in Blackburn (1991), but it has the drawback of requiring the introduction of the (otherwise irrelevant) machinery of modal completeness theory. Fortunately correspondence theory comes to the rescue with a general argument showing (at least for the case of finite $\mathcal{L}$) that $L^{KR\Box}$ validity is a r.e. notion. The argument is due to van Benthem (1984) who observes that when working with *elementary* classes of frames (that is, frames defined by a single L^1 formula) it is not necessary to give an explicit axiomatisation to show that modal validity is r.e.: if φ is the L^1 wff that defines the elementary class, and if ϕ is a modal formula such that $ST(\phi) \in L^1$ then ϕ is a validity iff $\varphi \models \forall x ST(\phi)$. But here '$\models$' denotes the first order consequence relation, and as this is an r.e. relation we would be through if we could show that the multiframes underlying our Kripke models form an elementary class. This is trivial: we are working with the class of multiframes that are partial functional. Given that $\mathcal{L}$ is finite we need merely define:

$$\varphi = \bigwedge_{l \in \mathcal{L}} \forall xyz(xR_l y \wedge xR_l z \rightarrow y = z).$$

Thus we are working with an elementary class, namely the class that satisfies φ. Thus we conclude:

THEOREM 4.2. *If* $|\mathcal{L}| \geq 2$ *and* $\mathcal{A}$ *is countably infinite then the satisfiability problem for* $L^{KR\Box}$ *is* Π^0_1 *complete.* □

What are we to make of this undecidability result? The key technical point is that it is genuinely due to the interaction between the ability to state generalisations and the ability to enforce re-entrancy. The subsequent results elaborate on this theme and reveal an interesting difference between $L^{N\Box}$ and $L^{KR\Box}$. We begin by showing, using a filtration argument (see Fisher and Ladner (1979) for filtrations in Propositional Dynamic Logic), that the satisfiability problems for $L^{\Box}$ and $L^{N\Box}$ are decidable.

THEOREM 4.3. *If* ϕ *is a satisfiable* $L^{\Box}$ *or* $L^{N\Box}$ *formula, then* ϕ *is satisfiable in a model with at most* $2^{2|\phi|}$ *nodes.*

Proof: Suppose that ϕ is an $L^{\Box}$ wff, $\mathrm{M} = \langle W, \{R_l\}_{l \in \mathcal{L}}, V\rangle$, and $\mathrm{M} \models \phi[w_0]$. Let $Cl(\phi)$ be the smallest set that contains ϕ, and is closed under subformulas and single negations. Define an equivalence relation $\sim$ on W as follows:

$$w \sim w' \text{ iff } \forall \psi \in Cl(\phi)(\mathrm{M} \models \psi[w] \Leftrightarrow \mathrm{M} \models \psi[w']).$$

Let $W^F \subseteq W$ be such that W^F contains exactly one element from each equivalence class. Let V^F be the restriction of V to W^F, and define R_l^F as follows:

$$wR_l^F w' \text{ iff } \exists w''(wR_l w'' \wedge w' \sim w'').$$

Let $\mathrm{M}^F = \langle W^F, \{R_l^F\}_{l \in \mathcal{L}}, V^F \rangle$. M^F is a filtration of M through $Cl(\phi)$ in the sense of Hughes and Cresswell (1984), thus it follows immediately that M^F satisfies ϕ. Since the size of $Cl(\phi)$ is at most $2|\phi|$, the size of W^F is bounded by $2^{2|\phi|}$. Furthermore, M^F is an $L^\Box$ model, since the definition of R_l^F ensures that R_l^F is a partial function for any modality l.

Essentially the same argument works for wffs ϕ of $L^{N\Box}$. We need only observe that for all nominals i in $Cl(\phi)$, if $V(i) = \{w\}$ then $w \sim w'$ iff $w' = w$. In short, all nominals in $Cl(\phi)$ denote singletons in the filtrations, and all other nominals can be assigned arbitrary singletons of W^F, thus we again have a small model for ϕ. □

From theorem 4.3, it follows immediately that the satisfiability problems for $L^\Box$ and $L^{N\Box}$ are both decidable in nondeterministic exponential time. But we can improve these results. Using methods similar to Pratt (1979) and Halpern and Moses (1985) we sketch a construction of a deterministic exponential time algorithm for both $L^\Box$ and $L^{N\Box}$ satisfiability.

THEOREM 4.4. *The satisfiability problems for $L^\Box$ and $L^{N\Box}$ are decidable in EXPTIME.*

Proof: Let $Cl(\phi)$ be defined as in the proof of the previous theorem. Let S be the set of all subsets Γ of $Cl(\phi)$ that are maximally propositionally consistent, and are closed under reflexivity of $\Box$; that is, if $\Box\psi \in \Gamma$ then ψ is also in Γ. Suppose ϕ is satisfiable in model M. Let S_M be the set of subsets of $Cl(\phi)$ that actually occur in M, that is, $S_M = \{\Gamma \in S : \mathrm{M} \models \Gamma[w], \text{ for some } w \in \mathrm{M}\}$. Obviously, $S_M \subseteq S$, but we can say more about S_M. First of all, note that every element of S_M contains the same $\Box$ formulas. Furthermore, if ϕ contains a nominal m, there is exactly one set in S_M that contains m. Let $\Sigma \subseteq Pow(S)$, consisting of all maximal $S' \subseteq S$ such that:

1. $\forall \Gamma, \Gamma' \in S', \forall \Box\psi \in Cl(\phi) : \Box\psi \in \Gamma \Leftrightarrow \Box\psi \in \Gamma'$, and
2. For every nominal m occurring in ϕ, there is exactly one set $\Gamma \in S'$ such that $m \in \Gamma$.

If ϕ is satisfiable in **M**, then there exists a set $S' \in \Sigma$ such that $S_M \subseteq S'$. What can we say about the size of Σ? Since $Cl(\phi)$ contains at most $2|\phi|$ elements, there exist at most $2^{2|\phi|}$ maximal sets $\widehat{S} \subseteq S$ fulfilling the first condition. If ϕ contains k nominals, at most $|\widehat{S}|^k$ subsets of $\widehat{S}$ occur in Σ. Since k is bounded by $|\phi|$, the size of Σ is exponential in the length of ϕ.

For every $S_1 \in \Sigma$, we will construct a list of sets $S_1 \supset S_2 \supset S_3 \supset \cdots$ such that: if ϕ is satisfiable in a model M, and $S_M \subseteq S_1$, then $S_M \subseteq S_i$.

Suppose we have defined S_i. Call a set $\Gamma \in S_i$ inconsistent iff one of the following situations occurs:

1. $\neg\Box\psi \in \Gamma$, but for all $\Gamma' \in S_i$: $\psi \in \Gamma'$, or
2. For some modality l, $\langle l\rangle\psi \in \Gamma$ for some ψ, but there is *no* $\Gamma' \in S_i$ such that $\forall\langle l\rangle\xi \in Cl(\phi)(\langle l\rangle\xi \in \Gamma \Leftrightarrow \xi \in \Gamma')$.

If there are inconsistent sets in S_i, then we let S_{i+1} consist of all sets of S_i that are *not* inconsistent. Otherwise, ϕ is satisfiable iff $\phi \in \Gamma$ for some set $\Gamma \in S_i$, and for every nominal m occurring in ϕ, there is exactly one set $\Gamma \in S_i$ that contains m.

Since S_1 is of exponential size, and S_{i+1} is strictly included in S_i, the algorithm terminates after at most exponentially many cycles. Determining which sets in S_i are inconsistent takes polynomial time in the length of the representation of S_i. Thus, for every member of Σ, the algorithm takes at most deterministic exponential time. Since Σ is of exponential size, we can determine if if ϕ is satisfiable in EXPTIME. □

However as the next result shows, there is a clear sense in which this result cannot be improved.

THEOREM 4.5. *The satisfiability problems for $L^{\Box}$ and $L^{N\Box}$ are EXPTIME complete for $|\mathcal{L}| \geq 2$, and $\mathcal{A}$ countably infinite.*

Proof: The upper bounds follow from theorem 4.4. To prove the corresponding lower bounds, it suffices to give a polynomial time computable reduction from an EXPTIME hard set to $L^{\Box}$ satisfiability. We will use a suitable subset of Propositional Dynamic Logic. Let PDL$(a,*)$ be the bimodal propositional language with modalities $\langle a\rangle$ and $\langle a^*\rangle$. We interpret wffs of PDL$(a,*)$ on Kripke models $\mathbf{M} = \langle W, R_a, V\rangle$, where R_a is an arbitrary binary relation on W, in the usual way, the key clause being:

$$\mathrm{M} \models \langle a^*\rangle\phi[w] \text{ iff } \exists w'(wR_a^*w' \;\&\; \mathrm{M} \models \phi[w']).$$

where R_a^* denotes the reflexive, transitive closure of R_a. In Fisher and Ladner (1979), it is proven that the satisfiability problem for PDL$(a,*)$ is EXPTIME hard. In fact, from careful inspection of this proof, we can conclude that even the following set is EXPTIME hard: Let C consist of all PDL$(a, *)$ formulas ϕ such that $\phi = \phi_1 \wedge [a^*]\phi_2$, and

1. ϕ_1, ϕ_2 are $*$-less and have modal depth ≤ 1,
2. ϕ is satisfiable in a model where every node has at most two successors.

Define the reduction f from C to $L^\Box$ satisfiability as follows:

1. If ϕ is *not* of the form $\phi_1 \wedge [a^*]\phi_2$, where ϕ_1 and ϕ_2 are $*$-less and of modal depth ≤ 1, then $f(\phi) = \bot$
2. For ϕ_1, ϕ_2 $*$-less and of modal depth ≤ 1, $f(\phi_1 \wedge [a^*]\phi_2) = s(\phi_1) \wedge \Box s(\phi_2)$, where s is defined on $*$-less formulas as follows:
$$\begin{aligned} s(p) &= p \\ s(\neg\psi) &= \neg s(\psi) \\ s(\psi \vee \xi) &= s(\psi) \vee s(\xi) \\ s(\langle a \rangle \psi) &= \langle a_1 \rangle s(\psi) \vee \langle a_2 \rangle s(\psi) \end{aligned}$$

Since s is polynomial time computable on $*$-less formulas of modal depth ≤ 1, f is polynomial time computable. Now, it is straightforward to prove the following fact by induction. If $\mathrm{M} = \langle W, R_a, \pi \rangle$ is a PDL-model, and $\mathrm{M}' = \langle W, R_{a_1}, R_{a_2}, \pi \rangle$ is an $L^\Box$-model, such that $R_a = R_{a_1} \cup R_{a_2}$, then for all $*$-less PDL(a)-formulas ϕ, and for all nodes $w \in W$, $\mathrm{M} \models \phi[w]$ iff $\mathrm{M}' \models s(\phi)[w]$. By making use of this it is easy to prove that f is indeed a reduction from C to $L^\Box$ satisfiability. □

Note that once again this reduction depends on having an unlimited supply of propositional variables. The following theorem will dispose of this issue once and for all:

THEOREM 4.6 (Single variable reduction theorem). *If $|\mathcal{L}| \geq 1$, then there exist polynomial time reductions from the satisfiability problems for $L^\Box$ and $L^{KR\Box}$ over signature $\langle \mathcal{L}, \mathcal{A} \rangle$ to the corresponding satisfiability problems over signature $\langle \mathcal{L}, \{p\} \rangle$.*

Proof: Recall that we used the following reduction from propositional satisfiability to L satisfiability over signature $\langle \{a\}, \{p\} \rangle$ in theorem 3.4:

$$f(\phi(p_0, \ldots, p_k)) = \phi(p, \langle a \rangle p, \langle a \rangle \langle a \rangle p, \ldots, \langle a \rangle^k p).$$

If ϕ is satisfied in w, we build the corresponding model for $f(\phi)$ by replacing w by a list of nodes $w_0 R'_a w_1 R'_a \ldots R'_a w_k$ such that p is true in w_i iff p_i is true in w. We will use a similar encoding to to prove the theorem. Fix a signature $\langle \mathcal{L}, \mathcal{A} \rangle$, $\mathcal{L} \neq \emptyset$. We'll use a fixed element $a \in \mathcal{L}$ to encode nodes. Suppose $\mathrm{M} = \langle W, \{R_l\}_{l \in \mathcal{L}}, V \rangle$ is a model, and we use propositional variables $p_0, \ldots, p_k$. As a first attempt to obtain an equivalent model with one propositional variable, look at the encoding given above: replace each node w by a list of nodes $w_0 R'_a w_1 R'_a \dots R'_a w_k$ such that p is true in w_i iff p_i is true in w. This doesn't quite work: consider for instance the formula $\Box p_1$. The obvious translation would be $\Box\langle a\rangle p$. But this would mean that $\langle a\rangle p$ has to be satisfied in *every* world w_i. This is too strong a requirement: we just want $\langle a\rangle p$ to be satisfied in every world of the form w_0. We therefore need to be able to determine if we are at a world of the form w_0. We can't use a propositional variable for this: we have already used our sole propositional variable p. The solution is to use a slightly different encoding: we will replace each node w by a list of $2k+3$ nodes $w_0 R'_a w_1 R'_a \dots R'_a w_{2k+2}$ such that: p is true in w_i iff either $i \leq k$ and p_i is true in w, or $i = 2k+2$. Define:

$$\sigma_{0,k} = \bigwedge_{i=k+1}^{2k+1} \langle a\rangle^i \neg p \wedge \langle a\rangle^{2k+2} p.$$

Then $\sigma_{0,k}$ is true in every world w_0, and we will ensure that for every $i > 0$, $\sigma_{0,k}$ is false in w_i. Now we will show how to define the relations R'_l. If $l \neq a$, this is easy: we let R'_l consist of all pairs $\langle w_0, w'_0\rangle$ such that $\langle w, w'\rangle \in R_l$. We can't do this for R'_a, since every node w_0 already has w_1 as its R'_a successor. If $\langle w, w'\rangle \in R_a$, we will add $\langle w_{2k+2}, w'_0\rangle$ to R'_a, that is, we add an R'_a edge from the last node of the encoding of w to the first node of the encoding of w'.

Now we are ready to define the reduction:

$$f(\phi(p_0, \dots, p_k)) = \sigma_{0,k} \wedge g_k(\phi).$$

Here g_k is inductively defined as follows:

$$\begin{array}{lcl}
g_k(p_i) & = & \langle a\rangle^i p \\
g_k(\neg\psi) & = & \neg g_k(\psi) \\
g_k(\psi_1 \vee \psi_2) & = & g_k(\psi_1) \vee g_k(\psi_2) \\
g_k(\langle l\rangle\psi) & = & \langle l\rangle(\sigma_{0,k} \wedge g_k(\psi)) \text{ for } l \neq a \\
g_k(\langle a\rangle\psi) & = & \langle a\rangle^{2k+3}(\sigma_{0,k} \wedge g_k(\psi)) \\
g_k(\Box\psi) & = & \Box(\sigma_{0,k} \rightarrow g_k(\psi)) \\
g_k(\langle A\rangle \approx \langle B\rangle) & = & \langle A\rangle \approx \langle B\rangle[\langle a\rangle := \langle a\rangle^{2k+3}] \wedge g_k(\langle A\rangle T) \wedge g_k(\langle B\rangle T)
\end{array}$$

(The notation $[\langle a\rangle := \langle a\rangle^{2k+3}]$ denotes the result of substituting $\langle a\rangle^{2k+3}$ for $\langle a\rangle$.) Obviously, f is polynomial time computable. Furthermore, if ϕ does not contain path formulas, then neither does $f(\phi)$. It remains to prove that ϕ is satisfiable iff $f(\phi)$ is satisfiable.

Let $\mathrm{M} = \langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$. Define the corresponding model $\mathrm{M}_k = \langle \widehat{W}, \{\widehat{R}_l\}_{l\in\mathcal{L}}, \widehat{V}\rangle$ as follows:

$$\begin{array}{lcl}
\widehat{W} & = & \{w \in W : \mathrm{M} \models \sigma_{0,k}\} \\
\widehat{R}_l & = & R_l|(\widehat{W} \times \widehat{W}) \text{ for } l \neq a \\
\widehat{R}_a & = & (R_a)^{2k+3}|(\widehat{W} \times \widehat{W}) \\
\widehat{V}(p_i) & = & \{w : \mathrm{M} \models \langle a\rangle^i p\}
\end{array}$$

With induction the structure of ψ, it is easy to prove that for all formulas ψ with propositional variables in $\{p_0, \ldots, p_k\}$, and for all $w \in \widehat{W}$:

$$\mathrm{M} \models g_k(\psi)[w] \text{ iff } \mathrm{M}_k \models \psi[w]$$

Now suppose $\mathrm{M} \models f(\phi)[w]$. Then $w \in W'$, since $\mathrm{M} \models \sigma_{0,k}[w]$. Therefore, $\mathrm{M}_k \models \phi[w]$, and hence ϕ is satisfiable.

For the converse, suppose that ϕ is satisfiable. Let $\mathrm{M} = \langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$ be a model such that $\mathrm{M} \models \phi[v]$. Let $\mathrm{M}' = \langle W', \{R'_l\}_{l\in\mathcal{L}}, V'\rangle$ be the corresponding model with one propositional variable, as sketched before the definition of the reduction:

$$\begin{array}{lcl}
W' & = & \{w_0, \ldots, w_{2k+2} : w \in W\} \\
R'_l & = & \{\langle w_0, w'_0\rangle : wR_lw'\} \text{ (for } l \neq a) \\
R'_a & = & \{\langle w_i, w_{i+1}\rangle : i \leq 2k+1\} \cup \{\langle w_{2k+2}, w'_0\rangle : wR_aw'\} \\
V'(p) & = & \{w_i : i = 2k+2 \text{ or } (w \in V(p_i) \text{ and } i \leq k)\}
\end{array}$$

It is clear that M'_k is isomorphic to M, therefore $\mathrm{M}' \models \sigma_{0,k} \wedge g(\phi)[v_0]$. □

As in theorem 3.5, we can prove that if $\mathcal{L}$ contains at least two modalities, we can dispense with propositional variables all together. Recall that we used the following reduction in theorem 3.5:

$$f(\phi(p_0, \ldots, p_k)) = \phi(\langle b\rangle\top, \langle a\rangle\langle b\rangle\top, \langle a\rangle\langle a\rangle\langle b\rangle\top, \ldots, \langle a\rangle^k\langle b\rangle\top).$$

We can strengthen this. It is easy to see that the techniques of the previous theorem can be applied to prove the analogue of theorem 3.5. We leave the details to the reader.

THEOREM 4.7. *If $|\mathcal{L}| \geq 2$, then there exist polynomial time reductions from the satisfiability problems for $L^{\Box}$ and $L^{KR\Box}$ over signature $\langle\mathcal{L}, \mathcal{A}\rangle$ to the corresponding satisfiability problems over signature $\langle\mathcal{L}, \emptyset\rangle$.* □

Combining the previous theorem with the earlier obtained lower bounds, we can summarise the complexity results of this section as follows:

COROLLARY 4.1. *If* $|\mathcal{L}| \geq 2$, *and* $|\mathcal{A}| \geq 0$ *then the satisfiability problems for* $L^{\Box}$ *and* $L^{N\Box}$ *are EXPTIME complete, and the satisfiability problem for* $L^{KR\Box}$ *is* Π^0_1 *complete.* □

An interesting aspect of the results of this section is the wedge they drive between $L^{N\Box}$ and $L^{KR\Box}$. At first sight the difference seems puzzling: after all, both are languages in which generalisations can be stated and re-entrancy forced. A closer look shows that the two languages work very differently. We might say that whereas in $L^{KR\Box}$ we can state genuine generalisations involving re-entrancy, in $L^{N\Box}$ there is a clear sense in which re-entrancy is only expressed within a given model. $L^{N\Box}$ isn't powerful enough to force labelings. An example will make this clear. Consider the GPSG head feature convention again. We've already seen that its force is captured in $L^{KR\Box}$ by the following wff:

$$\Box(\mathit{phrasal} \rightarrow \langle\text{HEAD}\rangle \approx \langle\text{HEAD-DTR}\rangle\langle\text{HEAD}\rangle).$$

But when we attempt to capture its force using nominals we run into a problem: how can we label the desired re-entrancy point? It seems we must step beyond the boundaries of $L^{N\Box}$ and write an expression such as the following:

$$\Box(\mathit{phrasal} \rightarrow \exists i(\langle\text{HEAD}\rangle i \wedge \langle\text{HEAD-DTR}\rangle\langle\text{HEAD}\rangle i)).$$

Now, this expression clearly captures what is required, but unfortunately it's not an $L^{N\Box}$ wff but a wff of a more powerful language in which explicit quantification over nominals is possible. Such languages have been investigated before; Bull (1970) seems to have been the first technical investigation of nominals. Moreover Reape (1991) has used such a language to investigate problems in unification based grammar. However when used together with the universal modality, explicit quantification over nominals is (from the point of view of complexity theory at any rate) rather uninteresting: it is straightforward to show that strengthening $L^{N\Box}$ to allow explicit quantification over nominals results in a notational variant of L^1, the first order language of AVSs. Such a language thus has a Π^0_1 satisfiability problem, just as $L^{KR\Box}$ does.

In short, it is asking a lot to be able to express generalisations involving re-entrancy. The nearest we can get to it in a decidable framework seems to be $L^{N\Box}$. However, while generalisations are expressible in this language, these generalisations don't involve re-entrancy in any strong sense. It's precisely for this reason that we're not able to force a tiling in this language, but (alas) it's also precisely for this reason that it is not able express some linguistically useful principles such as the head feature convention.

5. THE MASTER MODALITY

In this section we consider the complexity of the satisfiability problems for $L^{[*]}$, $L^{N[*]}$ and $L^{KR[*]}$, our base languages extended with the master modality $[*]$. Gazdar *et al.* (1988) define the master modality as follows:

$$\begin{array}{lll} \mathrm{M} \models [*]\phi[w] & \text{iff} & \mathrm{M} \models \phi[w] \text{ and} \\ & & \mathrm{M} \models [*]\phi[w'], \text{ for all } w' : wR_l w', \text{ for some } l \in \mathcal{L}. \end{array}$$

As they only work with finite AVSs this definition is not circular, indeed it has the advantage of making the intended use of $[*]$ particularly clear: $[*]$ expresses recursive constraints over AVSs. (See Carpenter (1992) for a discussion of recursive constraints.) However it will make the following technicalities more straightforward if we extend the definition to cover arbitrary AVSs. We do this as follows.

$$\mathrm{M} \models [*]\phi[w] \text{ iff } \mathrm{M} \models \phi[w'], \text{ for all } w' \in W \text{ such that } w(\bigcup_{l\in\mathcal{L}} R_l)^* w'$$

That is, ϕ must be satisfied at all nodes w' that are reachable by any finite sequence of transitions (including the null transition) from w. Clearly this definition reduces to the previous one for finite AVSs. It's also worth mentioning that we have introduced a notational change; Gazdar *et al.* use $\Box$ for the master modality. We prefer to reserve this for the universal modality.

The most important thing to note about both semantic definitions given above is their infinitary force: L^1 is *not* the correspondence language for $[*]$. As with PDL, the natural correspondences are with classical languages in which infinite disjunctions are allowed; in effect we are working with a fragment of infinitary logic.

A number of logical results for $L^{[*]}$, including the construction of a complete tableaux system, have been proved by Kracht (1989). However his methods only yield a nondeterministic exponential time upper bound for the satisfiability problem; we improve on this below. Neither $L^{N[*]}$ nor $L^{KR[*]}$ seem to have been treated in the literature, though Gazdar *et al.* (1988) note that some re-entrancy coding mechanism would be desirable, and Kasper and Rounds (1990) mention the possibility of combining the two approaches. $L^{KR[*]}$ is this combination.

We begin our investigation with a lemma which enables us to utilise results from the previous section.

LEMMA 5.1. *Let ϕ be a formula that contains no occurrences of $\Box$ or $[*]$. Then $\Box\phi$ is satisfiable iff $[*]\phi$ is satisfiable.*

Proof: First suppose $\mathbf{M} = \langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$, and $\mathbf{M} \models \Box\phi[w_0]$. Then for all $w \in W$, $\mathbf{M} \models \phi[w]$, and therefore certainly $\mathbf{M} \models [*]\phi[w_0]$.

Conversely suppose $\mathbf{M} = \langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$, and $\mathbf{M} \models [*]\phi[w_0]$. Let W' equal $\{w \in W : w_0(\bigcup_{l\in\mathcal{L}} R_l)^* w\}$, and let $\mathbf{M}'$ be the restriction of $\mathbf{M}$ to W'. It follows by the usual generated submodel argument that for all formulas ψ without $\Box$ or $[*]$, and for all $w \in W'$: $\mathbf{M} \models \psi[w]$ iff $\mathbf{M}' \models \psi[w]$. It follows that $\mathbf{M}' \models \phi[w]$, for all $w \in W'$. But then $\mathbf{M}' \models \Box\phi[w_0]$. □

From this lemma, and the form of the reductions in the proofs of theorems 4.1 and 4.5, it follows immediately that the lower bounds for languages with □ go through for the corresponding languages with $[*]$:

COROLLARY 5.1. *The satisfiability problems for $L^{[*]}$ and $L^{N[*]}$ are EXPTIME hard. The satisfiability problem for $L^{KR[*]}$ is Π^0_1 hard.* □

But do we have the the same upper bounds? The answer is almost always 'yes', but there is one notable exception. If $\mathcal{L}$ is *finite*, and contains at least two elements, the complexity of the satisfiability problem for $L^{KR[*]}$ is much higher than that of the corresponding satisfiability problem for $L^{KR\Box}$: we will show that in this case $L^{KR[*]}$ satisfiability is Σ^1_1 complete instead of 'just' Π^0_1 complete.

LEMMA 5.2. *If ϕ is satisfiable in* $\mathbf{M}$, *then ϕ is satisfiable in a* countable *submodel of* $\mathbf{M}$.

Proof: Suppose $\mathbf{M} \models \phi[w]$. Let W' be $\{w' \in W : w(\bigcup_{l\in\mathcal{L}} R_l)^* w'\}$. It follows by induction on the degree of ϕ that $\mathbf{M}|W' \models \phi[w]$. But as all our relations are in fact partial functions, and as we only have countably many of them, W' must be countable. □

THEOREM 5.1. *If $\mathcal{L}$ is finite, and $|\mathcal{L}| \geq 2$, the satisfiability problem for $L^{KR[*]}$ is Σ^1_1 complete.*

Proof: The upper bound follows directly from lemma 5.2. To prove the corresponding lower bound, we will construct a reduction from the following Σ^1_1 complete tiling problem (see Harel (1986)):

N × N recurrent tiling: Given a finite set $\mathcal{T}$ of tiles, and a tile $T_1 \in \mathcal{T}$, can $\mathcal{T}$ tile $\mathbf{N} \times \mathbf{N}$ such that T_1 occurs in the tiling infinitely often on the first row.

That is, does there exist a function t from $\mathbf{N} \times \mathbf{N}$ to $\mathcal{T}$ such that: $\mathit{right}(t(n,m)) = \mathit{left}(t(n+1,m))$, $\mathit{up}(t(n,m)) = \mathit{down}(t(n,m+1))$, and the set $\{i : t(i,0) = T_1\}$ is infinite?

Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ be a set of tiles. We construct a formula ϕ_{rt} such that:

$$\langle \mathcal{T}, T_1 \rangle \in \mathbf{N} \times \mathbf{N} \text{ recurrent tiling } \text{ iff } \phi_{rt} \text{ is satisfiable.}$$

To ensure that ϕ_{rt} forces a tiling of $\mathbf{N} \times \mathbf{N}$, we use the formula ϕ constructed in the proof of theorem 4.1. Let ϕ' be the result of replacing every occurrence of $\Box$ by $[*]$ in ϕ. Then, as in theorem 4.1, the following holds:

1. If ϕ' is not satisfiable, then $\mathcal{T}$ does not tile $\mathbf{N} \times \mathbf{N}$.
2. If $\mathbf{M} \models \phi'[w_0]$, then there exists a tiling t of $\mathbf{N} \times \mathbf{N}$, and a function f from $\mathbf{N} \times \mathbf{N}$ to W be such that $f(0,0) = w_0$, $f(n,m)R_r f(n+1,m)$ and $f(n,m)R_u f(n,m+1)$, and $\mathbf{M} \models t_i[f(n,m)]$ iff $t(n,m) = T_i$.

Now we force the recurrence: we will use a new propositional variable row_0, which can only be true at nodes of the form $f(n,0)$, and we will ensure that there exist an infinite number of nodes where row_0 holds and tile T_1 is placed. Define:

$$\phi_{rec} = ([*] \bigwedge_{l \in \mathcal{L}, l \neq r} [l][*]\neg row_0) \wedge row_0 \wedge [*](row_0 \rightarrow \langle * \rangle (row_0 \wedge t_1)).$$

Let ϕ_{rt} be the conjunction of ϕ' and ϕ_{rec}. In the same way as in theorem 4.1, we can now prove that $\langle \mathcal{T}, T_1 \rangle \in \mathbf{N} \times \mathbf{N}$ recurrent tiling iff ϕ_{rt} is satisfiable. □

In the previous proof it is essential that we can force a propositional variable to be true at w only if w is reachable from w_0 in a finite number of R_r steps. We can't force this in $L^{KR\Box}$, nor in $L^{KR[*]}$ if $\mathcal{L}$ is infinite. (Indeed the previous proof doesn't go through for $\mathcal{L}$ infinite as then ϕ_{rec} is not a formula.) As we shall now see, in the case where $\mathcal{L}$ is infinite, the satisfiability problem for a language with $[*]$ is never more complex than the satisfiability problem for the corresponding language with $\Box$.

THEOREM 5.2. *If $\mathcal{L}$ is infinite, then*

1. *The satisfiability problems for $L^{[*]}$ and $L^{N[*]}$ are EXPTIME complete.*
2. *The satisfiability problem for $L^{KR[*]}$ is Π^0_1 complete.*

Proof: The lower bounds follow from corollary 5.1. For the upper bounds, we will reduce the satisfiability problems for $L^{[*]}$, $L^{N[*]}$, and $L^{KR[*]}$ to the satisfiability problems for the corresponding languages with $\Box$. The claim then follows from theorems 4.5 and 4.1. To get rid of occurrences of $[*]$, we define function g from $\Box$-less formulas to formulas without $\Box$ or $[*]$ as follows:

$$g(p) = p \qquad g(\neg\psi) = \neg g(\psi) \quad g(\psi \vee \xi) = g(\psi) \vee g(\xi)$$
$$g(\langle l \rangle \psi) = \langle l \rangle g(\psi) \quad g([*]\psi) = p_{[*]\psi} \quad g(\langle A \rangle \approx \langle B \rangle) = \langle A \rangle \approx \langle B \rangle$$

We have to ensure that $p_{[*]\psi}$ mimics the behaviour of $[*]\psi$. In particular, if $\neg p_{[*]\psi}$ holds at some node, this node should have a (multi-step) successor where $g(\neg\psi)$ holds. We introduce new modalities $\langle\neg\psi\rangle$ for all formulas $[*]\psi \in Cl(\phi)$, and we will force that for every node w satisfying $\neg p_{[*]\psi}$, there exists a node w' such that $wR_{\neg\psi}w'$ and $g(\neg\psi)$ holds at w'. Let $\mathcal{L}'$ consist of the modalities occurring in ϕ, and the new modalities $\langle\neg\psi\rangle$ for $[*]\psi \in Cl(\phi)$. Since $\mathcal{L}$ is infinite, we may assume that $\mathcal{L}' \subset \mathcal{L}$. Our reduction f is defined as follows:

$$f(\phi) = g(\phi) \wedge \Box(p_{[*]\psi} \rightarrow g(\psi) \wedge \bigwedge_{l\in\mathcal{L}'} [l]p_{[*]\psi}) \wedge \Box(\neg p_{[*]\psi} \rightarrow \langle\neg\psi\rangle g(\neg\psi)).$$

Obviously, f is polynomial time computable. Furthermore, if ϕ doesn't contain nominals and/or path equations, then neither does $f(\phi)$. It remains to prove that ϕ is satisfiable iff $f(\phi)$ is satisfiable.

First suppose ϕ is satisfiable. By lemma 5.2, there exist a countable model $\mathrm{M} = \langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$, and a node $w_0 \in W$ such that $M \models \phi[w_0]$. Define a model $\widehat{\mathrm{M}}$ as follows: $\widehat{\mathrm{M}} = \langle W, \{\widehat{R}_l\}_{l\in\mathcal{L}}, \widehat{V}\rangle$, such that:

1. $\widehat{R}_l = R_l$ for l occurring in ϕ; $\widehat{R}_l = \emptyset$ for $l \notin \mathcal{L}'$
2. For $[*]\psi \in Cl(\phi)$, $\widehat{R}_{\neg\psi}$ is such that:

   a. $w\widehat{R}_{\neg\psi}w' \Rightarrow \mathrm{M} \models \neg\psi$, and $w(\bigcup_{l\in\mathcal{L}} R_l)^*w'$; and
   b. $\exists w' : w\widehat{R}_{\neg\psi}w'$ iff $\mathrm{M} \models \neg[*]\psi$.

3. $\widehat{V}(p) = V(p)$ for p occurring in ϕ; $w \in \widehat{V}(p_{[*]\psi})$ iff $\mathrm{M} \models [*]\psi[w]$.

Obviously, if $\mathrm{M} \models \phi[w_0]$, then $\widehat{\mathrm{M}}$ is well defined, and $\widehat{\mathrm{M}} \models f(\phi)[w_0]$.

For the converse, suppose $f(\phi)$ is satisfiable. let $\mathrm{M} = \langle W, \{R_l\}_{l\in\mathcal{L}}, V\rangle$, and $w_0 \in W$ be such that $\mathrm{M} \models f(\phi)[w_0]$. We may assume that $R_l = \emptyset$ for $l \notin \mathcal{L}$. It is easy to prove that for all formulas $\psi \in Cl(\phi)$ and for all $w \in W$, $\mathrm{M} \models \phi[w]$ iff $\mathrm{M} \models g(\phi)[w]$, and thus ϕ is indeed satisfiable. □

It remains to prove EXPTIME upper bounds for $L^{[*]}$ and $L^{N[*]}$ for finite $\mathcal{L}$.

THEOREM 5.3. *If $\mathcal{L}$ is finite, and $|\mathcal{L}| \geq 2$, then the satisfiability problem for $L^{[*]}$ is EXPTIME complete.*

The lower bound follows from corollary 5.1. For the corresponding upper bound, we will give a reduction from this satisfiability problem to the satisfiability problem for a suitable subset of Deterministic Propositional Dynamic Logic (DPDL). This proves the theorem, since the satisfiability problem for DPDL is in EXPTIME (see Ben-Ari *et al.* (1982)). Our DPDL subset is the multi-modal propositional language with modalities $\langle l \rangle$ for all $l \in \mathcal{L}$, and $\langle (\bigcup_{l \in \mathcal{L}} l)^* \rangle$, which we will abbreviate as $\langle * \rangle$. We interpret wffs of this language on Kripke models $\mathrm{M} = \langle W, \{R_l\}_{l \in \mathcal{L}}, V \rangle$, where R_l is a partial functional binary relation on W, in the usual way, the key clause being:

$$\mathrm{M} \models \langle * \rangle \phi[w] \text{ iff } \exists w'(w(\bigcup_{l \in \mathcal{L}} R_l)^* w' \ \&\ \mathrm{M} \models \phi[w']).$$

Let ϕ be an $L^{[*]}$ formula. It is obvious that ϕ is a satisfiable $L^{[*]}$ formula iff ϕ is a satisfiable DPDL formula. □

THEOREM 5.4. *If $\mathcal{L}$ is finite, and $|\mathcal{L}| \geq 2$, then the satisfiability problem for $L^{N[*]}$ is EXPTIME complete.*

The lower bound follows from corollary 5.1. For the corresponding upper bound, we will give a reduction from the satisfiability problem for $L^{N[*]}$ to the corresponding satisfiability problem for $L^{[*]}$. The theorem then follows from theorem 5.3. Suppose ϕ is an $L^{N[*]}$ formula, and $m_1, \ldots, m_k$ are all the nominals occurring in ϕ. We can view nominals as ordinary propositional variables, with the extra requirement that each nominal is satisfied exactly once. We can't quite force that, but it turns out that forcing the following requirements for every nominal m that occurs in ϕ are enough to obtain the required reduction.

1. All nodes where m holds are equivalent with respect to $Cl(\phi)$
2. If m is true, and $\neg[*]\psi, \psi$ hold at w for some $[*]\psi \in Cl(\phi)$, then there exists a node w' reachable from w by a non-m path such that $\neg\psi$ holds at w'

To force the second requirement, we introduce new propositional variables $m_{\langle * \rangle \neg\psi}$, for each $[*]\psi \in Cl(\phi)$, and each occurring nominal m. $m_{\langle * \rangle \neg\psi}$ will be true if $\neg\psi$ has to be fulfilled by a node reachable by a non-m path. Now define the reduction f:

$$\begin{aligned} f(\phi) = \phi \ \wedge \ & \textstyle\bigwedge_{\psi \in Cl(\phi)}([*](m \rightarrow \psi) \vee [*](m \rightarrow \neg\psi)) \\ \wedge \ & \textstyle\bigwedge_{[*]\psi \in Cl(\phi)}([*](m \wedge \neg[*]\psi \wedge \psi \rightarrow \bigvee_{l \in \mathcal{L}} \langle l \rangle m_{\langle * \rangle \neg\psi}) \wedge \\ & \qquad [*](m_{\langle * \rangle \neg\psi} \wedge \psi \rightarrow \textstyle\bigvee_{l \in \mathcal{L}} \langle l \rangle m_{\langle * \rangle \neg\psi}) \wedge \\ & \qquad [*](m \rightarrow \neg m_{\langle * \rangle \neg\psi}) \wedge \\ & \qquad [*](m_{\langle * \rangle \neg\psi} \rightarrow \langle * \rangle \neg\psi)) \end{aligned}$$

It is obvious that if ϕ is satisfiable in a model where every nominal m occurs exactly once, then $f(\phi)$ is satisfiable.

Conversely, suppose $f(\phi)$ is a satisfiable $L^{[*]}$ formula. Let $\mathrm{M} = \langle W, \{R_l\}_{l \in \mathcal{L}}, V \rangle$ be a model such that $\mathrm{M} \models f(\phi)[w]$. Define relation $\sim$

such that: $w \sim w' \Leftrightarrow (w = w')$ or $\mathrm{M} \models m[w]$ and $\mathrm{M} \models m[w']$ for some nominal m occurring in ϕ. It is easy to see that $\sim$ is an equivalence relation, and filtrating over $\sim$ (compare theorem 4.3) yields a model for ϕ. □

As in the case of languages with $\Box$, we can reduce the number of propositional variables. Define $g_k([*]\psi) = [*](\sigma_{0,k} \rightarrow g(\psi))$ in the construction of theorem 4.6, and define $f(\phi(p_0, \ldots, p_k))$ to be

$$\sigma_{0,k} \wedge g_k(\phi) \wedge [*](\sigma_{0,k} \rightarrow ([a]^{2k+3}\sigma_{0,k} \wedge \bigwedge_{l \neq a}[l]\sigma_{0,k} \wedge \bigwedge_{i=1}^{2k+2} \langle a \rangle^i(\neg\sigma_{0,k} \wedge \bigwedge_{l \neq a}[l]\bot)))$$

to get the analogue of theorem 4.6 for languages with $[*]$. The extra conjunct in f force more similarity between the original model and the encoded model: $[*]$ can force more structure than $\Box$. In a similar way, we can get the analogue of theorem 4.7. Details are left to the reader.

We can summarise the complexity results of this section as follows:

COROLLARY 5.2. *If $|\mathcal{L}| \geq 2$, and $|\mathcal{A}| \geq 0$ the satisfiability problems for $L^{[*]}$ and $L^{N[*]}$ are EXPTIME complete, and the satisfiability problem for $L^{KR[*]}$ is Π^0_1 complete for $\mathcal{L}$ infinite, and Σ^1_1 complete for $\mathcal{L}$ finite.* □

Clearly the results of this section are very bad; does this mean such infinitary extensions should be abandoned? We believe not: an interesting case for their linguistic interest is made by Keller (1991), who works with a language even *stronger* than $L^{KR[*]}$, namely PDL augmented with the Kasper Rounds path equality. Among other things Keller shows how this language can give a neat account of the LFG idea of *functional uncertainty*. Thus the idea seems linguistically interesting: the pressing task becomes the search for well behaved fragments.

Finally it should be remarked that Gazdar *et al.* (1988) emphasize a different application for $L^{[*]}$. Instead of viewing it as a grammar specification formalism, they use it to define syntactic categories; indeed the greater part of their paper is devoted to showing how a wide variety of treatments of syntactic category can be modelled and compared using $L^{[*]}$. An interesting corollary of this is that they are not particularly interested in the satisfaction problem: the problem of most concern to them is how expensive it is to check a category structure against some fixed category description ϕ. Clearly this is a very much simpler problem; in fact they show that it is solvable in linear time if ϕ is a wff of $L^{[*]}$. Their result extends to wffs of $L^{N[*]}$ and $L^{KR[*]}$.

6. CONCLUDING REMARKS

In this paper we have investigated the satisfiability problem for a variety of modal languages of AVSs. The following table summarises the results for the case of most interest in computational linguistics: both $\mathcal{L}$ and $\mathcal{A}$ finite $(|\mathcal{L}| \geq 2, |\mathcal{A}| \geq 0)$.

	L	L^N	L^{KR}
	NP complete	NP complete	NP complete
$\Box$	EXPTIME complete	EXPTIME complete	Π^0_1 complete
$[*]$	EXPTIME complete	EXPTIME complete	Σ^1_1 complete

As a final remark, let's see what happens if $|\mathcal{L}| = 1$. Intuitively, this should make things easier, and indeed it does. Consider for instance the languages with only $[*]$ and $\langle a \rangle$ as modalities. It is easy to see that a formula in these languages is satisfiable iff it is satisfiable on a (possibly infinite) model of the form $w_0 R_a w_1 R_a w_2 R_a \cdots$ or on a model of the form $w_0 R_a w_1 R_a \cdots R_a w_k R_a w_{k+1} R_a \cdots R_a w_m R_a w_k$. In this situation path equations or nominals don't make the situation more complex that for $L^{[*]}$.

In fact $L^{[*]}$ is very like propositional linear temporal logic with operators X (next time) and G (always in the future). Formulas of this language are interpreted on **N**, the natural numbers in their usual order, as follows: $X\phi$ holds at i if ϕ holds at $i + 1$, and $G\phi$ holds at i iff for all $i' \geq i$, ϕ holds at i'. Using the fact that satisfiability for this language is PSPACE complete (see Sistla *et al* (1985)), it is easy to prove that the satisfiability problems for the languages with only $\langle a \rangle$ and $[*]$ as modalities are PSPACE complete as well. Using similar methods, we get the same results for the corresponding languages with $\Box$. We leave the details to the reader. Combining these remarks with theorem 4.6, and theorem 3.4, we can summarise the results for $|\mathcal{L}| = 1$ as follows:

THEOREM 6.1. *If* $|\mathcal{L}| = 1$, *and* $|\mathcal{A}| \geq 1$, *the satisfiability problems for* L, L^N, *and* L^{KR} *are NP complete, and the satisfiability problems for* $L^{\Box}$, $L^{N\Box}$, $L^{KR\Box}$, $L^{[*]}$, $L^{N[*]}$, *and* $L^{KR[*]}$ *are PSPACE complete.* □

There remains much to do. In this paper we have confined ourselves to languages with full Boolean expressivity, hence the results of this paper are essentially limitative. An important problem to turn to next is the exploration of weaker fragments. Obvious choices would be fragments with only conjunction and disjunction, or fragments with only conjunction and the negation of atoms. Results for some such fragments exist in the literature, but a more detailed examination seems both possible and desirable. Further, it would be interesting to look for tractable fragments involving $\Box$ or $[*]$. A good way of approaching this topic would be to add strict implication $\Rightarrow$ as a primitive symbol to various fragments of L, L^N or L^{KR} (as we saw earlier, it the implicit combination of $\Box$ and $\rightarrow$ provided by $\Rightarrow$ that is the most important use of the universal modality) and then to look for restricted forms of strict implication that are useful but tractable.

It is the belief of the authors, however, that modal logic has more to offer computational linguistics than an analysis of unification formalisms. We've already seen a hint of this in Evan's use of $\Box$ to look at feature specification defaults, and in the the use of $L^{[*]}$ to specify grammatical categories. Moreover modalities figure in recent work in categorial grammar; see Roorda (1991) for example. However there seem to be further possibilities. A particularly interesting one concerns the organisation of computational lexicons. An important task in this application is the developed of formalisms for representing and manipulating lexical entries. DATR (see Evans and Gazdar (1989)) is such a formalism, and an examination of its syntax and semantics suggests that it is open to modal analysis. What sort of benefits might result from such an analysis? Complexity results and inference systems are obvious answers, but there is another possibility that might be more important: modal logic might provide 'logical maps' of possible extensions.

This point seems to be of wider relevance. In recent years modal logicians have explored a wide variety of enriched systems, some of which clearly bear on issues of knowledge representation. As has already been mentioned, Schild (1990) has made use of correspondences between core terminological logic and modal logic to obtain a number of complexity results for terminological reasoning. However more correspondences are involved. For example, terminological reasoning may also involve the 'counting quantifiers'; that is, we may want to perform numerical comparisons. The modal logic of such counting quantifiers (and a great deal more besides) has been investigated by van der Hoek and de Rijke (1991). Their work covers such topics as completeness, normal forms and computational complexity and is of obvious relevance to the knowledge representation community.

Finally, there may be deeper mathematical reasons for taking the modal connection seriously. Modal logic comes equipped not only with a Kripke semantics, but with an algebraic semantics, and *duality theory*, the study of the connections between the algebraic semantics and the Kripke semantics, is a highly developed branch of model logic; see Goldblatt (1989) for a detailed recent account. While some use of the algebraic semantics has been made in connection with Attribute Value structures (Reape (1991) for example, uses it to make connections with Smolka's work, and Schild (1990) utilises an algebraic approach to simplify his presentation) in general it seems that a tool of potential value has been neglected. It certainly seems sensible to explore the links with algebraic approaches to knowledge representation formalisms, such as that of Brink and Schmidt (1992).

ACKNOWLEDGEMENTS

The authors would like to thank Johan van Benthem, Claire Gardent and Gerald Gazdar for their comments on earlier drafts of this paper. A shorter version was presented at the Eighth Amsterdam Colloquium. On this occasion Kees Vermeulen pressed the issue of finite numbers of propositional variables, for which we're very grateful. Finally, comments from Johan van Benthem, Jochen Dörre, Peter van Emde Boas, Klaus Schild, Renate Schmidt and the anonymous referee on the penultimate version were very helpful. Patrick Blackburn would like to acknowledge the financial support of a Science and Engineering Research Council (United Kingdom) postdoctoral fellowship.

References

Balcázar, J., Díaz, J. and Gabarró, J., 1988, *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science 11, Berlin: Springer Verlag.

Ben-Ari, M., Halpern, J. and Pnueli, A., 1982, "Deterministic propositional dynamic logic: finite models, complexity and completeness", *Journal of Computer and System Sciences* **25**, 402–417.

Berger, R., 1966, "The undecidability of the dominoe problem", *Mem. Amer. Math. Soc.* **66**, 1–72.

Blackburn, P., 1989, "Nominal tense logic", to appear in *Notre Dame Journal of Formal Logic*.

Blackburn, P., 1991, "Modal logic and attribute value structures", to appear in *Diamonds and Defaults*, M. de Rijke, ed., Studies in Logic, Language and Information, Dordrecht: Kluwer.

Brink, C. and Schmidt, R., 1992, "Subsumption computed algebraically", *Computers Math. Applic.* **23**, 329–342.

Bull, R., 1970, "An approach to tense logic", *Theoria* **12**, 171–182.

Carpenter, B., 1992, *The Logic of Typed Feature Structures*, Cambridge: Cambridge University Press.

Evans, R., 1987, "Towards a formal specification for defaults in GPSG", in *Categories, Polymorphism and Unification*, E. Klein and J. van Benthem, eds., Edinburgh: Centre for Cognitive Science and Amsterdam: ITLI.

Evans, R. and Gazdar, G., 1989, "The semantics of DATR", *AISB-89, Proceedings of the Seventh Conference of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour*, A. Cohn, ed., London: Pitman.

Fischer, M. and Ladner, R., 1979, "Propositional dynamic logic of regular programs", *Journal of Computer and System Sciences* **18**, 194–211.

Gargov, G., Passy, Ş. and Tinchev, T., 1986, "Modal environment for boolean speculations (preliminary report)", in *Mathematical Logic and its Applications. Proceedings of the 1986 Gödel Summer School and Conference Bulgaria*, D. Skordev, ed., New York: Plenum Press.

Gargov, G. and Passy, S., 1988, "Determinism and looping in combinatory PDL", *Theoretical Computer Science* **61**, pages 259–277.

Gargov, G. and Goranko, V., 1989, "Modal logic with names", to appear in *Journal of Philosophical Logic*.

Gazdar, G., Klein, E., Pullum, G. and Sag, I., 1985, *Generalized Phrase Structure Grammar*, Oxford: Basil Blackwell.

Gazdar, G., Pullum G., Carpenter, R., Klein, E., Hukari, T. and Levine, R., 1988, "Category structures", *Computational Linguistics* **14**, pages 1–19.

Goldblatt, R., 1989, "Varieties of complex algebras", *Annals of Pure and Applied Logic* **44**, 173–242.

Goranko, V. and Passy, S., 1992, "Using the universal modality: gains and questions", *Journal of Logic and Computation* **2**, 5–30.

Halpern, J. and Moses, M., 1985, "A guide to the modal logics of knowledge and belief", pp. 480-490 in *Proceedings 9th International Joint Conference on Artificial Intelligence*.

Harel, D., 1983, "Recurring dominoes: making the highly undecidable highly understandable", pp. 177-194 in *Proc. of the Conference on Foundations of Computing Theory*, Springer Lecture Notes in Computer Science **158**.

Harel, D., 1984, "Dynamic logic", in *Handbook of Philosophical Logic* **2**, D. Gabbay and F. Guenthner, eds., Dordrecht: Reidel.

Harel, D., 1986, "Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness", *Journal of the ACM* **33(1)**, 224–248.

van der Hoek, W. and de Rijke, M., 1991, "Generalized quantifiers and modal logic", to appear in *Journal of Logic, Language and Information*.

Hughes, G. and Cresswell, M., 1984, *A Companion to Modal Logic*, London: Methuen & Co. Ltd.

Kaplan, K. and Bresnan, J., 1982, "Lexical functional grammar: a formal system for grammatical representation", pp. 173-281 in *The Mental Representation of Grammatical Relations*, Joan Bresnan, ed., Cambridge: MIT Press.

Karttunen, L., 1984, "Features and values", pp. 28-33 in *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics*, Stanford, California.

Kasper, R. and Rounds, W., 1986, "A logical semantics for feature structures", pp. 257-266 in *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, Columbia University, New York.

Kasper, R. and Rounds, W., 1990, "The logic of unification in grammar", *Linguistics and Philosophy* **13**, 33–58.

Keller, B., 1991, *Feature Logics, Infinitary Descriptions and the Logical Treatment of Grammar*, PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, United Kingdom.

Kracht, M., 1989, "On the logic of category definition", *Computational Linguistics* **15**, 111–113.

Ladner, R., 1977, "The computational complexity of provability in systems of modal logic", *SIAM Journal on Computing* **6(3)**, 467–480.

Moss, L., 1991, "Completeness theorems for logics of feature structures", Indiana University Logic Group Preprint No. IULG-91-2. To appear in *Proceedings of the MSRI Workshop on Logic From Computer Science*, Y. Moschovakis, ed., Berlin: Springer Verlag.

Passy, S. and Tinchev, T., 1985, "PDL with data constants", *Information Processing Letters* **20**, 35–41.

Passy, S. and Tinchev, T., 1991, "An essay in combinatory dynamic logic", *Information and Computation* **93**, 263–332.

Pereira, F and Shieber, S., 1984, "The semantics of grammar formalisms seen as computer languages", pp. 123-129 in *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational*

Linguistics, Stanford, California.

Pollard, C. and Sag, I., *Information-Based Syntax and Semantics: Volume 1 - Fundamentals*, CSLI Lecture Notes **13**, Stanford.

Pratt, V., 1979, "Models of program logics", pp. 115-122 in *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science.*

Prior, A., 1967, *Past, Present and Future*, Oxford: Oxford University Press.

Reape, M., 1991, *An Introduction to the Semantics of Unification-Based Grammar Formalisms*, DYANA deliverable R3.2.A, Edinburgh: Centre for Cognitive Science.

Robinson, R., 1971, "Undecidability and nonperiodicity for tilings of the plane", *Inventiones Math.* **12**, 177–209.

Roorda, D., 1991, "Dyadic modalities and Lambek calculus", to appear in *Diamonds and Defaults*, M. de Rijke, ed., Studies in Logic, Language and Information, Dordrecht: Kluwer.

Rounds, W. and Kasper, R., 1986, "A complete logical calculus for record structures representing linguistic information", in *Proceedings of the 15th Annual Symposium on Logic in Computer Science*, Cambridge MA.

Schild, K., 1990, "A correspondence theory for terminological logics: preliminary report", in *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia.

Shieber, S., 1986, *An Introduction to Unification-Based Approaches to Grammar*, CSLI Lecture Notes **4**, Stanford.

Sistla, A. and Clarke E, 1985, "The complexity of propositional linear temporal logics", *J. Assoc. Comput. Mach.* **32**, 733-749.

Smolka, G., 1989, "Feature constraint logics for unification grammars", IWBS Report 93, Stuttgart: IBM Deutschland.

van Benthem, J., 1984, "Correspondence theory", in *Handbook of Philosophical Logic* **2**, D. Gabbay and F. Guenthner, eds., Dordrecht: Reidel.