Centrum voor Wiskunde en Informatica

REPORT *RAPPORT*

Why combine logics?

P. Blackburn and M. de Rijke

Computer Science/Department of Software Technology

CS-R9533 1995

Report CS-R9533
ISSN 0169-118X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Why Combine Logics?

Patrick Blackburn[1] and Maarten de Rijke[2]

[1] *Computerlinguistik, Universität des Saarlandes*
Postfach 1150, D-66041 Saarbrücken, Germany
patrick@coli.uni-sb.de
[2] *Dept. of Software Technology, CWI, P.O. Box 94079,*
1090 GB Amsterdam, The Netherlands
mdr@cwi.nl

Abstract

Combining logics has become a rapidly expanding enterprise that is inspired mainly by concerns about modularity and the wish to join together tailored made logical tools into more powerful but still manageable ones. A natural question is whether it offers anything new over and above existing standard languages.

By analysing a number of applications where combined logics arise, we argue that combined logics are a potentially valuable tool in applied logic, and that endorsements of standard languages often miss the point. Using the history of quantified modal logic as our main example, we also show that the use of combined structures and logics is a recurring theme in the analysis of existing logical systems.

AMS Subject Classification (1991): 00A71, 03B45
CR Subject Classification (1991): F.4.0, F.4.1, I.2.4
Keywords & Phrases: Combinations of logics, complex structures, mathematics of modeling, modularity, modal and temporal logic, representation languages, transfer theorems.
Note: This report will appear in a special issue of *Studia Logica* on Combination of Logics, guest-edited by Dov Gabbay and Fiora Pirri.

1. Introduction

Combined logics have recently been attracting interest. Many applications of logic concern composite domains. In such cases it may be natural to work with combined languages, whose sublanguages are tailored to the requirements of the various subdomains. Or perhaps one is faced with the task of analysing a complex pattern of inference. Decomposing it into the interaction of simpler, more specialised, patterns may be the key to success. In general, whenever a problem offers some notion of 'modularity' or 'granularity', it is tempting to use a divide and conquer strategy, and the idea of combining logics arises naturally.

This idea is open to criticism. Does it offer anything new? There are many standard logical systems suitable for talking about composite structures; for example, quantified modal logic and sorted first-order logic. It makes little sense (a critic might argue) to dream up 'combined logics' when such tools are available.

Some of the objections are justified. Logical combination is a relatively new idea; it has not yet been systematically explored, and there is no established body of results and techniques. Nonetheless, we shall argue that the critical reaction is misguided. The purpose of the present paper is to show that combined logics are a potentially valuable tool in applied logic, and that endorsements of standard languages often miss the point.

Our argument takes the following form. We begin by introducing three types of composite structure — refinement structures, classification structures and fully fibred structures — together with various combined languages for talking about them. As we introduce each system we consider the likely response of the critic: in particular, we note the standard languages that (on the face of it, anyway) are suitable competitors. We then examine the standard approaches more closely. We argue that although one *can* work with such approaches there is no reason to think that one *should*, and indeed, good reasons for believing one *should not*. We then change tack. Using the history of quantified modal logic as our main example, we point out that the use of combined structures and logics is a recurring (and revealing) theme in the analysis of existing logical systems. We conclude the paper with a sketch of the existing literature and note possible research directions.

For the most part we consider combinations of propositional modal languages. This largely reflects our interest in applied logic; propositional modal languages are playing an increasingly important role in subjects as diverse as computational linguistics and concurrency theory. Moreover, their syntactic simplicity make them a natural medium for logical combination. The requisite modal background can be found in any standard text (Popkorn (1994) is particularly recommended), and we write $\mathbf{M}, n \models \phi$ for ϕ is true in the (Kripke) model $\mathbf{M}$ at node n.

The perspective of *modal correspondence theory*, and in particular the *standard translation* of modal logic into first-order logic, will influence the later discussion. For example, suppose we are working with the modal language $\mathcal{L}$ containing a single unary modality $\langle R \rangle$ and built over an *Atom* indexed collection of atomic sentences. A model $\mathbf{M}$ for $\mathcal{L}$ consists of a non-empty set $\mathcal{W}$, a binary relation $\mathcal{R}$ on $\mathcal{W}$, and a unary relation $\mathcal{P}_\alpha$ for every element α of *Atom*. In the modal tradition such structures tend to be called 'Kripke models' — but of course they are perfectly ordinary relational structures, and can be talked about using a first-order language containing a binary relation symbol R and a unary relation symbol P_α for every α in *Atom*. There is a straightforward translation of $\mathcal{L}$ into this first-order language, the standard translation:

$$\begin{aligned}
ST(p_\alpha) &= P_\alpha x, \\
ST(\neg\phi) &= \neg ST(\phi) \\
ST(\phi \wedge \psi) &= ST(\phi) \wedge ST(\psi) \\
ST(\langle R \rangle \phi) &= \exists y\, (xRy \wedge [y/x]ST(\phi))
\end{aligned}$$

In the first clause, p_α is the atomic symbol indexed by α, and P_α is the corresponding first-order predicate symbol. In the final clause, y is the first variable (in some fixed enumeration of the first-order variables) that is different from x and does not occur in $ST(\phi)$, and $[y/x]$ means uniformly substitute y for all free occurrences of x. Note that for any modal formula ϕ, $ST(\phi)$ contains exactly one free variable, namely x. It is clear that for all models $\mathbf{M}$, all modal formulas ϕ, and all elements w of $\mathcal{W}$,

$$\mathbf{M}, w \models \phi \text{ iff } \mathbf{M} \models ST(\phi)[w],$$

where $[w]$ means assign the element w to the free variable of the formula $ST(\phi)$. In short, propositional modal languages are essentially first-order fragments in an elegant (if not particularly deep) disguise. The standard translation extends straightforwardly to first-order modal logics, thus modal analyses can be seen as a way of picking out various first-order fragments; see van Benthem (1984) for further discussion. As will become clear, this view of modal logic influences the way we think about combined logics.

2. Combining logics

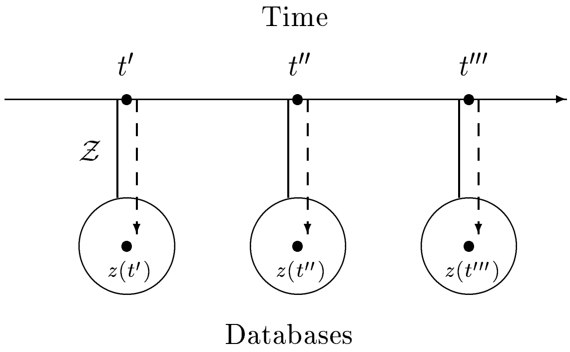
We begin by examining a number of applications where combined logics arise. Our discussion focuses on three commonly occurring types of composite structure — *refinement structures*, *classification structures* and *fully fibered structures* — and the intuitions they help capture.[1] In addition, for each type of structure we introduce a suitable combined language, and note which standard language(s) are potentially applicable.

We begin with refinement structures. Consider the problems raised by reasoning about temporal databases — for example, the 'belief states' of an assembly line robot over an eight hour period. Such belief states are a collection of facts (arm position, component position, and so on) stored in a database. As databases can be regarded as relational structures, there are many candidate tools (for example, first-order logic and various doxastic logics) for reasoning about this component. However the robot's beliefs will typically change over time. Some of these changes may be more or less cyclic (for example, arm position), but others are less predictable (the placement of components on the assembly line can never be perfect, and the robot must be capable of dealing with a reasonable range of variation) so there is a non-trivial temporal reasoning component to the task as well. Now, there should be little difficulty in finding a suitable model of the system's assumed temporal structure and one can probably find (or easily devise) a temporal logic for performing the temporal inferencing. But the reasoning task that faces us is neither purely temporal nor purely doxastic. What is its logical characterisation? The required account — whatever it may be — must in some sense have a composite character, for the two modes of reasoning are interleaved.

When reasoning about designed systems such as the assembly line robots, it is probably best to think of the problem in structural terms. (After all, one is likely to have a very complete system description — perhaps even a full formal specification.) The key question to pose is: how is the flow of information in the system regulated? That is, it is not enough to ask how the databases or the temporal representation are stored, one must know how the various components are linked, and what function each part plays in the whole. In many cases the structures may be connected in highly non-trivial ways, and the best way of looking at it may well be couched in terms of communication: structures receiving and sending information from and to each other, while adhering to various access restrictions.

In the robot example, however, the information flow may well be straightforward. Let us assume that each database is simply a read-only 'file' indexed by some timestamp. That is, we will view them as passive information stores, there to be consulted from the temporal structure: they themselves contain no temporal information, nor do they have 'access rights' to the temporal level. To put it another way, we are thinking of each database as an *expansion* or *refinement* of the temporal nodes, as the following diagram suggests:

[1] The terminology is taken from Blackburn and de Rijke (1994).



Here a temporal structure is *refined* by a collection of belief structures. (Technically, we have fibered a temporal structure over a collection of belief structures.) Above, there is a temporal structure $(\mathcal{T}, <)$, where $<$ is a linear ordering of $\mathcal{T}$. Below, there is a collection of belief structures. (These are triples $(\mathcal{W}, \mathcal{R}, \{\mathcal{P}_\alpha\}_{\alpha \in Atom})$ where $\mathcal{W}$ is a set of belief states, $\mathcal{R}$ is a transitive ordering of $\mathcal{W}$, and each $\mathcal{P}_\alpha$ is a unary predicate on belief states that tells us how the atomic information is distributed.) A function $\mathcal{Z}$ links the two domains: each time t is associated with a belief model $\mathcal{Z}(t)$ representing the state of the database at that point. Note that each $\mathcal{Z}(t)$ contains a distinguished node $z(t)$.

The logics of such refinement structures have been explored by Finger and Gabbay (1992, 1995). The first task is to find a suitable description language. These structures obviously provide a semantics for both a temporal language $\mathcal{L}^T$ (for example, the usual language of tense logic with operators P and F for scanning the past and future respectively) and a doxastic language $\mathcal{L}^B$ (for example, the unimodal language mentioned in the paper's introduction), but how should these be combined? The refinement intuition provides a clear answer. As each database is to be thought of simply as 'information available by lookup', the following language is natural: use the $\mathcal{L}^B$ sentences as the atomic sentences of $\mathcal{L}^T$. That is, build the $\mathcal{L}^T$ sentences in the usual way, but out of *structured* atomic sentences, namely the $\mathcal{L}^B$ sentences. Let's call this language $\mathcal{L}^T(\mathcal{L}^B)$, the language $\mathcal{L}^T$ *layered over* the language $\mathcal{L}^B$. We evaluate the sentences ϕ of $\mathcal{L}^T(\mathcal{L}^B)$ language in the obvious way. Suppose we want to evaluate ϕ at a time t. In general, ϕ will contain occurrences of temporal operators, and we can start evaluating ϕ and its subformulas in the usual manner. Eventually we come to what (in a standard language) would be the 'atomic' level; but as our 'atoms' are structured we have further work to do. We zoom in from the time of evaluation to the distinguished node in the associated belief model and start evaluating the 'structured atom' there in the usual way.

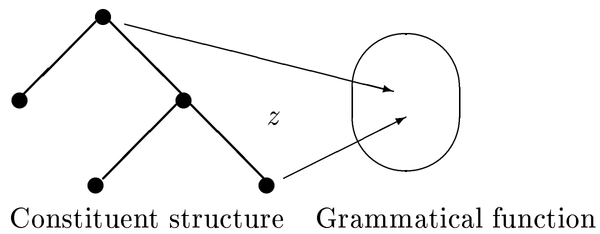
In short, we now have a temporal database language which decomposes into temporal and doxastic languages in the simplest manner imaginable. The layering process (which prevents temporal operators from occurring under the scope of belief operators) neatly encapsulates the hierarchical nature of the information flow between the two subdomains. The simplicity of this form of combination gives rise to pleasant logical properties. Finger and Gabbay (1992, 1995) show that many properties (such as completeness and decidability) can be lifted from the component structures to the combined one. The main technique used to prove such results rests on two ideas: to *hide* low level information from the doxastic structure while working in the top level temporal structure, and to ensure that one has the ability to move freely around in the top level temporal structure. The first idea is implemented by viewing low level doxastic formulas as atomic formulas while working in the top level structure, and to delay *unpacking* or evaluating these atoms until one moves down to the low level doxastic structures. The second idea is implemented by making sure that the top level temporal language is equipped with the universal modality A:

$$\mathbf{M}, n \models A\phi \ \text{ iff } \ \mathbf{M}, n' \models \phi, \quad \text{for all nodes } n'.$$

Intuitively, this operator (together with the dual 'somewhere' operator E) enables the top level to search for hidden inconsistencies emerging from the bottom level. This enables 'layered' completeness results to be proved: build verifying models for consistent sentences by making use of the completeness theorems for the two component languages. The presence of the universal modality ensures that the resulting models can be glued together without difficulty.

Are there standard alternatives to layered languages? Yes, two obvious ones. First, one can view the entire refinement structure as a single relational structure: as domain of quantification take the disjoint union of all the temporal and belief structure nodes, and as relations the temporal precedence relation, the belief structure relations, a unary relation $\mathcal{Z}(t)$ for each $t \in \mathcal{T}$, and all the distinguished points. We can then talk about these structures using a first-order language of appropriate signature. (For example, the first-order language containing the non-logical symbols of $\mathcal{L}^T$'s correspondence language, the non-logical symbols of $\mathcal{L}^B$'s correspondence language, a unary predicate Z_t for picking out each $\mathcal{Z}(t)$, and constants for the distinguished points.) There are obvious variations on this theme; for example, we could use a sorted version of this language by introducing distinct variables for the temporal and belief structure nodes. The second choice is equally transparent. Fibered structures are simply models for first-order modal languages. The canonical choice in the present case is the first-order modal language built from $\mathcal{L}^T$ together with the correspondence language of $\mathcal{L}^B$.

Let us consider how logics may be combined to talk about *classification structures*. The clearest example of classification structures is provided by Lexical Functional Grammar (Kaplan and Bresnan (1982)), one of the most influential theories of current generative syntax. LFG is virtually unique in generative grammar in viewing sentences structure in terms of two independent levels of information: constituent structure and grammatical function. Constituent structures are essentially the parse trees of context free grammars. Grammatical function concerns such concepts as subject, object, and indirect object, and is represented using a collection of finite partial functions called a feature structure. However, the crucial ideas in LFG revolve around the interrelationship between these two domains:



Here we have the basic LFG architecture: a finite tree linked by a partial function $\mathcal{Z}$ to a feature structure. Syntactic explanations in LFG use the *classifications*[2] that grammatical relations induce on trees: although tree nodes t and t' may be distinct, if $z(t) = z(t')$ they are functionally identical. That is, the task of the grammatical function information is to induce equivalence relations on the level of constituent structure, and these equivalence relations are the basic tool of syntactic explanation in LFG. Note that the intuitions underlying classification and refinement are quite distinct: the role of the feature structure in LFG is not in any sense to 'refine' the information on tree nodes. Indeed, a key point of classification structures is that the classifier (here, the feature structure) provides a global information store accessed by the classifyee (here, the tree). Classification is about synchronising two levels of information, a more subtle notion of information flow than refinement.

The key expressive power that we need when working with classification structures is the ability to enforce this synchronisation across levels. LFG implements this using phrase structure rules annotated with equations. For example, if the phrase structure rule VP $\longrightarrow$ V is annotated with the equation $\uparrow = \downarrow$, this means that if we move from a tree node t that is marked V up to its mother node t' (marked with a VP) and then zoom into the feature structure, we arrive at the same point we would have reached by zooming in directly from t. This can be formulated in a combined language quite naturally.[3] As our component languages take a suitable propositional modal language $\mathcal{L}^{Tree}$ for talking about parse trees (for example, the language explored in Blackburn, Meyer-Viol and de Rijke (1994)) and a similar language $\mathcal{L}^{Feat}$ for talking about feature structures. How should we put the languages together? The crucial requirement, the ability to express synchronisation, can be implemented by borrowing the idea of intersective program constructors from Propositional Dynamic Logic (PDL, see Harel (1984)). That is, we allow the tree modalities and the feature structure modalities to be combined inside complex, PDL style, modal operators. In this language, the effect of the earlier LFG annotated rule is captured by $\langle (up; zoom\_in) \cap zoom\_in \rangle \top$.

The more complex form of interaction between the component structures modeled by classification structures shows up in the logical behaviour of these languages. For example, it is fairly simple to encode well-known undecidable problems using logics of classification structures, even though the component logics may be extremely simple: see Blackburn and de Rijke (1994) for an encoding of Type 0 grammars. Completeness results (when obtainable) often require non-standard tools, for example Gabbay (1981) style irreflexivity rules.

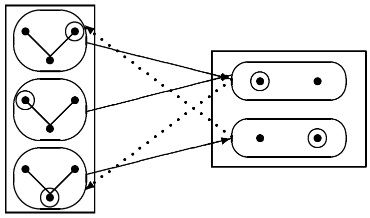
However, note that we are not forced to use this combined logic; once more there are standard alternatives. For example, one can view classification structures as structures for first-order (or: sorted first-order) logic with equality, and use the equality symbol to enforce the desired synchronisation between the subdomains. In some cases it is also possible to use first-order modal logic. To see this, note that when the function z from the tree to the feature structure is total, there is a simple way to view classification structures as fibered structures. Proceed as follows. For each node t in the tree $\mathcal{T}$, let $\mathcal{C}_t$ be the feature structure augmented by the distinguished point $z(t)$. Let $\mathsf{C} = \{\mathcal{C}_t : t \in \mathcal{T}\}$. Now fibre $\mathcal{T}$ across C: define $\mathcal{Z}(t) = \mathcal{C}_t$, and we are back in the setting of fibered structures, the home of first-order modal logic. Indeed, we are in a relatively well behaved part: as all the $\mathcal{C}_t$ are built over the same nodes, C is a class of constant domain structures.

[2]The terminology is borrowed from Seligman (1990), where the idea of abstract classification systems is systematically explored.

[3]For a more detailed discussion of this example, see Blackburn and Gardent (1994).

In the two examples we have looked at so far there is an obvious directionality to information flow. In refinement structures one component controls the flow; the other is merely a passive store. Even with classification structures the information flow is essentially one way: one structure induces an equivalence relation on the other. Recently, Dov Gabbay has advocated the idea of *fully fibering* two sets of semantic entities over each other; see Gabbay (199-). A fully fibered structure consists of two classes of models, each class with its own language, plus a function between the classes that enables you to evaluate formulas belonging to one language inside the structures belonging to the other. Unrestricted information flow is possible in such structures; no hierarchy is built in.

As an example, we fully fiber finite trees and finite equivalence relations over each other. We assume that we have two unimodal languages, $\mathcal{L}^{Tree}$ for talking about trees, and $\mathcal{L}^{Equiv}$ for talking about equivalence relations.



Fibering a tree and an equivalence relation

A *model-state* pair is a pair $(\mathbf{M}, s)$ where $\mathbf{M}$ is a model based on a finite tree or on a finite equivalence relation, and s is an element of $\mathbf{M}$. Let M_T, M_E be non-empty sets of model-state pairs whose first component is a finite tree or a finite equivalence relation, respectively, and such that if $(\mathbf{M}, s) \in \mathsf{M}_T \cup \mathsf{M}_E$ and $s' \in \mathbf{M}$, then $(\mathbf{M}, s') \in \mathsf{M}_T \cup \mathsf{M}_E$. The full (or mutual) fibering is performed as follows. Let F be a pair of functions (F_T, F_E) with $F_T : \mathsf{M}_T \to \mathsf{M}_E$ and $F_E : \mathsf{M}_E \to \mathsf{M}_T$ such that model-state pairs which are mapped onto each other agree on all atomic symbols common to both languages. Finally, for F a fibering function, the F-fibered structure over M_T and M_E is the triple (W_F, R_F, V_F) such that

1. W_F is $\mathsf{M}_T \cup \mathsf{M}_E$,
2. R_F is $\{((\mathbf{M}_1, s_1), (\mathbf{M}_2, s_2)) : \mathbf{M}_1 = \mathbf{M}_2 \text{ and } s_1 R s_2\}$,
3. V_F is simply the union of the component valuations.

$\mathcal{L}^{Tree}$ sentences are interpreted in M_T (and $\mathcal{L}^{Equiv}$ sentences in M_E) standardly. To evaluate a tree sentence while evaluating in M_E, we apply F to the current model-state pair, and continue evaluating at the associated model-state pair in M_T (and similarly when we hit an $\mathcal{L}^E$-subformula while evaluating in M_T).

Fully fibered structures arise in more sophisticated databases (see Finger and Gabbay 1995). On the technical side, iterated hide-and-unpack techniques have proved useful in establishing completeness results (Fine and Schurz 1991, Kracht and Wolter 1991), although other approaches such as Seligman (1995)'s bilingual sum have also been used to lift properties of the component logics to the combined ones. As with the earlier refinement and classification structures, the practical need for fully fibered structures and logics seems fairly well established. But, again, what is wrong with standard approaches? First-order modal logic is not obviously applicable here, but ordinary first-order logic certainly is. Why not use it?

3. Standard methods?

Is there anything to be gained by working with standard languages? Let's return to the refinement example. We defined a language $\mathcal{L}^T(\mathcal{L}^B)$, the language of propositional tense logic layered over a unimodal belief language, to talk about simple temporal databases. But there is an obvious alternative: use the first-order language built from (a) all the non-logical symbols of correspondence languages of $\mathcal{L}^T$ and $\mathcal{L}^B$; (b) for each $t \in T$, a one place predicate symbol Z_t; and (c) for each $t \in T$ a constant symbol z_t. The intended interpretation of the Z_t and z_t should be clear: each constant z_t names the distinguished point $z(t)$; and Z_t is interpreted by $\{w \in W : \mathcal{Z}(t) = (\mathcal{W}, \mathcal{R}, \{\mathcal{Q}_{p_\alpha}\}_{\alpha \in Atom})\}$. One can quibble over the details (for example, it might be useful to add a unary predicate T to pick out the elements of $\mathcal{T}$) but the basic point should be clear: it is straightforward to regard fibered structures as relational structures, and once this is done, one can work with the first-order language of that signature. Let us call such a language a *first-order language of fibered structures*, or FOLFS.

But simply using FOLFS hardly counts as an analysis of *refinement*. Practically any structure one is likely to encounter in the course of applied logic (or indeed, while doing mathematics) can be regarded as a relational structure — the very generality of this observation ensures that more work remains to be done. *Logical* analysis has twin goals: to arrive at a suitable semantic analysis of a problem (that is, to isolate the kinds of mathematical structures that underly the problem) and *in addition*, to devise an appropriate language for dealing with these structures. In many cases, perhaps most, the required mathematical structures will turn out to be some class of relational structures.[4] But this fact does not ensure that the first-order language of this signature is the best syntactic choice! For the analysis of mathematical applications, first-order approaches may simply be too weak. (Better analyses may be provided by infinitary languages or languages with non-standard quantifiers; see Barwise (1985) for further discussion.) On the other hand, for applications in computer science or computational linguistics, the first-order language may well be too strong. (Depending on the demands of the problem, the Horn clause fragment, the equational fragment or (via the standard translation) the modal fragment may well be better choices.) The first-order language language provides an important starting point — but it would hardly be an exaggeration to describe the task of the applied logician as being to explore exactly what *departures* from this language are required.

In the case of refinement such considerations are particularly relevant. We are trying to model a simple intuition: refinement of atomic information. The fibered structures themselves capture some, but not all, of this intuition. This should be clear. As we have seen, classification structures can be viewed as fibered structures; and (by taking a disjoint union) one can view refinement structures as classification structures. Nonetheless, the intuitions underlying classification and refinement are very different. We modeled refinement *logically*: a blend of semantic and syntactic ideas was used.

Use of the entire FOLFS for simple temporal databases seems to offer no comparable advantages. Given the simplicity of the example, it is natural to view the task at hand as being to

[4]Though by no means always. For some problems in computer science, for example, category theory seems to be a more appropriate setting.

find fragment(s) of the FOLFS that capture the refinement intuition. The important question is: which fragment? And how should one go about finding it? The layered language approach can be seen (via correspondence theory) as providing a principled answer to these questions.

Much the same criticism can be leveled against first-order analyses of classification or full fibering. Classification attempts to analyse problems in terms of synchronisation of information between structures, while full fibering attempts to do so in terms of free communication between structures. There is no good reason for thinking that first-order languages are the best way (or even a particularly good way) for getting to grips with these ideas. In short, the criticism that faces the first-order approach is not that it's wrong, but that it's all-too-trivially applicable.

With this criticism in mind, let us turn to the second standard approach: first-order modal languages. As we have seen, such languages can be used to talk about refinement structures and (in some cases) classification structures. On the face of it, they may seem a better choice than full first-order logic. For a start, from the viewpoint of correspondence theory they are simply (lightly disguised) fragments of the relevant FOLFS, thus they have some claim to being a fine-grained approach. Moreover, first-order modal languages are the languages usually associated with fibered structures — on the face of it, they have a special claim to consideration. Nonetheless, first-order modal logic doesn't seem to be a good way of getting to grips with the ideas underlying either refinement or classification.

Consider the first-order modal approach to our database example. The required language is the following: as atomic formulas take all the atomic formulas of $\mathcal{L}$'s correspondence language, and then close up under $\neg$, $\wedge$, $\exists$, F and P in the usual way. The interpretation is standard. But while one could use this language, it is difficult to see why one would want to. The key intuition underlying refinement is that there is a very limited, hierarchical, interaction between the information in the individual databases and the temporal structure. Database information is regarded simply as a refinement of the temporal nodes. This fact is obscured if we view refinement structures through the lens of first-order modal logic. First-order modal logic allows us to express complex interactions of database and temporal information, for we can arbitrarily iterate quantifiers and tense operators: $\exists x F \cdots \exists y F\, \phi$. But if a problem can be successfully analysed using a layered language, it means that we can actually make do with the fragment of first-order modal logic that arises as the image of the following translation:

$$\begin{array}{rcl} MT(\phi) & = & ST(\phi),\ \text{if } \phi \text{ belongs to } \mathcal{L}^B \\ MT(\#\phi) & = & \#MT(\phi),\ \text{if } \# \text{ is a unary connective} \\ MT(\phi \oplus \psi) & = & MT(\phi) \oplus MT(\psi),\ \text{if } \oplus \text{ is a binary connective} \end{array}$$

The key point to note about this fragment is its simplicity. In particular, quantifiers never take wide scope over modal operators: it is not possible to 'quantify in', the source of most difficulties in first-order modal logic.[5] We are using (the modal fragment of) a fragment in which these difficulties cannot even arise: $\mathcal{L}^T$ layered over the correspondence language of $\mathcal{L}^B$. Thus one could certainly give 'a first-order modal treatment of refinement' — but the first step in such a treatment would be to to throw most of the standard language away.

What of first-order modal approaches to classification? Recall that when the function z between the classifier and classifyee is total, there is a simple way to regard classification structures as fibered structures. Moreover, from the point of view of first-order modal logic,

[5]These problems are discussed in more detail in the following section.

the resultant structures are particularly natural, having constant domains. So first-order modal logic seems a possibility. Let's see how it fares.

Consider again the example of LFG style annotated phrase structure rules. We showed that we could capture their effect by combining the tree and feature structure modalities PDL style — that is, inside the diamonds. Then, using the intersection constructor we can formulate 'path equations':

$$\langle (up; zoom\_in) \cap zoom\_in \rangle \top.$$

How could we capture this in first-order modal logic? The ideal way would be by means of the following equation:

$$z_t = \langle up \rangle z_t.$$

(Recall that the z_t are constants naming the distinguished points.) This equation is simple, and clearly analogous to the PDL expression. Unfortunately, it's nonsense: the expression on the right hand side is not a term. Even in the constant domain setting, first-order modal logic has no mechanism for directly stating equalities between the nodes in different states. Rather, one must proceed indirectly:

$$\exists x \, (x = z_t \wedge \langle up \rangle \exists y \, (y = z_t \wedge x = y)).$$

So one can formulate classifications as theories in first-order modal logic. But embedding quantifiers under modal operators to navigate around classification structures is hardly the most natural way to proceed. Classification revolves around the idea of expressing path equations, and the combined approach focuses directly on this requirement. Using first-order modal logic requires one to first think about classification in terms of fibered structures, and then to reformulate the central intuition clumsily. Even a straight first-order approach approach seems preferable: there one can at least state the required equations directly.[6]

Let us summarise the discussion so far. Applying logic is not usually going to be a straightforward matter of dusting off well-known tools. Rather, the process runs more like this. First, one works out which are the mathematical structures underlying the problem; these will often be complex, composite structures. Secondly, the function(s) played by the various subcomponents must be identified. Although all the different components may be 'just' relational structures, they may have very different roles to play in regulating the flow of information. Thus although one usually can invoke standard approaches, it is probably more sensible to think about (a) the descriptive needs of each component separately and (b) how these are best to be fitted together. This is as much an art as a science — but certainly a blind insistence that only the standard languages are appropriate hardly seems a sensible response. In a nutshell: we should combine logics, because applying logic suggests that we need to.

4. Analyzing syntax

The reader who has followed the argument so far may be willing to agree to this much: if one wants to use logical languages to talk about the composite structures that arise in certain

[6]It's worth remarking, however, that it might be interesting to think about the classification structures in terms of the functional frames discussed in the next section.

applications, perhaps there is a point to combining logics. However, such a reader may suspect that this is the *only* place in which logical combination is likely to prove useful. The purpose of the present section is to dispel this notion. While the most convincing examples of logical combination reflect applied concerns, applying logic is rarely a straightforward business. It can be approached in many ways, and some of these give rise to combined logics in a more indirect fashion than anything we have yet encountered.

The examples of applied logic considered so far were model theoretically driven: one started with a class of intended models of an obviously composite nature (for example, temporal structures fibered over belief models), and only then turned to syntactic matters (finding a suitably expressive language, axiomatising the theory of the intended structures, investigating computability issues, and so on). This semantically driven approach is important, but one can begin with syntax instead. For example, thinking about some problem may lead to the conviction that a certain type of language is an especially appropriate analytic tool (a classic example is Carnap's (1947) advocacy of higher order intensional logic for the analysis of natural language expressions), or that an adequate logical analysis of some concept must respect certain principles (for example, the relevant logicians' insistence that only relevant premises may contribute to proofs). In this form of applied logic, the idea of combining logics is less self-evidently present. Nonetheless, here too, it emerges quite naturally: semantic analyses in terms of composite structures may reveal the combined nature of existing systems.

We consider the development of quantified modal logic. This example is a fascinating one for at least three reasons. Firstly, it leads with seeming inevitability to the kinds of structures we have previously discussed. Secondly, the questions we have raised concerning how the flow of information between structures is to be regulated (that is, which entities in which sub-structures can access one another, and how) are raised in a very direct fashion and underly virtually all the technical developments of the subject. And last comes the irony: the example is not usually perceived in terms of combined logics at all! Let us examine its history through the lens of combined logics.

The way quantificational modal logic is usually presented (and indeed, the way its pioneers seem to have conceived of it) is as follows. Suppose we wish to analyse some domain (for example, the semantics of a certain class of natural language expressions). We observe that some features of the domain can be dealt with in a first-order (or higher order) classical language, but that because of certain phenomena (notably, intensional expressions such as 'believes that' or 'knows that', and mechanisms such as tense which are sensitive to utterance time) a more adequate analysis eludes us. Now comes the syntactic temptation. Intensional and temporal devices can be regarded as 'sentential operators': somehow they take a sentence and transform its meaning, thus syntactically they seem much on a par with negation. Obeying Oscar Wilde's dictum[7] we extend the classical language with a number of unary operators, formulate plausible principles governing the new operators and their interactions — and lo-and-behold! — we have the requisite modeling tool.

Or do we? For all its apparent simplicity, the above procedure is highly problematic.[8] While we now have a notation which (putatively) allows us to formulate intensionalised or temporalised statements of classical logic, it is far from clear how to formulate the required

[7] "The only way of getting rid of temptation is to yield to it."

[8] As the more sophisticated pioneers were aware. For example, Carnap knew of the difficulties underlying his semantic program, and tackled many of them successfully; Carnap (1947) still repays careful study.

logics of these richer languages. And if this is unclear, what genuine claim can the notation lay to being a formalisation of intensionality or temporality?

The principle difficulties are not hard to find: how should the new operators interact with the classical first-order language? Because quantifiers can take wider scope than operators, we can 'quantify in', binding variables in an intentional or temporal context. This gives rise to many difficulties. Perhaps the best known concerns the validity of the Barcan formula:

$$\Diamond \exists x\, \phi \rightarrow \exists x\, \Diamond \phi.$$

Should this be accepted as an interaction principle? It embodies a non-trivial claim about the flow of information between domains, namely if at some (possibly distinct state) there is an individual satisfying ϕ, then an individual can be found *in the present state* which satisfies ϕ elsewhere. Historically, two problems made such claims difficult to solve: intuitions about what the 'correct answer' was, could (and often did) conflict; and there was no compelling technical analysis to provide a court of appeal. Indeed, one of the reasons for the widespread acceptance of Kripke semantics was precisely that it provided an intuitively clear technical framework in which to conduct such discussions.

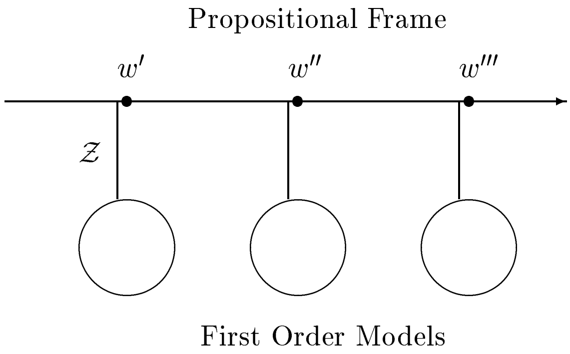
But what has all this to do with *logical combination*? Simply this. Although the pioneers of quantificational modal logic don't seem to have thought of its development as a problem of logical combination,[9] the problems that faced them (namely, how to merge various forms of intensional reasoning with classical reasoning) can naturally be viewed this way. Moreover, as subsequent history makes clear, it is not at all clear how best to make the required combinations. The 'sentence operator' inspired strategy does not result in a simple 'extension' of classical logic, but in a subtle system that defies straightforward analysis. With the benefit of hindsight (and in particular, correspondence theory), the underlying cause of these difficulties is not hard to find: first-order modal logic combines the modal language for one class of structures with the correspondence language for another. This results in a system containing two very different modes of quantification: classical style, with explicit variables and binding; and modal style, which dispenses with this explicit apparatus in favour of operators. The interplay between the logics is subtle and can give rise to unintended effects. For example, the explicit classical variables can have unintended effects on the implicit modal variables; among other things this leads to difficulties with lambda conversion in the higher order case (see Muskens (1995) for further discussion). Such a mode of combination is certainly interesting — if only because it leads to such difficult puzzles — but it is not really motivated by anything beyond the rather thin sentence operator metaphor. It is certainly not an obviously correct solution, even for the original problems it was intended to solve.

In short, the pioneers were dealing with a problem of logical combination, and did so in an unwittingly brutal fashion. Much of the subsequent history of quantificational modal logic can be seen as the step-by-step clarification of the true depths of the problem. Let us examine the matter a little further.

The single most important development was the arrival of Kripke semantics. This makes it abundantly clear that the technical problem is how best to combine systems of propositional modal logic with systems of quantificational logic. The point hardly needs elaborating. No modern day student trained in propositional modal logic (in particular, the use of frames

[9] There are some interesting exceptions. For example, in his 1984 survey article, Richmond Thomason examines combinations of tense and modality in terms reminiscent of the present discussion.

$(\mathcal{W}, \mathcal{R})$) and first-order logic (in particular, the use of models $(\mathcal{D}, \mathcal{I})$, where $\mathcal{I}$ is the function interpreting the non-logical symbols on $\mathcal{D}$) would be surprised when confronted with the familiar fibred structures for interpreting first-order modal logic:



With such a background, the idea of fibering frames over first-order models seems natural, perhaps even inevitable. And indeed — up to a point — the idea has great explanatory power. Many of the puzzles concerning quantifying in can now be discussed in manner that is both technically and intuitively satisfying. For example, the validity of the Barcan formula reduces to whether or not the collection of first-order models are built over a constant domain.

Nonetheless, the classical fibred analysis of quantified modal logic cannot be considered wholly satisfactory. In spite of the (genuine) advances brought about by the advent of Kripke semantics, James Garson opened his 1984 survey with the following words:

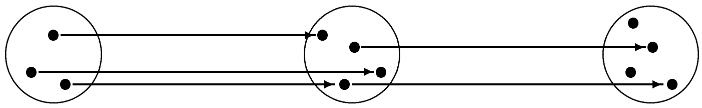
> The novice may wonder why quantified modal logic (QML) is considered difficult. QML would seem to be easy: simply add the principles of first-order logic to propositional modal logic. Unfortunately, this choice does not correspond to an intuitively satisfying semantics. From the semantical point of view we are confronted with a number of decisions concerning the quantifiers, and these in turn prompt new questions about the semantics of identity, terms and predicates. Since most of the choices can be made independently, the number of interesting quantified modal logics seems bewilderingly large.

Indeed, the situation turned out to be far more than bewildering: it became quite dramatic. During the 1980's it was discovered that first-order modal logic abounds in surprising incompleteness results: there are propositional modal logic which are complete with respect to some class of frames, and yet whose first-order companion is not. That is, one can begin with a 'well behaved' (frame complete) propositional modal logic, extend it to a first-order modal system in the obvious way, and end up with a 'badly behaved' frame incomplete logic. The first such results are due to Ono (1983), and Ghilardi (1991) proves that this is the case for *all* propositional modal logics between **S4.3** and **S5**!

What is of interest for the present discussion is the types of combined structures needed to prove such results. Here we will follow van Benthem's (1993) reformulation in terms of orthodox relational structures. (Ghilardi (1991) and Shehtman and Skvortsov (1990) are couched in category theoretic terms.)

The basic ingredient needed are what van Benthem terms *functional frames*. As with the orthodox Kripke semantics, these are composed out of a collection of first-order models —

but we link them up differently. A functional model $\mathbf{M}$ consists of a $\mathcal{W}$ indexed collection of first-order models together with a family of functions $\mathcal{F}$ between the domains of these models. (That is, if $f \in \mathcal{F}$ then for some $w, w' \in \mathcal{W}$, $dom(f) = \mathcal{D}_w$ and $cod(f) = \mathcal{D}_{w'}$.)



That is, whereas in the standard Kripke semantics for first-order modal logic, the models are connected 'from the top' (that is, they just dangle from the fibering propositional frame), in the functional frame semantics the models are linked much more intimately: we have a direct local specification (given by the class of functions) of which individuals in the various substructures a given individual can access.

The satisfaction definition for the modalities is as follows. Let $\mathbf{M}$ be a functional frame, g an assignment of values to variables, and $w \in W$. Then:

$$\mathbf{M}^g, w \models \Box\phi \quad \text{iff} \quad \mathbf{M}^{f \circ g}, w' \models \phi,$$

for all functions $f \in \mathcal{F}$ with domain $\mathcal{D}_w$ and codomain $\mathcal{D}_{w'}$. This satisfaction clause is intrinsically more fine grained than the orthodox semantics in terms of fibered models. As van Benthem emphasizes, this shows up even for the models containing only a single world. Whereas the global fibering semantics validates $\phi(x) \rightarrow \Box\phi(x)$ in such structures, it is easy to construct singleton functional frames which falsify it.

The analysis of first-order modal incompleteness is the most important technical development in the area since the birth of Kripke semantics. And the story is by no means over. At present it isn't clear which of several new technical settings offers the best hope for further advances. What *is* clear is that further progress involves recognising a fundamental point: quantified modal logics are systems in which two logics have been combined in a highly subtle way. The syntactic choices made by the pioneers were not innocent: combining explicit and implicit forms of quantification is a tricky business. Philosophical logicians have traditionally discussed the resulting difficulties with reference to the problems of identity, predication, and so on, and have used tools ranging from free logic to truth-value gaps to try and solve them. It would be interesting to view these issues in information theoretic terms — how exactly is the flow of information regulated? — and to attempt analyses using the idea of logical combination.

To conclude this section, some general remarks. First, the idea of analysing existing logical systems using composite structures is by no means confined to quantified modal logic. To give a recent example, Meyer and Mares (1994) have analysed the notion of entailment using such tools. As they note, 'entailment' is a composite notion: it can be thought of as 'necessary implication'; to understand it one must understand necessity, implication, and how to graft them together. They explore this idea using composite structures: 'galaxies' of relevant implication models linked by an **S4** relation. Their analysis leads them to believe that **NR**, the classic analysis of entailment, is incomplete; moreover, they are able to propose two systems (**CNR** and **CR4**) which have a genuine claim to capturing the logic of entailment.

Second, the style of analysis that proceeds from syntax to semantics is typical of much recent work in computer science and natural language semantics. In the computer science case this is perhaps to be expected: often the syntax of a programming language together with a specification of its behaviour on some architecture has been fixed, and one wishes to find a more abstract mathematical model of its behaviour. (Typical reasons for wanting such models are to investigate issues such as modularity and concurrency.) Much recent work in this vein has led to analyses couched in terms of composite structures; *evolving algebras* (see Gurevich (1991)) are a particularly striking example. What is striking is the way such structures allow successful analyses of warts-and-all accounts of the (often highly intricate) information flows found in *real* programming languages.

A similar trend can be discerned in recent analyses of systems of dynamic semantics for natural language, currently probably the most influential approach to natural language semantics. The basic idea of dynamic semantics is to explain the meanings of natural language expressions in terms of their 'context change potential'. Utterances are viewed very much like the statements of a programming language; they are state transformers. Important work in this tradition includes Discourse Representation Theory (DRT, Kamp and Reyle (1993)), and Dynamic Predicate Logic (Groenendijk and Stokhof (1991)).

However, while the applications in natural language semantics have been impressive, logical understanding of these systems has lagged. It has proved difficult to devise revealing proof methods for dynamic systems. The trouble is that while the syntax of these systems is fixed (for example, orthodox first-order syntax, or DRT syntax), and *part* of the semantics also (namely, a quasi first-order semantics), the dynamic aspects have (arguably) not been completely modeled. Recent work by Vermeulen and Visser (1995) attempts to capture this residue by making systematic use of composite structures. Essentially, one level of structure models content, the other the (changing) context in which representations are built. The heart of the analysis is an account of how change on one level induces change on the other — and how to keep track of what information belongs together. The principle tool used is the Grothendieck construction from category theory.

5. Conclusion

The argument we have presented can be summed up very simply. Combined logics are of interest for at least two reasons: they seem a natural way to approach many problems in applied logic; and when viewed correctly, existing systems may reveal themselves to be combined logics. The practical utility of combined logics seems clear. In particular (a theme that has been implicit throughout the paper) thinking in terms of combined logics may give rise to propositional analyses in cases where (at first glance) explicit quantificational apparatus seems called for. But while their practical utility seems clear, the theoretical underpinnings are largely unmapped. To close this paper, we briefly survey what is known about combinations of propositional modal languages.

Most work in this area has concentrated on so-called *transfer properties*. The setting here is the following. Let $\mathbf{L}$ be a modal logic in a given modal language $\mathcal{L}$. Let $\mathcal{L}'$ be an extension of $\mathcal{L}$ with additional modal operators, and let $\mathbf{L}'$ be the minimal extension of $\mathbf{L}$ to the richer language $\mathcal{L}'$ that includes all the basic axioms and rules of inference that are valid on structures for the richer language. Which properties of $\mathbf{L}$ are inherited by $\mathbf{L}'$? Properties one typically considers include completeness, decidability, finite model property (fmp), and interpolation.

The general message is that virtually all nice properties transfer provided that there is little or no interaction between the modal operators in the original language $\mathcal{L}$ and the operators that are in the extension $\mathcal{L}'$ but not in $\mathcal{L}$. As soon as non-trivial interactions are allowed transfer may fail. The earliest papers dealing with transfer issues (Fine and Schurz (1991), Finger and Gabbay (1992), Kracht and Wolter (1991)) all deal with combining logical systems without interaction. For example, in the first and third of the above papers the authors study the effect of taking together modal logics in disjoint languages. It is shown that completeness, decidability and fmp all transfer from the component logics to the combined one. More recently, a number of authors have looked at transfer results if the component logics are allowed to interact. Wolter (1995) has studied the problem of transferring properties from a modal logic **L** to its minimal tense extension, that is: from **L** to the smallest tense logic (with forward and backward looking operators for each of the operators in the language of **L**) that includes **L**. For many 'natural' extensions of **K4**, completeness and fmp do indeed transfer. But in general, neither completeness nor fmp transfers from a modal logic to its minimal tense extension. Further negative examples in the presence of interaction were obtained recently by Hemaspaandra (1994, 1995). She investigates the effect on the satisfiability problem of enriching modal languages. In particular, she analyzes the effect of enriching modal languages with the universal modality discussed earlier and the reflexive, transitive closure modality. (That is, suppose we have a modality $\langle R\rangle$ that explores a relation $\mathcal{R}$. Enriching this language with a reflexive, transitive closure modality means adding a modality $\langle R^*\rangle$ that explores $\mathcal{R}$'s reflexive, transitive closure.) She shows that the increase in the complexity of the satisfiability problem can be as large as it gets: from NP-complete to highly undecidable; so, in particular, decidability certainly doesn't transfer.

A topic that certainly needs to be investigated further is the algebraic side of modal combination. Although this paper has taken a predominately correspondence theoretic view of modal languages, one of the mathematical reasons why modal logics have proved so interesting is because they can also be viewed as equational theories for Boolean algebras with operators. An important perspective to investigate is thus: how should such theories be combined? The only work in this direction we are aware of is Eiben et al. (1995); clearly much remains to be done.

We conclude the paper on a speculative note. In theoretical computer science the idea of communicating structures has been around for a long time (see Milner (1989) for the background ideas), and the recent trend towards agent oriented and component based technologies in software certainly seem related to the work of this paper. We believe that a formal logical study of these techniques should prove useful in creating and understanding valuable proposals; as we've stressed in this paper, the required logical tools should allow one to be explicit about links between the various domains involved; Blackburn and de Rijke (1994) contain an early proposal.

Acknowledgment. Maarten de Rijke was supported by the Netherlands Organisation for Scientific Research (NWO) (project NF 102/62-356 'Structural and Semantic Parallels in Natural Languages and Programming Languages').

REFERENCES

1. J. Barwise. 1985. Model-theoretic logics: background and aims. In J. Barwise and S. Feferman (eds) *Model-Theoretic Logics*, Springer, New York, pages 2–23.
2. J. van Benthem. 1984. Correspondence theory. In D. Gabbay and F. Guenthner (eds) *Handbook of Philosophical Logic (II)*, Reidel, Dordrecht, pages 167–247.
3. J. van Benthem. 1993. Beyond accessibility. In M. de Rijke (ed) *Diamonds and Defaults*. Kluwer Academic Publishers, Dordrecht, pages 1–18.
4. P. Blackburn and C. Gardent. 1994. A specification language for Lexical Functional Grammars. To appear in *Proceedings of the 7th Conference of the European Chapter of the Association of Computational Linguistics*, Dublin.
5. P. Blackburn, W. Meyer-Viol, and M de Rijke. 1994. A proof system for reasoning about constituent structures. Submitted manuscript.
6. P. Blackburn and M. de Rijke. 1994. Zooming in, zooming out. Report CS-R9462, CWI, Amsterdam.
7. R. Carnap. 1947. *Meaning and Necessity.* The University of Chicago Press, Chicago.
8. K. Fine and and G. Schurz. 1991. Transfer theorems for stratified multi-modal logics. In *Proceedings of the Arthur Prior Memorial Conference*. To appear.
9. M. Finger and D.M. Gabbay. 1992. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information* 1:203–233.
10. M. Finger and D.M. Gabbay. 1995. Notes on several methods for combining temporal logic systems. To appear.
11. D.M. Gabbay. 1981. An irreflexivity lemma with applications to axiomatisations of conditions on linear frames. In U. Mönnich (ed), *Aspects of Philosophical Logic*, Reidel, Dordrecht, pages 67–89.
12. D.M. Gabbay. 199-. *Fibered Semantics*. Manuscript, Imperial College.
13. J. Garson. 1984. Quantification in modal logic. In D. Gabbay and F. Guenthner (eds) *Handbook of Philosophical Logic (II)*, Reidel, Dordrecht, pages 249–307.
14. G. Gazdar, E. Klein, G. Pullum, I. Sag. 1985. *Generalised Phrase Structure Grammar*. Basil Blackwell, Oxford.
15. S. Ghilardi. 1991. Incompleteness results in Kripke semantics. *Journal of Symbolic Logic*, 56:517–538.
16. J. Groenendijk and M. Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy*, 14:39–100.
17. Y. Gurevich. 1991. Evolving algebras; a tutorial introduction. *Bulletin EATCS* 43:264–284.
18. D. Harel. 1984. Dynamic logic. In D. Gabbay and F. Guenthner (eds) *Handbook of Philosophical Logic 2*, Reidel, Dordrecht, pages 497–604.
19. E. Hemaspaandra. 1994. Complexity transfer for modal logic. *Proceedings of the Ninth IEEE Symposium on Logic In Computer Science (LICS'94)*, 1994.

20. E. Hemaspaandra. 1995. The price of universality. To appear.

21. H. Kamp and U. Reyle. 1993. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht.

22. R. Kaplan and J. Bresnan. 1982. Lexical-functional grammar: a formal system for grammatical representation. In J. Bresnan (ed) *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, pages 173–281.

23. M. Kracht and F. Wolter. 1991. Properties of independently axiomatisable bimodal logics. *Journal of Symbolic Logic*, 56:1469–1485.

24. G. Eiben, Á. Kurucz and A. Jánossy. 1994. Combining logics via combining algebraic theories. To appear.

25. R.K. Meyer and E.D. Mares. 1994. The semantics of entailment 0. In K. Došen and P. Schröder-Heister (eds) *Substructural Logic*, Oxford University Press, pages 239–258.

26. R. Milner. 1989. *Communication and Concurrency*. Prentice-Hall.

27. R. Muskens. 1995. *Meaning and Partiality*. Studies in Logic, Language and Information. CSLI Publications, Stanford. To appear.

28. H. Ono. 1983. Model extension theorem and Craig's interpolation theorem for intermediate predicate logics. *Reports on Mathematical Logic*, 15:41–58.

29. S. Popkorn. 1994. *First Steps in Modal Logic*. Cambridge University Press.

30. J. Seligman. 1990. *Perspectives: a Relativistic Approach to the Theory of Information*. PhD Thesis, University of Edinburgh.

31. J. Seligman. 1994. Combining logics with ease. To appear.

32. V. Shehtman and D. Skvortsov. 1990. Semantics of non-classical first-order predicate logics. In P. Petkov (ed) *Mathematical Logic*, Plenum Press, New York, pages 105–116.

33. E. Spaan. 1993. *Complexity of Modal Logics*. PhD Thesis, University of Amsterdam.

34. R. Thomason. 1984. Combinations of tense and modality. In D. Gabbay and F. Guenthner (eds) *Handbook of Philosophical Logic (II)*, Reidel, Dordrecht, pages 135–165.

35. K. Vermeulen and A. Visser. 1995. Dynamic bracketing and discourse representation. To appear.

36. F. Wolter. 1995. A counterexample in tense logic. To appear.