JOHAN VAN BENTHEM

LANGUAGE IN ACTION

1. LOGIC MEETS INFORMATION PROCESSING

A noticeable tendency in current logical research is the move away from still reflection of abstract truth to a concern with the structure of *information* and the mechanism of its *processing*. Thus, two aspects come to the fore which used to be thought largely irrelevant for logical analysis, namely the actual linguistic detail of presentation of premises and the actual procedures for setting up arguments. Accordingly, logical analyses will now have to operate at a level where, e.g., the *syntax* of occurrences of propositions matters – and likewise it will be occupied, not just with declarative structure, but also with matters of argumentive *control*. In the form of a slogan, many people nowadays believe that

Natural Language is a Programming Language

for effecting cognitive transitions between information states of its users

The purpose of this paper is to signal this trend, as it occurs across a number of rather different research communities, discuss a number of recurrent strands, and then propose a suitably general framework of 'dynamic logic' (broadly conceived) for pursuing these explicitly. Thus, its main novelty – if any – does not lie in any central technical contribution, but rather in the perspective offered.

What we can observe in the literature is a number of independent attempts to create conceptual frameworks allowing us to capture significant features at this level, while still retaining a workable logical theory. One thing which many of these newer approaches have in common is the failure of certain so-called *structural rules* found in standard logic, such as

$$\text{Monotonicity: } \frac{X \Rightarrow A}{X,\ Y \Rightarrow A}$$

Journal of Philosophical Logic **20**: 225–263, 1991.

or even

$$\text{Contraction: } \frac{X,\ Y,\ Y \Rightarrow A}{X,\ Y \Rightarrow A}$$

In standard calculi, these seem harmless, and evident, book-keeping rules: now, their failure becomes a very general *symptom* (though by no means the essence) of operating at a finer-grained level of logical analysis. Thus, we shall be exploring a landscape, so to speak, of logic underneath the usual classical or intuitionistic base systems.

The sources of these newer systems are diverse. Some motivations are *proof-theoretic*, with prime examples in Relevance Logic (cf. Dunn 1985), which drops Monotonicity, or Linear Logic (cf. Girard 1987), which also drops Contraction of premises. Even more radically, from a *linguistic* perspective, one must also drop the structural rule licensing Permutation of premises (cf. Lambek 1958). Put differently, Relevant Logic is still concerned with *sets* of premises, Linear Logic with *bags* (or 'multi-sets'), and a congenial linguistic paradigm like Categorial Grammar (cf. van Benthem 1988a) in general with ordered *sequences* of premises. And the latter level of detail also arises with motivations of a more *computational* nature, in what may be called the Dynamic Logic of inference and interpretation (cf. Harel 1984, van Benthem 1988b). More information on these various approaches will be found in the course of this paper.

Now, the purpose of this work is not to start with any sacrosanct calculus of 'sub-standard inference', trying to understand its secrets, but rather to explore a number of *models* for the structure of information and its processing, reflecting various intuitions that we have about the latter. These will come in the following forms. First we consider *language models* ('L-models') focussing on syntax and occurrence. Then, we move on to more abstract *information models* ('I-models'), arising from L-models through a certain measure of collapse of syntactic detail. Alternatively, this part may be seen as broadening the scope of Intuitionistic Logic, being the traditional 'guardian' of information and verification in the setting of the foundations of mathematics. Finally, processing of information will be central in more 'dynamic' *relational models* ('R-models') reflecting the control structures of programming transitions between information states.

In all these cases, some common logical issues arise. Most conspicuously, there is a proliferation of *logical constants*, beyond the classical core, and a corresponding number of options for designing logical calculi of inference and notions of valid consequence. We shall propose some systematic perspectives on the options involved – whilst also investigating the potential of existing research programs (Categorial Grammar, Modal Logic, Relation Algebra) to adapt to this wider purpose.

The various kinds of model will be compared and integrated at the end, in an attempt to create one basic picture, or framework for a logical theory of information. The resulting system is a 'dynamic logic' of information processing, inspired by similar calculi in the semantics of programs – an analogy which seems only natural, given the earlier conception of natural language as a vehicle for cognitive programming. In particular, this system allows for the coexistence of earlier 'static' views of propositions and the newer 'dynamic' ones. But again, the proposal made here does not stand or fall with the adoption of some unique preferred 'base calculus' of information-processing oriented logic.

To return to our opening sentence, there are many observable 'tendencies' in any given science at any given time: and most of them prove ephemeral fashions. Nevertheless, there is some reason to believe that we are better off here. Basing logic on the processing of information merely continues a historical development already begun in constructive logics, albeit in a more radical manner. Moreover, it is a good sign that the new perspective, once grasped, allows one to make sense of various scattered precursors in the literature, including such diverse topics as Quantum Logic (cf. Dalla Chiara 1985), where testing one occurrence of a premise may not yield the same result as testing another – or the study of the Paradoxes (cf. Fitch 1952), where problematic arguments rest essentially on such classical structural rules as Contraction on Liar sentences at different stages of the paradoxical reasoning.

2. THE LANGUAGE PARADIGM

Although information is certainly something more abstract than concrete linguistic form, it nevertheless proves useful to start with a study of Syntax in order to get below the surface of standard logic.

2.1. *Categorial Grammar*

In the research program of Categorial Grammar, natural language is described by an assignment of *types* to expressions, which are constructed from certain primitive types (representing, e.g., truth values, entities or states) by further operations such as

$a\backslash b$ (left-looking functor)

b/a (right-looking functor)

$a \bullet b$ (concatenation product)

The basic idea then is that an expression can be recognized in type a if the corresponding sequence of types for its component words admits of a *derivation* to the type a. (Extensive motivation for this linguistic paradigm may be found in the anthologies Buszkowski *et al.*, eds., 1988, and Oehrle *et al.*, eds., 1988.) What counts as an admissible derivation here may be specified as follows:

> $a\backslash b$, b/a satisfy the obvious laws of function/argument application,

as in

$$t \ ((t\backslash t)/t) \ t \Rightarrow t \ (t\backslash t) \ \Rightarrow t,$$

and $a \bullet b$ satisfies the obvious laws of concatenation.

An elegant powerful calculus to this effect has already been proposed in Lambek 1958. It may be described as a system of Gentzen sequents having the usual *logical rules* while lacking all the *structural rules* of standard logic. Here are the basic laws of this 'Lambek Calculus':

axiom: $a \Rightarrow a$

rules:

$$\frac{X \Rightarrow a \quad Y, b, Z \Rightarrow c}{Y, X, a\backslash b, Z \Rightarrow c} \qquad \frac{a, X \Rightarrow b}{X \Rightarrow a\backslash b}$$

$$\frac{X \Rightarrow a \quad Y, b, Z \Rightarrow c}{Y, b/a, X, Z \Rightarrow c} \qquad \frac{X, a \Rightarrow b}{X \Rightarrow b/a}$$

$$\frac{X \Rightarrow a \quad Y \Rightarrow b}{X, Y \Rightarrow a \bullet b} \qquad \frac{X, a, b, Y \Rightarrow c}{X, a \bullet b, Y \Rightarrow c}$$

That none of the usual structural rules should hold is easily seen by keeping the intended linguistic interpretation in mind.

There is also an 'undirected' variant of this calculus, in which we allow the structural rule of Permutation of premises, whilst collapsing the two directed functors $a\backslash b$, b/a into one:

$$a \rightarrow b.$$

The latter notation is not arbitrary: there is an obvious analogy between function types and logical *implications*. Thus, parsing natural language expressions with the help of categorial grammars is a form of implicational *deduction*, bringing together grammatical parse trees and logical proof trees. And this phenomenon of 'Parsing as Deduction' even induces a further equation, namely that of 'Formal Linguistics as Proof Theory'. For, as was already shown by Lambek, the new weaker categorial calculi can still be studied by the usual logical proof-theoretic methods as to their mathematical properties. (On the resulting research program, see Buszkowski 1982, van Benthem 1986, 1987, 1991.)

What should be observed is that, from the linguistic perspective, there need not be one single 'best' categorial calculus. Lambek's own system is certainly a natural candidate, but various syntactic phenomena may require strengthenings or weakenings. Thus, a better picture is that of a *Categorial Hierarchy* of calculi underneath, but ascending up to the standard systems of logic. This variation is precisely an asset in making intra-linguistic, or cross-linguistic, comparisons of complexity between syntactic phenomena.

Still, viewed as systems of logic, these calculi are rather poor, employing only a few of the basic logical constants. But there has been a tendency in the recent linguistic literature to introduce further operations on types (cf. Moortgat 1988 on gapped constituents; or the use of disjunction of the feature structures in recent computational linguistics). We shall presently return to this issue of linguistically motivated further operations on types.

2.2. *Language Models*

That a richer logic of types should lie behind Categorial Grammar becomes clear once we realize that the basic structures in formal linguistics are *families* of *languages*

$$\{L_a \mid a \in A\}$$

over some finite alphabet of symbols, which are closed under certain natural operations (cf Hopcroft and Ullman 1979). The latter may be systematized roughly as follows:

$$\begin{array}{lll} \text{Boolean operations:} & -, \cap, \cup, & \bot, \top \\ \text{Order operations:} & \bullet, \backslash, /, & 1, * \end{array}$$

As for the latter, we have (with juxtaposition indicating concatenation)

$$\begin{array}{lll} L_a \bullet L_b & = \{xy \mid x \in L_a, y \in L_b\} & \text{(product)} \\ L_a \backslash L_b & = \{x \mid \forall y \in L_a \colon yx \in L_b\} & \text{(left inverse)} \\ L_b / L_a & = \{x \mid \forall y \in L_a \colon xy \in L_b\} & \text{(right inverse)} \\ L_1 & = \{\langle\,\rangle\} & \text{(empty sequence)} \end{array}$$

and the Kleene star denotes finite *iteration* as usual.

These operations are natural, e.g., in the sense that the family of all *regular* languages is closed under them. By an *L-model* we shall mean any family of languages over some finite alphabet having this closure property. A sequent of types

$$X \Rightarrow a$$

will be called *valid* in such models if, for every interpretation $[\![\ ]\!]$ sending primitive types to arbitrary languages and complex types to the obvious compounds, it holds that

$$[\![\bullet X]\!] \subseteq [\![a]\!].$$

Here, '$\bullet X$' denotes the concatenation product of X (with the stipulation that $\bullet\varnothing = 1$).

Valid principles on this account will include all Boolean laws, as well as typical Lambek principles such as

$$a \bullet (a\backslash b) \Rightarrow b, \quad \text{or} \quad a \Rightarrow (b/a)\backslash b.$$

In fact, here is a straightforward observation:

PROPOSITION. *The Lambek Calculus is sound for interpretation in L-models.*

The converse is still an open question: as we shall see later on.

Of course, the L-interpretation also produces further validities for other logical constants. For instance, it is of interest to compare the behaviour of our two 'conjunctions': with $\bullet$ satisfying the Gentzen laws of the Lambek Calculus, and Boolean $\wedge$ rather the following two:

$$\frac{X, A \Rightarrow B}{X, A \wedge C \Rightarrow B} \qquad \frac{X \Rightarrow A \quad X \Rightarrow B}{X \Rightarrow A \wedge B}$$

Note also that the Kleene star behaves somewhat like an S4 *modality* in that we have the validity of

$$a \Rightarrow a^*, \quad a^*, a^* \Rightarrow a^*.$$

In fact, this operator licences a structural rule which is not valid as such in the Lambek Calculus, viz. Contraction:

$$\frac{X, a^*, a^* \Rightarrow B}{X, a^* \Rightarrow b}$$

The analogy with modality is not perfect, however, in that iteration respects neither conjunction nor disjunction:

$$(a \cup b)^* \not\Rightarrow a^* \cup b^*, \quad a^* \cap b^* \not\Rightarrow (a \cap b)^*.$$

Even so, whatever its precise family resemblances, the logic of L-models has an independent interest as an object of investigation. For, the ordering operations seen central to syntax, and even iteration becomes quite natural once we move from single sentences to *texts* (e.g., texts themselves have the iterate t^* of the sentence type t).

REMARK. In this connection, even logical proof-theoretial structures, such as the above sequents are already text objects, with the *comma* as a separator. And the latter operator itself needs interpretation before it makes sense to discuss validity or non-validity of such principles as the usual structural rules. Thus, in a sense, the popular observation about loss of these rules is too simple-minded, e.g., contraction fails if we interpret the comma as a concatenation product – but it would remain valid if we had treated the comma via Boolean conjunction. □

Finally, it is also quite possible to introduce further operations on L-models. For instance, two useful operations are

$$\pi(L) := \text{all } \textit{permutations} \text{ of sequences in } L,$$

$$\iota(L) := \text{all } \textit{mirror images} \text{ of sequences in } L.$$

These will again exemplify logical laws, such as

$$\pi(a \cup b) \Leftrightarrow \pi(a) \cup \pi(b), \quad \pi\pi(a) \Leftrightarrow \pi(a),$$

$$\iota(-a) \Leftrightarrow -\iota(a), \quad \iota\iota(\alpha) \Leftrightarrow a.$$

2.3. *Numerical Models*

When full Permutation is allowed, the only information left about a string of symbols is the number of occurrences of each basic symbol in it. Thus undirected categorial calculi invite consideration of *numerical models*, defined as follows:

There is a family of sets of vectors in $\mathbf{N}^k$ (where k is the size of the alphabet) which is closed under the Boolean operations as well as vector addition and its converses providing the obvious interpretation for the order operations, e.g., now

$$L_{a \bullet b} = \{x + y \mid x \in L_a, y \in L_b\}.$$

If we want to have a smooth notion of $L_{a \to b}$, however, we shall need unlimited *subtraction*: which would require having the integers $\mathbb{Z}$ rather than the natural numbers $\mathbf{N}$ (even though this move would lose us a straightforward linguistic interpretation).

Again, the earlier soundness result extends to N-models, for categorial calculi admitting a Permutation rule. For the special case of the undirected Lambek Calculus, this observation generalizes the use of so-called primitive type *counts* (cf. van Benthem 1986) as a check on derivability. Thus, after all, the above, seemingly uninterpretable mathematical generalization toward negative numbers makes sense: as type counts may be negative.

Example. Consider two primitive types e, t. Assign the following two singleton sets of vectors:

$$L_e = \{\langle 1, 0\rangle\}, \quad L_t = \{\langle 0, 1\rangle\}.$$

Then, for each complex type a, the inductively computed integer value of L_a becomes a singleton set $\{\langle x, y\rangle\}$ with

$$x \text{ is the } e\text{-count of } a, \quad y \text{ is the } t\text{-count of } a.$$

E.g., $((e, (e, t)), t)$ goes to $\{\langle +2, 0\rangle\}$, $((e, t), (e, e))$ goes to $\{\langle +1, -1\rangle\}$. □

But also, we can quickly check further non-derivabilities which did not show up in the pure count system.

EXAMPLE. The sequent $((e, t), t) \Rightarrow t$ has equal counts on both sides: which is the necessary condition for Lambek derivability induced by count. Nevertheless, it is not derivable – as may be established by proof-theoretic analysis. But now, we can also provide a counterexample, with

$$L_e = \mathbf{N}, \quad L_t = \varnothing.$$

(Observe that $L_{(e,t)} = \varnothing$, $L_{((e,t),t)} = \mathbf{N}$.) Thus, the implicational logic on numerical models has at least 'truth-value counterexamples' – and hence it must be contained in the classical conditional logic. We conjecture that it is in fact equal to the conditional logic axiomatized in the non-directed Lambek Calculus. □

It would be of interest to see if this numerical interpretation yields further algorithms for reducing the Gentzen search spaces encountered in categorial parsing. (See Moortgat 1988, on the use of the original counts for the latter purpose.)

2.4. *Logical Issues*

The above models suggest a number of systematic logical questions.

2.4.1. *Calculi of Inference*. Perhaps the most obvious formalism for describing L-models or N-models is that of a standard first-order logic over the appropriate similarity type. For instance, here are two relevant observations:

PROPOSITION. *The first-order theory of concatenation on expressions from a* one-*symbol alphabet is equivalent to Additive Arithmetic. The first-order theory on* two *symbols, however, becomes equivalent to the True Arithmetic of addition and multiplication. Thus, the one-symbol case is decidable, whereas two symbols introduce highly non-effective complexity.*

REMARK. Current practice in mathematical linguistics, of concentrating on one-symbol 'pilot cases', may therefore be misleading.

Proof. For the first assertion, it suffices to equate sequences with their length, and observe that concatenation becomes addition then.

For the second assertion, it suffices to provide a first-order definable encoding from $(\mathbf{N}, +, \bullet)$ into the two symbol syntactic structure. (The converse embedding from syntax into numbers is provided by the usual techniques of arithmetization.) For the universe, take the subdomain of all sequences consisting only of occurrences of the first symbol, say a. Then, addition has an obvious definition via concatenation. As for multiplication, the following trick employs only concatenation-definable notions:

> for two a-sequences x, y, construct the sequence z as follows
> $- \; byb \; - \; byyb \; - \cdots by \cdots y$,
> where the parts $-$ stand for successive non-empty initial segments of x, with the interleaved parts receving an additional copy of y in each step.

The product value may then be read off at the end. □

REMARK. As Kees Doets has pointed out, this result was already found by Quine 1946.

By contrast, the first-order theory of N-models is embeddable into additive arithmetic in an obvious way, and hence it is decidable.

Next, one can go up to higher formalisms, such as the *monadic second-order* logic over L-models or N-models, allowing quantification over their subsets. This is what is needed for expressing the earlier validity of propositional principles such as

$$a \bullet (a\backslash b) \Rightarrow b:$$

which corresponds to the validity of the second-order principle

$$\forall A \forall B\, \forall x\colon \exists y \exists z(x = yz \wedge Ay \wedge \forall^A u\colon Buz) \rightarrow Bx.$$

But note that this is only a small Horn-type fragment of the full second-order formalism, which need not be subject to general complexity results about the latter. For instance, the question as to effective axiomatizability of the *universal* (Π^1_1)-*fragment* of monadic second-order logic for $\backslash$, $/$, $\bullet$ over L-models appears to be open.

In any case, this type of logic seems to deserve investigation, also over numerical N-models. For instance, how does it change across successive 'dimensions' for our occurrence vectors?

EXAMPLE. To get a feel for logics like this, it is useful to show, e.g., that the following set of formulas is satisfiable in $\mathbf{N}^1$:

$$\{p, -(p \bullet p), p \bullet p \bullet -p\}. \qquad \square$$

Against this background, it is instructive to mention some completeness results in the tradition of Categorial Grammar.

Categorial calculi like Lambek's may be modelled 'cheaply' via some suitable notion of *algebra* (obtainable via a Lindenbaum construction). In particular, one may use so-called 'residuate semigroups'. An improvement was obtained in Došen 1985, using 'residuate semigroups spread over partially ordered semigroups', i.e., structures

$$M = (|M|, \leqslant, \bullet, \backslash, /)$$

defined over some partially ordered semigroup $(|G|, \leqq, \bullet)$ as follows:

$$\begin{aligned}
|M| &= \{A \subseteq |G| \mid \forall b \in A, a \leqq b\colon a \in A\} \\
A \bullet B &= \{c \in |G| \mid \exists a \in A, b \in B\colon c \leqq a \bullet b\} \\
A\backslash B &= \{c \in |G| \mid \forall a \in A\colon a \bullet c \in B\} \\
B/A &= \{c \in |G| \mid \forall a \in A\colon c \bullet a \in B\}
\end{aligned}$$

Next, Buszkowski 1986 did even better, by proving the equivalence between Lambek-derivability and validity over 'residuate semigroups spread over semigroups', where the partial order $\leq$ is just *identity*. The latter kind of structure comes already quite close to our L-models, which are spread over *free* semigroups. As Buszkowski has shown, the \, / Lambek calculus is complete with respect to the latter L-models; but the question is open when we add the product $\bullet$. Moreover, no results seem to be known for the case where we add the Boolean operators.

2.4.2. *The Proper Logical Constants.* What we can no longer assume in the new context is that the old set of logical operators from standard logic will be sufficient. And in fact, we saw several variants for the old conjunction, as well as various new kinds of operator. Now, can we find some *systematic perspective* on this, which will allow us to formulate issues of *expressiveness* and *functional completeness*?

One possible approach here is proof-theoretic. One can try to generalize the analysis of general 'formats' of introduction rules for operators (as in Zucker and Tragesser 1978), showing how some distinguished set of operators defines all possibilities. This suggestion has been taken up in Wansing 1989.

Another approach is model-theoretic, referring to the earlier L- or N-models. The set of all *a priori* possibilities may then be seen as embodied in the appropriate first-order language of concatenation. For instance, the Booleans are definable via

$$A \cap B: \quad \lambda x \bullet Ax \wedge Bx$$

and the ordering operations via, e.g.,

$$A \bullet B: \quad \lambda x \bullet \exists y \exists z (Ay \wedge Bz \wedge x = yz).$$

In general, there will be *infinitely* many non-equivalent possibilities.

Nevertheless, there are some natural special classes to be considered. For instance, we can look at special syntactic *fragments* of the relevant predicate logic. Just as has proven useful in Modal Logic, we may view our types of propositions as variable-free notations for first-order concatenation formules employing only some fixed finite number of variables (cf. Gabbay *et al.* 1980, Immerman and Kozen

1987). In the above cases, this number was at most 3. And such fragments, at least for pure *predicate*-based first-order languages, always admit of some finite functionally complete operator notation. In the present case, we should have to introduce a ternary concatenation predicate then (see also below), and determine successive complete operator sets for operations

$$\lambda x \bullet \phi(A, B, C, \ldots; x),$$

definable by a schema ϕ employing 1, 2, 3 . . . variables. (E.g., the Booleans are complete for the case where ϕ uses only x itself.)

Another approach would be to locate some special semantic characteristics of admissible logical constants, cutting down on the number of *a priori* possibilities. We shall return to these options in the following Section.

2.4.3. *Meanings of Derivations.* One basic tool in the study of categorial calculi, and indeed their application to natural language semantics, is the correspondence between categorial derivations and *terms* from a *lambda calculus* allowing function application and abstraction (as well as pairing and projection). (See van Benthem 1989a, 1991 for an exposition as well as some basic questions arising in this perspective.)

From this viewpoint, we are not only interested in valid inferences, but also in the different ways that may be available for deriving the inference (its 'readings', to extend a linguistic concept). As it stands, however, this correspondence only works for functional and product types. Can it also be extended to the other new logical operators encountered above?

In fact, this can be done relatively easily, by extending the Lambda Calculus with suitable operations matching additional operators such as Boolean conjunction or disjunction. Thus, we can code up derivations, and bring out intuitive differences such as that between the following two derivations for the same valid inference:

$$- A \cap (A \cup B), -A, -(A \cap - B)$$

$$- A \cap (A \cup B), B, -(A \cap - B)$$

Nevertheless, there remains the problem that such enriched lambda calculi do not seem to provide an *independent* intuition concerning our derivations: whence this perspective may lack the appeal which it had in the more restricted case.

REMARK. Other more linguistically inspired topics may have some meaning in this wider setting too. For instance, what would be the more general logical import of the notion of *recognizing power*? □

2.5. *Linear Logic*

An emphasis on computational processing and the proper level of syntactic detail involved therein is also characteristic of the current research line of so-called 'Linear Logic'. This area shows a number of striking resemblances with Categorial Grammar, especially in the extended sense developed here. Indeed, the latter may be viewed as a linguistic paradigm taking a rather liberal view of syntax (in being willing to countenance permutations) – while the former is a logical paradigm taking syntax rather more seriously than is done in standard approaches: which explains their rapprochement. The present Section will merely point at a number of analogies.

First, we need some concrete system for the purpose of comparison. Instead of providing a full motivation here, we refer to Girard 1987, Lafont 1988 for the proof-theoretical and computational background of the following basic calculus (which allows Permutation of premises):

axioms: $A \Rightarrow A \qquad \Rightarrow 1 \qquad X, \bot \Rightarrow A \qquad A \Rightarrow \top$

rules:

$$\frac{X \Rightarrow A \quad Y \Rightarrow B}{X, Y \Rightarrow A \bullet B} \qquad \frac{X, A, B \Rightarrow C}{X, A \bullet B \Rightarrow C}$$

$$\frac{X, A \Rightarrow B}{X, A \cap C \Rightarrow B} \qquad \frac{X \Rightarrow A \quad X \Rightarrow B}{X \Rightarrow A \cap B}$$

Note that these two conjunctions would collapse in the presence of the structural rules of standard logic.

$$\frac{X, A \Rightarrow B \quad X, C \Rightarrow B}{X, A \cup C \Rightarrow B} \qquad \frac{X \Rightarrow A}{X \Rightarrow A \cup B}$$

$$\frac{X \Rightarrow A \quad Y, B \Rightarrow C}{X, Y, A \to B \Rightarrow C} \qquad \frac{X, A \Rightarrow B}{X \Rightarrow A \to B}$$

Next comes Girard's 'modality', allowing for a translation from classical logics into linear systems (by absorbing some behaviour of the earlier structural rules into their logical rules):

$$\frac{X \Rightarrow A}{X, !B \Rightarrow A} \qquad \frac{X, !B, !B \Rightarrow A}{X, !B \Rightarrow A}$$

$$\frac{X, A \Rightarrow B}{X, !A \Rightarrow B} \qquad \frac{!A \Rightarrow B}{!A \Rightarrow !B}$$

The intuitive motivation is that $!A$ stands for arbitrary finite iterations of A-type propositions. (We skip the rules here for the dual operator?.) Finally we add

$$\frac{X \Rightarrow A}{X, 1 \Rightarrow A}.$$

This calculus as well as several variants has been under intensive investigation recently (cf. Abrusci 1988a, Sambin 1988). For present purposes, it will suffice to note the obvious resemblance, and indeed identity, of its $\bullet$, $\rightarrow$ fragment with that of the undirected Lambek Calculus. This analogy can be extended, as it turns out, to a useful comparison between the existing proof theory for categorial grammar and that for linear logic. (For a first survey, see Ono 1988.)

A case in point are the earlier completeness theorems for categorial calculi (cf. Section 2.4.1). Roughly speaking, the completeness results in Sambin 1988, Abrusci 1988a seen comparable to Došen's kind of theorem, be it for a richer kind of language. (Note their use of 'closure' in the definition of a product.) It seems that one cannot do better than this in general, because of an example in Abrusci 1988a: which gives a distributivity principle that is non-derivable in the basic linear logic, even though it is valid in every 'simple' intuitionistic topophase structure (with the identity closure).

The exact extent of the analogies between the metatheory of categorial grammar and linear logics remains to be established.

To conclude here, it may be of interest to observe a connection with our earlier L-models.

PROPOSITION. *The following correspondence provides a sound interpretation for (part of) linear logic*:

$$(\textit{Boolean})\ \bot : \bot,\ \top : \top,\ \cap : \bigcap,\ \cup : \bigcup,$$

$$(\textit{order})\quad \bullet : \bullet,\ 1 : 1.$$

There is no direct analogue of linear $\rightarrow$ here: but \ and / are just right for the directed, non-permuted calculus of linear logic proposed in Abrusci 1988b. Moreover, there is no analogue for linear ! as it stands, although Kleene iteration seems to have a similar infinitary flavour, while validating the structural rule of Contraction – as was already observed before. Nevertheless, * failed to validate some of the necessary laws for modalities. But probably, some suitable defined iterative operator on languages will do the job.

3. THE INFORMATION PARADIGM

Now, let us abstract away from syntactic occurrences, and move to a level of analysis which is traditionally considered appropriate for locating information structures.

3.1. *Information Models*

Here, we can take a lead from the tradition – as there are already established systems of logic based on information structures, a noteworthy example being Intuitionistic Logic. Again, disregarding any particular calculus to be modelled, the simplest structures to be described are partial orders

$$(I, \subseteq)$$

of 'information states' or stages ordered by 'inclusion' ('possible growth'). (See Troelstra and van Dalen 1988.) In a somewhat richer perspective, the also information-oriented Relevance Logic has even *lattices*

$$(I, \subseteq, \cap, \cup)$$

which also include *suprema* ('sums') and *infima* of information stages.

Another way of arriving at these *I-models* would start from the earlier L- or N-models, and impose successive conditions on their binary operation. In particular, writing the supremum $\cup$ as addition $+$:

$$x + y = y + x$$

$$x + x = x,$$

so that it will induce a partial order in the usual way:

$$x \subseteq y \quad \text{iff} \quad x + y = y.$$

Now, a number of questions arises similar to those encountered in Section 2. For a start,

What kind of logical operators are suitable here?

As before, we can think of propositions as denoting sets of information stages: and logical operators are to relate these. Thus, a natural format of semantic truth conditions becomes a first-order language referring to the binary order $\subseteq$ as well as having unary predicates over stages.

EXAMPLE. The usual modalities are expressible as follows:

$$\overrightarrow{\Diamond}\, p\colon \lambda x \bullet \exists y (x \subseteq y \wedge Py)$$

$$\overleftarrow{\Diamond}\, p\colon \lambda x \bullet \exists y (y \subseteq x \wedge Py)$$

But we can also introduce more complex binary operators on propositions, such as, in particular, two relatives of conjunction and disjunction, respectively:

$$p \cup q\colon \lambda x \bullet \exists y \exists z\, (Py \wedge Qz \wedge \text{`}x \text{ is the supremum of } \{y, z\}\text{'})$$

$$p \cap q\colon \lambda x \bullet \exists y \exists z\, (Py \wedge Qz \wedge \text{`}x \text{ is the infimum of } \{y, z\}\text{'})$$

And using $\cup$, which reflects 'addition' of information stages (as far as defined), a notion of implication may be defined as before. □

Thus, there is a rich structure of logical constants, even on information models collapsing addition of identical states.

REMARK. In this perspective, Intuitionistic Logic has no favoured status as a logic of information, as it treats only part of the relevant operators. The reason why it does not 'see' these additional possibilities is that only *upward hereditary* propositions are considered intuitionistically (it is a 'logic of progress'), so that, e.g., $p \cup q$ and $p \wedge q$ will collapse. But, e.g., the above operator $\diamondsuit$ also envisages *retraction* of information. □

The next general question is this:

> What calculi of inference are appropriate over these information models?

Whatever choice is made, note that the analysis given in Section 2.4.1 still applies here: such calculi are likely to be fragments of the universal monadic second-order logic over lattices. As the latter form an elementary class, their universal second-order theory must be *recursively enumerable* (by a simple logical argument), and hence so are logics of *I*-models. Now, this *a priori* observation does not provide explicit useful axiomatizations. But in fact, of course, some calculi corresponding to the earlier categorial systems lie at hand: we can start with the usual system of intuitionistic or relevant logic, and then enrich them with further suitable operators.

3.2. *A Modal Perspective*

With information models in the above sense, we are back with standard logics, now not viewed as embodied in any particular proof calculus, but as a description for types of information. Thus, we can raise questions of *design*: did the founding fathers of Intuitionistic Logic, or Relevant Logic, really pick the appropriate logical constants? But also conversely, techniques developed for these systems may turn out to have wider applicability.

In fact, one obvious flexible formalism over information models is that of *Modal Logic*, whose theory on partial orders is S4 (if one considers the upward direction only). In general, again, the semantic format behind this is a first-order language over $\subseteq$ and unary predicates mirroring propositions. We shall use this format here to arrive at a more general technical perspective on the logic of information models. (See van Benthem 1989c for further details.)

First, on the issue of selection of appropriate logical constants, we can use the earlier method of *fixed variable fragments*. For instance, at the lowest levels, there is a simple classification. To see this, fix any I-model. All possibilities may be enumerated as follows:

PROPOSITION. *All operations of the form* $\lambda x \bullet \phi(x; A, B, C, \ldots)$ *employing only* one *variable are definable by means of Boolean combination of the predicates* $A, B, C, \ldots$ *. With* two *variables, the following set of modal operators is functionally complete*:

$$\Diamond_1 p\colon \lambda x \bullet \exists y(x \subsetneq y \wedge Py)$$

$$\Diamond_2 p\colon \lambda x \bullet \exists y(y \subsetneq x \wedge Py)$$

$$Ip\colon \quad \lambda x \bullet \exists y(x \not\subseteq y \wedge y \not\subseteq x \wedge Py).$$

In general, however, no finite functionally complete set of operators exists – as may be shown using the methods of Immerman and Kozen 1987.

Now, the usual modal formalism may be translated into the *two-variable* fragment of this first-order language. But even there, it distinguishes itself by further special semantic behaviour, which has some independent interest. This takes the form of *invariances* for modal formulas $\phi(x)$.

> $\phi(x)$ is invariant in passing from any model to the *generated submodel* containing x and being closed under $\subseteq$-successors and -predecessors.

This seems reasonable from the viewpoint of 'search' through information patterns. Note that it rules out the above 'incomparability' operator I.

Another invariant of the modal formalism brings out a question which ought to arise with any kind of semantic modelling. It is one thing to introduce a general class of information structures, but it remains to provide them with a suitable *criterion of identity*: Which models are really 'the same'? What the model theory of Modal Logic suggests here is in fact quite close to a notion which has recently become prominent in the algebra of processes in computer science (and even in non-standard set theories), namely 'bisimulation'. Let us say that a relation C between two models $M1$, $M2$ (with the latter being viewed as families of propositions spread over a pattern of information states) is a *bisimulation* if it satisfies the following conditions:

(i) C-correlated states verify the same propositions,

(iia) if $w_1 C w_2$ and $w_1 \subseteq v_1$,
then there exists v_2 with $w_2 \subseteq v_2$, $v_1 C v_2$,

(iib) likewise, in the opposite direction,

(iii) and analogously for $\subseteq$ predecessors.

Then, modal formulas ϕ have this property:

$\phi(x)$ is invariant for bisimulations,
i.e., if $w_1 C w_2$, then $M_1 \models \phi[w_1]$ iff $M_2 \models \phi[w_2]$.

Together, these two invariances pick out precisely the modal formulas inside the full first-order language, as is proved in van Benthem 1985.

What will happen next is that we can turn up the magnification of our 'logical lenses', so to speak, by moving on to richer modal formalisms: using 3-configurations of states in the first-order description language.

EXAMPLE. As in Temporal Logic, operators are possible of the Since/Until type: "until verifying p, q was encountered".

$$Upq\text{: } \lambda x \bullet \exists y(x \subseteq y \wedge Py \wedge \forall z(x \subseteq z \subseteq y \rightarrow Qz)).$$

This requires essentially 3 variables instead of 2. Clauses of this level of complexity occur, e.g., in the process of updating information states, when we want to make assertions q about the first state including the present one where some proposition p has become true:

$$U(p \wedge q) \neg p. \qquad \square$$

The latter fragment of the first-order language over I-models has its characteristic semantic invariance too: this time, in terms of 'strong bisimulations' respecting also the relation of *betweenness* among states.

EXAMPLE. Increased Discrimination.
The following two I-patterns are bisimulation equivalent, but no strong bisimulation connects them:

□

Finally, the full framework introduced in Section 3.1 naturally calls for configurations of 4 variables. This shows when we introduce a binary modality $\bigcup$ as follows:

$$\bigcup pq\colon \lambda x \bullet \exists y \exists z (Py \wedge Qz \wedge x = \sup(y, z)).$$

(The intuitive motivation here is like that for the 'information piece' semantics of Relevant Logic proposed in Urquhart 1972.) This is a genuine modality, in that, e.g., Distribution holds:

$$\bigcup (p_1 \vee p_2) q \leftrightarrow \bigcup p_1 q \vee \bigcup p_2 q$$
$$\bigcup p (q_1 \vee q_2) \leftrightarrow \bigcup p q_1 \vee \bigcup p q_2$$

Here too, there is a suitable notion of simulation invariance. As the language has become stronger, requirements on identification will go up too. What is needed now is a bisimulation satisfying two further back-and-forth clauses:

> if xRx^1 and $x = \sup(y, z)$, then there exist y^1, z^1 such that $x^1 = \sup(y^1, z^1)$ and yRy^1, zRz^1;
> and analogously in the opposite direction.

As above, one can prove a general characterization result here (witness the earlier reference van Benthem 1989c).

Thus, the modal perspective gives us a hierarchy of ever finer-grained descriptions of information patterns.

Again, given any choice of modal operators, there is the matter of axiomatization of valid consequence. Actually, there are various options as to the latter notion. On the usual modal account, validity of a sequent $X \Rightarrow \phi$ would be explained as follows:

> In all information models, at all states verifying each formula in X, ϕ is verified too.

The resulting notion satisfies the classical structural rules, such as Monotonicity and Contraction: because it treats its premises as a *Boolean conjunction*. But, another notion of validity would be more in line with the spirit of Section 2, taking a more 'dynamic' view of adding the information supporting the premises:

> In all information models, for all sets of information pieces $\{i_x | x \in X\}$ verifying the corresponding formulas in X, the *sum* $+\{i_x | x \in X\}$ verifies ϕ.

The latter notion of consequence loses some structural rules: in particular, Monotonicity. The reason is that it treats its premises via non-Boolean *additive conjunction*.

Nevertheless, there is no compelling reason to axiomatize two notions of consequence here. For, the second approach may be *defined* in terms of the first: via the standard validity of

$$\bigcup X \Rightarrow \phi.$$

Thus, complete axiomatizations for even dynamic notions of consequence may be sought using standard methods from Modal Logic and Relevance Logic: provided that we take those earlier enterprises in a suitably liberal spirit.

4. THE DYNAMIC PARADIGM

Now, we turn from information structures to information processing. Again, logical phenomena will turn up similar to those encountered before.

4.1. *Cognitive Transitions*

In many recent publications, one can find attempts at formulating logics reflecting more dynamic procedural aspects of interpretation and inference, thus mixing 'declarative' and 'imperative' aspects within a single system of deduction. Examples are Gärdenfors 1988, Groenendijk and Stokhof 1988, Veltman 1989 (see also the survey in van Benthem 1988b).

One way of describing the underlying patterns here is as follows. In standard logic, propositions stand for sets of possible worlds, or more generally, situations verifying them. Put differently, they are *properties* of information states. But now, we look at the effects of adding a proposition to an information state. Thus, dynamically, a proposition acts as a transformer on states, and its denotation will now rather be a binary *relation* (its 'successful transitions'). The latter pattern can be discerned across various proposals.

EXAMPLE 1. In the semantics of programming languages, a program π denotes a binary relation between computer states. Traditional logical formalisms still serve here as a means of making static assertions about such states (witness the usual 'correctness assertions' in the theory of program behaviour).

EXAMPLE 2. But, following Barwise 1987 or Groenendijk and Stokhof 1988, one can also interpret the latter formalisms dynamically, as transition relations between assignments which may change in the course of evaluation.

As in earlier Sections, this perspective brings up the question as to selection of an appropriate set of logical operators, which is going to be richer than that of standard logic. For instance, there are two options for 'conjunction' of propositions now: one is sequential *composition* of relations, the other is Boolean *intersection*, which has a more parallel flavour. Moreover, again, new operators may arise, such as a modality looking forward:

$$[\![\Box\phi]\!] \;=\; \{(a, a) \mid \text{for all assignments } b\text{: } (a, b) \in [\![\phi]\!]\}.$$

In addition, there is the issue of an appropriate definition of *valid consequence*. For instance, one possibility would be to have the composed transition relation for the premises be part of that for the conclusion:

$$\phi_1, \ldots, \phi_n \models^1 \psi\text{:} \quad [\![\phi_1 \bullet \cdots \bullet \phi_n]\!] \subseteq [\![\psi]\!].$$

Another option is to 'process the premises, and then see':

$$\phi_1, \ldots, \phi_n \models^2 \psi\text{:} \quad \text{If } (x, y) \in [\![\phi_1 \bullet \cdots \bullet \phi_n]\!], \text{ then } (y, z) \in [\![\psi]\!] \text{ for at least one } z.$$

Note that, as before, such notions of consequence will tend to lose classical structural rules: e.g., $\models^2$ does not satisfy Permutation or Monotonicity.

EXAMPLE 3. Assignments are not the only possible 'states'. In the folklore idea of propositions as transformations, states might be sets of models or possible worlds, and each successive proposition picks out some further subset of the current one. This idea was taken up formally in Heim 1982 or in Veltman's recent update system. Again, we find a proliferation of useful logical constants as well as notions of inference, each with different formal properties.

REMARK. The diversity of logical constants may now be understood as the effect of having both more purely descriptive notions and operations of *control.*

REMARK. The proliferation of notions of inference may be curbed again by trying to locate some reasonable base conditions which any sensible notion of 'inference' should satisfy: in the spirit of the program of Makinson 1988. In fact, one reason for the diversity here is that we are mixing notions of inference with other cognitive activities.

4.2. *Relational Algebra*

The above analysis gives us what might be called *R-models* for new logical formalisms: i.e., families of transition relations on some carrier set of states, closed under suitable operations.

But then, we can use the existing work on Relational Algebra to get a better grasp of the logical structure here (cf. Jónsson 1984). In the theory of binary relations, one tries to generalize the Boolean Algebra of unary propositions by creating a suitable richer similarity type. Notably one has

Boolean operations: $-$, $\cap$, $\cup$, $\perp$, $\top$

Order operations: $\bullet$ as well as converse $^{\smile}$

and one special relation, viz. *identity*: id.

But, one can introduce also other operations, such as analogues of our earlier slashes

$$R\backslash S \;=\; \{(x, y) \mid \forall z\colon (z, x) \in R \Rightarrow (z, y) \in S\}$$
$$S/R \;=\; \{(x, y) \mid \forall z\colon (y, z) \in R \Rightarrow (x, z) \in S\}$$

Here are the basic laws of relation algebras:

(i) all Boolean identities

(ii)
$$(R \cup S)^{\smile} = R^{\smile} \cup S^{\smile}$$
$$(-R)^{\smile} = -R^{\smile}$$
$$R^{\smile\smile} = R$$
$$id^{\smile} = \mathrm{id}$$

(iii)
$$(R \cup S) \bullet T = R \bullet T \cup S \bullet T$$
$$R \bullet (S \cup T) = R \bullet S \cup R \bullet T$$
$$R \bullet (S \bullet T) = (R \bullet S) \bullet T$$
$$R \bullet \mathrm{id} = \mathrm{id} \bullet R = R$$

(iv)
$$(R \bullet S)^{\smile} = S^{\smile} \bullet R^{\smile}$$
$$R^{\smile} \bullet -(R \bullet S) \leqslant -S$$

This framework may be used to analyze the earlier systems. First, interpreting validity of sequents now as

$X \Rightarrow a$ is valid if, for all relational interpretations $[\![\ ]\!]$,

$$[\![\bullet X]\!] \subseteq [\![a]\!],$$

one gets failures of all the structural rules of standard logic: Monotonicity, Permutation or Contraction. (E.g., $a, a \Rightarrow a$ is only valid for *transitive* relations.) Moreover, again, alternatives may be *defined* here. For instance, a modality $\Diamond R$ may be introduced as follows:

$$T \bullet R^{\smile}$$

defines $\{(x, y) \mid \exists z\colon (y, z) \in R\}$. But then, the notion $\models^2$ introduced earlier on can be reduced to

$$\bullet X \Rightarrow \Diamond a.$$

REMARK. The operation of converse $^{\smile}$ does not seem to have any reasonable analogue at the level of types or propositions. Nevertheless, it does occur at the level of texts, and conscious operations on information states: witness the 'revisions' and 'contractions' of Gärdenfors and Makinson 1988. Moreover, even the model class semantics of Veltman invites consideration of further operations than just 'updating', rather requiring 'stepping back' undoing the effects of some earlier transformation. □

Now the proposed use of this framework *vis-à-vis* concrete systems of dynamic interpretation and inference is as follows. It provides a basic level common to all approaches, enabling us to theorize about their general structure. On top of this, one can then determine which additional properties of some proposed system are due to further special features (e.g., the use of special types of transition relation only, or the selection of some special format of inference).

4.3. *Logical Issues*

As before, there are some broad logical issues now which deserve investigation.

First, there is the issue of *functional completeness*. In fact, the above similarity type for relational algebra is well-chosen in the following sense (cf. Maddux 1983):

> each first-order definable operation $\lambda xy \bullet \phi(x, y, R)$ on binary relations employing only *three* variables $\{x, y, z\}$ can be written using only $-$, $\cap$, id, $\bullet$, $^{\smile}$.

Of course, this still leaves many related questions; but it does show some stability.

Next, there is the matter of logical *calculi of inference*. At least, we have an observation similar to those made in Section 2:

PROPOSITION. *Under the relational Boolean/order interpretation, both the Lambek Calculus and (the relevant part of) Linear Logic are sound.*

What this amounts to is an embedding of these calculi into Relational Algebra. It is an open question, however, if this interpretation is *complete*, even for the Lambek Calculus.

One problem with this reduction is that Relational Algebra itself has its problems. Notably, the earlier set of basic principles is not a complete axiomatization of the class of all validities on set-representable relation algebras (the latter class is known to be non-finitely axiomatizable). In view of these, and other technical complications, it may be useful eventually to adopt a somewhat less orthodox form of Relational Algebra. In particular, one might think of transition relations more abstractly, as being sets of 'arrows', not necessarily to be identified with their end-points. And if we do that, an earlier perspective returns (cf. Section 3.2).

The point is that the calculus of Boolean operations, composition $\bullet$ and converse $^{\cup}$ may be viewed as a *modal logic*. (Note that $\bullet$ and $^{\cup}$ both satisfy Distribution.) So, we can introduce corresponding relations

$Cxyz$: 'arrow z is the composition of x and y'

Fxy : 'arrow y is the converse of x'

as well as a special property

Ix: 'x is an identity arrow'.

The various axioms of Relational Algebra then express conditions on what may be called 'arrow frames'

(W, I, F, C).

EXAMPLE. Here is the list of relevant correspondences:

$(R \cap S)^{\cup} \geqslant R^{\cup} \cap S^{\cup}$: $\forall xyz\,(Fxy \wedge Fxz \rightarrow y = z)$

$(-R)^{\cup} \geqslant -R^{\cup}$: $\forall x \exists y Fxy$

i.e., F is a function;

$R^{\cup\cup} \leqslant R$: $\forall xy(Fxy \rightarrow Fyx)$

i.e., F is idempotent.

Then, C is associative:

$$((R \bullet S) \bullet T) \;=\; (R \bullet (S \bullet T)):$$

$$\forall xyzuv\, (Cxyz \wedge Czuv \rightarrow \exists w\, (Cyuw \wedge Cxwv))$$

$$\forall xyzuv\, (Cyzu \wedge Cxuv \rightarrow \exists w\, (Cxyw \wedge Cwzv))$$

Moreover, it interacts with F as follows:

$$(R \bullet S)^{\cup} \;=\; S^{\cup} \bullet R^{\cup} : \forall xyz\, (Cxyz \rightarrow CF(y)F(x)F(z)),$$

and the rather forbidding final axiom expresses a principle in the same vein:

$$R^{\cup} \bullet -(R \bullet S) \leqslant -S\text{: } \forall xyz\, (Cxyz \rightarrow CF(x)zF(y)). \quad \square$$

At least, these correspondences give us a more concrete view of the meaning of the earlier set of basic axioms for Relational Algebra.

Again, this modal logic of relations can be studied in greater technical detail (cf. van Benthem 1989b). For instance, the preceding example is a special case of a general definability theorem for modal formulas expressing first-order conditions on relational frames.

The exact connection between the modal perspective on information structures and that on transition relations remains to be explored.

5. COMPARISONS

The emergence of various formal analogies between the earlier kinds of models invites comparison.

First, in one direction, there is a natural embedding.

5.1. *From L-Models to R-Models*

Given a family of languages on a universe of expressions, one can map each language to a binary relation as follows

$$\rho(L_a) \;=\; \{(x, xy) \mid y \in L_a\}.$$

Moreover, let us restrict the universe of admissible pairs to those of the form (x, y) where x is an initial segment of y. Then, we can make the following observation:

ρ is a *homomorphism* with respect to $-$, $\cap$, $\bullet$, $\langle\rangle$ and $/$.

Illustration:

$\rho(L_{a\cap b}) = \rho(L_a \cap L_b) = \{(x, xy) | y \in L_a \;\&$
$y \in L_b\} = \rho(L_a) \cap \rho(L_b)$,
for the case of $\rho(L_{-a})$, the special restriction on admissible pairs is needed,
$\rho(L_{a \bullet b}) = \{(x, xyz) | y \in L_a, z \in L_b\} = \rho(L_a) \bullet \rho(L_b)$,
$\rho(L_{a/b}) = \{(x, xy) | y \in L_{b/a}\} = \rho(L_b)/\rho(L_a)$, by a simple calculation,
$\rho(\{\langle\rangle\}) = \{(x, x) |$ all expressions $x\} =$ id. □

Since ρ is also injective, this yields an isomorphic embedding, and we have found a

PROPOSITION. *For the* $\{-, \cap, \bullet, \langle\rangle, /\}$ *fragment, the universal first-order theory of R-models is contained in that of L-models.*

One immediate question is if this result can be extended so as to include the converse implication \, for which the above representation does not work.

Moreover, the analogy does not necessarily extend to additional operators.

EXAMPLE. If we map the earlier inversion on languages to relational converse, then, e.g., $(x \bullet y)^{\cup} = y^{\cup} \bullet x^{\cup}$ will be valid in both cases, but, e.g., $x^{\cup} \bullet -(x \bullet y) \leqslant -y$ will not: a linguistic counter-example is $x = \{a\}$, $y = \{b, ab\}$. □

Next, we consider the opposite direction.

5.2. *From R-Models to L-Models*

Given any relation, we can choose to re-interpret it as a set of 'symbols' (x, y), which can be 'concatenated' in the usual way. But, no homomorphic preservation of structure takes place here. Typically, the problem is that the composition $R \bullet S$ will not correspond to the concatenation product of R and S: as certain arrows may not match. Therefore, a representation will only succeed for very special 'uniform' relations. Or on the other side, one might have to consider syntax models allowing for possible *restrictions* on concatenation (certain symbols would not admit of justaposition). The latter idea might have some independent interest all the same.

In any case, there are principles which are valid on all L-models, but not on all R-models, reflecting the above difference.

$$L_a \bullet L_b = \varnothing \text{ implies that } L_a = \varnothing \textit{ or } L_b = \varnothing$$

but

$$R_a \bullet R_b = \varnothing \text{ does not imply } R_a = \varnothing \textit{ or } R_b = \varnothing.$$

Now this is still a 'higher' example, above the level of algebraic identities. (We can make our model comparisons, of course, at different levels of their logical 'description languages'.) But, given the earlier emphasis on Gentzen sequents, i.e., Horn clauses, i.e., algebraic identities, here is a more telling illustration.

EXAMPLE. The following principle is valid in all L-models, but not in all R-models:

$$((-(x \bullet x) \cap x) \bullet (-(x \bullet x) \cap x)) \cap \mathrm{id} = \bot.$$

On L-models. Suppose that some string a is in the intersection. I.e., $a = \langle\rangle \in (-(x \bullet x) \cap x) \bullet (-(x \bullet x) \cap x)$: so that $\langle\rangle \in -(x \bullet x) \cap x$: $\langle\rangle \in x$: whence $\langle\rangle = \langle\rangle\,\langle\rangle \in x \bullet x$, which is a contradiction.
On our R-models. Here is a relational counter-example:

$$R = \{\langle 1, 2\rangle, \langle 2, 1\rangle\}, \quad \mathrm{id} = \{\langle 1, 1\rangle, \langle 2, 2\rangle\}.$$

In fact, it is possible to rework this example into one not containing id: so that it is already the combination of $\bullet$ with Boolean operations which creates the difficulty. □

The upshot of this discussion seems to be that R-models form the more general class of structures for our analysis of information and its processing.

REMARK. It would be of interest, nevertheless, to extend the above comparisons so as to include the case of N-models, and I-models. A case in point are the earlier analogies between the previous uses of Modal Logic: once as a theory of information models, and once as a generalized form of Relation Algebra. □

6. COMBINATION: DYNAMIC LOGIC OF INFORMATION

Despite its technical interest, the preceding Section does not reflect the point of view advocated in this paper. In fact, there seems to be little evidence in favour of a reductionist stance, trying to reduce one intuitive perspective on information to another. Therefore, our final offering is a paradigm allowing for the coexistence of several strands in the story so far.

6.1. *Intensional Type Theory*

Consider a standard type theory, with primitive types t (truth values), s (indices, states, possible worlds), e (entities), allowing for the formation of functional and product types.

First, take the s, t fragment only ('propositional dynamic logic' in higher orders). Classical propositions have type (s, t), being 'static' sets of states, whereas dynamic relational propositions have the type $(s, (s, t))$. Now, instead of choosing between these two perspectives, we can have both, and study *transformations* between them.

For instance, there is an obvious map from $D_{(s,(s,t))}$ to $D_{(s,t)}$, being the *diagonal* function

$$\Delta(R) \quad = \quad \{x \mid (x, x) \in R\}$$

mapping R to its fixed points, which serve as the obvious associated truth set. And conversely, each static proposition gives rise to a *test* relation

$$?(P) \quad = \quad \{(x, x) \mid x \in P\}.$$

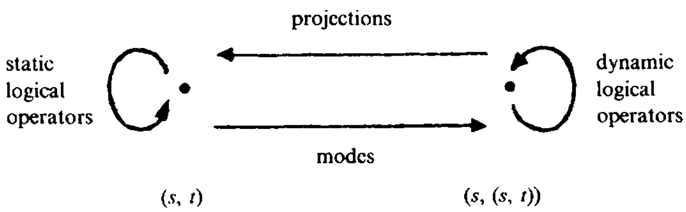
Note, e.g., that the composition $\Delta ?$ cancels out to identity on $D_{(s,t)}$. Moreover, these maps preserve a good deal of structure on their domains: for instance, Δ is a Boolean *homomorphism* and ? at least 'continuous' in a sense to be explained below.

Thus, the better perspective is rather to accept two distinct domains, and study two additional types:

$((s, (s, t)), (s, t))$: static 'projections' for dynamic propositions,

$((s, t), (s, (s, t)))$: dynamic 'modes' of using static propositions.

It is precisely this interplay between the two perspectives ('proposition as descriptions' versus 'propositions as programs') which gives Dynamic Logic its utility – e.g., when proving *correctness assertions* about program behaviour. And the same interaction seems useful for 'cognitive programming'. Thus, our world picture becomes this:



More interesting examples will only come to the fore, however, if we endow the base domain D_s with additional structure: say, at least the information ordering $\subseteq$ of Section 3. Then we also get dynamic modes such as the following:

$\lambda x_s \bullet \lambda y_s \bullet x \subseteq y \wedge P(y)$ 'indeterministic update'

$\lambda x_s \bullet \lambda y_s \bullet y \subseteq x \wedge \neg P(y)$ 'indeterministic downdate'

$\lambda x_s \bullet \lambda y_s \bullet x \subseteq y \wedge \neg \exists z (x \subsetneq z \subseteq y \wedge P(z))$ 'minimal update'

The resulting structure may be studied in a modal dynamic logic, having enough type-theoretical structure.

To repeat, the point here is not that propositions really 'are' dynamic entities. In fact, if static propositions did not exist already, they would have to be *invented*, in order to account for the common core in certain dynamic operations like updating, revision or querying. The present type-theoretical perspective allows us to study all possible semantic mechanisms involved here, without becoming committed to exclusive claims.

6.2. *Logical Issues*

There is much fine-structure to be investigated in the above general framework. The following discussion may serve as a first illustration (for more detail, as well as technical proofs, see van Benthem 1989c).

In general, it will become preferable now to formulate our semantic notions in a way which will apply to all relevant types at once, so as not to miss useful analogies. For instance, the earlier concern with *logical constants* can now be approached quite generally, using the notion of *invariance for permutations* of the individual states in D_s (cf. van Benthem 1986). With operators on (s, t), this will leave just the Booleans: but already with operators on $(s, (s, t))$, all the usual notions from Relation Algebra pass the test. But other items can be 'logical' in this sense too: for instance, the earlier-mentioned transformations Δ and ? both are.

Then, given the importance of Boolean structure, it makes sense, e.g., to locate all Boolean *homomorphisms* in our transformer types.

EXAMPLE. There is exactly one permutation-invariant homomorphic projection, namely the diagonal Δ. There are exactly two permutation-invariant homomorphic modes, viz.

$$\lambda P_{(s,t)} \bullet \lambda x_s \bullet \lambda y_s \bullet P(x), \quad \lambda P_{(s,t)} \bullet \lambda x_s \bullet \lambda y_s \bullet P(y). \qquad \square$$

A more liberal requirement would be *continuity*, in the sense of commuting with arbitrary unions of arguments (a notion which is ubiquitous in computational information-oriented settings). Then, e.g., also projections like *domain* or *range* of relations, and modes like the test ? qualify.

To arrive at a more genuinely modal analysis, one will have to take the inclusion structure on D_s seriously. Permutation invariance does not hold for modal operators, like $\Diamond$, or modal modes, like updating. The reason is that they may destroy all information about the ordering pattern $\subseteq$. Truly modal notions will only be invariant, then, for *inclusion automorphisms* of D_s.

On the other hand, one can also strengthen modal invariances, by developing proper type-theoretical generalizations of the various notions of bisimulation found in Section 3. For instance, van Benthem 1989c has one such proposal for bisimulation invariance across all types, which allows us to locate the analogues of the basic modal operators among transformers, and other relevant types.

Another general perspective from earlier Sections was the use of a first-order description language in the background. This is also quite possible here. In fact, at least the move from (s, t) to also including $(s, (s, t))$ type entities is entirely natural from the point of view of the earlier monadic $\subseteq$-language. One now merely admits formulas having *two* free variables instead of one: a move which had already been considered in the area for various technical reasons. And then, many of the earlier notions and results apply without any great effort.

The possibility of transcription into first-order formalisms, then, which has always been well-known for ordinary Intensional Logic, is not affected at all by including the new dynamic perspective. Hence, despite frequent misconceptions on this score, at least technical reductions to classical 'static' systems are always possible.

On the other hand, the new setting certainly suggests many *new questions* which would not easily have come up without it. For instance, one might be interested in classifying propositions as to their 'informational content' (upward persistent, closed under sums, etcetera) or propositional relations as to their various kinds of behaviour along the information ordering (upward looking, idempotent, or other important special classes).

Finally, as another item for the agenda, an eventual system will also have to incorporate the *individual* domain D_e needed to get at individual *predication* and *quantification*. What remains to be seen, however, is whether any essentially new *dynamic* phenomena will come to light here. One possible relevant example is the notion of *querying*, where our information state increases by growing acquaintance with individuals and their relationships.

7. CONCLUSIONS

A number of general points behind the story of this paper may be worth setting out separately, now that we have come to the end.

There is perhaps one obvious omission to be addressed right away. Although the word "information" has occurred throughout this paper, it must have struck the reader that we have had nothing to say on what information *is*. In this respect, our theories may be like those in physics: which do not explain what "energy" is (a notion which seems quite similar to "information" in several ways), but only give some basic laws about its *behaviour* and *transmission*.

The eventual recommendation made here has been to use a broad type-theoretic framework for studying various more classical and more dynamic notions of proposition in their interaction. This is not quite the viewpoint advocated by many current authors in the area, who argue for a whole-sale *switch* from a 'static' to a 'dynamic' perspective on propositions. This is not the place, however, to survey the conceptual arguments for and against such a more radical move.

This still leaves many questions about possible reductions from one perspective to another. For instance, it would seem that classical systems ought to serve as a 'limiting case', which should still be valid after procedural details of some cognitive process have been forgotten. There are various ways of implementing the desired correspondence: e.g. by considering extreme cases with $\subseteq$ equal to identity, or, in the pure relational algebra framework by considering only pairs (x, x). What we shall want then are reductions of dynamic logics, in those special cases, to classical logic. But perhaps also, more sophisticated views are possible. How do we take a piece of 'dynamic' prose, remove control instructions and the like, and obtain a piece of 'classical' text, suitable for inference 'in the light of eternity'?

There is also a more technical side to the matter of 'reduction'. By now, Logic has reached such a state of 'inter-translatability' that almost all known variant logics can be embedded into each other, via suitable translations. In particular, once an adequate semantic has been given for a new system, this usually induces an embedding into standard logic: as we know, e.g., for the case of Modal Logic. Likewise, all systems of dynamic interpretation or inference proposed so far admit of direct embedding into an ordinary 'static' predicate logic having explicit transition predicates (cf. van Benthem 1988b). Thus, our moral is this. The issue is not whether the new systems of information structure or processing are essentially beyond the expressive resources of traditional logical systems: for, they are not. The issue is rather which interesting phenomena and questions will be put into the right focus by them.

The next broad issue concerns the specific use of the perspective proposed here, *vis-à-vis* concrete proposals for information-oriented or dynamic semantics. The general strategy advocated here is to locate some suitable base calculus and then consider which 'extras' are peculiar to the proposal. For instance, this is the spirit in which modal S4 would be a base logic of information models, and intuitionistic logic the special theory devoting itself to upward persistent propositions. Or, with the examples in Section 4.1, the underlying base logic is our relational algebra, whereas, say, ordinary updates then impose special properties, such as 'idempotence':

$$xRy \Rightarrow yRy.$$

Does this kind of application presuppose the existence of one distinguished base logic, of which all others are extensions? This would be attractive – and some form of relational algebra or linear logic might be a reasonable candidate. Nevertheless, the enterprise does not rest on this outcome. What matters is an increased sensitivity to the 'landscape' of dynamic logics, just as with the 'Categorial Hierarchy' in Categorial Grammar (cf. van Benthem 1989a, 1991) where the family of logics with their interconnections seems more important than any specific patriarch.

Finally, perhaps the most important issue in the new framework is the possibility of new kinds of questions arising precisely because of its differences from standard logic. Notably, given the option of regarding propositions as *programs*, it will be of interest to consider systematically which major questions about programming languages now make sense inside logic too.

EXAMPLE. *Correctness*. When do we have

$$[\![\pi]\!]([\![A]\!]) \subseteq [\![B]\!]$$

for (s, t) propositions A, B and a dynamic $(s, (s, t))$ proposition π?

Program Synthesis. Which dynamic proposition will take us from an information state satisfying A to one satisfying B? (This question needs refinement, lest there be trivial answers.)

Determinism. Which propositions as programs are deterministic, in the sense of defining single-valued functions from states to states?

Querying. What does it mean to ask for information in the present setting? (Again, individual types referring to e will be crucial here.)

This is not merely an agenda for wishful thinking. Within Logic, there are various ways of introducing such concerns into semantics, especially, using tools from *Automata Theory*. (See van Benthem 1989c for further discussion of such computational perspectives in 'cognitive programming'.) □

At least if one believes that 'dynamics' is of the essence in cognition (rather than a mere interfacing problem between the halls of eternal truth and the noisy streets of reality), the true test for the present enterprise is the development of a significant new research program not merely copying the questions of old.

8. REFERENCES

Abrusci, V. M., 1988a, 'Sequent Calculus for Intuitionistic Linear Propositional Logic', report 1, Department of Philosophy of Science, University of Bari.

Abrusci, V. M., 1988b, 'A Comparison between Lambek Syntactic Calculus and Intuitionistic Linear Propositional Logic', report 2, Department of Philosophy of Science, University of Bari.

Avron, A., 1988, 'The Semantics and Proof Theory of Linear Logic', *Theoretical Computer Science* **57**, 161–184.

Barwise, J., 1987, 'Noun Phrases, Generalized Quantifiers and Anaphora', in P. Gardenfors, ed., 1987, 1–29.

Benthem, J. van, 1985, *Modal Logic and Classical Logic*. Bibliopolis, Napoli/The Humanities Press, Atlantic Heights.

Benthem, J. van, 1986, *Essays in Logical Semantics*, D. Reidel, Dordrecht.

Benthem, J. van, 1987, 'Categorial Grammar and Type Theory', report 87-07, Institute for Language, Logic and Information, University of Amsterdam. Also appeared in *Journal of Philosophical Logic* **19**, 115–168.

Benthem, J. van, 1987*, 'Polyadic Quantifiers', report 87-04, Institute for Language, Logic and Information, University of Amsterdam. Also appeared in *Linguistics and Philosophy* **12**, 437–464.

Benthem, J. van, 1988a, 'The Lambek Calculus', in R. Oehrle *et al.*, eds., 1988, 35–68.

Benthem, J. van, 1988b, 'Semantic Parallels in Natural Language and Computation', report 88-06, Institute for Language, Logic and Information, University of Amsterdam. Also appeared in H.-D. Ebbinghaus *et al.*, eds., 1989, 331–375.

Benthem, J. van, 1988c, 'Logical Constants across Varying Types', report LP-88-05, Institute for Language, Logic and Information, University of Amsterdam. Also appeared in *Notre Dame Journal of Formal Logic* **30**, 315–342.

Benthem, J. van, 1989a, 'The Fine-Structure of Categorial Semantics', report LP-89-01, Institute for Language, Logic and Information, University of Amsterdam. (To appear in M. Rosner, ed., *Lugano Workshop on Computational Linguistics and Formal Semantics*, Cambridge University Press.)

Benthem, J. van, 1989b, 'Relational Algebra from The Perspective of Modal Logic', Institute of Language, Logic and Information, University of Amsterdam.

Benthem, J. van, 1989c, 'Modal Logic as a Theory of Information', to appear in J. Copeland, ed., *Proceedings Prior Memorial Conference*, Christchurch, New Zealand.

Benthem, J. van, 1991, *Language in Action: Categories, Lambdas and Dynamic Logic*, North-Holland, Amsterdam (Studies in Logic).

Buszkowski, W., 1982, *Lambek's Categorial Grammars*, dissertation, Institute of Mathematics, Adam Mickiewicz University, Poznan.

Buszkowski, W., 1986, 'Completeness Results for Lambek Syntactic Calculus', *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **32**, 13–28.

Buszkowski, W., Marciszewski, W., and Benthem, J. van, eds., 1988, *Categorial Grammar*, John Benjamin, Amsterdam and Philadelphia.

Dalla Chiara, M.-L., 1985, 'Quantum Logic', in Gabbay & Guenthner, eds., 1985, 427–469

Došen, K., 1985, 'A Completeness Theorem for the Lambek Calculus of Syntactic Categories', *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **31**, 235–281.

Dunn, M., 1985, 'Relevance Logic and Entailment', in Gabbay & Guenthner, eds., 1985, 117–224.

Ebbinghaus, H.-D., *et al.*, eds., 1989, *Logic Colloquium. Granada 1987*, North-Holland, Amsterdam, (Studies in Logic).

Fitch, F. B., 1952, *Symbolic Logic: an Introduction*, New York.

Gabbay, D. and Guenthner, F., eds., 1984, *Handbook of Philosophical Logic*, vol. II (Extensions of Classical Logic), D. Reidel, Dordrecht.

Gabbay, D. and Guenthner, F., eds., 1985, *Handbook of Philosophical Logic*, vol. III (Alternatives to Classical Logic), D. Reidel, Dordrecht.

Gabbay, D., Hogger, C. and Robinson, J., eds., 1990, *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press, Oxford.

Gabbay, D., Pnueli, A., Shelah, S., and Stavi, Y., 1980, 'On the Temporal Analysis of Fairness', *ACM Symposium on Principles of Programming Languages*, 163–173.

Gärdenfors, P., ed., 1987, *Generalized Quantifiers. Linguistic and Logical Approaches*, D. Reidel, Dordrecht.

Gärdenfors, P., 1988, *Knowledge in Flux: Modelling the Dynamics of Epistemic States*, Bradford Books/MIT Press, Cambridge (Mass.).

Gärdenfors, P. and Makinson, D., 1988, 'Revision of Knowledge Systems Using Epistemic Entrenchment', in M. Vardi, ed., 1988, 83–95.

Gargov, G., Passy, S., and Tinchev, T., 1987, 'Modal Environment for Boolean Speculations', in D. Skordev, ed., 1987, 253–263.

Girard, J.-Y., 1987, 'Linear Logic', *Theoretical Computer Science* **50**, 1–102.

Groenendijk, J. and Stokhof, M., 1988, 'Dynamic Predicate Logic', Institute for Language, Logic and Information, University of Amsterdam. (To appear in *Linguistics and Philosophy*.)
Harel, D., 1984, 'Dynamic Logic', in Gabbay & Guenthner, eds, 1984, 497–604.
Heim, I., 1982, *The Semantics of Definite and Indefinite Noun Phrases*, dissertation, Department of Linguistics, University of Massachusetts, Amherst.
Hopcroft, J. and Ullman, J., 1979, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading (Mass.).
Immerman, N. and Kozen, D., 1987, 'Definability with Bounded Number of Bound Variables', *Proceedings IEEE* 1987, 236–244.
Jónsson, B. and Tarski, A., 1951, 'Boolean Algebra with Operators I', *American Journal of Mathematics* **73**, 891–939.
Jónsson, B., 1984, 'The Theory of Binary Relations', Department of Mathematics, VanderBilt University, Nashville (Tenn.).
Koymans, R., 1989, *Specifying Message Passing and Time-Critical Systems with Temporal Logic*, dissertation, Department of Computer Science, Technological University, Eindhoven.
Lafont, Y., 1988, The Linear Abstract Machine', *Theoretical Computer Science* **59**, 157–180.
Lambek, J., 1958, 'The Mathematics of Sentence Structure', *American Mathematical Monthly* **65**, 154–170.
Maddux, R., 1983, 'A Sequent Calculus for Relation Algebras', *Annals of Pure and Applied Logic* **25**, 73–101.
Makinson, D., 1988, 'General Non-Monotonic Logic', to appear in Gabbay *et al.*, eds., 1990.
Moortgat, M., 1988, *Categorial Investigations. Logical and Linguistic Aspects of the Lambek Calculus*, Foris, Dordrecht (GRASS series).
Oehrle, R., Bach, E., and Wheeler, D., eds, 1988, *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht (Studies in Linguistics and Philosophy).
Ono, H., 1988, 'Structural Rules and a Logical Hierarchy', Faculty of Integrated Arts and Sciences, Hiroshima University.
Quine, W. V. O., 1946, 'Concatenation as a Basis for Arithmetic', *Journal of Symbolic Logic* **11**, 105–114.
Sambin, G., 1988, 'Intuitionistic Formal Spaces and their Neighbourhood', Mathematical Institute, University of Padova.
Skordev, D., ed., 1987, *Mathematical Logic and its Applications*, Plenum Press, New York
Troelstra, A. and Dalen, D. van, 1988, *Constructivism in Mathematics: an Introduction*, North-Holland, Amsterdam (Studies in Logic).
Urquhart, A., 1972, 'Semantics for Relevant Logics', *Journal of Symbolic Logic* **37**, 159–169.
Vardi, M., ed., 1988, *Proceedings 2d Conference on Theoretical Aspects of Reasoning about Knowledge*, Morgan Kaufmann Publishers, Los Altos.
Veltman, F., 1989, 'Update Semantics', Institute for Language, Logic and Information, University of Amsterdam.
Wansing, H., 1989, 'The Adequacy Problem for Sequential Propositional Logic', Institute for Language, Logic and Information, University of Amsterdam.
Zucker, J. and Tragesser, R., 1978, 'The Adequacy Problem for Inferential Logic', *Journal of Philosophical Logic* **7**, 501–516.

Institute for Language, Logic and Information,
University of Amsterdam,
The Netherlands