Institute for Language, Logic and Information

LOGIC AND THE FLOW OF INFORMATION

Johan van Benthem

ITLI Prepublication Series
for Logic, Semantics and Philosophy of Language LP-91-10

University of Amsterdam

Instituut voor Taal, Logica en Informatie
Institute for Language, Logic and Information

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Plantage Muidergracht 24
1018TV Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP Amsterdam

LOGIC AND THE FLOW OF INFORMATION

Johan van Benthem
Department of Mathematics and Computer Science
University of Amsterdam

ITLI Prepublications
for Logic, Semantics and Philosophy of Language
ISSN 0924-2082

Received November 1991

to appear in
D. Prawitz, B. Skyrms and D. Westerståhl, eds.
Proceedings 9th International Congress of Logic, Methodology and Philosophy of Science, Uppsala 1991
North-Holland, Amsterdam

LOGIC AND THE FLOW OF INFORMATION

Johan van Benthem

Institute for Logic, Language and Computation
University of Amsterdam

september 1991

to appear in

D. Prawitz, B. Skyrms & D. Westerståhl, eds.

Proceedings 9th International Congress of Logic, Methodology and Philosophy of Science. Uppsala 1991

North-Holland, Amsterdam

I From Propositions to Procedures

At the core of standard logic is the notion of a declarative sentence, whose truth conditions in varying situations are the prime target of investigation. Of course, actual linguistic communication involves transient discourse and cognitive change, but this dynamics remains an 'extrinsic' feature of the use that is made of logical propositions. But gradually, a reversal of priorities is taking place in the literature, and many authors have focused instead on the potential for information change inherent in propositions. There are different sources for this trend (which can be traced back as an undercurrent far back into this century). In particular, in linguistics, dynamic information flow occurs at various levels. In categorial parsing, categories serve as procedures acting on each other consecutively to produce sentence meanings (Moortgat 1988, van Benthem 1991), at the sentence level, processing of anaphoric dependencies involves shifting variable assignments, quite like the workings of imperative computer programs (Barwise 1987, Groenendijk&Stokhof 1991), and finally, discourse has an obvious dynamic global structure with a sequential game-like character (Hintikka 1973, Hintikka&Kulas 1983). Taken together, these observations suggest that natural languages are more like programming languages, serving various cognitive purposes, than like standard declarative formal languages. This view reflects a more general move in contemporary philosophy, away from static 'knowledge' to dynamic 'cognition', putting cognitive procedures like updating, retraction or revision of information at centre stage (Gärdenfors 1988, Harman 1985) rather than static representational structures. (Of course, such an interest in cognitive change still presupposes some account of standard cognitive content.) In other words, one is moving from 'extrinsic' dynamics to 'intrinsic' dynamics at the core of a logic of information flow.

The purpose of this paper is to put forward a general perspective on these matters, inspired by dynamic logic in computer science (Harel 1984), and then to identify some salient general logical issues emerging behind many specific systems for linguistic and general cognitive purposes published so far. Notably, our 'procedural turn' leads to a reappraisal of traditional notions like 'logical constant' and 'valid inference', while also raising new issues of 'logical architecture'. We shall consider both a very general procedural logic and various possible specializations, showing that traditional methods of analysis still apply, when given an appropriate new twist. Our presentation follows the main lines of van Benthem 1991, Part VI (to which we refer for many details in what follows), while adding some further refinements and results obtained in the meantime. We shall not propose any specific system for performing the new cognitive tasks, but rather concentrate on foundational issues concerning their design.

Dynamics Within Classical Logics

An immediate entry to procedural thinking takes its departure from any basic text book in standard logic. Consider Tarski's well-known definition for truth of a formula ϕ in a model $M = (D, I)$ under some variable assignment a. Its atomic clause involves a static test whether some fact obtains, but intuitively, the clause for an existential quantifier $\exists x$ involves shifting an assignment value for x until some verifying object has been found. But then, we may also make the latter process explicit, by assigning to each formula a binary relation consisting of those transitions between assignments a which result in its succesful verification. Moreover, eventually, other components of the truth definition admit of dynamization too. For instance, shifting interpretation functions I are involved in ambiguous discourse (van Deemter 1991) or questioning, and eventually, shifting model structures M make sense too in the dynamics of domain change across sentences (Westerståhl 1984). One immediate question arising on this point of view is how to interpret the standard logical constants. Some stipulations seem clear: for instance, most people would agree on letting conjunction stand for sequential 'composition' of transition relations, while disjunctions would amount to some kind of 'choice'. But we shall analyze the options in a more principled way later on.

Another point of departure from standard logic lies in what are the best-known classical information-oriented model structures, namely the possible worlds models for intuitionistic logic proposed by Kripke. Here, worlds stand for information states, and the intuitive picture is that of a cognitive agent traversing such states in the quest for knowledge. Again, intuitively, intuitionistic formulas refer to transitions in this information pattern. E.g., to see that $\neg\phi$ holds, one has to inspect all possible extensions of the current state for absence of ϕ. As before, this dynamics may also be made an explicit part of the logic, by creating a system of cognitive transitions, such as 'updates' taking us to some minimal extension where a certain proposition has become true. Standard intuitionistic logic is then a forward-looking system, of Dutch mathematicians who never forget and never err, whereas ordinary mortals will display a zigzagging traveling pattern of cognitive advances and retreats, including 'downdates' and 'revisions'. Providing an explicit dynamic system here may even be viewed as taking the original 'constructivist' motivation to its logical conclusion.

Dynamics of Inference

Finally, let us take a look at what this dynamic view of logic would mean for the archetypal inferential setting:

$$\frac{P_1 \quad \ldots \quad P_n}{C} \qquad \text{conclusion } C \text{ follows from premises } P1, \ldots, Pn$$

What is the sense of this when all propositions involved are procedures changing information states? One natural explanation would be as follows. The premises of an argument invite us to transform our initial information state, and then the resulting transition has to be checked to see whether this 'warrants' the conclusion (in some suitable sense). Later on, we shall give a number of ways in which this may be made precise. For the moment, one consequence of this view needs to be pointed out. If the sequence of premises is a complex instruction for achieving some cognitive effect, then its presentation will be crucial. The sequential order of premises matters, the multiplicity of their occurrence matters, and each premise move has to be relevant. And this will bring us into open conflict with even the most basic 'structural rules' of standard logic (allowing us to disregard such aspects in classical reasoning). Think of meeting a date, where one has all the right moves available: flowers, tickets, sweet talking, kisses, and imagine the various ways in which succesful seduction might fail by Permutation of actions, Contraction of identical actions, or Monotonic Insertion of arbitrary additional actions. Of course, in certain settings, deviations from classical reasoning will be slight, for instance, when all premise actions correspond to tests, or steady updates. But in general cognition, our information may be more complex, with the information prior to inference also containing retractions ("no, forget about A after all") or qualifications ("unless B , that is") . And then, a more delicate dynamic logic becomes imperative.

II General Dynamics: Relational Algebra and Arrow Logic

Relational Algebra as Procedural Logic

Underlying many specific systems of dynamic logic is the usual mathematical concept of a 'state space'

$$(S, \{R_p \mid p \in P\})\ .$$

States may range here from cognitive constructs such as assignments or sets of possible worlds to real physical situations. Over these, there will be an 'atomic repertoire' of basic actions that can be performed, such as shifting the value in some register, adding or removing a world, kicking some round object. What atomic actions are appropriate depends on the particular choice of states, of course. Moreover, in particular settings, certain broad constraints on possible actions may be imposed from the start. For instance, updates are often assumed to satisfy a principle of 'idempotence': repeating them is unnecessary , that is $\forall xy\colon Rxy \rightarrow Ryy$. (This makes updating different from physical activities like kicking, or explaining something to one's students.)

On top of the atomic repertoire, which is taken for granted, there is a 'procedural repertoire' of various operations for creating compound actions, which we use for designing our programs or plans. Examples of such procedural operations are sequential composition, choice, iteration, but the literature also knows more exotic proposals. One example is the negation test of Groenendijk&Stokhof 1991, which reads as follows:

$$\neg R \quad = \quad \{ (x, x) \mid \text{for no } y, (x, y) \in R \} .$$

Another case are the directed functions of categorial grammar, whose procedural force is as follows:

$$A\backslash B \quad = \quad \{ (x, y) \mid \text{for each } z \text{ with } (z, x) \in A, \ (z, y) \in B \}$$

$$B/A \quad = \quad \{ (x, y) \mid \text{for each } z \text{ with } (y, z) \in A, \ (x, z) \in B \}$$

Of course, the basic choices made may also influence our freedom here. For instance, if all admissible actions are to be idempotent, then composition need not always be a safe combination, while choice or iteration do preserve idempotence.

What is happening here is a move from a standard Boolean Algebra of propositions to a Relational Algebra. The standard procedural repertoire in relational algebras is as follows:

Boolean operations	$-$ (complement)	$\cap$ (intersection)	$\cup$ (union)
Ordering operations	o (composition)	$^{\vee}$ (converse)	

with a distinguished diagonal relation Δ standing for the sweet achievement of 'dolce far niente'. (At the other extreme, the Boolean structure also provides a universal relation T of 'random activity'.) These operations are definable in a standard predicate logic with variables over states:

$-R$	:	$\lambda xy\bullet \neg Rxy$
$R\cap S$	:	$\lambda xy\bullet Rxy\wedge Sxy$
$R\cup S$	:	$\lambda xy\bullet Rxy\vee Sxy$
RoS	:	$\lambda xy\bullet \exists z\,(Rxz\wedge Szy)$
$R^{\vee}$	:	$\lambda xy\bullet Ryx$

The expressive power of this formalism shows in that it can define many other proposed procedural operators. In particular,

$\neg R$	:	$\Delta\cap-(R\,o\,T)$
$A\backslash B$	:	$-(A^{\vee}\,o\,-B)$
B/A	:	$-(-B\,o\,A^{\vee})$

The literature on Relational Algebra contains many results concerning axiomatization of valid identities between such relational expressions, as well as expressive power of various choices of operators (see Németi 1991). Some of these become relevant to our general procedural logic, as will be shown below.

Modal Arrow Logic and Categorial Grammar

In the long run, existing Relational Algebra would not be our favourite candidate for analyzing dynamic logic. Transition relations record rather little about the internal structure of processes, and some more delicate form of 'process algebra' (Milner 1980) will probably be needed sooner or later. Moreover, there are a number of mathematical complications in the subject as it exists, having to do with an insistence on set-theoretic relations consisting of ordered pairs. But intuitively, dynamic relations rather seem to consist of 'transitions' or 'arrows' as objects in their own right. Therefore, our preference would be to use a more abstract Arrow Logic (van Benthem 1989, Venema 1991). This may be viewed as a modal logic over 'arrow frames'

$$(W, C, F, I)$$

with a set W of 'arrows', a ternary relation C of 'composition', a binary relation F of 'conversion' and a unary predicate I for 'identical arrows'. The basic truth definition then explains the notion $M, x \models \phi$ (formula ϕ holds for the arrow x), so that formulas will describe sets of arrows, i.e., 'transition relations' in the new sense. For instance, some key clauses will be as follows:

$M, x \models A \cap B$	iff	$M, x \models A$ *and* $M, x \models B$
$M, x \models A \circ B$	iff	*there exist* y, z *such that* Cx,yz *and* $M, y \models A$ *and* $M, z \models B$
$M, x \models A^{\vee}$	iff	*there exists* y *such that* Fxy *and* $M, y \models A$
$M, x \models \Delta$	iff	Ix .

Arrow Logic is a minimal theory of composition of actions, which may be studied completely by well-known techniques from Modal Logic (cf. also Roorda 1991, Vakarelov 1991). Standard principles of Relational Algebra then express certain constraints on arrow patterns, which can be determined in the usual style through 'frame correspondences' (van Benthem 1985). For instance, an algebraic law like $(A \cup B)^{\vee} = (A^{\vee} \cup B^{\vee})$ is a universally valid principle of 'modal distribution' on arrow frames, but $(A \cap B)^{\vee} = (A^{\vee} \cap B^{\vee})$ expresses the genuine constraint that the conversion relation F be a partial function, whose idempotence would be expressed by the modal axiom $A^{\vee\vee} = A$. For technical convenience (and no more), we shall assume henceforth that there is an idempotent (and hence injective) conversion function f available in arrow frames.

It may be of interest now to see what dynamic content is expressed by some basic categorial laws of natural language. Here is a sample, demonstrating the use of modal correspondence techniques.

Proposition. $A \bullet (A \backslash B) \Rightarrow B$ expresses that $\forall xyz$: $Cx,yz \rightarrow Cz,f(y)x$
$(B/A) \bullet A \Rightarrow B$ expresses that $\forall xyz$: $Cx,yz \rightarrow Cy,xf(z)$.

Together, these two principles express the basic 'rotations' that can be made in composition (e.g., they imply the familiar law $\forall xyz$: $Cx,yz \rightarrow Cf(x),f(z)f(y)$).

Proof. In its arrow transcription, the first categorial principle has the modal form $(A \circ \neg (A^{\vee} \circ \neg B)) \rightarrow B$. Consider any arrow frame satisfying the stated constraint on its composition relation C . Let the antecedent $(A \circ \neg (A^{\vee} \circ \neg B))$ be true at x (under some valuation), and suppose that B fails at x . By the truth definition for $\circ$, there exist arrows y, z with Cx,yz , A true at y and $\neg (A^{\vee} \circ \neg B)$ true at z . But Cx, yz implies Cz,f(y)x , and we have $A^{\vee}$ true at f(y) (by the truth definition for $^{\vee}$ and idempotence of f), $\neg B$ true at x . Therefore $A^{\vee} \circ \neg B$ is true at z : which is the required contradiction. Conversely, suppose that our categorial law holds in a frame. Consider any situation Cx,yz . Define the following valuation V on the relevant proposition letters: $V(A) = \{y\}$, $V(B) = W - \{x\}$. Evidently, B fails at x , and hence so does the modal antecedent $A \circ \neg (A^{\vee} \circ \neg B)$. That is, either A must fail at y (which is impossible by the definition of V) or $\neg (A^{\vee} \circ \neg B)$ fails at z . Then there must be u, v with Cz,uv , $A^{\vee}$ true at u , $\neg B$ true at v . As A is only true at y , injectivity of f implies that $A^{\vee}$ can only be true at f(y) , whence $u = f(y)$. As B is only false at x , also $v = x$. But then we have Cz,f(y)x , as desired.

The second categorial equivalence may be proved in the same manner. But there is a more general observation to be made. Both categorial principles shown above exhibit a special form: they are so-called 'Sahlqvist formulas' in this binary modal logic. This may be seen by rewriting, e.g., the first to $(A \circ \neg (A^{\vee} \circ \neg B)) \wedge \neg B \rightarrow \bot$, or equivalently to $(A \circ \neg (A^{\vee} \circ B)) \wedge B \rightarrow \bot$, where the antecedent has its positive occurrences of A, B only in 'existential surface positions'. Thus the 'substitution algorithm' of van Benthem 1985 applies, which produces first-order corresponding conditions automatically (cf. also Venema 1991 for this technique). For instance, in this particular case, the formula generates a prefix $\forall xyz$: $Cx,yz \rightarrow$, and a substitution '$A = \{y\}, B = \{x\}$' which produces exactly the above frame condition. □

Which frame conditions on C and f are expressed by further categorial laws, such as Geach Composition or Montague Raising? Often, the substitution algorithm will apply, but not all categorial laws have Sahlqvist forms, and hence not all of them need be first-order definable. Thus, the precise dynamic content of Categorial Grammar remains to be investigated. In particular, one would like to determine the precise procedural counterpart of the basic 'Lambek Calculus'.

One could also reverse the perspective here, trying to embed Arrow Logic into a Lambek Calculus with operators $\backslash$, / , o , suitably enriched with Boolean operators $\neg$, $\wedge$, $\vee$ and an 'identity constant' *id* . Its deductive principles are the obvious union of categorial and Boolean laws, together with some suitable axioms concerning *id* . For instance, one possible rendering of relational conversion is as follows:

$$R^{\vee} \quad = \quad \neg\,(R \backslash \neg\, id)\,.$$

In this way, procedural principles also acquire categorial content. Here is an illustration:

- The procedural principle $(R \cup S)^{\vee} = (R^{\vee} \cup S^{\vee})$ translates into the equivalence
  $\neg\,((R \vee S) \backslash \neg id) \quad \leftrightarrow \quad \neg\,(R \backslash \neg id) \vee \neg\,(S \backslash \neg id)$.
  The latter exemplifies a derivable law of the Boolean Lambek Calculus, viz.
  $(A \vee B) \backslash C \quad \leftrightarrow \quad (A \backslash C) \wedge (B \backslash C)$.
- The procedural principle $R^{\vee\vee} = R$ translates into a less straightforward categorial law concerning *id* . For instance, one half would state essentially that $\neg\,((\text{"}R^{\vee}\text{"} \circ \neg R) \wedge id)$.

Thus, there exists an evident duality between categorial logic and procedural logic, whose further exploration must be foregone here.

Conclusion. Relational Algebra is a useful paradigm for bringing out general options of design for procedural systems of logic. Nevertheless, some more abstract framework like Arrow Logic will be desirable eventually. Either way, procedural logic may be studied using standard semantic tools from Modal Logic.

III **Logical Constants as Operators of Control**

Logical constants in standard logic are the key operators forming new propositions out of old ones. In dynamic logic, logical constants will be the key operators of control, combining procedures. Now, much of the recent literature still has a conservative bias, in that the only issue raised is 'what the standard logical constants mean' in a dynamic setting. But in fact, the latter allows for finer distinctions than the standard one, so that there may not be any clear sense to this question. Thus, it has to be analyzed on its own merits. For instance, standard 'conjunction' really collapses various notions: sequential composition, but also various forms of parallel composition. Likewise, standard 'negation' may be either some test as above, or merely an invitation to make any move refraining from some forbidden action ("anything, as long as you leave your father alone"). And also, there will be natural logical operators in the dynamic setting which lack classical counterparts altogether, such as conversion or iteration of procedures.

Logicality as Permutation Invariance

Nevertheless, there is a general perspective relating the two notions. Intuitively, 'logical' operators do not care about specific individual objects involved in their arguments. This is also true for procedural operators. What makes, say, a complement $-R$ a logical negation is that it works uniformly on all 'arrow patterns' R, in contrast to a negative social operator like 'Dutch' whose action depends on the content of its relational arguments ("Dutch dining" means making one's guests pay for themselves, "Dutch climbing" is running to the top ahead of one's companions just at the finish). The common mathematical generalization involves *invariance under permutations* π of the underlying set of relevant individuals (here, states):

- Declarative propositions denote unary properties / sets of states, and hence propositional operators satisfy

$$\pi\,[O\,(X, Y, \ldots)] \quad = \quad O\,(\pi\,[X], \pi\,[Y], \ldots)$$

- Dynamic procedures denote binary relations / sets of ordered pairs of states, and hence procedural operators satisfy the same schema:

$$\pi\,[O\,(R, S, \ldots)] \quad = \quad O\,(\pi\,[R], \pi\,[S], \ldots).$$

A Procedural Hierarchy

This mathematical condition still leaves a host of possible relational operators. To get a finer view of the options, a more 'linguistic' perspective may be taken, scrutinizing the form of definition for relational operators. For instance, the earlier examples had definitions in a first-order language having variables over states and binary relation letters for procedures. Now, one reasonable measure of complexity is the number of variables essentially employed in such a defining schema, which tells us what is the largest configuration of states involved in determining the action of the operator. For instance, intersection of relations employed only two variables, whereas composition involved three. And the resulting 'finite variable levels' provide an obvious Procedural Hierarchy of complexity against which we can measure proposed procedural operations. (Of course, some infinitary version of the first-order language will be needed to include operators like iteration and its ilk.) Here are some facts about this hierarchy, provable using model-theoretic Ehrenfeucht-Fraissé games:

Proposition.
- The usual similarity type of Relational Algebra is functionally complete for all relational operators with a three-variable defining schema.
- Each n-variable level has a finite functionally complete set of operators.
- There is no finite functionally complete set of algebraic operators for the whole procedural hierarchy at once.

Even finite-variable layers still contain a host of less plausible operators, through contrived definitions. But then, further constraints may be imposed, say of some reasonable computational character. One example is the well-known condition of

Continuity $O(\ldots, \cup_{i \in I} R_i, \ldots) = \cup_{i \in I} O(\ldots, R_i, \ldots)$.

This forces the operation to determine its values 'locally', by inspecting single transitions (using the fact that $R = \cup \{\{(x, y)\} \mid Rxy\}$):

Proposition. • Each continuous operation can be written in an existential form (displayed here for the two-argument case $O(R, S)$ only)

$\lambda xy \bullet \exists zu\ (Rzu \wedge \exists vw\ (Svw \wedge$

'Boolean combination of identities in $\{x, y, z, u, v, w\}$ ' .

• For each fixed arity, there are only finitely many continuous permutation-invariant relational operators.

Examples of continuous operations are Boolean intersection and union, as well as relational composition and converse. A non-example is Boolean complement.

Continuity in this strong form rules out too much, although it does describe a special 'natural kind' of logical operator. (Belnap 1977 proposes a weaker notion of 'Scott continuity' admitting more candidates.) Therefore, other 'computational' constraints on logicality of procedural operators become of interest too. First, logical constants should not generate 'unfeasible' transitions:

Feasibility Transitions for defined relations must be reachable through some finite sequence of basic actions.

Like Continuity, this rules out complement, while accepting conjunctions, disjunctions or compositions of procedures.

Next, Feasibility can be strengthened by a constraint which illustrates a broadly applicable line of thinking. Consider the important notion of 'simulation' of one process via another, which is crucial to computation. One well-known candidate for this purpose is 'bisimulation' in the usual sense of having a relation C between states in two transition models $(S, \{R_p \mid p \in P\})$, $(S', \{R'_p \mid p \in P\})$ satisfying the following back-and-forth clauses:

- if $x\,C\,x'$, $x\,R_p\,y$, then there exists some y' with $y\,C\,y'$, $x'\,R'_p\,y'$
- if $x\,C\,x'$, $x'\,R'_p\,y'$, then there exists some y with $y\,C\,y'$, $x\,R_p\,y$

Logical constants should not 'disturb' such connections:

Simulability Any simulation for basic actions must automatically be one for complex actions defined by logical constants.

Logicality and Simulation

With bisimulation in the ordinary sense as a measure of process equivalence, the effect of Simulability will be to rule out essentially all but the 'regular program operations' $\cup$ (Boolean union), $\circ$ (composition), * (infinitary Kleene iteration), together with the following functions of 'domain' $\Diamond$ and 'counter-domain' $\neg$:

$$\Diamond (R) \quad = \quad \lambda xy \bullet \; x=y \wedge \exists z \, Ryz$$

$$\neg (R) \quad = \quad \lambda xy \bullet \; x=y \wedge \neg\exists z \, Ryz \, .$$

(The latter is the earlier 'test negation'. Note that in fact, $\Diamond R$ is definable as $\neg\neg R$.) This amounts to the repertoire of Propositional Dynamic Logic (cf. Section V below), couched in purely relational terms. By a straightforward induction, all 'regular modal procedures' defined in this way have the required property vis-à-vis bisimulations. Moreover, here is one kind of converse result (disregarding infinitary matters), adapting an observation from the modal folklore:

Proposition. Two states x , y in two finite transition models M_1, M_2 (respectively) can be connected by some bisimulation between M_1, M_2 iff they belong to the domains of the same regular modal procedures.

Proof. The new direction is from right to left. Define a binary relation C between the state domains of M_1, M_2 by setting u C v iff

u, v belong to the domains of the same regular modal procedures.

It suffices to show that C is a bisimulation. So, assume that u C v and consider any R-successor u' of u in M_1 , where R belongs to the atomic repertoire. We have to find some R-successor v' of v matching u' in C . Suppose now that each of the finitely many possible R-successors v' of v in M_2 fails to do the job. That is, there is some regular modal procedure π with either $u' \in$ domain (π) and $v' \notin$ domain (π) (1) , or vice versa (2) . But then, consider the following procedure:

R composed with all $\Diamond\pi$ of case (1) and all $\neg\pi$ of case (2) .

This is a regular modal procedure with u in its domain, and hence so is v . But this will require the existence of some R-successor of v in M_2 distinct from all v' above: a contradiction. (Unions of procedures have dropped out of the definition here, because of the symmetric form of the preservation condition. Compare also the next Subsection on the pure { $\neg$, $\circ$ } repertoire.) □

This Proposition expresses a certain 'maximality' of regular modal operations with respect to Simulability. More sophisticated versions of the result, without the restriction to finite models, may be derived using the preservation theorems for bisimulation invariance found in general Modal Logic (van Benthem 1985; cf. also Section VI).

Analyzing Special Repertoires

Whatever the most general notion of procedural 'logicality' may be, there are at least natural subkinds, such as the earlier continuous operators. Conversely, with special sets of operators from the literature, one can try to determine their specific semantic characteristics. An example is the procedural repertoire $\{ \neg , \circ \}$ of Groenendijk & Stokhof 1991. This seems more special than the regular modal operations, in that there is no explicit union of procedures. Nevertheless, the difference is a more delicate one. Again, this is seen most clearly in a two-level propositional modal logic, having both propositions and procedures in its language:

- The operations $\neg , \circ$ are both regular, and they suffice to embed the propositional component into the procedural one via the test mode ? :

$?(\phi \wedge \psi)$	=	$?(\phi) \circ ?(\psi)$
$?(\neg\phi)$	=	$\neg ?(\phi)$
$?(\langle\pi\rangle\phi)$	=	$\neg\neg(\pi \circ ?(\phi))$

Thus, at least at the level of propositions or their corresponding tests, this repertoire provides all Boolean operations, including union.

- Adding an explicit operation of union $\cup$ to the $\{ \neg , \circ \}$ repertoire results only in addition of outermost unions of $\{\neg , \circ\}$ programs, because of the valid equivalences

$(R \cup S) \circ T$	=	$(R \circ T) \cup (S \circ T)$
$R \circ (S \cup T)$	=	$(R \circ S) \cup (R \circ T)$
$\neg (R \cup S)$	=	$\neg (R) \circ \neg (S)$.

These two observations establish a virtual equivalence between a standard finitary propositional dynamic logic and a relational algebra based on the above two operations.

Next, here is a case of a genuinely different repertoire. In order to exclude unions (i.e., procedural 'choice') more radically, the following semantic characteristic may be used. Consider a 'direct product' of two transition models, whose domain is the Cartesian product of the two state sets, with this stipulation for its atomic repertoire:

$$(x, x')\ R\ (y, y') \quad \text{iff} \quad x R y \ \text{ and } \ x' R y' \ .$$

Arbitrary products may be defined in the same manner. Then, it is easy to show that

- All procedures formed from atomic ones using only the repertoire $\{\Diamond , \circ , \cap\}$ are *invariant for direct products* in the sense of the above equivalence; whereas the latter may fail for $\cup$ and $\neg$.

The reason is that, more generally, all first-order formulas constructed from atoms $\{ Rxy , x=y \}$ using $\wedge , \exists$ are invariant for direct products. A further admissible construction here is the universal quantifier $\forall$: but this does not seem to make much sense procedurally, and may be ruled out, e.g., by insisting on the earlier Continuity.

Without proof, we state a sample outcome of an earlier model-theoretic analysis here, using binary operations for convenience:

- The logical procedural repertoire satisfying Continuity plus Product Invariance consists of all operations definable by the following schema:
  $\lambda xy\bullet\ \exists zu\ (Rzu\wedge\ \exists vw\ (Svw\wedge$ 'conjunction of identities in $\{x, y, z, u, v, w\}$'.

Finally, a semantic analysis of special atomic repertoires may be of interest too. For instance, in the above-mentioned paper, basic actions π are all 'propositional tests' or 'random assignments', satisfying the identity $\pi \circ \pi = \pi$. Moreover, these are both symmetric relations. (These particular properties are not preserved by the procedural repertoire $\{\neg, \circ\}$, but others are.) Algebras over special kinds of binary relation have also been studied in the recent mathematical literature (cf. Németi 1990).

Conclusion. Logical constants in procedural logic are its basic operators of control, whose structure is much richer than that found in standard logic. The art is now to bring out further intuitions of logicality so as to motivate natural finite bases. For the latter task, ordinary model-theoretic analysis is still a useful tool.

IV Varieties of Inference

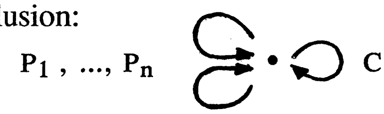
The standard explication of valid inference demands 'automatic transmission of truth':

"in every situation where all premises are true, so is the conclusion".

And with classical propositions, this is a most reasonable basic option (although natural modifications have been proposed since for heuristic purposes in Artificial Intelligence, witness Makinson 1991). If one wants to approximate this standard style in a dynamic setting, then the following seems appropriate. Each procedure may have its 'fixed points', being those states at which it loops (the state already 'satisfies' the goal of the procedure, so to speak) . Thus, we can formulate a first notion of inference:

classical style: in all models, each state which is a fixed point for all premises is also a fixed point for the conclusion:

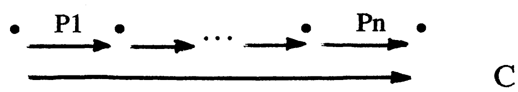
$\text{fix}(P_1) \cap \dots \cap \text{fix}(P_n) \subseteq \text{fix}(C)$



But there is also a genuine

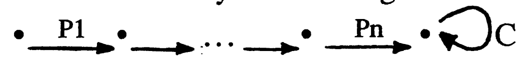
dynamic style: in all models, each transition for the sequential composition of the premises must be admissible for the conclusion:

$P_1 \circ \dots \circ P_n \subseteq C$



And some people prefer a compromise between the two styles:

mixed style "first process all premises consecutively, then test if the conclusion is satisfied by the resulting state":

range (P_1 o ... o P_n) $\subseteq$ fix (C) 

Thus, there appears to be a genuine variety of dynamic styles of inference, reflecting different intuitions and possibly different applications.

Capturing Styles via Structural Rules

One way of defining a basic 'style of inference' is through its general properties, expressed in the usual 'structural rules'. For instance, the above classical style has all the general properties of standard inference:

	/	$C \Rightarrow C$	**Reflexivity**
$X \Rightarrow D \quad Y, D, Z \Rightarrow C$	/	$Y, X, Z \Rightarrow C$	**Cut Rule**
$X, P_1, P_2, Y \Rightarrow C$	/	$X, P_2, P_1, Y \Rightarrow C$	**Permutation**
$X, P, Y, P, Z \Rightarrow C$	/	$X, P, Y, Z \Rightarrow C$	**Contraction**
$X, P, Y, P, Z \Rightarrow C$	/	$X, Y, P, Z \Rightarrow C$	**Contraction**
$X, Y \Rightarrow C$	/	$X, P, Y \Rightarrow C$	**Monotonicity**

By contrast, the dynamic style satisfies only Reflexivity and Cut. Indeed we have several representation results:

Proposition. • {Monotonicity, Contraction, Reflexivity, Cut} completely determine the structural properties of classical inference

• {Reflexivity, Cut} completely determine dynamic inference.

By way of illustration, we prove the second result in more detail. Let us look at models in which the 'propositions' involved in our sequents are interpreted as arbitrary binary relations, while a sequent is 'true' if the above inclusion holds for the composition of its premises in its conclusion. Then there is an obvious notion of semantic consequence $\Delta \models \sigma$ among sequents: truth of all sequents in Δ should imply that of σ.

Proposition. Reflexivity and Cut completely axiomatize valid consequence among dynamic sequents.

Proof. Evidently, Reflexivity and Cut are valid on the above semantic interpretation. Conversely, suppose that some sequent σ cannot be derived from a set Δ using these two principles. Then let all finite sequences of basic syntactic items occurring in sequents be our underlying state set, and define the following map * taking basic items to binary relations:

$$C^* = \{ (X, XY) \mid Y \Rightarrow C \text{ is derivable from } \Delta \text{ using Reflexivity and Cut} \}$$

Then we have that, for sequents $X = X_1, ..., X_n$,

$$X \Rightarrow C \text{ is derivable from } \Delta \quad \text{iff} \quad X_1^* \circ ... \circ X_n^* \subseteq C^* .$$

'If'. By Reflexivity, $X_1 \Rightarrow X_1$, ..., $X_n \Rightarrow X_n$ are all derivable from Δ . Therefore, the pairs $(\langle\rangle, X_1)$, (X_1, X_1X_2), ..., $(X_1...X_{n-1}, X_1...X_{n-1}X_n)$ belong to X_1^*, ..., X_n^*, respectively. So, $(\langle\rangle, X)$ is in the composition of the consecutive premise relations, and hence it belongs to C^* . But then, by definition, $X \Rightarrow C$ is derivable from Δ.

'Only if'. Consider any sequence of transitions according to the successive premises: (X, XY_1) ($Y_1 \Rightarrow X_1$ derivable) , (XY_1, XY_1Y_2) ($Y_2 \Rightarrow X_2$ derivable), etcetera, up to $Y_n \Rightarrow X_n$. Then, n successive applications of Cut to the derivable sequent $X \Rightarrow C$ will derive $Y_1...Y_n \Rightarrow C$, and hence $(X, XY_1...Y_n)$ is in C^*, by the definition of *.

The required counter-example now arises by observing that every sequent in Δ is derivable from it and hence true under the intended relational interpretation, whereas the original nonderivable sequent σ has become false. □

But new religions need not be defined by mere listing which old dogmas they accept or reject. Their point may be precisely that these old dogmas are too crude as they stand. Inferential styles may in fact modify standard structural rules, reflecting a more delicate handling of premises. For instance, the mixed style has none of the above structural properties (counter-examples are easy to produce), but it does satisfy

Left Monotonicity	$X \Rightarrow C$	/ $P, X \Rightarrow C$
Left Cut	$X \Rightarrow D \quad Y, X, D, Z \Rightarrow C$	/ $Y, X, Z \Rightarrow C$

These principles even characterize this style of inference:

Proposition. {Left Monotonicity, Left Cut} completely determine mixed inference.

Proof. It suffices to give the recipe for the main representation involved. This time, the following map $^\#$ from syntactic items to binary relations will work:

$$C^\# = \{ (X, X) \mid X \Rightarrow C \text{ is derivable} \} \cup \{ (X, XC) \mid \text{all sequences } X \} .$$

What may be shown now is the following equivalence:

Fact. $X \Rightarrow C$ is derivable iff it is valid under this interpretation in the mixed style.

'If'. The pairs $(<>, X_1), ..., (X_1...X_{n-1}, X_1...X_{n-1}X_n)$ belong to the successive premise relations. Because of mixed validity then, (X, X) must be in $C^{\#}$, which can only mean that $X \Rightarrow C$ is derivable.

'Only if'. Here is an example, with n=4. Consider the following sequence of 'mixed' transitions for the premises:

(U, UX_1), (UX_1, UX_1X_2), (UX_1X_2, UX_1X_2) (with $UX_1X_2 \Rightarrow X_3$),

$(UX_1X_2, UX_1X_2X_4)$.

Then we have $X_1...X_4 \Rightarrow C$ (by assumption), $UX_1...X_4 \Rightarrow C$ (Left Monotonicity), $UX_1X_2X_4 \Rightarrow C$ (Left Cut, using $UX_1X_2 \Rightarrow X_3$). That is, the final pair of objects $(UX_1X_2X_4, UX_1X_2X_4)$ is in $C^{\#}$. □

Switching Styles

Having different inferential styles available also raises a new issue. How are these styles going to co-exist? In particular, it is natural to ask whether reasoning according to one style may be systematically reduced to reasoning via another. Here a connection emerges with the earlier topic of logical constants. Often, one inferential style can be 'simulated' inside another, through the addition of suitable logical operators. One example is the above classical style. Let us introduce a relational fixed point operator Φ sending relations R to their diagonal $\lambda xy\bullet$ $(Rxy \wedge y=x)$. Then we have the evident equivalence

$P_1, ..., P_n$ imply C classically if and only if

$\Phi(P_1), .., \Phi(P_n)$ imply $\Phi(C)$ dynamically.

In the opposite direction, however, there is no similar formula-wise faithful embedding from the dynamic style into the classical style. The reason is that such an embedding would import classical Monotonicity into the dynamic style (adding translations of dynamic premises would not disturb the classical translation of a dynamic inference). Still there may be more global kinds of embedding that do the trick, translating whole sequents at once (van Benthem 1992 has a survey of various possibilities).

Another form of interplay between structural rules and logical constants arises as follows. One may wonder whether certain structural behaviour can be licensed, not for all propositions, but for special kinds only (cf. Girard 1987). For instance, in the dynamic style, let O be some operator that is to admit of arbitrary monotonic insertion:

$X, Y \Rightarrow C$ / $X, O(P), Y \Rightarrow C$.

It is easy to show that this can be the case if and only if $O(P)$ is a 'test' contained in the diagonal relation. Here is a slightly less trivial result:

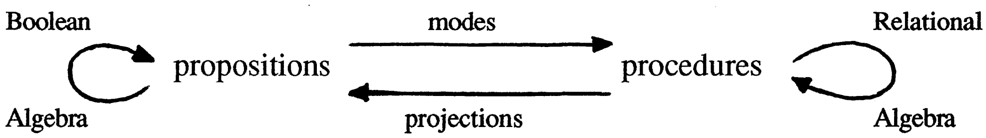
Proposition. An operator O allows unlimited contraction if and only if for all P, O(P) is either empty or it contains the diagonal relation.

Proof. 'Only if'. If O(P) is empty, then compositions including it are empty, and hence the conclusion of Contraction holds vacuously. If O(P) includes the identity relation, then any relation Y dynamically implies both $Y \circ O(P)$ and $O(P) \circ Y$, whence Contraction holds too.
'If'. Suppose that O(P) allows unlimited contraction under the dynamic interpretation of sequents. If O(P) is not empty, then there exist x, y with xO(P)y. Consider any state z. Let R be $\{(y, z)\}$. The sequent $O(P), R, O(P) \Rightarrow O(P) \circ R \circ O(P)$ is dynamically valid, and hence by Contraction, so is $O(P), R \Rightarrow O(P) \circ R \circ O(P)$. Hence (x, z) must be in $O(P) \circ R \circ O(P)$, which can only be the case if zO(P)z. □

Conclusion. There are different natural styles of dynamic inference, exemplifying various clusters of structural rules, that can be determined via representation theorems. What we need now is some 'abstract proof theory' telling us what clusters are especially natural or useful. Moreover, reductions between inferential styles may be investigated, as a means of understanding the systematic connections between the various options that exist for reasoning.

V **Logical Architecture**

<u>Combining Statics and Dynamics</u>
Standard propositions and dynamic propositions both have reasonable motivations. Therefore, there seems little to choose between. Indeed, it is better not to choose at all. Actual inference is a mixture of more dynamic sequential short-term processes and more standard long-term ones, not necessarily working on the same representations. And then, both kinds of system would actually have to be around. In that case, a two-level logical architecture arises:



In this picture, the connections between the two levels have become essential components in their own right. There will be 'modes' taking standard propositions P to procedures with that content, such as 'updating' to make P true, or 'testing' whether P holds already: $\lambda P \bullet \lambda xy \bullet Px \wedge y=x$. Running in the opposite direction, there will be 'projections' assigning to each procedure R a standard proposition recording some essential feature of its action, such as the earlier fixed point operator Φ seeing in which states R is already satisfied, or the taking of a set-theoretic range: $\lambda R \bullet \lambda x \bullet \exists y\, Ryx$, seeing where R might still lead.

Thus, we have acquired several new kinds of operator which may be analyzed from a logical point of view, just as much as the preceding ones. This is in fact a quite general phenomenon. If logical architecture becomes important, in systems having several logical calculi at the same time, then there is also an issue of what may be called 'logical management': what is the structure of the possible connections?

Type-Theoretic Analysis

Despite this diversity, there is also a clear mathematical uniformity. Most of the earlier techniques used in analyzing logical constants make sense for the new categories of operators too, when viewed in a suitable *type-theoretic* perspective. For instance, test or range functions are both 'permutation-invariant' in an obvious extended sense. Moreover, both of them are 'continuous' in that they commute with arbitrary unions of their arguments. All possibilities of this kind can be classified by previous reasoning:

Fact. • The only permutation-invariant continuous modes are those definable by a schema of the form

$\lambda P \bullet \lambda xy \bullet \exists z\, (Pz \wedge$ 'Boolean Condition on $\{=, x, y, z\}$') .

'Test' is one example here: $\lambda P \bullet \lambda xy \bullet \exists z\, (Pz \wedge z=x=y)$.

All other possibilities are simple variations.

• The only permutation-invariant continuous projections are those definable by a schema of the form

$\lambda R \bullet \lambda x \bullet \exists yz\, (Ryz \wedge$ 'Boolean Condition on $\{=, x, y, z\}$')

'Fixed Points' is one example here: $\lambda R \bullet \lambda x \bullet \exists yz\, (Ryz \wedge x=y=z)$, and so are 'Domain' ($\lambda R \bullet \lambda x \bullet \exists yz\, (Ryz \wedge x=y)$) and 'Range' .

The other options are again simple variations.

Another management question concerning projections is whether there exists some map from procedures to statements preserving all relevant logical structure. In particular, one might expect that composition of procedures will reduce to conjunction of the corresponding statements. But here, a negative result arises (which may be proved using the techniques of (van Benthem 1986, chapter 3)):

Fact. • There is only one logical Boolean homomorphism from procedures to propositions, namely the diagonal fixed point map Φ .

The proof is based on a recipe for 'deflating' (logical) Boolean homomorphisms in a type ((a, t), (b, t)) to arbitrary (logical) maps in the type (b, a) , and then counting the mathematical possibilities there.

• The diagonal map Φ does not transform o into $\cap$:

$\Phi(\{(1,2)\} \circ \{(2,1)\}) = \{1\} \neq \emptyset = \Phi(\{(1,2)\}) \cap \Phi(\{(2,1)\})$.

This issue is related to one raised earlier. The two-level system has inference going on both between propositions and between procedures. Say, propositions have standard inference and procedures the earlier dynamic inference. Then, we want to see whether one mechanism can be related systematically to the other. One direction here is easy: standard inference may be simulated within the dynamic style using the test mode ? :

$$P_1, \ldots, P_n \models_{class} C \qquad \text{iff} \qquad ?(P_1), \ldots, ?(P_n) \models_{dyn} ?(C) .$$

In the opposite direction, say, the fix point operator 'fix' will not work, for reasons explained earlier: a similar reduction would make dynamic reasoning monotonic.

Static Tracing of Dynamic Procedures

But we can also analyze the situation somewhat differently, using a well-known concept from computer science. Let us trace a procedure through propositions describing successive images of sets of states under its action. Define 'strongest postconditions' as follows:

$$SP(A, R) = R[A] .$$

Likewise, there are 'weakest preconditions':

$$WP(R, A) = R^{-1}[A] .$$

Then, we can reduce dynamic validity as follows:

Proposition. $R_1, \ldots, R_n \models_{dyn} S$ if and only if

$SP(A, R_1 \circ \ldots \circ R_n) \models_{class} SP(A, S)$ for arbitrary sets A .

Proof. The 'only if' direction follows from the definition of $\models_{dyn}$ and the monotonicity of SP in its right-hand argument. The 'if' direction follows by considering any pair (x, y) in $R_1 \circ \ldots \circ R_n$, and then applying the second condition to the set $A = \{x\}$. □

Now, it becomes of interest to have a good way of computing weakest preconditions and strongest postconditions. Here are some inductive clauses:

$$SP (A, R \circ S) = SP (SP (A, R), S)$$
$$WP (R \circ S, A) = WP (R, WP (S, A))$$
$$SP (A, R \cup S) = SP (A, R) \vee SP (A, S)$$
$$WP (R \cup S, A) = WP (R, A) \vee WP (S, A)$$
$$SP (A, R^{\vee}) = WP (R, A)$$
$$WP (R^{\vee}, A) = SP (A, R)$$

This calculation may be extended to cover the earlier regular modal operations of 'domain' $\Diamond$ and 'counter-domain' $\neg\Diamond$, with clauses such as

$$SP (A, \Diamond(R)) = A \wedge WP (R, T)$$
$$SP (A, \neg\Diamond(R)) = A \wedge \neg WP (R, T)$$
$$WP (\Diamond(R), A) = A \wedge WP (R, T)$$
$$WP (\neg\Diamond(R), A) = A \wedge \neg WP (R, T)$$

There are no obvious inductive clauses, however, for intersection or complement of programs (let alone, homomorphic behaviour). For instance, we do not have

' $SP (A, R \cap S) = SP (A, R) \wedge SP (A, S)$ '.

This situation may again be understood by the earlier type-theoretic analysis. SP and WP are transformations from the type of relations to that of functions from statements to statements: i.e., they live in the intensional type

$((s, (s, t)), ((s, t), (s, t)))$,

having definitions, respectively,

$\lambda R \bullet \lambda A \bullet \lambda x \bullet \exists y (Ay \wedge Ryx)$ and $\lambda R \bullet \lambda A \bullet \lambda x \bullet \exists y (Ay \wedge Rxy)$.

In principle, the type $((s, t), (s, t))$ has more room than (s, t) to acommodate relations (indeed, SP is a bijection between relations and *continuous* maps from sets to sets). Again, we can analyze this larger class of transformations for its most interesting logical inhabitants, and then we find an explanation for the above poverty:

Proposition.
- The only logical homomorphisms in the type of SP are those defined by a schema of the form $\lambda R \bullet \lambda P \bullet \lambda x \bullet R (F (P, x))$, with F a logical map from sets and individuals to pairs of individuals.
- The latter are all 'products' of two logical maps from sets P and individuals x to individuals, of which there are essentially just two: 'right projection' to x and 'definite description' $\iota x \bullet P$ (in case P is a singleton).

Proof. This may be shown again by the earlier recipe of 'homomorphic deflation', now from type $((s, (s, t)), ((s, t), (s, t))) = ((s \bullet s, t), ((s, t) \bullet s, t))$ to $((s, t) \bullet s, s \bullet s)$, followed by an analysis of invariant candidates. □

Conclusion. It is possible, and indeed desirable, to have logical architectures combining both dynamic and standard inferential styles. This is also a key idea in dynamic logic as found in computer science. What we add here is 'management' as a separate concern: one wants to see to which extent independent reasoning inside the various levels can be related, and for that purpose, the connections between the two levels become independent objects of logical study in their own right. Logical uniformity in exploring this wider domain is guaranteed by taking a suitable type-theoretic perspective.

VI Further Informational Structure: Dynamic Modal Logic

Cognitive Procedures over Information Patterns

The notions and issues introduced so far are purely procedural, and have nothing to do with information per se. A modest basic step introducing more informational structure consists in endowing state spaces (now thought of as patterns of information states) with the inclusion relation also found in models for intuitionistic or relevant logics:

$$(S, \subseteq, \{R_p \mid p \in P\}) .$$

Then, new notions emerge in all components of the earlier architecture, such as

- a new propositional operator
  $\Diamond_{up}(P) \;=\; \lambda x\bullet\ \exists y\,(x \subseteq y \wedge Py)$ 'upward modality'
- a new procedural operator
  $\text{forw}(R) \;=\; \lambda xy\bullet\ (Rxy \wedge x \subseteq y)$ 'forward part'
- a new projection
  $\text{fut}(R) \;=\; \lambda x\bullet\ \exists y(x \subseteq y \wedge \exists z\, Ryz)$ 'future domain'.

In particular, new *modes* may be defined creating dynamic procedures out of standard propositions. Some prominent examples are as follows:

$\lambda P\bullet\ \lambda xy\bullet\ x \subseteq y \wedge Py$ 'loose updating'

$\lambda P\bullet\ \lambda xy\bullet\ x \subseteq y \wedge Py \wedge \neg\exists z\,(x \subseteq z \subset y \wedge Pz)$ 'strict updating'

Moreover, going through the information pattern in the opposite direction, there are two obvious counterparts for processes of 'cognitive retreat':

$\lambda P\bullet\ \lambda xy\bullet\ y \subseteq x \wedge \neg Py$ 'loose downdating'

$\lambda P\bullet\ \lambda xy\bullet\ y \subseteq x \wedge \neg Py \wedge \neg\exists z\,(y \subset z \subseteq x \wedge \neg Pz)$. 'strict downdating'

Note that downdating is not a converse of updating. The proper duality is this:

$$\text{downdate}(P, \subseteq) \;=\; \text{update}(\neg P, \supseteq) .$$

This formalism is rich enough to perform all cognitive tasks covered in the well-known system of Gärdenfors 1988. In particular, procedures of 'revision' may be described by combination of updates and downdates. Here is another example. Possibly counterfactual conditional statements $A \rightarrow B$ are often explained via the Ramsey Test: "Assume the antecedent. If this leads to inconsistency, make a minimal adjustment in the background theory so as to restore consistency. Then see if the result implies the consequent" . In our models, this procedure becomes: "Downdate strictly with respect to $\neg\Diamond A$. Then update strictly with respect to A . Finally, check if B holds."

Exploring Dynamic Modal Logic

One useful general calculus in this case will be some system of dynamic logic serving as a common generalization of standard modal logic and the earlier relational algebra. That is, the language will have standard propositional and modal operators

$\{ \neg, \wedge, \vee, \Diamond_{up}, \Diamond_{down} \}$

as well as the usual relational repertoire

$\{ -, \cap, \cup, \circ, ^{\vee}, \Delta \}$

plus suitable modes

{test, loose and strict updates as well as downdates}

and projections

{fixed point, domain, range}.

This also contains the usual weakest preconditions $<\pi> P$ via $\mathrm{dom}(\pi \circ ?(P))$.

There are various connections between these operators, witness valid identities like

$\subseteq$	=	upd (T)	Δ	=	? (T)
upd (P)	=	$\subseteq \circ ?(P)$	range (π)	=	dom ($\pi^{\vee}$)
$\Diamond_{up}$ (P)	=	dom (upd (P))	fix (π)	=	dom ($\Delta \cap \pi$)

But we are not after a least redundant version of this modal system $S4^2$, which may be viewed as a dynamic version of the standard modal logic S4.

The universal validities of this logic form a general theory of cognitive processes. Notably, one can study various combinations of updating and downdating, noting that:

- update (P) o update (P) = update (P)
- and the same holds for strict updates, and for downdates
- an update followed by a downdate need not be idempotent, witness a situation like $\bullet \neg P$ —— $\bullet \neg P$ —— $\bullet P$

The system also handles various earlier notions of inference. E.g., the dynamic variant $P_1, ..., P_n \models_{dyn} C$ amounts to validity of the implication $<P_1 \circ ... \circ P_n> q \rightarrow <C>q$. And more complex relations between cognitive processes can be formulated too, such as $[\pi_1]<\pi_2> \phi$ (process π_1 'enables' process π_2 to achieve result ϕ ') .

There are some obvious technical questions concerning this dynamic version of the standard modal logic S4 . By general reasoning, its set of universal validities must be recursively enumerable (due to the embedding into first-order logic presented below). De Rijke 1991 presents a complete axiomatization using a 'difference operator' D (stating "truth in at least one different state"), whose corresponding 'difference relation' $\neq$ is definable in this system as $-\Delta$. Then, characteristic deductive principles are definitory axioms for the main modes, such as

$$(q \wedge \neg Dq) \rightarrow (\langle \text{strict-upd}(P) \rangle A \leftrightarrow \Diamond_{up} (A \wedge P \wedge \Box_{down} (\Diamond_{down} q \rightarrow \neg P))) .$$

Moreover, there is the issue whether the system is *decidable*. Modal S4 has the latter property, full relational algebra over arbitrary relations does not: but what about this intermediate case, which handles only special updating, downdating and test relations?

Without going into deductive detail, one can analyze the characteristic properties of some modes and projections in direct semantic terms (assuming that the operators on propositions and procedures already have their standard interpretation):

Proposition. 'Test' is the only permutation-invariant continuous operator satisfying

$$?\,(P) \quad \leq \quad \Delta \qquad *$$

$$?\,(\neg P) \quad = \quad \Delta \cap -\,?\,(P) \qquad **$$

Proof. By Continuity, it suffices to determine the behaviour of 'test' on singleton arguments $\{x\}$. And since test values are subrelations of the diagonal, by * , it suffices to specify the individual states in the image. By permutation invariance, there are only four basic options here: $\{x\}$ itself (1) , $\{y \mid y \neq x\}$ (2) , the unit set (3) and the empty set (4) . Moreover, the choice will be made uniformly for all states x , again by permutation invariance. Now, outcome (4) would result in any test relation being empty: which contradicts ** . Outcomes (3) and (2) may also be ruled out, by observing that they would allow for two distinct sets $\{x\}$, $\{y\}$ to have overlapping test values, whence some value $?\,(\neg P)$ would not be disjoint from $?\,(P)$: another contradiction with ** . Thus, only the standard interpretation (1) remains. □

A similar kind of argument characterizes a key projection:

Proposition. 'Domain' is the only permutation-invariant continuous operator satisfying the principles

$$\text{dom}\,(\,?\,(P)\,) \quad = \quad P \qquad \#$$

$$\text{dom}\,(0) \quad = \quad 0 \qquad \#\#$$

$$\text{dom}\,(\,R \circ S\,) \quad = \quad \text{dom}\,(\,R \circ ?\,(\text{dom}\,(S))\,) \qquad \#\#\#$$

There is still an 'asymmetry' in the design of the modal system $S4^2$, whose static component is too weak as compared with the dynamic one. This may be seen by considering an earlier reduction from dynamic behaviour of regular procedures to static propositions, via 'weakest preconditions' such as:

$$WP\ (\ ?\ (P)\ ,\ A\ ) \quad = \quad P \wedge A$$
$$WP\ (\ upd\ (P)\ ,\ A\ ) \quad = \quad \Diamond_{up}\ (P \wedge A)$$
$$WP\ (\ downd\ (P)\ ,\ A\ ) \quad = \quad \Diamond_{down}\ (\neg P \wedge A)$$

In order to describe the strict variants, a more complex modal language is needed over information patterns, employing two well-known 'temporal' operators:

$$WP\ (\ strict\ upd\ (P)\ ,\ A\ ) \quad = \quad UNTIL\ (P \wedge A\ ,\ \neg P)$$
$$WP\ (\ strict\ downd\ (P)\ ,\ A\ ) \quad = \quad SINCE\ (\neg P \wedge A\ ,\ P)\ .$$

Clearly then, $S4^2$ is only at the bottom end of a ladder of dynamic modal logics. This observation brings us to an even broader formalism over information models.

Dynamic Modal Logics as Fragments of First-Order Logic

As in standard modal logic, there is a straightforward *translation* from the new dynamic propositions and procedures to unary and binary formulas in a standard first-order predicate logic over partial orders of states with unary predicates:

$(p)^*$	Px
$(\neg\phi)^*$	$\neg\ (\phi)^*$
$(\phi\wedge\psi)^*$	$(\phi)^* \wedge (\psi)^*$
$(\Diamond_{up}\phi)^*$	$\exists y\ (x \subseteq y \wedge [y/x](\phi)^*)$
$(\Diamond_{down}\phi)^*$	$\exists y\ (y \subseteq x \wedge [y/x](\phi)^*)$
$(?\phi)^\#$	$x=y \wedge (\phi)^*$
$(upd\ (\phi)\ )^\#$	$x \subseteq y \wedge [y/x](\phi)^*$
$(strict\ upd\ (\phi)\ )^\#$	$x \subseteq y \wedge [y/x](\phi)^* \wedge \neg\exists z\ (x \subseteq z \subset y \wedge [z/x](\phi)^*)$
$(downd\ (\phi)\ )^\#$	$y \subseteq x \wedge [y/x](\phi)^*$
$(strict\ downd\ (\phi)\ )^\#$	$y \subseteq x \wedge \neg[y/x](\phi)^* \wedge \neg\exists z\ (y \subset z \subseteq x \wedge \neg[z/x](\phi)^*)$
$(\Delta)^\#$	$x=y$
$(-\pi)^\#$	$\neg\ (\pi)^\#$
$(\pi_1 \cap \pi_2)^\#$	$(\pi_1)^\# \wedge (\pi_2)^\#$
$(\pi_1 \circ \pi_2)^\#$	$\exists z\ (\ [z/y](\pi_1)^\# \wedge [z/x](\pi_2)^\#\ )$
$(\pi^{\vee})^\#$	$[y/x,\ x/y]\ (\pi)^\#$
$(fix\ (\pi))^*$	$[x/y]\ (\pi)^\#$
$(dom\ (\pi))^*$	$\exists y\ (\pi)^\#$

As the first-order theory of partial orders with monadic predicates (an elementary class of models) is recursively enumerable, so is our dynamic logic $S4^2$. And the same holds for any first-order reducible strengthening thereof.

This brings us to a general question of logical design. Modal logics, whether 'static' or 'dynamic', may be viewed as fragments of a full first-order logic over information patterns. And the question is what kinds of fragment are natural for present purposes. Now, several earlier observations may be brought to bear. First, the above translation may be seen to involve essentially only *three* variables over states in any formula. Thus, one view of the matter would be to have a full three-variable fragment, considering all unary and binary first-order formulas $\phi(x)$, $\pi(x, y)$ constructed using only the three variables $\{x, y, z\}$. This establishes a certain 'harmony' between the minimal procedural repertoire found in Relational Algebra and the three-variable {Since, Until} language which has been so prominent in temporal logic. (Conversely, this harmony also amounts to a kind of 'functional completeness' for the dynamic part, which should be strong enough to achieve everything expressible in the static part.) Basically, what we are studying here is the behaviour of cognitive procedures whose action can be described using configurations of no more than three states at any one time. That is, we can specify goal states, while imposing conditions on intermediate states encountered en route.

This three-variable fragment also has a purely semantic characterization (van Benthem 1991), in terms of the following notion. A 'k-partial isomorphism' between two first-order models is a family of partial isomorphisms of size at most k satisfying the usual Back and Forth properties for addition of new objects on both sides up to length k. Moreover, restrictions of partial isomorphisms in the family are to remain inside it. The relevant result is this

Theorem. A first-order formula having its free variables among $\{x_1, ..., x_k\}$ can be written using these variables only (free or bound) if and only if it is invariant in passing from one model and assignment to another model related to it by some k-partial isomorphism PI and using a PI-matching sequence of objects for the new assignment.

Specialization to the case k=3 then describes one very natural dynamic modal logic. But there is another, related perspective too. Upon closer inspection, translations of the above {Since, Until} language turn out to involve only part of the full three-variable first-order formalism. This point is even clearer with the basic modal language, which describes a special fragment of the two-variable first-order language over its models, having all quantifiers restricted to relational successors and predecessors, with only unary atoms. There is an independent semantic characterization of the latter fragment too (cf. van Benthem 1985), using an earlier semantic notion:

Theorem. A unary first-order formula $\phi(x)$ is equivalent to the translation of a basic modal formula if and only if it is invariant for standard bisimulation.

In the earlier terms, one now restricts attention to 'process simulation' via partial isomorphisms where the next choice in the Back and Forth moves is restricted to successors or predecessors of the previous selection. Moreover, comparisons between matching states concern only atomic propositions.

Inspection of the proof for this preservation theorem and its predecessor shows that there is a recurring pattern here. The crucial step in all cases runs as follows:

> Finite sequences of objects up to some fixed length in two suitably saturated models (e.g., finite ones, as in an earlier argument) are 'connected' if they satisfy the same formulas in the restricted fragment under consideration.
> Then it is shown how this connection is in fact an appropriate relation of 'bisimulation' between the two models.

This is the precise spot where expressive power of the language and semantic strength of bisimulation meet, suggesting a general 'recipe' for generating preservation results. Here are some illustrations, whose purpose is mainly to give an impression of the general method.

At level k=2 , there are basically two options for the above connection. The first has arbitrary Back and Forth moves from single state matches to matched state pairs, and the appropriate formalism needs a strong projection operator to ensure this:

$\lambda x \bullet \ \exists y \ \ \pi(x, y)$ Domain

If one also insists that matched pairs generate two matched individuals, then two modes are needed:

$\lambda xy \bullet \ \phi(x)$ $\lambda xy \bullet \ \phi(y)$ Raising

Finally, in order to ensure that matched sequences are truly (partial) isomorphisms, the formalism needs all Boolean operations, as well as relational conversion and identification of arguments. (One might economize on this repertoire, though, by weakening the requirements on 'partial isomorphism'.)

The second main option is to have Back and Forth clauses demanding extension by new individual matchings only, looking at $\subseteq$-successors and $\subseteq$-predecessors. Then essentially, just two weakened versions of the domain projection are needed:

$\lambda x \bullet \ \exists y \ (x \subseteq y \wedge \phi(y))$ $\lambda x \bullet \ \exists y \ (y \subseteq x \wedge \phi(y))$ Modality

At level $k=3$, the proper one for the above dynamic logic, similar options emerge. One notion congenial to the intended procedures might be called 'Path Simulation' :

There is a restriction-closed matching between individual states and pairs of states satisfying the following Back and Forth conditions:

- for matched states x, y, selecting a $\subseteq$-successor or $\subseteq$-predecessor z in either model produces an admissible matching xz, yu or vice versa with some $\subseteq$-corresponding state u on the opposite side.
- for matched pairs xy, zu, selecting a state v in between (along $\subseteq$) leads to a $\subseteq$-corresponding selection w on the opposite side, generating admissible matchings xv, zw and vy, wu (or vice versa).

Proposition. The complete first-order formalism for invariance under path simulation contains atoms $\{ Px, x \subseteq y \}$, all Boolean operations, restricted modal existential quantifiers $\exists y (x \subseteq y \wedge \pi(x, y))$, $\exists y (y \subseteq x \wedge \pi(x, y))$ (that is, domains of the earlier 'forward' and 'backward' parts of π), as well as a new binary modal quantifier $\exists z (x \subseteq z \subseteq y \wedge \pi_1(x, z) \wedge \pi_2(z, y))$ ('betweenness') .

Again this outcome may be varied, with weaker versions of simulation capturing restricted 'unary' quantifiers such as $\exists y (x \subseteq y \wedge \phi(y))$ and $\exists z (x \subseteq z \subseteq y \wedge \phi(z))$.

Conclusion. Dynamic modal logics are a joint generalization of standard intuitionistic or other information-oriented logics and relational algebra, encompassing most recent 'cognitive logics' for information processing. There is no single preferred such system, but options for design may be laid out in terms of invariance for 'bisimulation', used as a flexible model-theoretic technique. These logics provide a simple natural 'completion' of constructivist thinking, which can still be studied by standard modal techniques.

VII Towards More Realistic Systems

All information modelings considered so far have been concerned with transitions in the internal cognitive space of one agent. A general logical architecture for cognition will have to be extended in at least the following ways.

First, more sensitive notions of 'process' are to be introduced to get at finer dynamic phenomena. Examples are the 'failure paths' of Segerberg 1991, the 'full trace models' of Vermeulen 1989 or the 'process algebra' of Milner 1980, Bergstra & Klop 1984. These approaches may be developed in the general logical style advocated here.

Another finer perspective concerns computational complexity. For instance, the above 'modes' are ways of testing or realizing standard propositions that may still be of vastly different complexities. What we want is some understanding of 'minimal cost' for modes with respect to different standard propositions, allowing us to compare them. One relevant viewpoint here is that of 'semantic automata' (van Benthem 1986), which may be viewed as procedural mechanisms checking truth conditions of standard constructions, in particular, various quantifiers. (First-order quantifiers are of finite-state complexity, while computing higher-order ones may involve push-down storage. Moreover, even among first-order quantifiers, e.g., "some" is cheaper than "one".) One could devise similar 'graph automata' operating on the above inclusion patterns of information states. Basic moves are steps along $\subseteq$ or atomic tests $?(p)$, and then, one would want to make comparisons as to complexity of search for various tasks.

Then, the physical world environment is to brought into the picture if 'real correctness' of cognitive procedures is to be formulated. There are various proposals to this effect in the recent literature, witness Kamp 1979, 1984 about the triangle 'language – representation – real world' or Barwise 1991 on 'information links' between various abstract and concrete informational systems. The least that should be done to this effect in our case is the introduction of a real world structure in addition to the pattern of information states, with suitable links between these. For instance, one might work with some distinguished 'actual world' in ranges of possible worlds, with some suitable relation of 'instantiation' linking possible worlds to information states. This will enable us to formulate real correctness of cognitive procedures, for instance, by letting the real world be among the possible instantiations of the states along some trajectory in a cognitive state space.

Also, real cognition usually involves the interplay of various agents. This brings in the interplay between different cognitive spaces (and a real world environment). On the modal strategy advocated here, one would need distributed environments as in Halpern & Moses 1989, again with an appropriate general perspective on logical architecture and management. Interestingly, earlier game-theoretical approaches to logic and cognition, like that of Hintikka 1973, had this multi-agent perspective all along.

Finally, cognitive activity is certainly not restricted to the standard business of 'interpretation' and 'inference'. It also involves planning, learning, guessing, querying or searching vis-á-vis patterns of cognitive states in a real world environment. These activities too, with their salient structural properties, will have to be brought eventually within the compass of a genuine logic of information flow.

VIII References

Barwise, J. 1987 'Noun Phrases, Generalized Quantifiers and Anaphora' in P. Gärdenfors, ed., *Generalized Quantifiers. Logical and Linguistic Approaches*, Reidel, Dordrecht, 1-29.

Barwise, J. 1991 'Information Links' Department of Philosophy, Indiana University, Bloomington.

Belnap, N. 1977 'A Useful Four-Valued Logic' in M. Dunn & G. Epstein, eds., *Modern Uses of Multiple-Valued Logics*, Reidel, Dordrecht, 8-37.

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

Benthem, J. van 1986 *Essays in Logical Semantics* Reidel, Dordrecht, (Studies in Linguistics and Philosophy 29).

Benthem, J. van 1989 'Modal Logic and Relational Algebra' Institute for Language, Logic and Information, University of Amsterdam. (To appear in *Proceedings Malc'ev Conference*, Institute of Mathematics, USSR Academy of Sciences, Novosibirsk.)

Benthem, J. van 1991 *Language in Action. Categories, Lambdas and Dynamic Logic* North-Holland, Amsterdam, (Studies in Logic 130).

Benthem, J. van 1992 'The Landscape of Deduction' to appear in K. Dosen & P. Schröder-Heister, eds., *Substructural Logics*, Oxford University Press.

Bergstra, J. & J-W Klop 1984 'Process Algebra for Synchronous Communication' *Information and Control* 60, 109-137.

Deemter, K. van 1991 *On the Composition of Meaning* Dissertation, Institute for Language, Logic and Information, University of Amsterdam.

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

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

Groenendijk, J. & M. Stokhof 1991 'Dynamic Predicate Logic' *Linguistics and Philosophy* 14, 39-100.

Halpern, J. & Y. Moses 1989 'Knowledge and Common Knowledge in a Distributed Environment', revised and expanded version, IBM Research Report RJ 4421, (to appear in *Journal of the ACM*).

Harel, D. 1984 'Dynamic Logic' in D. Gabbay & F. Guenthner, eds., *Handbook of Philosophical Logic*, vol. II, Reidel, Dordrecht, 497-604.

Harman, G. 1985 *Change in View: Principles of Reasoning* The MIT Press / Bradford Books, Cambridge (Mass.).

Hintikka, J. 1973 *Logic, Language Games and Information* Clarendon Press, Oxford.

Hintikka, J. & J. Kulas 1983 *The Game of Language* Reidel, Dordrecht.

Kamp, H. 1979 'Instants, Events and Temporal Discourse' in R. Bäuerle et al., eds., *Semantics from Different Points of View*, Springer Verlag, Berlin, 376-417.

Kamp, H. 1984 'A Theory of Truth and Semantic Representation' in J. Groenendijk et al., eds., *Truth, Interpretation and Information*, Foris, Dordrecht, 1-41.

Makinson, D. 1991 'General Non-Monotonic Logic' to appear in D. Gabbay et al., eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press.

Milner, R. 1980 *A Calculus of Communicating Systems* Springer Verlag, Berlin.

Moortgat, M. 1988 *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*, Foris, Dordrecht.

Moschovakis, Y. 1991 'Sense and Reference as Algorithm and Value' Department of Mathematics, University of California, Los Angeles.

Németi, I. 1990 'Algebraizations of Quantifier Logics. An Introductory Overview', Institute of Mathematics, Hungarian Academy of Sciences, Budapest.

de Rijke, M. 1991 'A Modal Logic for Updating and Revising' Institute for Logic, Language and Computation, University of Amsterdam.

Roorda, D. 1991 *Resource Logics: Proof-Theoretical Investigations* Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

Saraswat, V., M. Rinard & P. Panangaden 1990 'Semantic Foundations of Concurrent Constraint Programming', Report SSL-90-86, Xerox Parc Research Center, Palo Alto.

Segerberg, K. 1991 — 'Logics of Action'
Abstracts 9th International Congress on Logic, Methodology and Philosophy of Science, Uppsala.

Spohn, W. 1988 — 'Ordinal Conditional Functions: A Dynamic Theory of Epistemic States', in W. L. Harper et al., eds., *Causation in Decision, Belief Change and Statistics* II, Kluwer, Dordrecht, 105-134.

Vakarelov, D. 1991 — 'Arrow Logics'
Abstracts 9th International Congress on Logic, Methodology and Philosophy of Science, Uppsala.

Veltman, F. 1991 — 'Defaults in Update Semantics'
Institute for Logic, Language and Computation, University of Amsterdam.

Venema, Y. 1991 — *Many-Dimensional Modal Logics*
Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

Vermeulen, C. 1989 — 'A Dynamic Analysis of Reasoning'
Master's Thesis, Philosophical Institute, University of Utrecht.

Westerståhl, D. 1984 — 'Determiners and Context Sets'
in J. van Benthem & A. ter Meulen, eds., *Generalized Quantifiers in Natural Language*, Foris, Dordrecht, 45-71.

The ITLI Prepublication Series

1986 86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem Logical Syntax
1987 87-01 Jeroen Groenendijk, Martin Stokhof Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem Polyadic quantifiers
87-05 Víctor Sánchez Valencia Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem Categorial Grammar and Type Theory
87-08 Renate Bartsch The Construction of Properties under Perspectives
87-09 Herman Hendriks Type Change in Semantics: The Scope of Quantification and Coordination
1988 LP-88-01 Michiel van Lambalgen *Logic, Semantics and Philosophy of Language:* Algorithmic Information Theory
LP-88-02 Yde Venema Expressiveness and Completeness of an Interval Tense Logic
LP-88-03 Year Report 1987
LP-88-04 Reinhard Muskens Going partial in Montague Grammar
LP-88-05 Johan van Benthem Logical Constants across Varying Types
LP-88-06 Johan van Benthem Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe A Blissymbolics Translation Program
ML-88-01 Jaap van Oosten *Mathematical Logic and Foundations:* Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen The Arithmetical Fragment of Martin Löf's Type Theories with weak Σ-elimination
ML-88-03 Dick de Jongh, Frank Veltman Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra Remarks on Intuitionism and the Philosophy of Mathematics
CT-88-01 Ming Li, Paul M.B.Vitanyi *Computation and Complexity Theory:* Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas
Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas Towards implementing RL
X-88-01 Marc Jumelet *Other prepublications:* On Solovay's Completeness Theorem
1989 LP-89-01 Johan van Benthem *Logic, Semantics and Philosophy of Language:* The Fine-Structure of Categorial Semantics
LP-89-02 Jeroen Groenendijk, Martin Stokhof Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03 Yde Venema Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04 Johan van Benthem Language in Action
LP-89-05 Johan van Benthem Modal Logic as a Theory of Information
LP-89-06 Andreja Prijatelj Intensional Lambek Calculi: Theory and Application
LP-89-07 Heinrich Wansing The Adequacy Problem for Sequential Propositional Logic
LP-89-08 Víctor Sánchez Valencia Peirce's Propositional Logic: From Algebra to Graphs
LP-89-09 Zhisheng Huang Dependency of Belief in Distributed Systems
ML-89-01 Dick de Jongh, Albert Visser *Mathematical Logic and Foundations:* Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03 Dick de Jongh, Franco Montagna Rosser Orderings and Free Variables
ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna On the Proof of Solovay's Theorem
ML-89-05 Rineke Verbrugge Σ-completeness and Bounded Arithmetic
ML-89-06 Michiel van Lambalgen The Axiomatization of Randomness
ML-89-07 Dirk Roorda Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08 Dirk Roorda Investigations into Classical Linear Logic
ML-89-09 Alessandra Carbone Provable Fixed points in $I\Delta_0+\Omega_1$
CT-89-01 Michiel H.M. Smid *Computation and Complexity Theory:* Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas On Space Efficient Simulations
CT-89-04 Harry Buhrman, Leen Torenvliet A Comparison of Reductions on Nondeterministic Space
CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree A Parallel Functional Implementation of Range Queries
CT-89-06 H.W. Lenstra, Jr. Finding Isomorphisms between Finite Fields
CT-89-07 Ming Li, Paul M.B. Vitanyi A Theory of Learning Simple Concepts under Simple Distributions and
Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet On Adaptive Resource Bounded Computations
CT-89-10 Sieger van Denneheuvel The Rule Language RL/1
CT-89-11 Zhisheng Huang, Sieger van Denneheuvel, Peter van Emde Boas Towards Functional Classification of Recursive Query Processing
X-89-01 Marianne Kalsbeek *Other Prepublications:* An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers New Foundations: a Survey of Quine's Set Theory
X-89-03 A.S. Troelstra Index of the Heyting Nachlass
X-89-04 Jeroen Groenendijk, Martin Stokhof Dynamic Montague Grammar, a first sketch
X-89-05 Maarten de Rijke The Modal Theory of Inequality
X-89-06 Peter van Emde Boas Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project
1990 *Logic, Semantics and Philosophy of Language*
LP-90-01 Jaap van der Does A Generalized Quantifier Logic for Naked Infinitives
LP-90-02 Jeroen Groenendijk, Martin Stokhof Dynamic Montague Grammar
LP-90-03 Renate Bartsch Concept Formation and Concept Composition
LP-90-04 Aarne Ranta Intuitionistic Categorial Grammar
LP-90-05 Patrick Blackburn Nominal Tense Logic
LP-90-06 Gennaro Chierchia The Variablity of Impersonal Subjects
LP-90-07 Gennaro Chierchia Anaphora and Dynamic Logic
LP-90-08 Herman Hendriks Flexible Montague Grammar
LP-90-09 Paul Dekker The Scope of Negation in Discourse, towards a flexible dynamic Montague grammar
LP-90-10 Theo M.V. Janssen Models for Discourse Markers
LP-90-11 Johan van Benthem General Dynamics
LP-90-12 Serge Lapierre A Functional Partial Semantics for Intensional Logic
LP-90-13 Zhisheng Huang Logics for Belief Dependence

The ITLI Prepublication Series

LP-90-14 Jeroen Groenendijk, Martin Stokhof Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast Awareness, Negation and Logical Omniscience
LP-90-17 Paul Dekker Existential Disclosure, Implicit Arguments in Dynamic Semantics
ML-90-01 Harold Schellinx *Mathematical Logic and Foundations* Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema Relational Games
ML-90-04 Maarten de Rijke Unary Interpretability Logic
ML-90-05 Domenico Zambella Sequences with Simple Initial Segments
ML-90-06 Jaap van Oosten Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen Randomness in Set Theory
ML-90-11 Paul C. Gilmore The Consistency of an Extended NaDSet
CT-90-01 John Tromp, Peter van Emde Boas *Computation and Complexity Theory* Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Edith Spaan, Leen Torenvliet Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets Greatest Fixed Points of Logic Programs
CT-90-08 Fred de Geus, Ernest Rotterdam, Sieger van Denneheuvel, Peter van Emde Boas Physiological Modelling using RL
CT-90-09 Roel de Vrijer Unique Normal Forms for Combinatory Logic with Parallel Conditional, a case study in conditional rewriting
X-90-01 A.S. Troelstra *Other Prepublications* Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04 Annual Report 1989
X-90-05 Valentin Shehtman Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone Provable Fixed points in $I\Delta_0+\Omega_1$, revised version
X-90-12 Maarten de Rijke Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra Lectures on Linear Logic
1991 LP-91-01 Wiebe van der Hoek, Maarten de Rijke *Logic, Semantics and Philosophy of Langauge* Generalized Quantifiers and Modal Logic
LP-91-02 Frank Veltman Defaults in Update Semantics
LP-91-03 Willem Groeneveld Dynamic Semantics and Circular Propositions
LP-91-04 Makoto Kanazawa The Lambek Calculus enriched with additional Connectives
LP-91-05 Zhisheng Huang, Peter van Emde Boas The Schoenmakers Paradox: Its Solution in a Belief Dependence Framework
LP-91-06 Zhisheng Huang, Peter van Emde Boas Belief Dependence, Revision and Persistence
LP-91-07 Henk Verkuyl, Jaap van der Does The Semantics of Plural Noun Phrases
LP-91-08 Víctor Sánchez Valencia Categorial Grammar and Natural Reasoning
LP-91-09 Arthur Nieuwendijk Semantics and Comparative Logic
LP-91-10 Johan van Benthem Logic and the Flow of Information
ML-91-01 Yde Venema *Mathematical Logic and Foundations* Cylindric Modal Logic
ML-91-02 Alessandro Berarducci, Rineke Verbrugge On the Metamathematics of Weak Theories
ML-91-03 Domenico Zambella On the Proofs of Arithmetical Completeness for Interpretability Logic
ML-91-04 Raymond Hoofman, Harold Schellinx Collapsing Graph Models by Preorders
ML-91-05 A.S. Troelstra History of Constructivism in the Twentieth Century
ML-91-06 Inge Bethke Finite Type Structures within Combinatory Algebras
ML-91-07 Yde Venema Modal Derivation Rules
ML-91-08 Inge Bethke Going Stable in Graph Models
ML-91-09 V.Yu. Shavrukov A Note on the Diagonalizable Algebras of PA and ZF
ML-91-10 Maarten de Rijke, Yde Venema Sahlqvist's Theorem for Boolean Algebras with Operators
ML-91-11 Rineke Verbrugge Feasible Interpretability
ML-91-12 Johan van Benthem Modal Frame Classes, revisited
CT-91-01 Ming Li, Paul M.B. Vitányi *Computation and Complexity Theory* Kolmogorov Complexity Arguments in Combinatorics
CT-91-02 Ming Li, John Tromp, Paul M.B. Vitányi How to Share Concurrent Wait-Free Variables
CT-91-03 Ming Li, Paul M.B. Vitányi Average Case Complexity under the Universal Distribution Equals Worst Case Complexity
CT-91-04 Sieger van Denneheuvel, Karen Kwast Weak Equivalence
CT-91-05 Sieger van Denneheuvel, Karen Kwast Weak Equivalence for Constraint Sets
CT-91-06 Edith Spaan Census Techniques on Relativized Space Classes
CT-91-07 Karen L. Kwast The Incomplete Database
CT-91-08 Kees Doets Levationis Laus
CT-91-09 Ming Li, Paul M.B. Vitányi Combinatorial Properties of Finite Sequences with high Kolmogorov Complexity
CT-91-10 John Tromp, Paul Vitányi A Randomized Algorithm for Two-Process Wait-Free Test-and-Set
CT-91-11 Lane A. Hemachandra, Edith Spaan Quasi-Injective Reductions
CL-91-01 J.C. Scholtes *Computational Linguistics* Kohonen Feature Maps in Natural Language Processing
CL-91-02 J.C. Scholtes Neural Nets and their Relevance for Information Retrieval
CL-91-03 Hub Prüst, Remko Scha, Martin van den Berg A Formal Discourse Grammar tackling Verb Phrase Anaphora
X-91-01 Alexander Chagrov, Michael Zakharyaschev *Other Prepublications* The Disjunction Property of Intermediate Propositional Logics
X-91-02 Alexander Chagrov, Michael Zakharyaschev On the Undecidability of the Disjunction Property of Intermediate Propositional Logics
X-91-03 V. Yu. Shavrukov Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic
X-91-04 K.N. Ignatiev Partial Conservativity and Modal Logics
X-91-05 Johan van Benthem Temporal Logic
X-91-06 Annual Report 1990
X-91-07 A.S. Troelstra Lectures on Linear Logic, Errata and Supplement
X-91-08 Giorgie Dzhaparidze Logic of Tolerance
X-91-09 L.D. Beklemishev On Bimodal Provability Logics for Π_1-axiomatized Extensions of Arithmetical Theories
X-91-10 Michiel van Lambalgen Independence, Randomness and the Axiom of Choice
X-91-11 Michael Zakharyaschev Canonical Formulas for K4. Part I: Basic Results
X-91-12 Herman Hendriks Flexibele Categoriale Syntaxis en Semantiek: de proefschriften van Frans Zwarts en Michael Moortgat
X-91-13 Max I. Kanovich The Multiplicative Fragment of Linear Logic is NP-Complete
X-91-14 Max I. Kanovich The Horn Fragment of Linear Logic is NP-Complete
X-91-15 V. Yu. Shavrukov Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic, revised version
X-91-16 V.G. Kanovei Undecidable Hypotheses in Edward Nelson's Internal Set Theory
X-91-17 Michiel van Lambalgen Independence, Randomness and the Axiom of Choice, Revised Version
X-91-18 Giovanna Cepparello New Semantics for Predicate Modal Logic: an Analysis from a standard point of view