Institute for Language, Logic and Information

GENERAL DYNAMICS

Johan van Benthem

ITLI Prepublication Series
for Logic, Semantics and Philosophy of Language LP-90-11

University of Amsterdam

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

Logic, Semantics and Philosophy of Language:

LP-88-01 Michiel van Lambalgen — 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

Mathematical Logic and Foundations:

ML-88-01 Jaap van Oosten — 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

Computation and Complexity Theory:

CT-88-01 Ming Li, Paul M.B.Vitanyi — 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

Other prepublications:

X-88-01 Marc Jumelet — On Solovay's Completeness Theorem

1989

Logic, Semantics and Philosophy of Language:

LP-89-01 Johan van Benthem — 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

Mathematical Logic and Foundations:

ML-89-01 Dick de Jongh, Albert Visser — 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 $I\Delta_0+\Omega_1$

Computation and Complexity Theory:

CT-89-01 Michiel H.M. Smid — 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

Other Prepublications:

X-89-01 Marianne Kalsbeek — 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 SEE INSIDE BACK COVER

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

GENERAL DYNAMICS

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

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

Received August 1990

To appear in *Theoretical Linguistics*
(Ph.A. Luelsdorff, ed., special issue on
'Complexity in Natural Language')

1 The Interplay of Truth and Action

Many of our central cognitive notions have a dual character. For instance, *judgment* stands for both an intellectual action and the content of that action, and likewise *reasoning* denotes both an intellectual process and its products. This interplay between 'static' contents and 'dynamic' actions is also to be observed in our ordinary use of the term 'natural language', which can be either a static mathematical structure of words and rules, but also a dynamic social activity with many systematic conventions that are not necessarily encoded in explicit syntax. In the mainstream of contemporary Logic and Linguistics so far, static aspects have been predominant, witness the emphasis on isolating so-called 'truth conditions' for linguistic discourse. Here, what may be called Boolean propositional structure is paramount, with various logical operators creating complex forms of description, such as negation, conjunction or disjunction. In other words, the emphasis lies on "that" or "whether" certain statements are true about a situation, not so much on "how" they come to be seen as true. To some, this 'declarative' bias, as opposed to a 'procedural' one, is even a laudable hall-mark of logical approaches as such. But, in recent years, there has been a growing tendency in logical and linguistic research to move dynamic considerations of cognitive action to the fore, trying to do justice to the undeniable fact that human cognitive competence consists in procedural facility just as much as communion with eternal truths.

The purpose of this paper is to propose a general model for construing the interplay of content and action, which may be discerned behind many current proposals in the field. Its mathematical core is not new at all, being derived from existing systems of Relational Algebra and Dynamic Logic. What we want to show, however, is its wide applicability and unifying power across a great variety of apparently diverse research lines in the literature. Moreover, such a general model allows us to start raising systematic questions of 'dynamic logic' across various fields. In particular, we shall make technical proposals for studying such issues as the proper choice of fundamental dynamic logical constants, charting the variety of dynamic forms of inference, and the systematic interplay between newer forms of procedural logic and the original standard systems. Finally, the model also invites reflection in the end on a number of counts. We shall discuss a number of pertinent questions, such as the ontological status of 'transitions' as a new basic category, or the complexity of more procedural systems of logic.

In all this, our focus is on language and cognition, but not exclusively so: the 'general dynamics' of this paper applies also to non-linguistic activities, such as digging or playing. Indeed, we feel that this is the proper strategic depth from which to start logical analysis of these phenomena, before looking at the peculiarities of cognitive action, or within the latter realm again, at specifics of lexically or syntactically encoded cognitive action. Restricting oneself to the latter from the start is like working with a one-dimensional projection of reality, deliberately throwing away valuable clues out of respect for academic conventions. Moreover, at the present level of abstraction, many similarities become visible between Logic, Linguistics, Philosophy and Computer Science, which may become of mutual benefit. Thus, a side purpose of this paper is precisely to effect a quantifier interchange here: each piece of our picture is already in the hands of a number of colleagues, but we would like a number of colleagues to hold all of them together.

2 The Procedural Structure of Dynamics

2.1 Relational Algebra of States and Transitions

The most general model of dynamics is simply this: some system moves through a space of possibilities. Thus, there is to be some set S of relevant *states* (cognitive , physical , etcetera) and a family $\{ R_a \mid a \in A \}$ of binary *transition relations* among them, corresponding to actions that could be performed to change from one state to another.

Procedural aspects of action or cognition then have to do with the way in which such transitions can be combined to obtain certain desired effects. For instance, one can think of an instruction manual for building a model airplane, or of a computer program guiding some computation, or even a linguistic text as it modifies the information state of its reader. For the moment, we do not constrain these examples any further.

'Logic' now enters as a study of the structure of complex procedures on binary relational models and their general effects. If one wishes, this may be contrasted with the 'standard approach', where logical systems strive for description of *unary* propositional *properties* of states, which can only be tested for truth or falsity. The difference is reflected, for instance, in the greater number of natural operators for constructing procedures arising in the dynamic perspective, over and above the Boolean Algebra of mere sets of states. It is easy, however, to exaggerate here: and our view will be that the standard approach retains its value too, both intuitively and in the technical sense of being able to *reduce* the dynamic one by 'translation' if the need arises.

What are the most general operations on actions? Specific examples, that occur over and over again, are *sequential composition* and *choice*. But others occur frequently too, such as *undoing* an action.

A convenient formalism for studying such general operations is that of *Relational Algebra*, being the study of logical operations on binary relations initiated by Ernst Schröder and Alfred Tarski (see Jónsson 1984, Németi 1990). Basic operations here are the following:

$-$, $\cap$, $\cup$	set-theoretic Booleans: complement, intersection and union
o , $^{\vee}$	ordering operations: composition and converse
Δ	the identity relation.

More precisely,

$$R \circ S \quad = \quad \{ (x, y) \mid \exists z: \ Rxz \ \text{and} \ Szy \}$$
$$R^{\vee} \quad = \quad \{ (x, y) \mid Ryx \}$$

For instance, $\cup$ models choice, o sequential composition and $^{\vee}$ 'reversal' for binary relations.

These operations O form a completely general procedural apparatus, which is 'logical' in the sense of being independent from any specific structure of states. Technically, this may be seen in their so-called *invariance under permutations* of states. For instance, in the binary case, we have that:

For every permutation π of the state set S ,
$$\pi[O(R, S)] \quad = \quad O(\pi[R], \pi[S]) .$$

Therefore, not surprisingly, operations from relational algebra return across many concrete systems of dynamics making more specialized choices for the space S .

Next, what are the basic inferential properties of the above logical operators? A well-known list of axioms for these operations looks as follows. One starts with all basic axioms of Boolean Algebra, and then adds the following equations

$$R \circ \Delta = R = \Delta \circ R$$
$$R^{\vee\vee} = R$$
$$(-R)^{\vee} = -R^{\vee}$$
$$(R \cup S)^{\vee} = R^{\vee} \cup S^{\vee}$$
$$(R \circ S) \circ T = R \circ (S \circ T)$$
$$R \circ (S \cup T) = (R \circ S) \cup (R \circ T)$$
$$(R \cup S) \circ T = (R \circ T) \cup (S \circ T)$$
$$(R \circ S)^{\vee} = S^{\vee} \circ R^{\vee}$$
$$(R^{\vee} \circ -(R \circ S)) \cup -S = -S .$$

Nevertheless, it is known that no finite axiomatization can capture all valid principles of Relational Algebra in its set-theoretic guise: an observation to which we shall return in Section 6.

It may not be clear a priori, however, why the 'static' mathematical notion of validity chosen here, namely that of universally true algebraic identities, should be suitable in all dynamic settings. Indeed, the literature shows that a dynamic approach generates plausible new varieties of consequence, just as it generates new kinds of logical operators (see van Benthem 1989A, D). For instance, genuine dynamic validity of an inference might consist in the following sequential prescription:

'first process all premises successively,
then see if the resulting transition is succesful for the conclusion'.

But as we shall see later in Section 5, at least in principle, the latter notions can usually be reduced to the above format.

Evidently, relational algebra as introduced here is a subsystem of *first-order predicate logic*. What this means is that algebraic calculations may always be replaced by predicate-logical reasoning using explicit variables for states. (In fact, Relational Algebra is known to be an undecidable fragment of the latter formalism.) What this gives us is at least a general reduction from dynamic reasoning to standard formalisms, which may be useful for technical purposes, such as establishing standard meta-properties or finding initial estimates of complexity.

In addition to first-order operations, however, there are also other natural *infinitary* operations on binary relations. These correspond to 'unlimited' structures in action, such as endless repetition. The most prominent technical example is that of *transitive closure*:

$$R^* = \{ (x, y) \mid \text{there exists some finite sequence of successive } R \text{ transitions linking } x \text{ to } y \} .$$

This operation will also satisfy various algebraic principles, witness the following sample

$$R \cup R^* = R^*$$
$$R^{**} = R^*$$
$$(R \cup S)^* = (R^* \circ S^*)^* .$$

Digression. Relations and Functions.

Some current dynamic frameworks are couched in terms of *functions* rather than relations, reflecting the intuitive idea of 'transformations' acting on states. In principle, there is no conflict with our approach so far. In one direction, functions are nothing but *deterministic total* relations, and hence the functional perspective is subsumed in the present one. But also conversely, every binary relation R on S induces a function R^* from pow(S) to pow(S) , by setting

$$R^*(X) = R[X] \qquad (= \{ y \in S \mid \exists x \in X\ Rxy \}) .$$

Moreover, nothing is lost in this larger setting, as these 'lifted' functions can still be uniquely retrieved via a well-known mathematical property:

Fact. A function $F : pow(S) \rightarrow pow(S)$ is of the form R^* for some binary relation R on S if and only if F is *continuous*, in the sense of commuting with arbitrary unions of its arguments.

The reason is that continuous maps F can be computed 'locally', from their values at singleton arguments only. The existence of such 'liftings' and 'lowerings' between various levels of set-theoretic representation can also be studied more systematically (see van Benthem 1986, chapter 3). As another example, here is a similar reduction of maps on sets of states to 'propositions' in the standard style, being subsets of S :

Fact. A unary operation F on pow(S) is both continuous and *introspective* (i.e., $F(X) \subseteq X$ for all $X \subseteq S$) if and only if it represents some unary property P via the rule

$$F(X) = X \cap P .$$

2.2 Dynamic Logic at Two Levels

Although relational algebra is powerful enough in principle for embedding standard propositions as special relations, there is much to be said for adopting a two-level perspective merging propositional dynamics with statics. For instance, let us also allow unary properties P, Q, ... of states henceforth. Then, the following picture arises, connecting 'propositions as tests' on states with 'propositions as programs' for changing states:

unary properties	→	'modes'	→	binary relations
of states	←	'projections'	←	between states

propositional operators (Boolean Algebra)	program operators (Relational Algebra)

The motivation for this dual system is partly technical, having to do with mathematical elegance (see Section 4.1), but mainly conceptual: there remains a clear independent intuition behind classical propositions. And there may also be a practical advantage here. Procedural effects are 'local', leaving no traces in short-term memory, while classical propositions may be closer to eventual stored content that can be recalled afterwards. Thus, operating jointly with two (or even more) levels of 'short-term logic' and 'long-term logic' may be eventually what is needed in cognitive practice.

Indeed, the above picture suggests a number of *mechanisms* for switching from one perspective to another, which turn out to correspond to plausible moves that we make in practice. For instance, classical propositions give us 'descriptive contents' for different kinds of action on states, such as

testing P	?P	$\lambda xy.\ y{=}x \wedge Py$
realizing P	!P	$\lambda xy.\ Py$.

And the latter proposition-driven *modes* of action deserve independent logical investigation as such. For instance, even in ordinary argument, there are clear shifts of mode. Intuitively, in the transition "so" from premises to conclusion, a shift is felt from a processing mode for the successive premises to a testing mode for the conclusion.

Likewise, in the opposite direction, there are various natural *projections* from programs to statements, providing classical properties recording the behaviour of binary relations between states, such as

domain R	$\lambda x.\ \exists y\ Rxy$
diagonal R	$\lambda x.\ Rxx$,

where the latter gives the 'fixed points' of the transformation R .

A simple mathematical analysis of special transitions between the two sides, preserving basic logical structure, tells us the following (cf. van Benthem 1986):

Proposition. Among the logical (i.e., permutation-invariant) modes, exactly two are *Boolean homomorphisms*, namely

$\lambda P.\ \lambda xy.\ Py$	('left expansion')
$\lambda P.\ \lambda xy.\ Px$	('right expansion')

Among the logical projections, only one is a Boolean homomorphism, namely

$\lambda R.\ \lambda x.\ Rxx$	('diagonalization').

The other transformations mentioned above, such as 'domain' or 'test', are not entirely without preservation behaviour either. Both are *continuous*, in the earlier sense of commuting with arbitrary unions of their arguments. In fact, all continuous logical modes and projections can easily be classified: they form a finite set.

Such mathematical properties are not ad-hoc. They also make sense in other domains: in particular, for logical operations within the two domains of propositions themselves. For instance, of the earlier basic operations of relational algebra, Boolean intersection and union are continuous in both arguments, and so are composition and converse. The only exception is Boolean negation, which has a special position in other respects too. Thus, we can trace analogies in the logical behaviour of expressions across different categories.

The above system may be compared to 'Dynamic Logic' in the technical sense of the word, a branch of applied Intensional Logic developed by Vaughan Pratt and Rohit Parikh (see Harel 1984). The core system of propositional dynamic logic has 'formulas' and 'programs', defined via a mutual recursion. Basic operators on formulas are the Boolean connectives, while programs are joined by means of the so-called 'regular operations':

$;$	(sequential composition)
$\cup$	(Boolean choice)
$*$	(Kleene iteration) .

Finally, there is a 'test' mode ? taking formulas to programs, and a 'projection' modality $<>$ taking programs π and formulas φ to formulas $<\pi>\varphi$. (A dual universal modality [ ] may be defined in the usual manner.) This formalism is capable of expressing many standard operators on programs, such as Conditional Choice or Guarded Iteration:

IF ε THEN $\pi 1$ ELSE $\pi 2$	$(\varepsilon?;\pi 1) \cup ((\neg\varepsilon)?;\pi 2)$
WHILE ε DO π	$(\varepsilon?;\pi)^*;(\neg\varepsilon)?$

Moreover, it expresses various useful types of statement about execution of programs, such as Correctness, Termination or Enabling:

$\varphi \rightarrow [\pi]\psi$	precondition φ implies postcondition ψ after every succesful execution of program π
$<\pi>T$	program π terminates
$[\pi 1]<\pi 2>\varphi$	program $\pi 1$ 'enables' program $\pi 2$ to produce effect φ.

One virtue of the restricted set of regular program operations, as compared to full relational algebra, is that the resulting logic is *decidable*. Moreover, it manages with a perspicuous set of principles (on top of the minimal propositional modal logic), the so-called 'Segerberg Axioms':

$$< \pi 1;\pi 2\varphi > \varphi \quad \leftrightarrow \quad < \pi 1 > < \pi 2 > \varphi$$

$$< \pi 1 \cup \pi 2 > \varphi \quad \leftrightarrow \quad < \pi 1 > \varphi \vee < \pi 2 > \varphi$$

$$< \varphi ? > \psi \quad \leftrightarrow \quad \varphi \wedge \psi$$

$$< \pi^* > \varphi \quad \leftrightarrow \quad \varphi \vee < \pi > < \pi^* > \varphi$$

$$(\varphi \wedge [\pi^*](\varphi \rightarrow [\pi]\varphi)) \rightarrow [\pi^*]\varphi$$

Most of these equivalences are direct 'reductions' determining the behaviour of program constructions. The final Induction Axiom for iteration, however, reflects the more complex infinitary behaviour of this notion. This logical calculus generalized various earlier natural systems of reasoning, such as 'regular algebra', 'propositional modal logic' and 'Hoare calculus' for correctness assertions.

Like Relational Algebra, Dynamic Logic is an open-ended enterprise, in that many further operations may be studied. For instance, on top of the above, one can add the earlier *converse* operation on programs, running them backwards. This gives rise to what may be called a two-sided 'tense-logical' variant of the system. Moreover, in the literature, the missing Booleans $\cap$ and $-$ have been brought in after all, be it at the price of loss of decidability (cf. Goranko & Passy 1990, Vakarelov 1990). And finally, there has been a good deal of research on various less standard operators of 'parallel execution' in this framework.

3 Concrete Dynamic Systems

3.1 The Structure of States

So far, we have only discussed general procedural structure. A specific dynamic system will usually specify a more detailed set of states, in order to introduce meaningful *basic actions*. These can have quite diverse forms, such as

real action	put the block on the table
games	serve the ball
computation	assign some value to a register
information flow	update the current state.

We shall review a number of more concrete examples below, with 'states' being either concrete physical states, or more abstract procedural ones, or mixtures of both (the most frequent case in practice).

Not only different types of activity induce different kinds of state. Even one and the same activity may be studied using states with different degrees of detail. For instance, already in relational algebra, one can make a standard shift from the original states S to *finite sequences* of such states, being the 'traces' of some relational process. Then, the algebraic operators acquire obvious new meanings: for instance, composition becomes 'concatenation' of traces (see Section 6 below). But of course, one can also give states various components, recording different aspects of the process taking place.

Supplying further detail about the set S of states does not only show up in basic actions: it may also produce further complex procedural operations. For instance, suppose that cognitive states are ordered by some pattern $\subseteq$ of *inclusion* by informational content. Then, there arises a much richer array of

- 'propositional modes', such as

updating	up(P)	$\lambda xy.\ x \subseteq y \wedge Py$
minimal updating	μ-up(P)	$\lambda xy.\ x \subseteq y \wedge Py \wedge \neg\exists z\,(x \subseteq z \subset y \wedge Pz)$
downdating	down(P)	$\lambda xy.\ x \supseteq y \wedge \neg Py$

- 'unary propositional operators', such as

possibility	poss(P)	$\lambda x.\ \exists y\,(x \subseteq y \wedge Py)$

- or 'binary operators' in general, such as

upward part	upw(R)	$\lambda xy.\ Rxy \wedge x \subseteq y$.

This study can be undertaken in two ways. One is to retain the above relational algebra, while adding a constant relation symbol $\subseteq$ explicitly (in addition to the already available identity Δ). The other option is to leave the inclusion ordering implicit, more like accessibility in Modal Logic, and develop a generalization of relational algebra to 'information models', where logical operators no longer satisfy invariance with respect to all permutations of states, but only with respect to those which are also $\subseteq$-*automorphisms* (cf. van Benthem 1989C).

More concretely, the above models may be used to create dynamic variants of standard systems of intuitionistic or modal propositional logic. For instance, one can read the modal operator $\Diamond$ *itself* as a name for the inclusion relation. Moreover, atomic propositions p can be interpreted as standing for updates, either general or 'minimal' in the above sense. Then, there will be further complex algebraic operations corresponding to connectives, such as sequential composition, or testing for domainhood of some relation. (An implementation of this idea for the related case of temporal logic may be found in van Benthem 1989E, 1990.) In a sense, this comes closer to the original intuitions behind possible worlds semantics for intuitionistic logic, as being concerned with the *growth* of knowledge for an epistemic agent. But now, we want this implicit ideology to have its explicit reflection in the design of our logical system itself. Here is a concrete system of 'dynamic modal logic' to illustrate this new perspective on standard intensional systems (cf. van Benthem 1989C for further details):

Propositions now denote transition relations in a standard possible worlds model M = (W, R, V) of 'information states' ordered by some relevant relation R ('inclusion' $\subseteq$ as above, or 'cognitive preference', etcetera):

First, here are some atomic actions:

$[[q?]] = \{ (w,w) \mid w \varepsilon V(q) \}$

$[[q!]] = \{ (w, v) \mid v \varepsilon V(q) \}$

$[[up(q)]] = \{ (w, v) \mid Rwv \text{ and } v \varepsilon V(q) \}$

$[[\mu\text{-}up(q)]] = \{ (w, v) \mid Rwv \text{ and } v \varepsilon V(q) \text{ and no } u$ strictly in between w and v is in $V(q) \}$

Perhaps surprisingly, the modality itself may be viewed as an instruction standing for an atomic move (say, 'random extension'):

$[[\Diamond]] = R$

Then, the procedural repertoire on top of this contains at least 'sequential conjunction' and a test of 'strong failure':

$[[\varphi \wedge \psi]] = [[\varphi]] \circ [[\psi]]$

$[[\neg\varphi]] = \{ (w, w) \mid \text{for no } v, (w, v) \varepsilon [[\varphi]] \}$.

This is just one basic system. In general, 'dynamic modal logic' is more of a family of related systems, differing in their choices of repertoire for atomic actions and general procedural operations. For instance, one can also have downward atomic actions of liberal or strict downdating. And one may add further procedural relational operations too, such as Boolean complement or intersection (standing for *parallel* conjunction):

$[[\varphi \mathbin{\wedge\!\!\!\wedge} \psi]] = [[\varphi]] \cap [[\psi]]$.

And finally, a full dynamic logic on this basis is possible, having both static and dynamic formulas, incorporating the initial atomic modes uniformly for the former, e.g., allowing updates for all complex formulas.

The theory of even this full relational version of basic modal logic may still be studied by standard analogies. For instance, its transition predicates allow of a 'standard translation' into the full first-order language over information models, which explains a number of general features of its logical behaviour. Indeed, e.g., its 'pure test fragment' may still be embedded back into its standard counterpart (van Benthem 1989E). Even so, many of its properties, especially concerning decidability, are still largely unknown.

3.2 General Activities

Let us now briefly consider a number of dynamic 'genres', seeing how the above structures are exemplified in them. The first batch concerns non-linguistic, or non-primarily linguistic phenomena.

• Real action in the world changes actual physical states.

Basic actions are defined either by intrinsic properties of a process, such as "moving", or by a desired end state: "to put a block on the table". In both cases, we recognize our earlier modes: either the resultative ! describing the resulting state, or some imperative linked to a pure description of the desired transition itself.

The algebra of actions includes all Boolean operations: "walk and whistle" (intersection), "walk, don't run" (complement), "take it or leave it" (union), and also all ordering operations: "hit and run" (composition), "put it back" (converse).

Finally, real action involves a natural interplay between instructions and descriptive tests, as shown in "take the tram if you are late" or "run until you are safe" . Not surprisingly then, Pratt 1980, Segerberg 1980 and Moore 1984 have used Dynamic Logic as a general theory of action (see also Meyer 1988). Note that there are also various natural relations between actions that can be expressed in this framework, ranging from 'implying' to 'enabling' in an earlier sense.

• Playing games is another form of action with a clear procedural structure.

Here, the notion of state can be more complex, involving both physical components and more ephemeral obligations and commitments of participants. Examples range from sports to cognitive games, such as those introduced into Logic by Lorenzen 1962, Hintikka 1973B.

Again, basic actions include both kinds of transition ("serving a ball", "drawing an object to be inspected") and tests ("the ball is in", "a player is over-committed").

As for operations on games, the most natural ones seem to be those of *choice* and sequential *composition*. Already in a two-person game, the former leads in fact to two options, depending on whose choice is involved. To see this, identify such games with Parikh 1984 as relations between states representing a win for player I if she plays her best strategy. Then starting with a choice for I amounts to Boolean union, but starting with a choice for player II to Boolean intersection. Moreover, the two roles suggest a natural operation of 'role switching', which amounts to switching of winning and losing states: i.e., to Boolean complement in this representation.

Of course, subtler representations for games are possible too: modelling them, for instance, with both possible transitions and sets of distinguished winning states. Then, e.g., role switching might affect only the latter coordinate. But this is just an earlier general point about all our examples: activities and processes can be modelled at different levels of 'grain size'.

• Perhaps the most important physical process of state change, as far as logical theory is concerned, is that of computation.

In the simplest set-up, states are *environments* mapping variables ('identifiers') in a programming language to values in some data structure. These represent snapshots of the registers during computation.

Basic actions are then *assignments* x:=t and tests ε? for so-called 'Boolean conditions' ε .

The procedural structure of programs has been investigated since Floyd 1967, with operators including the earlier-mentioned sequential composition, conditional choice and guarded iteration. All of these fell within relational algebra, be it that the latter involves infinitary iteration. The latter is often taken to be a special feature of computation: but see van Benthem 1989D for more general cognitive uses of such operators, for instance, when recording continuing (as opposed to singular) commitment in a dialogue.

The semantics of programming contains many cues for a more detailed general logic of dynamics. We mention two simple features that connect up with Section 1.

First, in Computer Science, there is sometimes a preference for an alternative operational description of what a program does: namely, as a 'predicate transformer' mapping 'preconditions' into associated 'strongest postconditions'. What this amounts to is nothing but the functional representation of Section 1.

Then, even for sequential imperative programming, there are standard techniques that reduce relational assertions to statements in some standard logical formalism over states (via 'transition predicates' explicitly manipulating their variables). This will lead us into first-order predicate logic, or in the presence of infinitary program constructions, into logics having infinite conjunctions and disjunctions (cf. Goldblatt 1982 for a principled defense of the latter habitat).

Finally, again, there is no single notion of state serving all computational purposes. For instance, Pnueli 1977 has a mixture of physical environments and internal recording of 'control'. Likewise, one may want to preserve more information about the traces of computations, as in Process Logic (Harel, Kozen & Parikh 1982) or Process Algebra (Milner 1980, Bergstra & Klop 1984).

3.3 Linguistic Activities

In the field of natural language, dynamic ideas have been around for a long time. For instance, in Semantics, incremental growth of discourse representations was already advocated, amongst others, in Seuren 1975, with modern implementations in Heim 1982, Kamp 1984, Seuren 1986. And in Pragmatics, the tendency dates back to at least Stalnaker 1972, who proposed 'context change' as a driving force. This is no coincidence, since a more procedural orientation goes naturally with the older historical perspective of global 'language games'.

Here again, there is a great variety of states and prominent operations, depending on the particular speech act or linguistic 'mode', such as evaluating a sentence against a given model, constructing some partial model, querying a certain predicate, etcetera. We shall survey a few families of linguistic procedures demonstrating this dynamic potential.

3.3.1 Tarskian Variations

The standard truth definition explains the notion $\mathbb{D}, I, b \models \varphi$: that is, 'formula φ is true in structure $\mathbb{D}$ under interpretation I and variable assignment b'. In this scheme, various 'parameters' may be varied dynamically. For instance, evaluation of formulas may be viewed as a process which changes the assignment b . This idea has been proposed by Heim 1982, Barwise 1987 and Groenendijk & Stokhof 1989, who provide different technical implementations. But then, one can also change the interpretation function I : for instance, when incorporating the answer to a *question* "who wants an ice cream?" into one's partial picture of the model. And even the whole structure $\mathbb{D}$ itself may be changed, for instance, when learning about new individuals and new facts. All these possibilities are demonstrated rather nicely in the graphics of the computer program 'Tarski's World' (Barwise & Etchemendy 1988).

Here is a basic system of this kind, analyzed in our general framework. The 'dynamic predicate logic' of Groenendijk & Stokhof has assignments for its states, and essentially two kinds of basic action:

x:= -	random assignment to the variable x
P?	tests for atomic assertions P .

Its further operations are then just two: sequential composition, as well as a test of strong failure

$$\sim R \quad = \quad \{ (x, x) \mid \neg \exists y \, Rxy \} .$$

The latter can be defined as follows in relational algebra:

$$\Delta \cap -(T \circ R^{\vee}) .$$

Thus, dynamic predicate logic is a relational algebra of random assignment with a limited number of algebraic operations on propostions.

Note how there is no independent instruction for quantifiers in this set-up. The existential quantifier corresponds directly to random assignment, while the universal quantifier may be viewed as a test defined out of this. What this suggests is that there need not be any 'quantified dynamic logic' over and above the 'propositional' analysis that we have already given in Section 1.

Finally, from the point of view of 'design' then, it would also make sense to study the full relational algebra of random assignment, including all Boolean operations, whether or not this happens to be realized in natural languages.

Next, as an example of changing interpretations, one can think of imperative instructions of the form

Px!

which drive a 'state' consisting of a pair (b, I) to a new (b, I') in which we have added the object b(x) to the interpretation $I(P)$: $I'(P) = I(P) \cup \{x\}$.
We shall not go into such systems here: the main principle will be clear.

Finally, more intricate semantic settings are conceivable too. For instance, in Hintikka's game-theoretical semantics, the main idea in evaluation is finite sampling from a given structure. In that case, states may consist of partial assignments plus some finite subset of the domain of individuals where they take their values. Thus, $\exists x$ will become an instruction for drawing one more object from the background. (In more psychological terms: one shifts an item from long-term to short-term memory.) Then, further distinctions arise for other quantifiers. For instance, a universal quantifier may now be read as either referring to the sample, or to the total domain. Or more generally, we can now make a plausible distinction between local procedures operating on the sample and global procedures operating on the whole structure.

3.3.2 Information Flow

The preceding example was concerned with dynamics of interpretation. Now, let us consider the dynamics of information flow, accompanying the uses of natural language. There is a wide variety of models in the literature capturing parts of this phenomenon, which we shall first review in broad outline.

A very general perspective was proposed in Gärdenfors 1984, which starts from a bare set of cognitive states and a family of propositions as operations on these, and then explains logical operations on the latter by means of mathematical structures from *Category Theory*. In particular Gärdenfors has composition, but also equivalence (due to the assumed existence of 'equalizers'). Then, he imposes very strong constraints, such as Idempotence (i.e., $f(f(x)) = f(x)$ for all f and x) and Symmetry of composition, in order to pave the way toward standard intuitionistic logic. These constraints are removed in van Benthem 1989D, which views idempotence or symmetry as special denotational properties which propositions may or may not have. Moreover, an inclusion order $\subseteq$ is postulated among cognitive states, which allows for the formulation of finer propositional properties, such as that of being an 'update function', or certain plausible forms of 'monotonicity' (i.e., the property that a function f preserve the inclusion order among states). Finally, more concrete notions of state may be found in Gärdenfors 1988, namely both deductively closed sets of statements and probability distributions over a language. In the latter setting, a more detailed logical theory is developed of various modes for standard propositions: not just *updating*, but also *revision* and *contraction* (the earlier 'downdating').

But there are many more concrete models too. For instance, in the literature on temporal logic, already Allen 1983 has a semantics in terms of temporal data networks that get updated, and which produce the outcome 'true' for a statement in a state just in case its processing leaves that network state unaffected. And likewise, the temporal discourse theory of Kamp 1979 has a similar flavour, with states being partial temporal data bases. This idea then became generalized in the Discourse Representation Theory of Kamp 1984, where 'discourse representation structures' serve as cognitive states, which are transformed by each successive proposition that gets incorporated.

A simpler, 'non-representational' model of information stems from the logical folklore, with recent explicit expression in Heim 1982, Veltman 1988. Here, states are merely classes of valuations, models or indices, which get narrowed down as further propositions are learnt. In a sense, this was already the original leading motivation behind possible worlds semantics as an account of information (although an alternative dynamic view of the framework takes the indices themselves to be information states). This model can be enriched to accommodate various epistemic operators. For instance, in Conditional Logic, implications have been traditionally explained in terms of 'preferences' among models: 'the conclusion must be true in all *most preferred* worlds where the antecedent is true'. But then, it is reasonable to think of a state as a set of worlds together with a preference order over them. And certain operations will affect, not so much the descriptive content of a state, but rather our preference pattern over the worlds in it. This is the main idea, for instance, in Spohn 1988, Veltman 1989, Sandewall 1989, when explaining the cognitive effect of adopting certain heuristic default rules.

Finally, however, there is an alternative approach too, where information states would not be sets of models, but rather indices within one single possible worlds model (cf. Vermeulen 1989, as well as the dynamic modal logic of Section 2.2). This would be appropriate for, say, intuitionistic logic, where worlds have traditionally been thought of as representing stages of knowledge. In that case, cognitive dynamics consists in moving through one such model, going from world to world. (An idealized mathematician will always move forward along the inclusion order, ordinary human beings can also be observed plodding on more zigzagging tracks.)

What has been described so far is merely a number of possible notions of cognitive *state*. In addition, as before, a system of information flow must specify *basic actions* and *procedural operations* over these. We shall merely give some examples here of what may be encountered. In 'constructive' accounts of cognitive states, basic actions will consist in adding or removing pieces of code, perhaps followed by closure under some deductive algorithm. It seems fair to say that little systematic theory has been developed concerning these matters so far. In the more global account with sets of models, things are a little easier. Basic actions may be intersection of these models with some fixed set, thus increasing descriptive content, or changing the preference pattern over them in some prescribed manner. As for procedural operations, it may be of interest to list a few found in Veltman 1988, which is concerned with the 'deterministic' case, where all relevant relations are functions:

$$[[q]]\,(X) = X \cap Q$$

$$[[\varphi \wedge \psi]]\,(X) = [[\psi]]([[\varphi]]\,(X))$$

$$[[\varphi \vee \psi]]\,(X) = [[\varphi]]\,(X) \cup [[\psi]]\,(X)$$

$$[[\neg\varphi]]\,(X) = X - [[\varphi]]\,(X)$$

$$[[\Diamond\varphi]]\,(X) = \begin{cases} X & \text{if } [[\varphi]]\,(X) \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

All these operations move an agent 'forward', to ever smaller sets of possibilities. Backward movement would come in with an 'epistemic retreat', as produced by the qualifier "unless φ ", which adds the whole range $[[\varphi]]$ to the information state again. Moreover, relations instead of functions over states would arise here with a 'revision variant' of the modality, where one would have to retreat to 'some' superset of X for which $[[\varphi]](X)$ is still non-empty.

Without the modality, this is still a purely descriptive system, where propositions φ map states X to their intersection with the standard models for φ. (The technical reason is that all such functions are still 'continuous', in the sense of Section 1.)

Once a preference relation comes in, as outlined above, more complex instructions may be formulated. For instance, van Benthem 1989D takes incoming propositions with a 'heuristic' surplus of information. That is,

[[q]] (X) consists of only the *most preferred* states in $X \cap Q$.

Then, even in the purely 'descriptive' propositional case, order of presentation becomes important: there will already be differences between the dynamic effects of $p^\circ q$, $q^\circ p$ or $p \cap q$. Also, even atomic functions [[q]] will no longer be continuous.

4 General Procedural Logic

In the above framework, a number of general questions of logical dynamics arises, of which we discuss a few of the more salient ones: namely, expressive power, interaction between truth-conditional and procedural views, varieties of inference and natural kinds of informational 'aspect'.

The results thus obtained may be applied again to the concrete systems listed in Section 3.

4.1 Logicality and Invariance

Traditional Boolean Algebra comes with a rather clear-cut idea as to what are the central truth-functional operations on propositions, which is then sanctioned by the usual Functional Completeness theorem. By contrast, Relational Algebra provides a plethora of possible operations on relations. So, one immediate foundational question is what are most natural operations on propositions, the 'logical constants' so to speak, in a dynamic setting. This question probably has no single definitive answer (cf. van Benthem 1989A, C). But what we can do in the present setting is provide an important type of systematic approach.

One basic idea in semantics is that we can measure the expressive power of a formalism against its sensitivity to comparisons between similar models that 'simulate' each other to some extent. Now, one reasonable notion of simulation for dynamic processes derives from ordinary Model Theory. If we want to compare two structures of states-cum-transitions, it must be possible to trace similar processes on either side across suitably connected states and transitions. Technically,

> A binary relation C from S1 to S2 linking points to points and ordered pairs to ordered pairs is a *2-simulation* between two models $(S1, \{R1_a \mid a\varepsilon A\})$, $(S2, \{R2_a \mid a\varepsilon A\})$ if it satisfies
> 'Partial Isomorphism':
>
> matching points (in ordered pairs) satisfy corresponding relational transitions $R1_a$, $R2_a$,
>
> as well as 'Zigzag':
>
> if xCy and $z\varepsilon S1$, then $\exists u\varepsilon S2$: xz C yu,
> and vice versa.

Then, a formula $\varphi(x, y)$ may be called *invariant for 2-simulation* if its truth value is unaffected by passing from evaluation in one model to a corresponding pair of states in a simulated one.

At least within first-order logic, the syntactic effects of this notion can be described very concretely (cf. van Benthem 1990A):

Proposition. A first-order formula $\varphi(x, y)$ is invariant for 2-simulation if and only if it can be written using the two variables x and y only (free or bound).

Examples are the defining schemata for the following operations from our earlier relational algebra:

all Booleans	$\lambda xy. \neg Rxy$
	$\lambda xy. Rxy \wedge Sxy$
conversion	$\lambda xy. Ryx$
identity	$\lambda xy. x=y$
domain	$\lambda xy. \exists x\, Ryx$ (no fresh variable z is needed!)

By contrast, composition involves three variables essentially:

$$\lambda xy.\ \exists z\,(Rxz \wedge Szy)\,.$$

More concretely, this tells us the following about the algebra of relations. If one wants procedural structure that involves essentially just computation over two 'registers' x, y when traversing states, then the following limited set of operations suffices:

Proposition. Each first-order formula $\varphi(x, y)$ with two free variables using only x, y at all can be written in a variable-free notation using the following algebraic operators:

Booleans	$-, \cap, \cup$	Conversion	$\check{}$
Identity	Δ	Domain	Do .

The complete Relational Algebra of Section 1 moves us one step up, as was already observed by Tarski:

Proposition. The operators $\{ -, \cap, \cup, \check{}, \Delta, \circ \}$ form a complete variable-free notation for all first-order formulas $\varphi(x, y)$ that can be written using only three variables x, y, z .

Again, there is also a model-theoretic characterization of the three variable formalism, this time in terms of *3-invariance* with respect to simulation of states, pairs of states and triples of states. (See van Benthem 1990A for more general results.)

These propositions are instances of a more general effective correspondence, between 'finite variable fragments' and complete finite sets of algebraic operators (cf. Gabbay 1981, van Benthem 1989C). In general then, the following *Procedural Hierarchy* arises:

level k=1 is that of the Boolean operations
levels k=2, 3 have been described above

Corresponding invariances will provide ever finer views of processes. For instance, 2-simulation preserves only the structure of successive transitions, while 3-simulation already allows us to decompose a transition into components and make comparisons on both sides. From here, the finite-variable hierarchy extends indefinitely:

> The full first-order language over state structures is not exhausted by any of its k-variable fragments, and hence no finite set of algebraic operations is functionally complete for all of relation algebra in this wider sense.

This, then, is our proposed general background against which to discuss choices of logical operations in a dynamic setting. The Procedural Hierarchy of logical constants allows us to determine the fine-structure of various combinations of instructions.

Example. This perspective may be applied to determine the 'fine-structure' of proposed dynamic operations. For instance, the 'strong negation' of Groenendijk & Stokhof 1989 lies within the 2-variable fragment, since it may be written as:

$$\sim R \quad = \quad \lambda xy.\ y{=}x \land \neg\exists x\ Ryx\ .$$

Also, even though composition is essentially of 3-variable complexity, taking *domains* of compositions may already be done using two variables only:

$$Do(R \circ S) \quad := \quad \lambda x.\ \exists y\ (Rxy \ \land\ \exists x\ Syx)\ \ (!)\ .$$

Then, in the dynamic version of modal logic considered earlier, ordinary updates lie within the two variable fragment:

$$\lambda xy.\ x \subseteq y\ \land\ Py \qquad \text{, or algebraically}$$

$$\subseteq\ \cap\ Do(P?)\ .$$

But minimal updates require three variables essentially:

$$\lambda xy.\ x \subseteq y\ \land\ Py\ \land\ \neg\exists z\ (x \subseteq\neq z\ \land\ z \subseteq\neq y\ \land\ Pz) \qquad \text{, or algebraically}$$

$$\subseteq\ \cap\ Do(P?)\ \cap\ -\ ((\subseteq \cap - \Delta)\ \circ\ (P?\ \circ\ (\subseteq \cap - \Delta)))\ .$$

Some technical details on the Procedural Hierarchy are given in Appendix 6.1.

4.2 Dynamic Logic Revisited

From the above point of view, it seems natural to consider both unary statements $\varphi(x)$ and binary ones $\varphi(x, y)$ in technical co-existence. This brings us again to the earlier Dynamic Logic.

The notion of invariance can also be specialized to explain the specific 'modal' operations available in the basic propositional dynamic logic. For this purpose, one needs so-called *modal zigzags* (or 'bisimulations' in the computational literature): that is, 2-simulations where the back-and-forth clauses only refer to taking R_a-successors of C-connected states. This amounts to tracing a process only via successive atomic actions. Again, a characterization result exists (cf. van Benthem 1985):

Theorem. The first-order formulas $\varphi(x)$ that are invariant for bisimulation are precisely those that occur as translations of modal formulas (in a poly-modal logic with possibility operators for each atomic relation).

Of the first-order relational algebraic operations, this still leaves composition and union: as these admit of obvious reductions. But statements involving, e.g., Boolean *intersection* or *complement* of relations need no longer be preserved in this fashion.

What we see here is an instance of a more general phenomenon. The usual notions of *Modal Logic* may be generalized to a dynamic setting - and then, they provide us with specific 'semantic constraints' on algebraic operations, that bring out some characteristic differences among the various inhabitants of Relational Algebra. Another illustration is modal 'Locality', being the restriction of evaluation to iterated successors and predecessors of the point of evaluation only, which distinguishes negation from 'positive' Booleans like union or intersection.

Also, here is an example of a natural 'model shifting' constraint, relating evaluation of instructions in one model to that in its extensions:

> Which relations $[[\pi]]$ expressed in our language have the following 'Persistence Property':
>
> > if M1 is a submodel of M2, then $[[\pi]]_{M1}$ is the set-theoretic restriction of $[[\pi]]_{M2}$ to the domain of M1 ?

This time, the Booleans operations are completely harmless, but it is rather *composition* which may fail to be persistent: a transition might be decomposable into certain steps in the larger model which are not available in the smaller. And a similar observation applies to the earlier testing mode for negations $\sim\varphi$.

These semantic constraints are of a structural nature, and do not depend on any specific (first-order) formalism. In particular, infinitary operations can share them too. Invariance for bisimulation provides a good illustration. In fact, we have the following joint observation, whose proof is by a simple mutual induction, showing that basic behaviour for atomic propositions and atomic programs extends to all complex ones:

Fact. Let C be a modal zigzag between two models M1, M2 for propositional dynamic logic. Then,

1. C-corresponding states x, y verify the same formulas φ ,
2. all programs π have the back-and-forth property:
   if xCy and $x[[\pi]]_{M1}x'$, then there exists some y' with $y[[\pi]]_{M2}y'$ and yCy' .

The infinitary construction π^* presents no special difficulty here. In fact, it may be shown that modal zigzags exist between two models if and only if these have the same theory in an infinitary standard formalism, namely, the above poly-modal logic provided with arbitrary infinite conjunctions and disjunctions.

Remark. It remains to be understood precisely what makes the regular fragment of the latter formalism so special. By Kleene's Theorem, regular programs π correspond one-to-one with *finite state automata*, recognizing, in this case, 'languages' consisting of all finite sequences of atomic transitions that form the traces of succesful computations for π . Thus, regular programs have a relatively simple computational behaviour. Moreover, regular operations correspond to natural ways of combining processes, such as putting them in 'series' or in 'parallel'. Can we give similar computational analyses for other fragments of relational algebra? For instance, complement corresponds to switching accepting and rejecting states (like the earlier role switching in games), while intersection involves running two processes simultaneously. Thus, one might get a handle on the relative *complexity* of various procedural constructs per se, rather than of global classes of programs.

Digression. The earlier perspective of Section 2 also suggests certain changes in Dynamic Logic as it is usually conceived. These include adding

- a fixed program Δ ('skip') for the identity relation.
- the 'realizing' *mode* ! (sending the property P to the relation $\lambda xy.\ Py$),
  or equivalently,
  a fixed program U for the universal relation ('random change').
  (The resulting extension has been axiomatized and shown decidable in Goranko&Passy 1990.)
- the *projection* of diagonalization sending relations R to properties $\lambda x.\ Rxx$.
  (Gargov & Goranko 1990 axiomatize this, under the name of 'loop'.)

Finally, the Dynamic Logic perspective may be used to implement a two-level architecture for various dynamic systems from Section 3, computing both dynamic effects and their static 'projections':

.... procedural logic

projection

———————————— standard logic

Example. Computing Preconditions and Postconditions.

In computational practice, behaviour of programs is often explained in terms of changing preconditions and postconditions. From our dualistic point of view, this amounts to measuring standard 'unary' informational content along the way. In particular, starting from some set of states described by a predicate P , execution of a program π moves us to a new set of states (the 'image of P under π '), defined by the *strongest postcondition* of P under π :

$$SP(P, \pi) .$$

Conversely, one may also compute *weakest preconditions* for a resulting predicate under a program: $WP(\pi, P)$. The two are related by conversion:

$$SP(P, \pi) \quad = \quad WP(\pi^{\vee}, P) .$$

Now, weakest preconditions describe nothing but the inverse images of relations with respect to certain sets. But, this is precisely what is defined by the central notion of propositional dynamic logic, being $< \pi > \varphi$. And then, the reduction axioms of the latter system may be seen as recursive clauses for computing weakest preconditions. (No similar recursion holds for a pure 'domain' operator on relations: the unary predicate at the back is essential.) Again, this will work only for a limited fragment: there is no similar decomposition for such Boolean combinations as $< \pi 1 \cap \pi 2 >$ or $< - \pi >$, which explains their somewhat more complex status.

Applying all this to the earlier system of dynamic predicate logic, a simple effective mechanism results for computing changing 'classical' informational contents during the dynamic process. First, we note this

Proposition. Weakest preconditions decompose via the following recursion:

$$WP(At, P) \quad = \quad P \wedge At$$
$$WP(x{:=}\text{-}, P) \quad = \quad \exists x\, P$$
$$WP(\varphi 1 \circ \varphi 2, P) \quad = \quad WP(\varphi 1, WP(\varphi 2, P))$$
$$WP(\sim\varphi, P) \quad = \quad P \wedge \neg WP(\varphi, T) \qquad \text{(with 'T' for 'true')} .$$

These clauses may be computed directly or derived via transcription into propositional dynamic logic.

For strongest postconditions then, the above reduction works, since dynamic predicate logic is closed under conversion:

both atomic actions are symmetric

$$(At)^{\vee} \quad = \quad At \qquad\qquad (x{:=}\text{-})^{\vee} \quad = \quad x{:=}\text{-}$$

and conversion treats compounds as follows

$$(\varphi \circ \psi)^{\vee} \quad = \quad \psi^{\vee} \circ \varphi^{\vee} \qquad\qquad (\sim\varphi)^{\vee} \quad = \quad \sim\varphi .$$

More direct recursion clauses would look like this:

$$SP(P, At) \quad = \quad P \wedge At$$
$$SP(P, x{:=}\text{-}) \quad = \quad \exists x\, P$$
$$SP(P, \varphi \circ \psi) \quad = \quad SP(SP(P, \varphi), \psi)$$
$$SP(P, \sim\varphi) \quad = \quad P \wedge \neg WP(\varphi, T) \quad (!) .$$

Further details of this and other examples of the SP projection procedure may be found in Appendix 6.2.

4.3 Varieties of Dynamic Inference

What would be a proper notion of *valid consequence* in a dynamic setting? As with 'logical constants' in earlier Sections, the matter is no longer so clear-cut here as has been traditionally assumed for standard logic. There turns out to be a proliferation of attractive options.

One relatively canonical choice arises by the following recipe:

I 'process all premises successively,
then see if the resulting transition is succesful for the conclusion'.

In other words, *inclusion* and *composition* are the key notions here:

$$[[\varphi 1]] \circ \ldots \circ [[\varphi n]] \subseteq [[\psi]] .$$

This notion of validity is highly sensitive to the order and multiplicity of premises. In particular, the *structural rules* of standard logic (such as Permutation or Contraction of premises, or Monotonicity under additions; cf. van Benthem 1989D) no longer apply in their generality. One only retains the usual Cut Rule.

But there are other reasonable options too. For instance, Groenendijk & Stokhof 1989 have this notion:

II 'process all premises,
then see if a transition can still be made for the conclusion'.

This amounts to stating that the range of the composed premises should be contained in the domain of the conclusion. And likewise, one could merely test the conclusion in the latter range.

But, there are already further competitors around. For instance, Veltman 1990 also considers the following instructions:

III 'see if the conclusion 'holds' at all states where all premises 'hold',
(where a proposition *holds* at a state if that state is among its fixed points),

IV 'see if the conclusion holds at all states arrived at by processing the premises
starting from some state of no information'.

These various notions of inference may have quite different logical behaviour. For instance, some of them are at least monotonic for additions to the left-hand side of the premises, some allow permutation of premises after all, and so forth.

Here again, Relational Algebra provides a unifying perspective. First, there is an analogy here with current 'occurrence-based logics' such as Lambek Calculus or Linear Logic (cf. van Benthem 1989A). There, the basic systems have no structural rules at all, but one can often (though not always!) get the effects of certain structural rules (either the classical ones, or more refined versions, such as those discussed in Makinson 1989), by introducing suitable logical constants allowing special types of inference for specifically marked formulas.

And indeed, all of the above notions of consequence may be reduced to the original one I by introducing standard relational operations from earlier Sections. Here are some possible definitions:

II	$Do((\varphi 1 \circ ... \circ \varphi n)^{\vee}) \subseteq Do(\psi)$
	(the variant with test would have 'diag(ψ)' on the right)
III	$(\varphi 1 \circ ... \circ \varphi n) \cap \Delta \subseteq \psi$
IV	$Do((E \circ (\varphi 1 \circ ... \circ \varphi n))^{\vee}) \subseteq diag(\psi)$
	(where 'E' consists of a single loop at the empty information state) .

After this pruning process, what are the remaining broad options for dynamic logical consequence?

Here is one attractive viewpoint. The main option in our notions of inference may be localized in a matter of 'text logic', namely, the interpretation of the *commas* in inferences involving a sequence of premises. In classical logic, these are interpreted as an instruction for *parallel* processing, via Boolean intersection

$$[[\varphi 1]] \cap ... \cap [[\varphi n]] \subseteq [[\psi]] .$$

In the basic dynamic case, however, they were read as instructions for *sequential* processing, via relational composition. The two options turned out to differ in their logical behaviour, even at the level of purely structural rules.

There are also more technical reasons why these two notions are natural candidates. These will be set forth in Appendix 6.3 below, to which we also refer for missing explanations in what follows.

Notions of inference may be classified as to their global logical properties. For instance, classical consequence $\models$ has all the basic features of set inclusion. Structurally, it satisfies the following one-premise principles:

$A \models A$			Reflexivity
$A \models B$, $B \models C$	imply	$A \models C$	Transitivity .

Next, it also has such Boolean features as:

$A \models B$	implies	$(A \wedge C) \models B$	Left Monotonicity
$A \models B$	implies	$A \models (B \vee C)$	Right Monotonicity
$A \models B$, $A \models C$	imply	$A \models (B \wedge C)$	Conjunction.

These same properties also hold for the above dynamic inference I , with the Booleans now applying to binary relations. (Other options may have different behaviour here. For instance, II does not satisfy Conjunction.)

With multi-premise principles, classical and dynamic consequence will diverge. For instance, the former, but not the latter, satisfies the 'structural rules' of

Permutation

$X \models A$	implies	$\pi[X] \models A$ for any permutation π of X	

Contraction

$X, A, Y, A, Z \models B$	implies	$X, Y, A, Z \models B$	and
		$X, A, Y, Z \models B$	

Monotonicity

$X,Y \models B$	implies	$X, A, Y \models B$.

Remark. Note the different senses of 'monotonicity' involved here. Dynamic inference does admit Boolean strengthening of existing premises, but it does not allow arbitrary addition of new premises.

Nevertheless, there also remain general analogies at the multi-premise level. Classical and dynamic consequence are both *anti-additive* in their left-hand arguments:

$$X, R1 \cup R2, Y \models Z \quad \text{iff} \quad X, R1, Y \models Z \;\; \textit{and} \;\; X, R2, Y \models Z ,$$

and both are *multiplicative* in their right-hand argument:

$$X \models R1 \cap R2 \quad \text{iff} \quad X \models R1 \;\; \textit{and} \;\; X \models R2 .$$

Interestingly, most important structural properties of the above kind can still be studied within Relational Algebra itself. The reason is this. Such properties usually have a universal 'clausal form', as exemplified in the schema of Conjunction:

$$\text{if } X \models \varphi \text{ and } X \models \psi \text{ , then } X \models \varphi \wedge \psi \text{ .}$$

This says that, in any model, if the premises are included in two conclusions, then they are already included in their Boolean intersection. (This is stronger than the 'derived rule' that universal validity of the two premises implies that of the conclusion!) Now, such a principle is not an algebraic identity itself, but rather a 'conditional inclusion'. But, as we have seen before, statements of inclusion may be reduced to identities, already in Boolean Algebra. And also, unlike Boolean Algebra, Relational Algebra has the following reduction property:

Proposition. All universal clauses involving algebraic identities are reducible to equivalent identities.

Combining this observation with the earlier reduction strategy employing additional relational operators, we see that a large part of the structural properties of varieties of dynamic inference lies encoded within Relational Algebra itself.

4.4 Aspectual Calculus for Information

Finally, many special inferential properties of dynamic propositions can be studied in a similar manner.

For instance, certain relations R are 'tests', in that they do not change states:

$$\forall x \forall y\,(Rxy \rightarrow y{=}x) \qquad \text{i.e.,} \quad R \subseteq \Delta \text{ .}$$

Other relations are 'idempotent', in that repeating them will produce no further effect:

$$\forall x \forall y\,(Rxy \rightarrow Ryy) \qquad \text{i.e.,} \quad Do(R) \subseteq diag(R) \text{ .}$$

The latter property (the 'quasi-reflexivity' of possible worlds semantics) is plausible for updates. Such special relations may be recognized by their inferential behaviour. For instance, tests are 'monotonic', in that they can be inserted in sequences of premises without disturbing any dynamic conclusions in sense I . And the latter property is also characteristic for them: monotonic relations in this sense must be tests.

Many standard properties of binary relations make sense for the purpose of dynamic classification. For instance, both tests P? and realizations P! are idempotent. But the former have other well-known modal properties too, such as Symmetry and 'Euclidity':

$$\forall x \forall y \forall z\, ((Rxy \wedge Rxz) \rightarrow Ryz)\,.$$

The latter rather has

$$\forall x \forall y \forall z\, (Rxy \rightarrow Rzy)\,.$$

This picture becomes richer in the special case of information structures carrying an inclusion order $\subseteq$. Then, many other interesting properties arise, such as that of being 'updating' or 'progressive': i.e., being a subrelation of the inclusion order. (For the special case of functions on information states, there are even some natural conditions of preserving inclusion, or infima and suprema in the information ordering.) Here, we only want to point at a general analogy. The ordering $\subseteq$ may be compared to inclusion of matter or time, as in Dowty 1979 or Krifka 1989: information is a kind of 'stuff' too. Thus, we have what linguists would call an 'aspectual calculus' for various kinds of informative behaviour of propositions. In particular, we may investigate syntactic or semantic criteria for recognizing whether a given complex proposition is going to have specific desirable dynamic properties. Here is a question illustrating this kind of concern:

Which syntactic forms in the relational algebra over $\subseteq$ are progressive ?
Which syntactic operations create, or endanger, this property?

5 Conclusion

The main point of this paper has been the following: Relational Algebra and Dynamic Logic provide a convenient setting for bringing out essential logical features of action and cognition. To conclude, here is some clair-obscur to this clear picture.

In reality, there are mathematical alternatives too, witness the earlier-mentioned category-theoretic approach of Gärdenfors 1984. And even closer to the present paradigm, variations are possible. For instance, it would also be possible to assign each proposition its 'trace set', being those finite sequences of atomic transitions traversed during its succesful computations. The latter assignment also records intermediate stages (for instance, for compositions), that are lost in our more 'extensional' account of dynamics. In fact, 'state models' $(S, \{R_a \mid a \in A\})$ as used so far can be mapped to 'trace models' S^* , whose domain consists of all *finite sequences* of states from S . Then, atomic relations R_a may be mapped onto sets of sequences, for instance

$$R_a \quad \mapsto \quad \{ xy \mid Rxy \},$$

or

R_a $\mapsto$ 'all sequences whose beginning and end form a transition in R_a'.

Evaluation of complex relational expressions in trace models may then follow its own intuitions. In particular, composition will now refer to *concatenation* of sequences having matching ends and beginnings:

[[RoS]] = all sequences in [[R]] concatenated with a matching sequence in [[S]] .

Boolean operations will retain their usual meaning here.

The latter approach takes us closer to the 'language models' studied in van Benthem 1989A, 1990A: trace models really consist of families of 'transition languages' for processes. The resulting logic is related to relational algebra, but also to categorial 'occurrence logics'.

Moreover, the modelling given so far may eventually prove not radical enough. After all, if one is willing to believe that logical dynamics is of equal importance to logical statics, then perhaps, the two aspects ought to be accorded equal ontological status. What this means again is that *transitions* would come to be viewed, not as ordered pairs of states, but rather as independent basic objects in their own right. In that case, the relevant mathematical paradigm would no longer be Relational Algebra with its intended set-theoretic semantics, but rather some form of 'arrow logic' as advocated in van Benthem 1989B. Its structures are of the form

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

where W is a set of arrows, C a ternary graph for the partial operation of composition of arrows, F a binary predicate for their conversion, and I the unary predicate of being an identity arrow.

In particular, then, transitions can be extensionally the same in having equal starting and finishing points, without being intensionally identical.

The latter move might also alleviate a problem of computational complexity. Relational Algebra is known to be a more complex theory than Boolean Algebra. As was already observed in Section 1, it is *undecidable*: such is the price of its greater expressive power. Thus, perhaps surprisingly, incorporating cognitive dynamics has not led so far to any gains in practicality or simplicity. But arrow logic is less costly (I. Németi, personal communication, has claimed its decidability).

Next, the word "general" has been used deliberately at various places in this paper to state our aims, as not being tied to specific linguistic forms or practices in logical semantics. For instance, it remains to be seen in the long run how much essential procedural structure can be captured in the standard semantic approach. Many important procedural aspects of cognition seem deeply tied up with the actual syntactic lay-out of arguments, or the actual course of games. What compositional semantics tends to do is provide only limited access here, by discarding the syntactic structure that produced its denotations as soon as possible. (Admittedly, for some purposes, that is precisely its strength.) There is a certain unnaturalness to this, in that humans do have short-term recall of syntactic structure: so, why not make free use of it? In this respect, recent proof-theoretic approaches, like that of Kracht 1988 or Ranta 1990, may have their advantages, identifying states with the available proof-theoretic 'context', which allows for a more flexible dynamics, switching between what was said and how it was said.

Likewise, once more, we want to stress our rather free-wheeling attitude toward 'natural language' in the current investigation. As was stated before, the crucial issue is to understand human action and cognition, whether or not this is fully encoded linguistically. It would be a kind of wilful suppression of the evidence to stick with the 'letter' of cognition, rather than its 'spirit' here (compare Levelt 1989 on this issue). And indeed, the procedural perspective is rather unorthodox, in that one is treating lexical expressions, such as "and", "if" or "not", on a par with discourse particles like "so", or even just punctuation signs, such as commas or dots. (The latter idea is not new in Logic: see Jennings 1986, Dosen 1989 for a view of logical constants themselves as 'punctuation marks'.) The latter theme could be elaborated more systematically: for instance, certain typographical devices are obvious 'mode switchers', witness question or exclamation marks, or semicolons inside a sentence. And from graphics, the border line to acoustics, or even just abstract game conventions 'in the air', becomes quite fluid.

Thus, one has to admit the justice of an unconventional position which was already advocated in Hintikka 1973A, Barwise & Perry 1983: studying cognition entirely at the level of syntax means using an 'existential projection' which might be even more complex than the full picture (compare the non-decidability of many recursively enumerable predicates). And in the end, logical analysis becomes concerned, as it should be, with just any rational clue in discourse and argumentation.

6 Appendices

6.1 The Finite Variable Hierarchy

Here are some details behind the assertions in Section 4.1.

- Explicit Syntactic Description of the Two-Variable Fragment.

Consider any formula ϕ (x, y) written with the two variables x, y only. Without loss of generality, it may be supposed that ϕ involves only atoms, $\neg$, $\wedge$ and $\exists$. If there are no quantifiers, then a Boolean combination suffices of cases

$\lambda xy \bullet Rxy$	R	$\lambda xy \bullet Ryx$	$R^{\vee}$
$\lambda xy \bullet Ryy$	$Do(R \cap \Delta)$	$\lambda xy \bullet Rxx$	$(Do(R \cap \Delta))^{\vee}$

Now, consider any innermost quantifier occurring in a position

$\exists y\ \alpha$ (x, y) where α is quantifier free.

That is, we may assume that α already has a relational transcription α^*. But then, $Do(\alpha^*)$ will do. This procedure may be repeated until one reaches the outside formula.

This explains the 2-variable completeness of the operator set

$$\{ -, \cap, \cup \quad ^{\vee} \quad \Delta \quad Do \}.$$

- Semantic Characterization of k-Variable Fragments by Means of Invariance.

First-order model theory has the notion of <u>partial isomorphism</u> between two models $\mathbb{M}$ and $\mathbb{N}$, via some non-empty family $\mathbb{PI}$ of finite partial isomorphisms between their domains, satisfying the Back and Forth extension properties:

If the partial isomorphism (X, Y) (viewed as a pair of matching sequences) is in $\mathbb{PI}$, and a is any object in $\mathbb{M}$, then there exists some object b in $\mathbb{N}$ such that (Xa, Yb) is also in $\mathbb{PI}$. And analogously in the opposite direction.

One important observation is that corresponding sequences X, Y verify the same first-order formulas in the two models.

Now, for k-variable fragments, this notion may be restricted in an obvious manner to partial isomorphisms of length <u>at most</u> k, to obtain a notion of k-<u>partial isomorphism</u>. And then, a straightforward induction shows that

Proposition. Formulas from the k-variable fragment are invariant for matching sequences in k-partial isomorphism.

There is also a converse (cf. van Benthem 1990B).

Theorem. Any formula $\phi = \phi(x_1, ..., x_k)$ in the full first-order language (possibly employing other bound variables besides those displayed) which is invariant for k-partial isomorphism is logically equivalent to a formula constructed using $x_1, ..., x_k$ only.

Proof Sketch. This may be proved using the following claim:

- Each invariant formula ϕ follows from the set $\mathbb{k}(\phi)$ of its own k-variable consequences.

By the Compactness Theorem, ϕ must then be equivalent to some finite conjunction of k-variable formulas.

The proof of the claim runs roughly as follows:

Suppose that $\mathbb{M}, a \models \mathbb{k}(\phi)$. Then, there exists some model $\mathbb{N}, b$ that is k-elementarily equivalent to $\mathbb{M}, a$ in which ϕ holds. But now,

- Any two models that are elementarily equivalent with respect to k-variable formulas have saturated elementary extensions that are k-partially isomorphic via the family of all their pairs of sequences up to length k verifying the same type in the k-variable language.

Take two such elementary extensions, say $\mathbb{M}^*, \mathbb{N}^*$. Then we have

$\mathbb{N}, b \models \phi$ (by assumption) $\mathbb{N}^*, b \models \phi$ (by elementary extension)
$\mathbb{M}^*, a \models \phi$ (by k-invariance) $\mathbb{M}, a \models \phi$ (by elementary descent) .

- Lack of Functional Completeness for the Full Relational First-Order Language.

We merely demonstrate the kind of argument that is required for a negative conclusion like this.

As is well-known from standard logic, two models are indistinguishable by first-order sentences up to quantifier depth n iff the second player in an *Ehrenfeucht comparison game* over these structures has a winning strategy in any play over n rounds. Now, this analysis has been refined in Immerman & Kozen 1987, to show that

> Indistinguishability by sentences up to depth n employing only some fixed set of k variables amounts to the existence of a winning strategy for the second player in a modified Ehrenfeucht game over n rounds, where each player receives k *pebbles* at the start, and can only select objects by putting one of these pebbles on them.

Now, any finite operator formalism has a first-order transcription involving only some fixed finite number k of variables over states. Therefore, if any such formalism were functionally complete, this would mean that the full relational first-order language would actually be logically equivalent to one of its own k-variable fragments. But the latter kind of reduction is impossible:

Consider the first-order sentence ϕ stating the existence of at least k+1 distinct points all unrelated in the relation R. Consider also two models consisting of k and k+1 R-unrelated points, respectively, with all other relations empty too. Evidently, the second player has a winning strategy in the Ehrenfeucht comparison game between such models with k pebbles, over an arbitrary finite number of rounds. So, no k-variable sentence distinguishes between these two models: whereas ϕ can.

6.2 Two-Level Logic

- A Sample Computation of Postconditions for Dynamic Predicate Logic.

Here is an illustration, with 'trace points' indicated in bold-face subscripts:

$\exists x(Ax \circ Bx) . \exists x\, Cx . \sim\exists x\, Dx$
$\exists x_{\mathbf{1}} (Ax_{\mathbf{2}} \circ Bx_{\mathbf{3}}) . \exists x_{\mathbf{4}}\, Cx_{\mathbf{5}} . (\sim\exists x\, Dx)_{\mathbf{6}}$

SP	0	T	1	$\exists x T{=}T$	2	Ax
	3	$Ax \wedge Bx$	4	$\exists x\,(Ax \wedge Bx)$ *	5	$\exists x(Ax \wedge Bx) \wedge Cx$
	6	$\exists x(Ax \wedge Bx) \wedge Cx \wedge \neg\exists x\, Dx$.				

At the trace point *, the initial $\exists x$ becomes an ordinary quantifier after all.

- Postconditions for an Update System.

Similar calculations are possible for other dynamic systems, even when based on slightly different principles. An illustration is the propositional update semantics presented in Section 3 (cf. Veltman 1986). Here, the relevant clauses for on-line computation of classical content are as follows:

$SP(P, q) = P \wedge q$
$SP(P, \varphi \wedge \psi) = SP(SP(P, \varphi), \psi)$
$SP(P, \varphi \vee \psi) = SP(P, \varphi) \vee SP(P, \psi)$
$SP(P, \neg\varphi) = P \wedge \neg SP(P, \varphi)$
$SP(P, \Diamond\varphi) = P$ if $SP(P, \varphi)$ is consistent, $\perp$ otherwise.

- Postconditions for Dynamic Modal Logic.

One can also compute weakest preconditions for a dynamic modal logic in one of its standard relatives, provided that a suitably strong formalism is chosen for the latter. In particular, consider a language with a future modality Fut and a past modality Past , as well as binary operators Since and Until . (Indeed, the latter suffice, as they already define the former.)

Proposition. Weakest preconditions for dynamic modal logic may be computed via the following recursion:

$WP(p?, A)$	$=$	$p \wedge A$
$WP(\text{up } p, A)$	$=$	$\text{Fut}\,(p \wedge A)$
$WP(\mu\text{-up } p, A)$	$=$	$\text{Until}\,(p \wedge A, \neg p)$
$WP(\text{down } p, A)$	$=$	$\text{Past}\,(\neg p \wedge A)$
$WP(\mu\text{-down } p, A)$	$=$	$\text{Since}\,(\neg p \wedge A, p)$
$WP(\phi \wedge \psi, A)$	$=$	$WP(\phi, WP(\psi, A))$
$WP(\neg\phi, A)$	$=$	$A \wedge \neg WP(\phi, \text{TRUE})$.

The proof is by a straightforward calculation. Note how a 3-variable standard formalism is needed for tracing a 2-variable dynamic counterpart.

Thus, we can use standard modal formalisms for keeping track of the long-term effects of dynamic ones.

6.3 Structural Rules and Representations

- Relational Algebra Embeds its own Universal Clauses.

A universal clause of the form

'universal prefix $\forall R1 \ldots \forall Rn$, followed by a disjunction of equalities and inequalities between terms in R1, ..., Rn '

may be reduced as follows. First, already by Boolean Algebra, it suffices to consider equalities of the form 'A=1' . Next, inequalities may be replaced by equalities, because of the key equivalence

$$A \neq 1 \quad : \quad 1 \circ - A \circ 1 = 1 \qquad (\text{'}A\#=1\text{'}) .$$

But then, even one equality suffices for the remaining disjunction of equalities, thanks to the following chain of equivalences

$$A=1 \vee B=1 \qquad \neg(A \neq 1 \wedge B \neq 1) \qquad \neg(A\#=1 \wedge B\#=1)$$

$$A\# \cap B\# \neq 1 \qquad (A\# \cap B\#)\#=1 .$$

- Systematic Perspectives on Basic Notions of Inference.

The variety of dynamic notions of inference may be approached more systematically than has been done in the main text.

For instance, all possible candidates satisfying the earlier two constraints of 'anti-additivity' and 'multiplicity' can be analyzed still further, using the mathematical property of *permutation invariance.* Arguably, proposed notions of inference should be permutation-invariant 'meta-relations' between the binary relations corresponding to their component propositions. Then, only a finite number of possible interpretations remains for the comma: most prominently, Boolean conjunction, disjunction and relational composition:

Proposition. In any set-algebra of binary relations, the only permutation-invariant anti-additive and multiplicative inference relations are those definable by a conjunction of schemata of the form

$$\forall x_1 y_1 \varepsilon \varphi_1 \ldots \forall x_k y_k \varepsilon \varphi_k : \alpha \rightarrow \beta ,$$

where α is some Boolean condition on identities in the variables x_i, y_j and β is some atom of the form $\psi x_i y_j$.

The above two notions of classical and dynamic consequence are characteristic of anything that may be encountered in this general area. For, many of the above schemata are reducible to these two, using suitable algebraic compounds of the φ_i and ψ . Moreover, their shape suggests one further plausible restriction. The above schemata are all *universal clauses*. But intersective or compositive inference is even defined in the more restricted *Horn clause* format. For instance, with two premises, they read as follows:

$$\forall xy\ \varepsilon\ \varphi_1\ \forall zu\ \varepsilon\ \varphi_2 : (x{=}z \wedge y{=}u) \rightarrow \psi xy ,$$
$$\forall xy\ \varepsilon\ \varphi_1\ \forall zu\ \varepsilon\ \varphi_2 : y{=}z \rightarrow \psi xu .$$

And all candidates in this Horn format are algebraically reducible to just these two.

- Natural Clusters of Structural Rules.

Another way of stating the special status of classical intersective and dynamic compositive inference arises via purely structural rules only, without any Boolean connectives. Natural clusters of structural rules determine natural kinds of reasoning, and hence, they provide a convenient focus for logical research.

For a start, standard consequence is characteristic for the usual set of structural rules, which lump premises together into mere sets of formulas, namely:

Reflexivity	Cut
Monotonicity	Contraction .

(We omit Permutation here, as it is derivable from these four.)

Proposition. The classical structural rules axiomatize precisely the theory of the set-theoretic relation $a_1 \cap \ldots \cap a_k \subseteq b$.

Proof. The reason is the following representation result. Let R be any abstract relation between finite sequences of objects and single objects satisfying the classical structural rules. Now, define

$$a^* \quad = \quad \{ A \mid A \text{ is a finite sequence of objects such that } A\, R\, a \} .$$

Then, it is easy to show that

1 if $a_1, .., a_k\, R\, b$, then $a_1^* \cap \ldots \cap a_k^* \subseteq b^*$,
using Cut and Contraction, while
2 if $a_1^* \cap \ldots \cap a_k^* \subseteq b^*$, then $a_1, .., a_k\, R\, b$,
follows by Reflexivity and Monotonicity.

Actually, a similar representation would also have worked for the earlier inference notion III in terms of mere testing of the premises.

On the other hand, dynamic consequence is characteristic for the opposite extreme, leaving the ordering of premises completely intact.

Proposition. Reflexivity and Cut axiomatize precisely the theory of the set-theoretic relation $a_1 \circ \ldots \circ a_k \subseteq b$.

Proof. This time, a relational representation is needed for any abstract relation R like above satisfying only the two mentioned constraints. Again, consider the universe of all finite sequences of objects B , and set

$$a^{\#} \quad = \quad \{ (B, BA) \mid \text{for any finite sequence } A \text{ such that } A\, R\, a \} .$$

Then, by Cut alone, we have that, if $a_1, .., a_k$ R b , then $a_1^{\#} \circ ... \circ a_k^{\#} \subseteq b^{\#}$, while Reflexivity already implies the converse, when applied to the sequence of transitions $(\langle\rangle, a_1), (a_1, a_1a_2), ..., (a_1...a_{k-1}, a_1...a_k)$.

- Subtleties of Preferential Reasoning.

Other natural clusters of structural rules will arise with further notions of logical consequence. These also illustrate a new phenomenon, namely that adherence to standard principles need not be an all-or-nothing matter.

For instance, universes of models graded by a preference order as in Section 3 lead to a different broad option, namely that of 'preferential entailment':

"The conclusion should be true in all *most preferred* models for the premises".

In the latter case, those structural rules for standard consequence which reflect its set-like treatment of the premises are not at issue, such as Permutation, Contraction and *Expansion* (allowing duplication of occurrences of available premises). Moreover, Reflexivity holds even in the following strong form:

$X \models A$ whenever A occurs in X .

But, with this kind of more heuristically oriented reasoning, not only Monotonicity but also the Cut Rule ('Transitivity') becomes invalid in general. One only retains these principles in 'cautious' forms, allowing only addition of premises which follow from the original ones, or removal of premises already derivable from the remaining ones. These form a dual pair:

$X \models A$ and $X \models B$ imply $X, A \models B$

$X \models A$ and $X, A \models B$ imply $X \models B$.

Proposition. Preferential entailment has its structural rules axiomatized completely by

Contraction	Expansion	Permutation
Strong Reflexivity	Cautious Monotonicity	Cautious Cut.

The key idea of the required representation is this. Objects are now mapped to the family of all 'states' to which they belong. Here, a *state* is any set X of objects for which we have 'harmony':

$b \in X$ iff $X \models b$ for all objects b .

The relevant preference order over states is just ordinary set-theoretic inclusion ('the smaller the better').

7 References

J. Allen, 1983, 'Maintaining Knowledge about Temporal Intervals', *Communications of the Association for Computing Machinery* 26, 832-843.

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

J. Barwise & J. Etchemendy, 1988, 'Tarski's World', Academic Software.

J. Barwise & J. Perry, 1983, *Situations and Attitudes*, Bradford Books / MIT Press, Cambridge (Mass.).

J. van Benthem, 1985, *Modal Logic and Classical Logic*, Bibliopolis, Naples.

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

J. van Benthem, 1989A, 'Language in Action', Report LP-89-04, Institute for Language, Logic and Information, University of Amsterdam. (To appear in *Journal of Philosophical Logic*.)

J. van Benthem, 1989B, '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.)

J. van Benthem, 1989C, 'Modal Logic as a Theory of Information', Report LP-89-05, Institute for Language, Logic and Information, University of Amsterdam. (Revised version to appear in J. Copeland, ed., *Proceedings Arthur Prior Memorial Conference*, Christchurch, New Zealand.)

J. van Benthem, 1989D, 'Semantic Parallels in Natural Language and Computation', in H-D Ebbinghaus et al., eds., *Logic Colloquium. Granada 1987*, North-Holland, Amsterdam, 331-375.

J. van Benthem, 1989E, 'Time, Logic and Computation', in J. W. de Bakker et al., eds., *Linear Time, Branching Time and Partial order in the Semantics of Concurrency*, Springer Verlag, Berlin, 1-49.

J. van Benthem, 1990A, *Language in Action: Categories, Lambdas and Dynamic Logic*, to appear with North-Holland, Amsterdam, (Studies in Logic).

J. van Benthem, 1990B, 'Temporal Logic', Institute for Language, Logic and Information, University of Amsterdam. (To appear in D. Gabbay et al., eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press.)

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

K. Dosen, 1989, 'Logical Constants as Punctuation Marks', *Notre Dame Journal of Formal Logic* 30, 362-381.

D. Dowty, 1979, *Word Meaning and Montague Grammar*, Reidel, Dordrecht.

R. Floyd, 1967, 'Assigning Meanings to Programs', *Proceedings AMS Symposia in Applied Mathematics* 19, Providence (R.I.), 19-31.

D. Gabbay, 1981, 'Functional Completeness in Tense Logic', in U. Mönnich, ed., *Aspects of Philosophical Logic*, Reidel, Dordrecht, 91-117.

P. Gärdenfors, 1984, 'Propositional Logic based on the Dynamics of Belief', *Journal of Symbolic Logic* 50, 390-394.

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

G. Gargov & V. Goranko, 1990, 'Extended Modal Logic, and Looping', Mathematical Institute, Kliment Ohridski University, Sofia.

R. Goldblatt, 1982, *Axiomatizing the Logic of Computer Programming*, Springer Verlag, Berlin.

V. Goranko & S. Passy, 1990, 'Using the Universal Modality: Gains and Questions', Mathematical Institute, Kliment Ohridski University, Sofia.

J. Groenendijk & M. Stokhof, 1989, 'Dynamic Predicate Logic', Institute for Language, Logic and Information, University of Amsterdam. (To appear in *Linguistics and Philosophy*.)

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

D. Harel, D. Kozen & R. Parikh, 1982, 'Process Logic: Expresiveness, Decidability, Completeness', *Journal of Computer and Systems Sciences* 25, 144-170.

I. Heim, 1982, *The Semantics of Definite and Indefinite Noun Phrases*, Department of Linguistics, University of Masssachusetts, Amherst.

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

J. Hintikka, 1973B, 'Quantifiers versus Quantification Theory', *Dialectica* 27, 329-358.

N. Immerman & D. Kozen, 1987, 'Definability with Bounded Number of Bound Variables', *Proceedings* IEEE 1987, 236-244.

R. Jennings, 1986, 'Logic as Punctuation', in W. Leinfellner & F. Wuketits, eds., *The Tasks of Contemporary Philosophy*, Schriftenreihe der Wittgenstein Gesellschaft, Hölder-Pichler-Tempsky Verlag, Wien.

B. Jónsson, 1984, 'The Theory of Binary Relations', Department of Mathematics, VanderBilt University, Nashville (Tenn.).

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

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

M. Kracht, 1988, ' How to say "It" ', Mathematisches Institut, Freie Universität, Berlin.

M. Krifka, 1989, 'Nominal Reference, Temporal Constitution, and Quantification in Event Semantics', in R. Bartsch et al., eds., *Semantics and Contextual Expression*, Foris, Dordrecht, 75-115.

W. J. Levelt, 1989, *Speaking. From Intention to Articulation*, The MIT Press, Cambridge (Mass.).

P. Lorenzen, 1962, *Metamathematik*, Bibliographisches Institut, Mannheim.

D. Makinson, 'General Non-Monotonic Logic', UNESCO, Paris. (To appear in D. Gabbay et al., eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press.)

J-J Meyer, 1988, 'A Different Approach to Deontic Logic: Deontic Logic viewed as a Variant of Dynamic Logic', *Notre Dame Journal of Formal Logic* 29.

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

R. Moore, 1984, 'A Formal Theory of Knowledge and Action', Artificial Intelligence Center, SRI International, Menlo Park.

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

R. Parikh, 1984, 'The Logic of Games and its Applications', Department of Computer and Information Sciences, Brooklyn College, SUNY, New York.

A. Pnueli, 1977, 'The Temporal Logic of Programs', *Proceedings 18th IEEE Symposium on Foundatuions of Computer Science*, 46-57.

V. Pratt, 1980, 'Application of Modal Logic to Programming', *Studia Logica* 39: 2/3, 257-274.

A. Ranta, 1990, 'Intuitionistic Categorial Grammar', Institute for Language, Logic and Information, University of Amsterdam. (To appear in *Linguistics and Philosophy*.)

E. Sandewall, 1989, 'The Semantics of Non-Monotonic Entailment Defined Using Partial Interpretations', in M. Reinfrank, J. de Kleer, M. Ginzberg & E. Sandewall, eds., 1989, *Non-Monotonic Reasoning*, Springer Verlag, Lecture Notes in Artificial Intelligence, 27-41.

K. Segerberg, 1980, 'Applying Modal Logic', *Studia Logica* 39:2/3, 275-295.

P. Seuren, 1975, *Tussen Taal en Denken*, Oosthoek, Scheltema en Holkema, Utrecht.

P. Seuren, 1986, *Discourse Semantics*, Blackwell, Oxford.

W. Spohn, 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.

R. Stalnaker, 1972, 'Pragmatics', in D. Davidson & G. Harman, eds., *Semantics of Natural Language*, Reidel, Dordrecht, 380-397.

D. Vakarelov, 1990, 'Describing Program Conjunction and Complement by means of Modal Rules', Mathematical Institute, University of Sophia.

F. Veltman, 1986, 'Update Semantics', Institute for Language, Logic and Information, University of Amsterdam.

F. Veltman, 1990, 'Defaults in Update Semantics', Institute for Language, Logic and Information, University of Amsterdam. (To appear in L. Ivanov et al., eds., *Proceedings Kleene Conference. Chaika 1990*, Plenum Press, New York.)

C. Vermeulen, 1989, 'A Dynamic Analysis of Reasoning', master's thesis, Philosophical Institute, University of Utrecht.

The ITLI Prepublication Series

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
Mathematical Logic and Foundations	
ML-90-01 Harold Schellinx	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
Computation and Complexity Theory	
CT-90-01 John Tromp, Peter van Emde Boas	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, Leen Torenvliet	Bounded Reductions
Other Prepublications	
X-90-01 A.S. Troelstra	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