DYNAMIC BITS AND PIECES

Johan van Benthem

January 1997

This report is the first of a planned sequence of annual updates of the book "Exploring Logical Dynamics", CSLI Publications, Stanford, summer 1996. It contains a number of further results obtained since its first publication. Sections 1, 2, 3 concern various issues in modal logic, 4, 5, 6 dynamic logic, and 7 temporal logic, while Sections 8, 9 digress into infinitary logic. Some corrigenda, and related results by others, have been included as well.

1 On the History of Bisimulation

Bisimulation is the characteristic semantic invariance for the language of modal logic. In computer science, it is also a central notion of process equivalence in its own right. At the request of some colleagues, I record a few personal notes about its history.

Modal Frames and *p*–Morphisms

When modal logic took off in the sixties, its practitioners focussed on semantic 'frames' $\mathbf{F}$ = (W, R) of worlds with some accessibility relation. Frames are the underlying structures of the usual Kripke models $\mathbf{M}$ = (W, R, V) which add a valuation V evaluating the proposition letters in all worlds. A modal formula ϕ holds in model $\mathbf{M}$ at world w ($\mathbf{M}$, w $\models \phi$) if it evaluates to true according to the usual truth definition. It is then true in a frame if it is true at all worlds under all valuations over that frame. (Note that this is second-order.) This led to an interest in truth-preserving operations on frames. Examples of these are: generated subframes, disjoint unions, and in particular, so-called *p*–morphic images, where a *p–morphism* f is an R–homomorphism from one frame onto another which satisfies the backwards condition that, whenever Rf(w)v, there exists u with Rwu such that f(u)=v . This notion is due to Krister Segerberg, and appeared in his dissertation "An Essay in Classical Modal Logic" (*Philosophical Studies*, Uppsala 1971). For intuitionistic propositional logic, though, a similar notion occurs earlier in D. de Jongh & A. Troelstra (1966), 'On the Connection of Partially Ordered Sets with Some Pseudo-Boolean Algebras', *Indagationes Mathematicae* 28, 317-329. The mathematical high-light of the frame tradition is the characterization of all modally definable classes of frames given in R. Goldblatt & S.K. Thomason 1975, 'Axiomatic Classes in Propositional Modal Logic' (J.N. Crossley, ed., *Algebra and Logic*, Springer Lecture Notes in Mathematics 450, Berlin, 163-173). The general result is somewhat cumbersome to state, but here is a beautiful special case. An elementary (that is, first-order definable) class of frames **K** is definable by a set of modal formulas iff **K** itself is closed under (1) generated subframes, (2) disjoint unions, and (of course) (3) *p*–morphic images, while the complement class $^{c}\mathbf{K}$ is closed under (4) 'ultrafilter extensions'. The first proof of this depended on Birkhoff's Theorem in universal algebra – the first purely model-theoretic proof (via saturated models) is in J. van Benthem 1993, 'Modal Frame Classes Revisited', *Fundamenta Informaticae* 18: 2/3/4, 307-317.

<u>Modal Models and Bisimulations</u>

My dissertation *Modal Correspondence Theory* (Mathematical Institute, University of Amsterdam, 1976 – published in expanded form as *Modal Logic and Classical Logic*, Bibliopolis, Napoli, 1983) contains what I believe to be the first occurrence of bisimulation. Overall, this work follows the frame trend, but it also considers modal models on their own (as a base for frame theory), and it asks what semantic invariance would be characteristic for modally definable classes of modal *models.* The natural translation from modal formulas to first-order formulas over models was known, and hence, the latter question is easily answered if we can only determine which first-order formulas are definable by modal ones. The answer requires generalization of (directed) *p*–morphisms between modal frames to a symmetric relation between models, and I defined '*p–relations*' (an awful name, enjoying a well-deserved oblivion) to that end. These are relations between worlds in two models which only connect worlds satisfying the same propositional atoms, and obeying the (nowadays) familiar bisimulation zigzag conditions for R-successors. I was thinking of *p*–relations as total relations between rooted models, and then used generated submodels to switch between arbitrary models and rooted ones in the usual way. Then, my main result was this. *A first-order formula (in the appropriate similarity type) is definable by a (translated) modal formula iff it is invariant for p–relations and generated submodels.* In modern jargon, the latter states *invariance for bisimulations*! The heart of the proof is a Lemma stating that two models **M**, x and **N**, y satisfy the same modal formulas (in x and y) iff they have elementary extensions $\mathbf{M}^{+}$, $\mathbf{N}^{+}$ that admit of a bisimulation between x and y . As a special case, this shows that finite models have the same modal theory iff they bisimulate – a result rediscovered around 1985 by Hennessy & Milner. My results were stated for a language with just one modality and its accessibility relation, but it was well-known around the time that extension to the polymodal case with many relations is entirely routine.

Newer Developments

Around 1990, I became interested in these matters again, partly by having heard about the work of Park and Hennessy & Milner. My recent book *Exploring Logical Dynamics* (CSLI Publications, Stanford, 1996) contains many subsequent developments, of which I mention a few. (**i**) Many different proofs have been found for the 'Modal Invariance Theorem' by now, including techniques like elementary chains, saturated models, and Ehrenfeucht games. In particular, Eric Rosen proved in 1995 that the result also holds in *finite model theory*. I suspect that more generally, unlike with full first-order logic, most of modal model theory is robust under the transition from ordinary model theory to finite model theory. (**ii**) One can vary the expressive power of modal languages, and then modify the matching 'simulations' so that the Invariance Theorem remains true. Here is a small example: a first-order formula is invariant under *p*–relations only ('total bisimulations') iff it can be defined using ordinary modal operators plus the 'universal modality' expressing truth "in all worlds". A broad investigation of this interaction is Maarten de Rijke's dissertation *Extending Modal Logic*, ILLC, Amsterdam 1993. Also of interest are studies of 'non-Boolean' languages, with non-symmetric simulations of rather new flavours (cf. Natasha Kurtonina's dissertation *Frames and Labels. A Modal Analysis of Categorial Deduction*, ILLC & OTS, Amsterdam & Utrecht 1995). Even so, we still do not understand the route 'from languages to simulations' in full generality. (**iii**) In computer science, the route has been the reverse. One studies processes via labeled transition systems (i.e., polymodal Kripke models) under various notions of simulation, and then asks for logical languages matching these. Formal outcomes are often the same, though! Comparisons between the two routes are in J. van Benthem & J. Bergstra, 'Logic of Transition Systems', *Journal of Logic, Language & Information* 3:4, 1995, 247–283. Also relevant is Marco Hollenberg's forthcoming dissertation (Utrecht, philosophy, 1997). (**iv**) One can also go upward to *infinitary* languages, starting from the folklore observation that two models **M**, x, **N**, y admit a bisimulation iff they have the same modal theory allowing infinitary conjunctions and disjunctions. The modal and computational traditions are merged in a non-well-founded set theory in J. Barwise & L. Moss, 1996, *Vicious Circles. On the Mathematics of Non-Well-Founded Phenomena*, CSLI Publications, Stanford. A related proposal is the reanalysis of modal invariance theorems as infinitary 'generalized interpolation theorems' found in J. Barwise & J. van Benthem, 'Interpolation, Preservation, and Pebble Games' (Report ML–1996–12, ILLC, Amsterdam). (**v**) Finally, the bisimulation analysis of 'modal statements' may be extended to 'modal programs', introducing *safety for bisimulation*. What one gets are (more or less) the regular operations plus an appropriate negation. (See my paper 'Programming Operations that are Safe for Bisimulation', Report 1993-

179, CSLI, Stanford. To appear in *Studia Logica*). This I view as the program core of dynamic logic, playing the same role as the usual core repertoire of propositional logic. Generalizations of this approach, covering most standard operations (also parallel ones) of Process Algebra, are in Marco Hollenberg 1996, 'Bisimulation Respecting First-Order Operations', Logic Group Preprint Series 156, Institute for Philosophy, Utrecht.

Where to Go From Here

I am interested in merges of modal logic, non-well-founded set theory, and brands of process algebra, because I think these all have the same flavour and aims. Over the past few years, our Dutch environment has organized some events to this effect, such as the two workshops documented in J. van Eijck & A. Visser, eds., 1994, *Dynamic Logic and Information Flow*, MIT Press, Cambridge (Mass.), and in A. Ponse, M. de Rijke & Y. Venema, eds., 1995, *Modal Logic and Process Algebra,* CSLI Lecture Notes, Stanford. But there is much more pre-established harmony, as one can see, e.g., in Rob van Glabbeek's work at Stanford. (Cf. R. van Glabbeek, 1990, 'The Linear Time – Branching Time Spectrum', CONCUR '90, Lecture Notes in Computer Science 458, Springer, Berlin, 278-297 – and R. van Glabbeek & G. Plotkin, 1995, 'Configuration Structures', Department of Computer Science, Stanford University. E.g., Rob independently discovered directed simulations for non-Boolean languages, in his case, for intuitionistic logic.) I even suspect that existing *category-theoretic* approaches to programming constructs are after essentially the same things, and have comparable results (cf. Albert Thijs' dissertation "Simulation and Fixpoint Semantics", computer science, Groningen, 1995). It would be nice to get yet more confluence in this field.

2 Another Bridge between Bisimulation and Elementary Equivalence

Modal logic resembles first-order logic, despite being much simpler combinatorially. To understand these analogies, one needs systematic 'bridges'. We use a new one here.

Consider the key result relating labeled transition systems to poly–modal formulas:

Modal Invariance Theorem For first-order formulas $\phi(x)$ the following are equivalent

(i) $\phi(x)$ is invariant under bisimulations

(ii) $\phi(x)$ is definable by a modal formula.

A key proof step for the MIT (cf. ELD, Chapter 4) replaces the 'linguistic' relationship of 'modal equivalence' between two Kripke models by a 'structural' one of bisimulation, among elementarily equivalent models (satisfying the same first-order sentences). Thus, we can pass back-and-forth between bisimulation and modal equivalence:

<u>First Switching Lemma</u> For rooted models $\mathbf{M}, x$, $\mathbf{N}, y$, the following are equivalent

(i) $\mathbf{M}, x$ and $\mathbf{N}, y$ satisfy the same modal formulas

(ii) $\mathbf{M}, x$ and $\mathbf{N}, y$ have elementary extensions $\mathbf{M}^+, x$ and $\mathbf{N}^+, y$, respectively, which bisimulate (with x connected to y).

In a picture, this observation gives us the following square of related notions:

$\mathbf{M}, x$	*modal equivalence*	$\mathbf{N}, y$
first-order equivalence		*first-order equivalence*
$\mathbf{M}^+, x$	*bisimulation*	$\mathbf{N}^+, y$

De Rijke 1993 uses walks through this diagram for a systematic comparison between modal and first-order logic. But other 'Gestalt switches' occur, too. One 'boosts' modal equivalence to first-order equivalence (Andréka, van Benthem & Németi 1996):

<u>Second Switching Lemma</u> For rooted models $\mathbf{M}, x$, $\mathbf{N}, y$, the following are equivalent

(i) $\mathbf{M}, x$ and $\mathbf{N}, y$ satisfy the same modal formulas

(ii) $\mathbf{M}, x$ and $\mathbf{N}, y$ have bisimilar models $\mathbf{M}^*, x$ and $\mathbf{N}^*, y$, respectively, which are elementarily (i.e., first-order) equivalent.

This time, the picture has turned around – allowing us different back-and-forth trips:

$\mathbf{M}, x$	*modal equivalence*	$\mathbf{N}, y$
bisimulation		*bisimulation*
$\mathbf{M}^*, x$	*first-order equivalence*	$\mathbf{N}^*, y$

One new application of this schema is the following alternative route toward the MIT.

<u>A Quick New Proof of the Modal Invariance Theorem</u>

Let $\phi(x)$ be a first-order formula which is invariant for bisimulation, and define $mod(\phi)$ to be the set of all modal consequences of ϕ. We show that $mod(\phi) \models \phi$, from which fact a modal equivalent for ϕ follows by Compactness (namely, as the conjunction of some finite subset of $mod(\phi)$). So, let $\mathbf{M}, x \models mod(\phi)$. By standard reasoning, the full modal type of $\mathbf{M}, x$ together with $\phi(x)$ is finitely satisfiable. Compactness then gives a model $\mathbf{N}, y$ for ϕ which is modally equivalent to $\mathbf{M}, x$. Now consider the two models $\mathbf{M}^*, x$, $\mathbf{N}^*, y$ given by clause (ii) in the Second Switching Lemma. As $\mathbf{N}^*, y$ is bisimilar to $\mathbf{N}, y$, ϕ holds there (by its bisimulation invariance). Hence, ϕ (being first-order) holds in the elementarily equivalent model $\mathbf{M}^*, x$, too, and thus also $\mathbf{M}, x \models \phi$ (again by ϕ's bisimulation invariance). ■

Coda   Generalized Translation

Analyzing the proof of the Second Switching Lemma more precisely (it employs Ehrenfeucht games with invariants that can be stated in a modal logic over trees), we can find out more. Here we take our cue from a result by Janin & Walukiewicz 1996 relating formulas from a monadic second-order logic over trees to formulas in the so-called modal 'μ–calculus'. The models $\mathbf{M}^*, \mathbf{N}^*$ are 'tree unravelings' of Kripke models or LTSs, with additional duplication of nodes (just for technical reasons). Now, let an *extended modal formula* be any formula constructed using Booleans plus ordinary modal operators, as well as the 'universal modality' expressing truth "in all worlds".

Fact   There exists an effective translation taking first-order formulas ϕ to extended modal formulas $\mu(\phi)$ such that, for all models $\mathbf{M}$, x and their duplicated tree unravelings $\mathbf{M}^*$, x, $\mathbf{M}, x \models \mu(\phi)$ iff $\mathbf{M}^*, x \models \phi$.

For a more precise formulation and a genuine proof of this result, see Hollenberg 1997. (Incidentally, for any two unraveled trees, modal equivalence in their roots implies equivalence with respect to extended modal formulas.) The Fact suggests an intriguing generalization of 'logical translation'. The MIT presupposes the well-known translation taking modal formulas to first-order ones, on the class of all LTSs. There is no effective converse translation, however – since this would reduce first-order logic (undecidable) to modal logic (decidable). But the Fact shows how we can open up the game, widening the relevant notion of translation to allow equivalences across *different* models.

3 Extending the Guarded Fragment to Betweenness and Pair Arrows

In modal logic, as in many other areas, there is always an option of either studying proposed systems as such, or translating them back into fragments of first-order logic, and then look at their properties in a standard light. A powerful part of first-order logic serving this purpose is the so-called 'Guarded Fragment'. We shall extend this here.

The Guarded Fragment of first-order logic generalizes many modal languages, allowing all quantifications of the form $\exists \mathbf{y}\ (Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y}))$, where the atom $Q\mathbf{xy}$ is the 'guard'. Here, variables in the finite sequences $\mathbf{x}, \mathbf{y}$ may occur in any multiplicity and order. The main result in Andréka, van Benthem & Németi 1996 (cf. ELD, chapter 4) says

Theorem   Universal validity in the Guarded Fragment (GF) is decidable.

Under the obvious first-order translations for their semantic truth conditions, this result explains and extends the decidability of a large class of standard modal languages, from basic modal and tense logic to even the polyadic version of first-order CRS.

Proof We recall the basic steps. Any satisfiable GF-formula ϕ has a finite 'quasi-model', of 'types' consisting of subformulas of ϕ, of some effectively computable size, which also conversely generates a model for ϕ. Thus, whether a guarded formula is satisfiable is equivalent to its having a finite quasi-model – a decidable property.

From Standard Models to Finite Quasi-Models Suppose that formula ϕ is satisfiable in standard model $\mathbf{M}$. Let V be the set of variables occurring in ϕ (free or bound). Henceforth, we restrict attention to the finite set Sub_ϕ consisting of ϕ and all its subformulas, closed under simultaneous substitutions using only variables in V, that do not change syntactic forms. (This is feasible, by the cited references.) Each variable assignment verifies a 'type' Δ of finitely many formulas from this set. Our quasi-model has a universe consisting of the finitely many types realized in $\mathbf{M}$. In this structure, for each guarded formula $\exists\mathbf{y}\ (Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y})) \in \Delta$, there exists a type Δ' with (i) $Q\mathbf{xy}$, $\psi(\mathbf{x}, \mathbf{y}) \in \Delta'$, (ii) Δ, Δ' agree on all 'unaffected' formulas with only free variables in $\mathbf{x}$.

Definition (i) Let F denote the finite set of all guarded formulas of length $\leq |\phi|$ that use only variables from V. Note that $\phi \in F$ and F is closed under taking subformulas and 'alphabetic variants'. (ii) An F-*type* is a subset Δ of F for which we have

(a)	$\neg\psi \in \Delta$	iff	*not* $\psi \in \Delta$	whenever $\neg\psi \in F$
(b)	$\psi\wedge\xi \in \Delta$	iff	$\psi \in \Delta$ *and* $\xi \in \Delta$	whenever $\psi\wedge\xi \in F$
(c)	$[\mathbf{u}/\mathbf{y}]\psi$	implies	$\exists y\ \psi \in \Delta$	whenever $\exists y\ \psi \in F$

$[\mathbf{u}/\mathbf{y}]\psi$ comes from ψ by replacing each free variable in $\mathbf{y}$ with the corresponding variable in $\mathbf{u}$, simultaneously. (iii) Let $\mathbf{y}$ be a sequence of variables, and Δ, Δ' types. Write $\Delta =_{\mathbf{y}} \Delta'$ if Δ, Δ' have the same formulas with free variables disjoint from $\mathbf{y}$. (iv) A *quasimodel* is a set of F–types S such that, for each $\Delta \in S$ and each guarded formula $\exists\mathbf{y}\ (Q\mathbf{xy} \wedge \psi) \in \Delta$, there is a type $\Delta' \in S$ with $Q\mathbf{xy}$ and $\psi(\mathbf{x}, \mathbf{y})$ in Δ' and $\Delta =_{\mathbf{y}} \Delta'$. We say that ϕ *holds in a quasi-model* if $\phi \in \Delta$ for some Δ in this model. ■

Clearly, if ϕ is satisfied by some model, then ϕ also holds in some quasi-model.

From Quasi-Models to Standard Models From any quasi-model $\mathbf{M}$, we can define a standard model $\mathbf{N}$. Call π a *path* if $\pi = \langle \Delta_1, \phi_1, ..., \Delta_n, \phi_n, \Delta_{n+1} \rangle$ where Δ_1, Δ_{n+1} are types in $\mathbf{M}$, each formula ϕ_i is of the form $\exists\mathbf{y}\ (Q\mathbf{xy} \wedge \psi) \in \Delta_i$ and Δ_{i+1} is an alternative type as described above (i.e., $Q\mathbf{xy}$, $\psi(\mathbf{x}, \mathbf{y})$ in Δ_{i+1} and $\Delta_{i+1} =_{\mathbf{y}} \Delta_i$). We say that the variables in $\mathbf{y}$ *changed their values* from Δ_i to Δ_{i+1} (the others did not). Finally, variable z is called *new in path* π if either $|\pi| = 1$ or z's value was changed at the last round in π . *Objects* in $\mathbf{N}$ are all pairs (π, z) with π a path, z new in π . Next, we *interpret predicates* over these objects. I(Q) holds of the sequence of objects $<(\pi_j, x_j)>_{j\in J}$ iff the paths π_j fit into one linear sequence under inclusion, with a maximal path π^* such that (i) the atom $Q<x_j>_{j\in J} \in \Delta^*$ (the last type on π^*) and for no (π_j, x_j) does x_j change its value on the further path to the end of π^* . Finally, we define an *assignment* s_π for each path. We set $s_\pi(x) =_{def} (\pi', x)$ with π' the unique subpath of π^* at whose end x was new, while it remained unchanged afterwards.

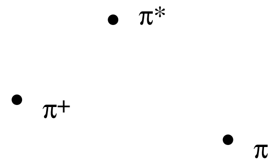
The correctness of this model construction shows at *last*(π), the last type on the path π :

<u>Truth Lemma</u> For all paths π in $\mathbf{N}$, and all formulas $\psi \in F$,

$$\mathbf{N}, s_\pi \models \psi \quad \text{iff} \quad \psi \in \mathrm{last}(\pi) .$$

<u>Proof</u> Induction on ψ . ***Boolean cases*** are immediate, by the closure conditions for $\neg$ and $\wedge$ on types. ***Atoms***: involve a straightforward calculation, via the linearity condition in the interpretation function I, plus the '$=_{\mathbf{y}}$ -clause' in quasi-models ensuring transfer of 'unaffected formulas' along paths. For later reference, we repeat the full argument for bounded ***Existential Quantifiers*** $\exists \mathbf{y}\,(Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y}))$. (i) First, suppose that $\exists \mathbf{y}\,(Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y})) \in \mathrm{last}(\pi)$. Then there is an extended path $\pi^+ =_{def} \pi$ *concatenated with* $< \exists \mathbf{y}\,(Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y})), \Delta' >$, where Δ' is a successor type for Δ chosen as above with $Q\mathbf{xy}, \psi(\mathbf{x}, \mathbf{y}) \in \Delta'$ (satisfying the transfer condition for unaffected formulas with free variables $\mathbf{x}$). All objects (π^+, y_i) with y_i in $\mathbf{y}$ are new here. By definition, the atomic guard I(Q) holds for the object tuples $s_{\pi+}(\mathbf{y})$, $s_{\pi+}(\mathbf{x})$ ($= s_\pi(\mathbf{x})$) . Also, by the inductive hypothesis, $\mathbf{N}, s_{\pi+} \models \psi(\mathbf{x}, \mathbf{y})$. Therefore, $\mathbf{N}, s_{\pi+} \models \exists \mathbf{y}\,(Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y}))$. By $\mathbf{x}$-invariance in the standard model $\mathbf{N}$, then, indeed $\mathbf{N}, s_\pi \models \exists \mathbf{y}\,(Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y}))$. (ii) Conversely, suppose that $\mathbf{N}, s_\pi \models \exists \mathbf{y}\,(Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y}))$. By the truth definition, there are objects $d_i = (\pi_i, u_i)$ with $\mathbf{N}, s_\pi{}^{\mathbf{y}}{}_{\mathbf{d}} \models Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y})$. (Here, $s_\pi{}^{\mathbf{y}}{}_{\mathbf{d}}$ is the assignment which is like s_π except for setting all y_i to d_i .) In particular, I(Q) holds of the objects $s_\pi(\mathbf{x})$, $\mathbf{d}_i$. This leads to a picture of forking paths. The $s_\pi(\mathbf{x})$ were all introduced by stage π^* inside π , and then the $\mathbf{d}_i$ were (either interpolated, or) added to form a maximal sequence π^+ with the atom $Q\mathbf{xy}$ true at the end. The fork is such that $\mathbf{x}$-values do not change any more from π^* onward, whether toward π or π^+ . (This is the only case where the atomic guard on our quantifiers comes in essentially.) We now analyse this situation a bit more carefully:

•



Now, the variables u_i do not have to be the y_i . Say, π^+ has $s_{\pi+}(u_i) = (\pi_i, u_i) = d_i$. Thus, the assignments $s_\pi{}^{\mathbf{y}}{}_{\mathbf{d}}$ and $s_{\pi+}$ agree on $\mathbf{x}$, and for all $y_i \in \mathbf{y}$ we have $s_\pi{}^{\mathbf{y}}{}_{\mathbf{d}}(y_i) = d_i = s_{\pi+}(u_i)$. Then, by $\mathbf{N}$, $s_\pi{}^{\mathbf{y}}{}_{\mathbf{d}} \models Q\mathbf{xy} \wedge \psi$ and the above observations, we have $\mathbf{N}$, $s_{\pi+} \models [\mathbf{u}/\mathbf{y}]Q\mathbf{xy}$, $\mathbf{N}$, $s_{\pi+} \models [\mathbf{u}/\mathbf{y}]\psi$. By the inductive hypothesis, $[\mathbf{u}/\mathbf{y}]\psi \in \text{last}(\pi^+)$. Also, from the initial description of π^+ , we see at once that $[\mathbf{u}/\mathbf{y}]Q\mathbf{xy} \in \text{last}(\pi^+)$ (by the interpretation of atomic predicates). By closure conditions (b), (c) for types, one gets $\exists \mathbf{y}\, (Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y})) \in \text{last}(\pi^+)$. Finally, since no changes in $\mathbf{x}$-values occurred on the fork from π^* , the transfer condition for unaffected formulas along successor types along paths ensures that this same formula is in last(π) . ■

Thus having a quasi-model implies having a real model, and the Theorem is proved. ■

The decidability of GF explains that of many other systems, from basic modal logic to CRS (predicate logic over 'generalized assignment models'), which can be effectively translated into it. But some natural decidable modal logics remain beyond its scope.

<u>Example 1</u> *Pair Arrow Logic*, i.e., relational algebra over arbitrary top relations (not just full Cartesian squares). Here, the GF strategy would use ternary guards $Uxyz$ for a composition, whereas pair arrow models in fact have the binary relativization $R \circ S =_{def} \lambda xy \bullet \exists z\, ((Uxz \wedge Uzy) \wedge Rxy \wedge Szy)$, with a composite guard $Uxy \wedge Uyz$.

<u>Example 2</u> *Temporal Logic*. E.g., the well-known UNTIL AB says $\exists y\, (x<y \wedge Ay \wedge \forall z\, ((x<z \wedge z<y) \rightarrow Bz))$. Its "betweenness" clause has a composite guard $x<z \wedge z<y$.

The point here cannot be that arbitrary conjunctions of atoms are acceptable guards. For, the latter can express undecidable logics. An known example is CRS plus the 'Patchwork Property' for glueing compatible available assignments into new ones. Given this warning, here is the proper generalization covering both the above examples. We call a quantification *loosely guarded* if it has the following format:

$$\exists \mathbf{y}\, (\,\&Q\mathbf{xy} \wedge \psi(\mathbf{x}, \mathbf{y}))$$

where $\&Q\mathbf{xy}$ is a conjunction of atoms with free variables $\mathbf{y}, \mathbf{x}$ in which every variable y in $\mathbf{y}$ co-occurs with every other variable in $\mathbf{y} \cup \mathbf{x}$ in at least one of the listed atoms, conjoined with a matrix formula $\psi(\mathbf{x}, \mathbf{y})$ from the Loosely Guarded Fragment.

Single atomic guards exemplify this, and so does the above $x<z \wedge z<y$ (with z in the role of $\mathbf{y}$). A typical non-example is transitivity $\forall y_1y_2y_3\, ((y_1<y_2 \wedge y_2<y_3) \rightarrow y_1<y_3)$, without co-occurrence of y_1, y_3 in a guard atom. The Patchwork Property is similar.

Theorem The Loosely Guarded Fragment is decidable.

Proof We analyse the above representation argument. The definition of quasi-models carries over without major changes, as does their representation via 'path models'. Here, we now allow path extensions via the new generalized form of bounded quantification. Again, the crucial result is the Truth Lemma, saying that guarded formulas hold under the assignment induced by a path iff they occur in the last set encoded in that path. The step from right to left here is as before. Thus, the key is a combinatoric aspect of the converse direction, whose main step was illustrated in the above picture. The argument for true existential formulas still works with a conjunction of atomic guards like above. We look at the maximal position π^* as before. For each new variable y , again given the truth condition for atomic statements, loose guardedness requires that the path of the new y-value fits linearly with the original path on which the x-values occurred. Therefore, it either lies on the latter, or it extends it starting from π^* . Moreover, the condition also applies to all new values y amongst each other - and hence, these form at worst some linear path π^+ extending π^*, up to some maximal node where the highest new y-value has been introduced. The rest of the argument is as before, since all relevant y-atoms hold at π^+, and no y-values change in going back towards π^*. Cases of mere interpolation of the new y-values on the old path π are merely simpler. (Here, we heavily use the constancy of relevant variable values in an atom along the path up to the highest variable mentioned. This requires some checking of cases.) ■

Is this result the best that we can do? Here is a new challenge. Consider Pair Arrow Models with a *polyadic composition* $R \circ S \circ T$ different from iterated binary composition. This employs clauses $\exists y_1y_2\, (Uxy_1 \wedge Uy_1y_2 \wedge Uy_2z \wedge Rxy_1 \wedge Sy_1y_2 \wedge Ty_2z)$, that are not loosely guarded. Test question then: is this polyadic arrow logic still decidable?

But there are other interesting open questions concerning the Guarded Fragment as a classical mirror of modal logic. For instance, modal and dynamic logic can be extended with arbitrary *fixed-point operators* $\mu p \bullet \phi(p)$ (where p occurs only positively in ϕ) to obtain the earlier-mentioned μ–*calculus*, which remains decidable. Likewise, does the Guarded Fragment remain decidable when we add first-order fixed-point operators?

4 Continuity as a Constraint on Program Operations

What are natural program operations? Formats in the literature often involve semantic invariances, such as 'safety for bisimulation'. But there are also semantic requirements of 'computability'. We discuss the syntactic repertoire of operations induced by one of these, viz. the requirement of Continuity, as distributivity over unions of finite sets.

Continuity of a function expresses 'computability' for its values from finite information about arguments. An abstract formulation of this is Scott's *Finite Distributivity* (FD): $F(X) = \cup \{ F(X_0) \mid X_0 \subseteq X \text{ finite}\}$. This implies that F is upward monotone in its argument, while all fixed points arising in this way emerge after ω iteration steps. What syntactic definitions of functions guarantee this pleasant behaviour? ELD, Chapter 11 gives a syntactic preservation theorem for first-order definable functions F given as first-order formulas $\phi(P)$ with a predicate letter P of the required arity.

<u>Theorem</u> A first-order formula $\phi(P)$ defines a finite–distributive operation on P iff ϕ is definable from (i) P–atoms and (ii) arbitrary P-free formulas, using only conjunction, disjunction and existential quantification.

<u>Proof</u> That all given syntactic forms define FD operations, follows by induction. Conversely, by Finite Distributivity, formula $\phi(P)$ implies the countable disjunction of all formulas of the form $\exists \mathbf{x}_1 \ldots \exists \mathbf{x}_k (P\mathbf{x}_1 \wedge \ldots \wedge P\mathbf{x}_k \wedge [\lambda u \bullet u{=}\mathbf{x}_1 \vee \ldots \vee u{=}\mathbf{x}_k / P]\, \phi)$. (Here the $\mathbf{x}$ are tuples of variables, appropriate to the arity of the relevant P-atoms.) Hence, by the Compactness Theorem, ϕ implies some finite subdisjunction δ of the latter. Moreover, by the monotonicity of ϕ , each disjunct of δ also implies ϕ . Thus, δ is the required definition, which indeed satisfies the given syntactic constraints. ■

This argument hinges on a substitution trick involving identity. In many practical settings, however, these would not be naturally available in our format of definition. Can we do without them? Our new observation here is that we can.

<u>Theorem</u> The preceding preservation result for first-order finite-distributive definitions $\phi(P)$ also holds in a predicate-logical language without identity.

<u>Proof</u> Consider all models $\mathbf{M}$ of $\phi(P)$ where P consists of some finite relation – over some finite subdomain $d_1, \ldots, d_n$ of distinct objects. By Finite Distributivity, each model for ϕ contains such a model, obtained by shrinking P to some finite subset of its original denotation. For any such $\mathbf{M}$, let $\underline{T}(\mathbf{M}, \phi)$ be the complete P-free-type of $d_1, \ldots, d_n$ (that is, all P-free formulas, in some fixed set of free variables $x_1, \ldots, x_n$, which are true of $d_1, \ldots, d_n$ in $\mathbf{M}$) together with a direct transcription of all true atomic

P-statements among $d_1, ..., d_n$. First, we observe that this information implies ϕ – by a routine argument, paying some extra attention to the syntactic form of our formulas:

Claim $\underline{T}(\mathbf{M}, \phi) \models \phi$

Proof Let $\mathbf{N}$ be any model satisfying $\underline{T}(\mathbf{M}, \phi)$ in objects $e_1, ..., e_n$. The latter need not be all distinct, as we have no identity statements in the language enforcing this. Now, let $\mathbf{N'}$ be obtained from $\mathbf{N}$ by shrinking the interpretation of P to just its tuples among the e–objects. Moreover, construct $\mathbf{M'}$ from $\mathbf{M}$ by (possibly) extending the denotation of P so that the match (d_i, e_i) $(1 \le i \le n)$ becomes a strong homomorphism. As ϕ is monotone, it still holds in $\mathbf{M'}$. Now, take new constants for each object in $\mathbf{N'}$ distinct from all e_i . Since $\mathbf{N'}$ satisfies $\underline{T}(\mathbf{M}, \phi)$, its P-free theory (without identity) is finitely satisfiable in the model $\mathbf{M'}$. Therefore, by a standard Compactness argument, there exists an elementary extension $\mathbf{K}$ of $\mathbf{M'}$ (in the P-free language *with* identity) which also verifies the P-free theory of $\mathbf{N'}$ (without identity). Now, copy the $\mathbf{M'}$–interpretation of P into $\mathbf{K}$, to obtain an expanded model $\mathbf{K'}$. Since P is definable in $\mathbf{M'}$ with identity and finitely many parameters, $\mathbf{K'}$ is an elementary extension for the full language of ϕ, and hence, this formula holds in $\mathbf{K'}$. Finally, the obvious match between interpreted $\mathbf{N'}$–constants and objects assigned to the above variables $x_1, ..., x_n$ is a strong homomorphism between $\mathbf{N'}$, $\mathbf{K'}$ for the identity-free language including P (even though it need not be injective either way). Therefore, we also get ϕ true in $\mathbf{N'}$. But then, by upward monotonicity plus the definition of $\mathbf{N'}$, ϕ is also true in $\mathbf{N}$. ■

The remaining argument is similar to the above. From the Claim, by the Compactness Theorem, the conjunction Δ of some finite subset of $\underline{T}(\mathbf{M}, \phi)$ implies ϕ . Choose such a formula Δ for each case, and take its existential closure with respect to the variables $x_1, .., x_k$. By the above construction, ϕ implies the disjunction of all these formulas, and once again by Compactness, it implies some finite subdisjunction of these. But then, since all the disjuncts implied ϕ , we have the required definition. ■

Finite Distributivity Requires Horn Clause Definitions

These results provide a useful normal form for first-order finite-distributive operations. For instance, it is precisely the format found in so-called 'completed logic programs'. To see this, let $\phi(P, x)$ define a finite-continuous unary operation on an argument P . Its fixed points satisfy the equivalence $\forall x\,(Px \leftrightarrow \phi(P, x))$. Via our syntactic format, the right-hand side becomes a finite disjunction of existentially quantified conjunctions α of P-atoms and P-free formulas. These conjuncts form a finite set of (possibly P-recursive) Horn clauses $\alpha(x) \rightarrow Px$ – where we drop the existential quantifiers as usual.

Thus, one interpretation of the above results is that they justify the *Horn clause format* of logic programming as provably the broadest one which guarantees ω–fixed-points.

Enumerating Admissible Infinitary Forms

The above argument gets simpler for the *infinitary* first-order language $L_{\infty\omega}$. For then, we can transcribe Finite Distributivity directly into a countable disjunction of cases, with which the formula ϕ (P) must then be equivalent. (A syntactic characterization for the identity-free infinitary language remains open.) An interesting special case are formulas ϕ (R, S, ..., x, y) defining k–ary program operations on binary relations. Successive elimination of the binary predicates R, S, ..., using our existential prefixes $\exists \mathbf{x}_1 \ldots \exists \mathbf{x}_k (R\mathbf{x}_1 \wedge \ldots \wedge R\mathbf{x}_k \wedge$ plus the above identity substitutions, yields a normal form stating the existence of some finite set of objects, with a number of specified binary relational links, followed by a pure identity formula concerning these objects. In $L_{\infty\omega}$, the latter can only state identities and non-identities among our objects, plus a cardinality statement via a countable Boolean combination of forms "there exist k objects", with k running from zero to 'infinity'. Disregarding the latter (which may be filtered out by further semantic requirements on definitions), we get a simple format for defining finite-distributive operations. They are countable disjunctions of descriptions of finite 'transition graphs' between the input argument x and the output argument y . This includes the usual operations of relational algebra (union, converse, composition), but also Boolean intersection, or more involved cases with branching transition graphs. What we do not get is a *functional completeness* theorem with only finitely many operations. By the Immermann Ehrenfeucht-plus-pebble technique (ELD, Chapter 5), no finite set of relational operators captures all the above finite-distributive operations.

Zooming in on the Regular Operations

The preceding concerns meet in the regular program operations $;\ \cup\ ^*$. These are both finite-distributive and *safe for bisimulation.* The functional completeness theorem of Section 7 below says all infinitary safe operations are definable by 'strong negation', composition, and infinitary union. Finite Distributivity rules out the negations. So, is their combination charcateristic for the regular operations? The answer is negative, as FD still admits arbitrary unions. Therefore, the two requirements combined still allow countable unions over *uncountably* many non-iterative patterns. (The definition might prescribe x-to-y R-sequences of lengths taken from any set of natural numbers.) Regular operations only refer to some finite *linear* route from input to output argument – and this is much more restrictive than the Distributivity investigated in this Section.

5 Resolution in Dynamic Logic as Task Calculus

Dynamic logic is a general logic of action, not just a program calculus. In particular, implications between standard correctness assertions function as a simple task calculus. We identify the latter as a simple fragment of modal logic, and discuss its peculiarities.

Hoare Calculus is usually presented as a system for proving correctness of programs, or more interactively, for developing correct programs. But computations are just one kind of action, and correctness assertions $\{A\}S\{B\}$ may just as well be read as descriptions of some available routine S that will produce effects described by postcondition B given resources described by precondition A. Our more general planning task does not consist in proving isolated correctness statements. It is rather one of logical derivation. Given a number of routines $\{A\}S\{B\}$, how can we put together some combination of them performing some new task, from a given precondition to a given postcondition? Such a more general 'calculus of tasks' (ELD, chapter 11) is a common interpretation of propositional dynamic logic. It only involves a small fragment of the latter system, however. We can take the conditions to be Boolean, and the given actions to be atomic. So our question is, what is a complete subsystem for planning derivations?

<u>Resolution and Monotonicity</u>

One natural analogy is with propositional *resolution*. We can normalize task statements – using the valid rules of Disjunction of Antecedents and Conjunction of Consequents, to conjunctions of universal 'action clauses' of the form $A \rightarrow_S B$, where A is a conjunction of literals, S is a program expression, and B is a disjunction of literals. What we need is a suitable style of reasoning on these clauses. Now, resolution is really a form of *Monotonicity*, a very general logical inference allowing insertion of suitable formulas in syntactically 'positive' positions. For instance, consider $\neg A \vee B$, $A \vee C$. The former says that A implies B. Therefore, we may substitute B for A in the positive occurrence of A in the second disjunction, to get the usual resolvent $B \vee C$. This is the 'upward' view. Alternatively, we can use a 'downward monotonic' inference where $\neg C$ implied A, substituting $\neg C$ for the negative occurrence of A in the first clause. With labeled action clauses $A \rightarrow_S B$, however, some complications arise. (**1**) First, consider analogues of standard propositional inferences. Let $A \rightarrow_S B$, $B \rightarrow_T C$. We want to conclude $A \rightarrow_{S;T} C$. But what is the precise mechanism producing the right programs in these conclusions? (**2**) Next, take two action premises $A \rightarrow_S B \vee C$, $B \wedge D \rightarrow_T E$. Given that the actions separate the Boolean atoms, is there some obvious format for an evident conclusion at all? Instead of giving a close analogue to resolution, we make a simple proposal based on 'plan trees' describing actions with conditions.

Plan Implications

Let us replace the above correctness statements $A \rightarrow_S B$ by Boolean implications of the form $P_S A \rightarrow B$ – or more generally, by 'plan implications'

$$\Pi \rightarrow B$$

where Π describes the succesful execution of some actions from given resources, using existential modalities $P_S A$ looking *backward* into the past of the current state. (Cf. the folklore observation that strongest postconditions are best expressed in a backward looking 'temporal' logic). In general, Π describes a finite tree of previous atomic actions, with literals true at its nodes. Thus, it may be constructed using only *literals, conjunctions and indexed program modalities* P_S. The conclusion B may be a *disjunction* of literals. One further elucidation is needed. As usual in Hoare Calculus, we read the premises as universally quantified, over all available states in our model. The above examples become (**1**) $P_S\, A \rightarrow B$, $P_T\, B \rightarrow C$, with conclusion $P_T\, P_S\, A \rightarrow C$ by downward Monotonicity. The passage to one complex program $P_{S;T}\, A \rightarrow C$ will come later. (**2**) From $P_S\, A \rightarrow B \vee C$, $P_T\, (B \wedge D) \rightarrow E$, downward Monotonicity yields $P_T\, (\neg\, C \wedge P_S\, A \wedge D) \rightarrow E$. This may be 'linearized' to $P_{\,(A)?\,;\,S\,;\,(\neg\, C\, \wedge\, D)?\;\,;\,T} \rightarrow E$.

Tree Calculus

We now present a very simple *Tree Calculus* which justifies these inferences. Given premises of the above form, plus a tree formula Π^*, apply the following three rules. In general, starting from $\{\Pi^*\}$, these will lead to the formation of a finite set of tree (formula)s $\{\Pi_1, ..., \Pi_k\}$, to be viewed as a *disjunction* of possible cases:

I If the tree for some premise $\Pi \rightarrow B$ 'fits inside' some tree Π_i, at any node position, then we may write B at that node.

II If a tree has a disjunction D at a node, we may replace it by a disjunction of trees with the successive D-literals at that node.

III If a contradiction occurs at a node, remove the tree.

A set of trees implies a disjunction B if B follows from the literals at each root. We revisit the above examples to demonstrate how this works. In particular, we show what is meant by 'fitting inside' – leaving further formal details to those inclined that way.

(**1**) Start: $\{P_T\, P_S\, A\}$

I: $\{P_T\, (P_S\, A \wedge B)\}$

I: $\{P_T\, (P_S\, A \wedge B) \wedge C\}$

The literal C at the root implies the desired conclusion.

(**2**) Start: $\{P_T(\neg C \wedge P_S A \wedge D)\}$.

I: $\{P_T(\neg C \wedge P_S A \wedge (B \vee C) \wedge D)\}$

II: $\{P_T(\neg C \wedge P_S A \wedge B \wedge D), P_T(\neg C \wedge P_S A \wedge C \wedge D)\}$

III: $\{P_T(\neg C \wedge P_S A \wedge B \wedge D)\}$

I: $\{E \wedge P_T(\neg C \wedge P_S A \wedge B \wedge D)\}$

The desired conclusion E follows from inspection of the root.

<u>Theorem</u> The Tree Calculus is complete for our task inference.

<u>Proof</u> Starting with the set $\{\Pi\}$ for the conclusion $\Pi \rightarrow B$, perform all possible inferences allowed by the calculus, using the given premises to perform substitutions. Moreover, remove trees which are subtrees of other ones. (These are implied anyway.) This process will stop after finitely many steps. Note in particular, that it only produces trees richer than the original one – which therefore imply it, in an obvious semantic sense. Now, suppose some tree Π_i in the resulting set has root literals whose conjunction does not imply B. Π_i gives a countermodel to the implication as follows. Take this tree *itself* as a model, with only the atomic relations described, and only those atomic propositions true at each node which were explicitly indicated at that position. Evidently, B fails at the root. But, each premise is true at every node in this model. For, if its antecedent tree is true at a node, then it 'fits' inside Π_i (this is because of the special form of the corresponding modal formulas), and hence, it would have given rise to a further I-move adding literals. In general, this will be a disjunction, whence a further II-move was applied, yielding trees with extra literals (as compared with Π_i). Not all of these can have been removed by III-moves, or Π_i would not have survived into the final set. But the other situations are impossible, too, as Π_i would then have been removed for not being maximal. The outcome must be that no antecedent of a premise is true at any node in our model – and hence all premises hold vacuously. ■

A complete calculus of task inference comes as no surprise. Inference between plan implications is *decidable*, even with premises read universally (cf. ELD, Chapter 7, Theorem 10). Our analysis leads to several further open questions of logical interest.

<u>Program Operations for Hoare-Style Conclusions</u>

First, is there a standard procedure for *linearizing* statements $\Pi \rightarrow B$ into more standard correctness assertions $A \rightarrow_S B$, of course, for suitable complex programs S? The matter is not entirely clear-cut. In particular, branching patterns in a tree may call for *parallel* program operators, going beyond dynamic logic. E.g., premises $A \rightarrow_S B$, $C \rightarrow_T D$ naturally suggest a conclusion $A \wedge C \rightarrow_U B \wedge D$ for some new program U. One valid option for this purpose is Boolean intersection $S \cap T$. But we can also use other parallel operators. The tree transcription of our premises suggests a conclusion $(P_S A \wedge P_T C) \to B \wedge D$, whose linearisation might read $\underline{\text{true}} \to_{((A)?\,;\,S)\,||\,((B)?\,;\,T)} C$. A third option would employ new n–ary modalities directly over tree-like structures (cf. Hollenberg 1996B for examples), which also support parallel program operations. We leave the proper design of a suitably expressive repertoire of program operations for our task calculus as an open question here. But even without such a program repertoire, *trees themselves* may be just as convenient representations of plans.

Synthesizing Plans

The Tree Calculus also helps in synthesizing plans out of premise routines. This time, we only have 'resource propositions' **A** and a 'goal' G , and the desired plan is a tree with leaves from **A** only which implies G . One procedure is to enumerate all possible resource-to-goal implications from the given premises (with accompanying plan trees). A finite upper bound to the number of these derived implications can be determined in advance (since it only depends on the proposition letters occurring in the problem). Then, we solve the standard propositional search problem from **A** to G using these derived implications. The associated plan with intermediate actions indicated arises from successive leaf substitution of trees for auxiliary implications.

Example

Let the resource proposition be A and the goal G . The available action premises are $P_S B \wedge C \to G$, $P_T B \to C$, $P_U A \to B$. We derive G from A as follows:

1	G from B, C
2	B from A
3	C from B
4	B from A

The associated trees will work out to (via their above normal form descriptions):

1	$P_S B \wedge C$
2	$P_S P_U A \wedge C$
3	$P_S P_U A \wedge P_T B$
4	$P_S P_U A \wedge P_T P_U A$ ■

Less blindly, we would need a search procedure providing guidance. And indeed, the preceding example is reminiscent of a logic programming derivation. Here we need a translated first-order version of our plan implications, in the standard modal fashion. Consider the earlier Example (**1**). Take first-order clause forms for its two premises:

$Ax \wedge Sxy \rightarrow By$ and $Bx \wedge Txy \rightarrow Cy$. From an assumption Au , the standard search procedure for a proof of the goal Cv will produce an outcome $Sus \wedge Tsv$ – whose quantified version $\exists s\,(Sus \wedge Tsv)$ is exactly the definition of program composition proposed earlier. The preceding example may be analyzed in a similar manner through its first-order transcriptions, trying to get Gv from instances of Au using the clauses

$$Bx \wedge Sxy \wedge Cy \rightarrow Gy \qquad Bx \wedge Txy \rightarrow Cy \qquad Ax \wedge Uxy \rightarrow By$$

Thus, standard proof search via first-order transcriptions may produce useable answers.

Another angle on this problem of synthesis is one of 'propositional completeness'. Note first that all valid consequences between plan implications reduce to valid propositional inferences by disregarding all action operators P_S . (The reason is simply that these consequences must also hold on models where all atomic relations coincide with the identity relation.) Conversely, consider any valid propositional inference from a set of implicational clauses to one implicational clause $D \rightarrow E$. Now, assume that the premise clauses all carry an action S producing their consequent from their antecedent.

Question Is there always a plan implication $\Pi \rightarrow E$ for a valid conclusion whose antecedent Π only employs conditions that occur in D ?

A positive answer expresses a kind of functional completeness for the programming repertoire encoded in our Tree Calculus. We proceed to discuss a case of plan inference where additional expressive power seems needed.

<u>Incorporating Negations and Converse</u>

The obvious dynamic version of the propositional law of Contraposition

$$A \rightarrow B \models \neg B \rightarrow \neg A$$

is the inference from

$$\text{from } P_S A \rightarrow B \quad \text{to} \quad P_{S^\vee} \neg B \rightarrow \neg A ,$$

involving a *relational converse* $S^\vee$. Contraposed once more, this implication reflects the well-known tense-logical inference from $P\,A \rightarrow B$ to $A \rightarrow G\,B$. This example shows that we need plan trees which also allow converse arrows, going to successors, rather than predecessors in the atomic relations. It may be checked that the above rules remain complete. E.g., dynamic contraposition remains derivable in this fashion.

6 Dynamic Logic over Sequences as Path Geometry

Dynamic Logic interprets programs as binary input–output relations between states. A richer semantics should employ complete finite traces of succesful computations. We explore the resulting dynamic logic of states and computation sequences – which naturally extends into a more general arrow–logic style geometry of points and paths.

The usual interpretation of Propositional Dynamic Logic in labeled transition systems $\mathbf{M}$ is a mutual recursion on $\mathbf{M}, s \models \phi$ (formula ϕ is true at s) and $\mathbf{M}, s_1, s_2 \models \pi$ (program π has a succesful execution starting from s_1 and ending in s_2). Thus, programs are interpreted as binary input–output relations, without the intermediate computation traces. But the later are surely the more intuitive interpretation of program execution. Accordingly, we can formulate a new truth definition $\mathbf{M}, \sigma \models \pi$, where σ is any sequence of states – so that programs now express properties of computations. Let $\sigma_1 \bullet \sigma_2$ be the result of concatenating two sequences, identifying the end of σ_1 with the start of σ_2 . (Unlike ordinary concatenation, this operation is only partial.)

$\mathbf{M}, \sigma \models a$	iff	$\sigma \in V(a)$
$\mathbf{M}, \sigma \models \pi_1 \,;\, \pi_2$	iff	$\sigma = \sigma_1 \bullet \sigma_2$ with $\mathbf{M}, \sigma_i \models \pi_i$ ($i = 1, 2$)
$\mathbf{M}, \sigma \models \pi_1 \cup \pi_2$	iff	$\mathbf{M}, \sigma \models \pi_1$ or $\mathbf{M}, \sigma \models \pi_2$
$\mathbf{M}, \sigma \models \pi^*$	iff	σ is a finite $\bullet$–concatenation of finite sequences satisfying π in $\mathbf{M}$
$\mathbf{M}, \sigma \models (\phi)\mathbf{?}$	iff	σ is a one-element sequence $\langle s\rangle$ such that $\mathbf{M}, s \models \phi$

The clauses for the statement part are as usual, with the following modality:

$\mathbf{M}, s \models \langle\pi\rangle\phi$	iff	there exists a sequence σ with $\mathbf{M}, \sigma \models \pi$ and endpoint s such that $\mathbf{M}, s \models \phi$

Our first observation is that this reinterpretation does not change the logic.

<u>Theorem</u> The PDL language interpreted over finite sequences has the same logic as standard PDL interpreted over binary transition relations.

<u>Proof</u> One easily checks that all principles in the well-known complete axiomatization of PDL are valid on the new sequence interpretation. For the converse direction, suppose that some formula fails in a binary standard model $\mathbf{M}$. We construct a sequence model $\mathbf{M}^{seq}$ as follows. The states remain the same, and we interpret each atomic relation a as the set of two-element sequences $\sigma = (s, t)$ such that $R_a st$. Then a straightforward simultaneous induction proves the following:

$\mathbf{M}, s \models \phi$	iff	$\mathbf{M}^{seq}, s \models \phi$
if $\mathbf{M}^{seq}, \sigma \models \pi$	then	$\mathbf{M}$, begin(σ), end(σ) $\models \pi$
if $\mathbf{M}, s_1, s_2 \models \pi$	then	there is a sequence σ with begin (σ) = s_1, end(σ) = s_2 , and $\mathbf{M}^{seq}, \sigma \models \pi$

(Note the analogy with the safety analysis of ELD, Chapter 5.) As a consequence, counter-examples to validity on standard models transfer to sequence models. ■

Language Extensions

This harmony changes when we take advantage of the richer structure of sequence models to interpret more expressive formalisms. For instance, sequences also support other operations, including standard (total) concatenation, juxtaposing the end of the first sequence with the beginning of the second. This would correspond to a new form of program composition – more like π_1 **; 1 ;** π_2 , where **1** is an arbitrary move. (Thus, the logic will encode a part of elementary syntax.) For more finely detailed properties of computations, a natural extension is the usual *temporal logic* of "Since" and "Until", which allows us to talk about what went on in between the input state and output state.

Open Question Axiomatize temporal PDL over sequence models.

Finally, we can also add Booleans to enrich the program class. (Németi 1991 shows the resulting algebraic structure is problematic – but the move seems natural from a logical point of view.). As with ordinary PDL, this move increases complexity.

Fact PDL over sequences with all Boolean operations is undecidable.

Proof We can embed full Relational Algebra as in standard PDL. The reason is that we can define the relevant algebraic operations by singling out the two-element sequences through the following definition (with id $=_{def}$ (true)**?**)

$$\neg \text{ id} \wedge \neg (\neg \text{ id} \mathbin{\mathbf{;}} \neg \text{ id})$$ ■

Remark (Infinite Sequences) In addition to changing the language, one can also change the ontology still further. For instance, Edsger Dijkstra's notion of 'total correctness' for a program π says that, starting from some state satisfying a given precondition, all execution sequences for π terminate, in a final state satisfying the given postcondition. This excludes infinite computation sequences, which then have to be semantic objects.

Arrow Logic Strategies

In order to restore decidability of PDL with all Booleans, we can follow the arrow logic strategy (ELD, Chapter 8) and work with models restricted to some set of 'admissible sequences'. The above truth definition can then be relativized in an obvious manner.

Open Question Prove decidability for relativized sequence PDL, and axiomatize it.

In particular, this system loses Associativity for composition. This reduces the power of other principles, such as induction. The latter effects already show in the arrow version of PDL (ELD, Chapter 8). *Dynamic Arrow Logic* was intended as an abstract version of binary relation algebra, but a more general interpretation for its semantics and valid laws suggests itself. States are 'points', and 'arrows' are abstract *paths* containing these. Thus, DAL is also an abstract theory of information states and computation paths. It would be of interest to investigate the deductive power of this system more practically.

<u>Example (Induction Principles)</u>
For its program iteration, DAL has the two axioms (i) $\pi \rightarrow \pi^*$, (ii) $\pi^* ; \pi^* \rightarrow \pi^*$, plus the induction rule (iii) *if* $\vdash \pi \rightarrow \alpha$ *and* $\vdash \alpha ; \alpha \rightarrow \alpha$, *then* $\vdash \pi^* \rightarrow \alpha$. These derive at least the rule form of standard PDL induction $(\phi \wedge [\pi^*] (\phi \rightarrow [\pi]\phi)) \rightarrow [\pi^*]\phi$ – using $B\phi$ $(E\phi)$ for " ϕ is true at the beginning (end)" :

$\vdash \pi \rightarrow (B\phi \rightarrow E\phi)$	(given)
$\vdash (B\phi \rightarrow E\phi) ; (B\phi \rightarrow E\phi) \rightarrow (B\phi \rightarrow E\phi)$	(derivable in Arrow Logic)
$\vdash \pi^* \rightarrow (B\phi \rightarrow E\phi)$	(DAL induction) ■

Induction reflects the finiteness of paths in our models. More technically, the semantic content of our theory would be clarified by a *representation theorem* for abstract DAL models in terms of relativized sequence models – in the style of Marx 1995.

Finally, a sweeping reinterpretation of all the above is as a form of *Modal Geometry* of points and paths. In the underlying abtract spaces, we have three basic notions:

point s lies on path π *s is the beginning of π* *s is the end of π*

These binary relations induce six forward and backward modalities (plus, of course, richer temporal and first-order languages). This leads to a new kind of geometry, where segments and lines need not be 'straight'. Much of its elementary first-order theory is decidable, as it translates into the Guarded Fragment – with the above three relations as atomic guards. Explicit axiomatizations will provide new modal geometries.

7 The Narrative Flow of Time

In temporal narrative, subsequent utterances build up a consecutive picture of events that occurred and states that obtained. This picture is obtained dynamically through the application of recurrent discourse rules (cf. Kamp & Reyle 1993, Ter Meulen 1995). E.g., successive past tenses introduce a linear sequence of events (the 'consecutio temporum' of traditional linguistics), while the ubiquitous connective "and" often means "and next". This dynamic semantics (cf. ELD, chapters 2, 12) has interesting features. We discuss these in connection with 'dynamic aspect trees' (DATs, Ter Meulen 1995). The same issues would arise in connection with Hans Kamp's more widely used DRTs.

Two Strategies of Dynamification

ELD, chapter 2, follows one particular strategy of dynamification for standard logics. Formulas are reinterpreted as evaluation or update procedures, involving state changes over standard models. Thus, as in the DPL treatment of anaphora, no separate level of syntactic representation is needed. But discourse representations as in DRT provide an alternative strategy. Its dynamics involves construction of successive syntactic (or mental) states, whose relation to standard models remains static. It is instructive to compare the two for the same logical system. Consider propositional temporal logic, with operators $F\phi$ (' at least once in the future') and $P\phi$ ('at least once in the past'). On the first strategy, its dynamic semantic involves transitions between points in time:

$\mathbf{M}, t_1, t_2 \models \phi$ iff there exists a succesful evaluation of ϕ starting from t_1 and ending in t_2

The temporal operators F, P can be read as existential quantifiers, denoting forward or backward moves along the temporal order. For instance, $\mathbf{M}, t_1, t_2 \models F$ iff $t_1 < t_2$. The resulting system is a dynamified version of (a bounded fragment of) first-order logic. Van Benthem 1995 shows how it can be translated into standard temporal logic, by a simple recursive definition of pre- and postconditions for these evaluations. (This is one case where dynamification does not increase complexity.) The second strategy would rather turn sequences of formulas into (descriptions of) 'small models', which are then to be related to real models as to their 'truth'. We shall see this at work with DATs. When stated at this level of generality, there may be no essential difference between the two dynamic strategies. With enough freedom in the definition of a 'model', one can incorporate representations into new-fangled denotations, and hence the second strategy is contained in the first. The converse route also seems feasible, turning computation traces into syntactic objects. An abstract mathematical equivalence seems plausible.

DATs in a Nutshell

Processing narrative discourse can be viewed as stepwise construction of tree patterns. Here is a simplified sketch. Each DAT is a finite graph, with nodes standing for temporal intervals. Nodes can carry propositional information, and stand in relations of precedence and inclusion to other nodes. We designate one node as the 'active' one (in the full system, there are more such roles). Verb tenses modify the current DAT. E.g., an event reported in the past tense PAST ϕ leads to attachment of a new active node to the right of the current active one, where ϕ is written. An auxiliary PERF ϕ leads to attachment of a new node to the left of the currently active one, with ϕ written on it. (Here, the old active node remains the active one.) Finally, a progressive PROGR ϕ creates a new node above the currently active one, with ϕ written on it (again, no shift in active node). Temporal adverbs ("always ϕ") involve propagation rules spreading propositional information around a DAT. Other rules spread information, too. Thus, PERF ϕ labels carry over to nodes to the right, while PROGR ϕ carries over to nodes underneath. This algorithm provides a dynamic semantics for temporal discourse, with update conditions modifying DATs, now viewed as constructive information states.

In a more standard semantics, DATs can now be related to temporal *interval models* $(I, <, \subseteq, V)$ via an obvious notion of 'embedding' sending nodes to intervals, and preserving all stated relationships, as well as the propositional information recorded. Together, all succesful embeddings for a DAT encode its classical truth-conditional content. Moreover, we may now define new styles of *dynamic inference*. For instance, say that conclusion ψ *follows from* premises $\phi_1, .., \phi_k$ if each succesful embedding of the DAT for the discourse $\phi_1, .., \phi_k$ validates ψ (viewed as an ordinary statement). Other such notions can be defined along the lines of ELD, chapter 7. Seligman & ter Meulen 1995 discuss logical features of this paradigm. Here, we add a few thoughts.

Connections with Temporal Interval Logic

As in ELD, chapter 2, a new dynamic system like this may be analysed by familiar logical techniques. For instance, the DAT constructions are reminiscent of standard temporal logic. What would be a temporal formalism expressive enough to capture the truth-conditional content of the above? For a start, the constructions given so far require only future and past $F\phi$, $P\phi$, as well as a progressive operator $\Pi\phi$ stating that ϕ holds in at least one superinterval of the current one. It is easy then to describe temporal formulas with the right meaning for each successive DAT (see below). As a result, one can explain most of the above spreading rules. For instance, this temporal logic will validate the rightward spread of statements $P\phi$ for perfect tense (by transitivity of $<$), as well as downward spread of progressive tense statements $\Pi\phi$ (by transitivity of $\subseteq$). Moreover, the 'monotonicity law' for intervals ($\forall xyz\, ((x<y \wedge z\subseteq y) \rightarrow x<z)$) implies

downward transfer of Perfect statements as well – another law of DAT construction. Other transfer rules in DATs have no such general structural temporal background, but reflect the lexical semantics of specific aspectual classes. E.g., we must have downward transfer of 'state propositions'. Temporal interval logic can also be used as an aspectual calculus handling the latter cases, through suitable axioms (van Benthem 1995).

Richer DAT systems need further temporal operators. An example is a construction putting two successive intervals under one current node ("and next"). This requires a binary modality $\phi \& \psi$ true at an interval if it has a subinterval satisfying ϕ preceding another subinterval satisfying ψ. Also, with further distinguished nodes present in DATs ('speech time', 'reference time', etcetera), the format of embeddings changes, and we will need a many-dimensional temporal logic keeping track of these. (Marx & Venema 1995 is an up-to-date treatment of many-dimensional temporal logic.) There are questions of explicit axiomatization for such DAT-induced temporal logics. Here, we only note that one of our earlier techniques is applicable, too. All we have said can be translated into the obvious first-order language over temporal interval models. But then, we can measure the complexity of the system by means of the resulting forms of quantification, using the earlier Guarded Fragment (section 3 above).

<u>Proposition</u> The temporal interval logic of P, F, Π and $\&$ is decidable.

<u>Proof</u> It suffices to show that the translations of the above operators all land up in the 'loosely guarded' extension of the Guarded Fragment (cf. Section 3 above). This is obvious for the first three operators, whose quantifier forms are guarded:

$$P\phi \quad \exists y\,(y<x \wedge \phi(y)) \qquad F\phi \quad \exists y\,(x<y \wedge \phi(y)) \qquad \Pi\phi \quad \exists y\,(x\subseteq y \wedge \phi(y))$$

For $\phi \& \psi$ we have the loosely guarded $\exists yz\,(y\subseteq x \wedge z\subseteq x \wedge y<z \wedge \phi(y) \wedge \psi(z))$. (These truth conditions stay loosely guarded, even with additional requirement found in DATs. For instance, progressive is taken to require that the superinterval y starts before x. This says that $\exists u\,(u\subseteq y \wedge u<x)$, which does not endanger loose guardedness.) ■

What this still leaves open is decidability of these languages over temporal interval models satisfying additional restrictions of transitivity and monotonicity.

<u>Structural Rules</u>

The above dynamic inference over DATs may be studied as an abstract reasoning style, just as in ELD, chapter 7. Then, we find that none of the usual structural rules are valid, not even in plausibly modified dynamic versions (such as those for Update-to-Test). For Permutation, Contraction, or Monotonicity this is clear, as the flow of temporal narrative will not tolerate such changes in 'the story'. But one might want to have at least some version of Reflexivity or Cut. This may involve changing valid inference after all, or having special structural rules for special types of temporal statement only. E.g., merely adding propositional information without temporal side effects will be an admissible form of Monotonicity. We leave these matters open here, and conclude with a more concrete, though highly simplified, logical calulus for DAT-like reasoning.

<u>A Simplified Logic of Tree Modification</u>

The main moves in the above can be viewed as rules for constructing LTSs by adding new nodes, and annotating existing ones with atomic propositions. Moreover, there were shifts in 'perspective', as the distinguished node of the LTS is allowed to wander. From a logical point of view, the most elegant instruction set is as follows.

Move 1	write p on the distinguished node
Move 2	adjoin an outgoing a-arrow to the distinguished node with a new node at the end, where we write p
Move 3	the same as Move 2, but making the new node the distinguished one
Move 4	adjoin an incoming a-arrow to the distinguished node with a new node at the beginning, where we write p
Move 5	the same as Move 4, but making the new node the distinguished one

Together, these moves build any directed acyclic graph with propositional annotations. It is easy to describe this process via transformation of modal graph formulas τ :

Move 1	go to $\tau \wedge p$
Move 2	go to $\tau \wedge \langle a \rangle p$
Move 3	go to $p \wedge \langle a^{\vee} \rangle \tau$
Move 4	go to $\tau \wedge \langle a^{\vee} \rangle p$
Move 5	go to $p \wedge \langle a^{\vee} \rangle \tau$

It is easy to describe the generally valid inferences associated with this tree calculus, in a standard modal logic with relations and their converse. Alternatively, we can redescribe these construction processes via binary transition relations on larger LTSs (cf. ELD, chapter 10). The format is $\mathbf{M}, s_1, s_2 \models \text{DAT}$ iff s_2 is a result of performing the instructions encoded in DAT, starting from s_1 . Thus, we have an analogy with the representation-free dynamics of our introduction after all. We leave its extent, and its general moral, for further investigation. Our claim is merely that representation-based dynamic formalisms can be profitably viewed as part of the broader ELD framework.

8 Characterizing Safety in $L_{\infty\omega}$

The modal characterization of assertions invariant for bisimulation has a counterpart in a description of all program operations that are safe for bisimulation. By a technique from Barwise & van Benthem 1996, both results can be lifted to infinitary logic, which is the language of choice for many process operations, as well as non-well-founded set theory.

The analysis of assertions in the Modal Invariance Theorem extends to programs in *dynamic logic*. Consider the following notion of invariance for program operations:

Definition An operation $O(R_1, ..., R_n)$ on programs is *safe for bisimulation* if, whenever C is a relation of bisimulation between two models for their transition relations $R_1, ..., R_n$, then it is also a bisimulation for the defined relation $O(R_1, ..., R_n)$.

It is easy to show that the regular operations of relational composition **;** and choice $\cup$ (Boolean union) have this property, and so do test relations (ϕ)**?** for modal formulas ϕ. Typically non-safe operations are program intersection and Boolean complement. But the following negation operation is safe: $\sim(R) = \{ (x, y) \mid x=y$ *and for no* $z : x R z \}$. All these operations are first-order definable in an obvious language over LTSs. Indeed, we have this counterpart to the above Modal Invariance Theorem (ELD, chapter 5):

Modal Safety Theorem A first-order operation $O(R_1, ..., R_n)$ is safe for bisimulation iff it can be defined using atomic relations R_axy and atomic tests (q)**?** for propositional atoms q in our models, using the three operations **;** , $\sim$ and $\cup$.

This result expresses functional completeness for dynamic counterparts of the Boolean primitives $\wedge$, $\neg$, $\vee$. New proofs are in Hollenberg 1995 giving safety over much broader notions of process equivalence. (Hollenberg 1996 extends MST to monadic second-order logic, following Janin & Walukiewicz 1996.) Now, it is natural to seek *infinitary* versions of MIT and MST. The usual regular program operations include Kleene iteration – and many further natural programming constructs are infinitary. Barwise & Moss 1996 show how infinitary modal logic ties in with non-well-founded set theory, and the first-order logic of bisimulation. So, consider an infinitary first-order language over possible worlds models with arbitrary set conjunctions and disjunctions. The infinitary modal language extends the basic one likewise. Clearly, infinitary modal formulas are invariant for bisimulation: infinitary conjunctions and disjunctions fall within the obvious inductive argument. What is more, we also have the converse result, even though its first-order proofs based on compactness and saturation fail for $L_{\infty\omega}$.

<u>Theorem</u> An infinitary first-order formula is invariant for bisimulations iff it is definable by an infinitary modal formula.

<u>Proof</u> One proof of this result is in ELD Chapter 10, using modified 'consistency families' to circumvent compactness. Another proof is in Barwise & van Benthem 1996. We will use techniques from the latter to also extend the Safety Theorem to $L_{\infty\omega}$. Therefore, we give a brief sketch of the relevant argument. It involves crucial use of the following remnant of compactness retained by the infinitary language:

<u>Boundedness Theorem</u> Let $\psi(<)$ be a formula of $L_{\infty\omega}$ with models whose domains can be well-orders $<$ of any size. Then ψ has a model where $<$ is no well-order.

Now, suppose that ϕ is invariant for bisimulation. We prove that

\# There exists an ordinal κ such that for all models **M**, s |= ϕ and all models **M'**, s' having the same modal theory as **M**, s *up to modal operator depth* κ , **M'**, s' |= ϕ

Modal operator depth is measured in the usual way. (Through infinitary combinations, it can run up to arbitrarily high ordinals.) The crucial property of this notion is this (compare the similar results for Ehrenfeucht games in standard logic, Doets 1996):

<u>Lemma</u> Two models **M**, s, **M'**, s' share the same modal theory up to depth κ iff there exists a descending chain of sets of 'partial bisimulations' between them of length κ , with zigzag conditions holding downward from levels $\beta+1$ to β .

From **#**, modal definability of ϕ follows easily. Consider the set (!) of all complete modal descriptions up to depth κ of all models for ϕ . Then ϕ is equivalent to the disjunction of all of these. (That it follows from each disjunct is the main content of **#**.)

<u>Proof of #</u> Suppose that for each ordinal κ , there are models **M**, s and **M'**, s' with (i) **M**, s |= ϕ , (ii) **M**, s and **M'**, s' have the same modal theory up to depth κ , but (iii) not **M'**, s' |= ϕ . By the above lemma, **M** and **M'** have a descending 'κ–tower' of partial bisimulations. Now, this situation may be coded up by an infinitary first-order formula $\Phi(<)$. (This trick comes from a well-known proof of Lindström's Theorem.) Using fresh predicate letters A, B, C^k, I, <, one states that $(\phi)^A$, $(\neg\phi)^B$, while C^k i **x y** is a (1+2k)-ary predicate defining a partial bisimulation of size k between matched members in the sequences **x**, **y** . Here, the variable i runs over an index set I linearly ordered by $<$, and we can also state the key zigzag properties. E.g., if C^k (i+1) **x y** and Au, $R_a(\mathbf{x})_j u$, then there exists v with Bv, $R_a(\mathbf{y})_j v$ such that C^{k+1} i **x**u **y**v . Now, the Boundedness Theorem says that $\Phi(<)$ has a model in which $<$ is not a well-order.

That model, must have at least one countably descending chain of indices. Collecting all finite partial bisimulations along its stages, we get a true bisimulation, without a bound on its zigzag properties. But then, we have two models A, B connected by a bisimulation which disagree on ϕ : which refutes invariance for bisimulation. ■

By similar reasoning, we now derive our main result.

<u>Theorem</u> A relational operation $O(R_1, ..., R_n)$ in $L_{\infty\omega}$ is safe for bisimulation iff it can be defined using atomic relations $R_a xy$ plus atomic tests (q)**?** , using only three operations **;** , $\cup$ and **~** , where the unions may now be infinitary.

<u>Proof</u> We recall the proof for the finitary first-order case (ELD, chapter 5), identifying the part where a new route is needed. The outermost argument remains the same, up to an important module. **I** For a start, specifying the relevant languages, it is clear that, if a relational operation defined by π (x, y) is safe for L-bisimulations, then the $L_{\infty\omega}$–formula $\exists y\ (\pi\ (x, y) \wedge Qy)$ is invariant for (L+Q)-bisimulations, where Q is a new unary predicate letter. But then, by the infinitary Modal Invariance Theorem, there is an equivalent infinitary modal formula $\phi(q)$. **II** Due to the simple occurrence of Q , the latter has a strong semantic property. Call ϕ (q) *continuous* in the proposition letter q if the following equivalence holds in each model (with some benign abuse of notation):

for each family of subsets $\{P_i\}_{i\in I}$, $\phi\ (\cup_{i\in I} P_i) \leftrightarrow \bigvee_{i\in I} \phi\ (P_i)$

From right to left, this is the well-known *monotonicity* whose syntactic correlate is obligatory positive occurrence for q – but the other half excludes a lot more. We want a syntactic preservation theorem for continuous modal formulas. This can be done – and the resulting normal forms are described in the main theorem below. **III** From these forms, one can extract the following explicit information. Any safe relation π (x, y) may be defined as an infinitary union of finite sequential compositions of successive atomic actions $R_a xy$ plus tests (α)**?** for some infinitary modal formulas α . **IV** Finally, the latter tests unpack to combinations of atomic tests by the valid equivalences

$(\bigvee_{i\in I} \phi_i)? = \cup_{i\in I} (\phi_i)?$ $\quad (\neg\phi)? = \sim (\phi)?$ $\quad (\langle a\rangle\phi)? = \sim\sim (a\ ;\ (\phi)?)$ ■

At this point, we prove an independent model-theoretic preservation theorem.

<u>Theorem</u> Up to logical equivalence, the q–continuous infinitary modal formulas ϕ (q) are just those that can be written as infinitary disjunctions of formulas of 'existential forms α_0 ™ q , α_0 ™ $\langle a_1\rangle$ (α_1 ™ q), α_0 ™ $\langle a_1\rangle$ (α_1 ™ $\langle a_2\rangle$ (α_2™ q)), etcetera, where all formulas α_i are q–free.

<u>Proof</u> All forms described are evidently continuous w.r.t. the proposition letter q . The hard part is the converse. Let us first analyse the models **M**, s where a continuous formula $\phi(q)$ holds. The denotation of q can be written as a union of singletons, and so, by continuity, ϕ will hold with q true in only one world t . (In case the denotation of q is empty, monotonicity wil keep it true for any singleton denotation {t} of q .) Moreover, we may assume that this single q–world lies at some finite successor distance from s , since we also have ϕ true at the submodel generated from the root. Thus, there is some finite sequence $s=s_1, ..., s_n=t$. Call a model **M'**, s' a κ–*relative of* **M**, s if it has a corresponding sequence $s_1', ..., s_n'$ leading to a q–world $t'=s_n'$, such that matched worlds s_i, s_i' satisfy the same infinitary q–free modal formulas up to operator depth κ . (Henceforth, we will refer to the relevant vocabulary as language L .) We prove this

<u>Lemma</u> There exists an ordinal κ such that, if **M**, s |= ϕ and **M'**, s' is a κ–relative of **M**, s , then **M'**, s' |= ϕ .

From this, the required definition for ϕ arises as a disjunction of all modal descriptions up to depth κ of finite q–paths in models **M**, s for ϕ as described just now. (This is a set, because of the restriction to fixed modal depth.) Clearly, ϕ implies this disjunction. But also conversely, whenever some disjunct holds, we are in a model which is a κ–relative of some such **M**, s , and the Lemma tells us that ϕ must hold.

<u>Proof of Lemma</u> The argument starts like in the earlier proof of the infinitary Modal Invariance Theorem. Assume that, for each ordinal κ , there are models **M**, s |= ϕ and **M'**, s' with κ–corresponding finite branches as above, such that ϕ fails in **M'**, s' . Now, code up this situation in one infinitary formula Ψ(A, B, **x**, **y**, C^k, I, <) which describes, in particular, the existence of a <–descending sequence (along the index set I) of partial L-bisimulations with the simulation sending the $(\mathbf{x})_i$ to the $(\mathbf{y})_i$ at the top. Moreover, we can state that in A, there is just one q–world. This formula then has models with well-orders of arbitrarily high cardinality for < . By the Boundedness Theorem, it must have a model where < is not a well-order. Using a countable descending chain of indices as before, such a model yields the following situation:

- a model **M**, s |= ϕ with finite action sequence $s=s_1, ..., s_n=t$ to its only q–world t
- a model **M'**, s' where ϕ fails, with action sequence $s_1', ..., s_n'$ to q–world $t'=s_n'$
- an L–bisimulation C between **M** and **M'** with $s_i \, C \, s_i'$ $(1 \le i \le n)$.

The remainder of the argument is as for the finitary Safety Theorem (ELD, chapter 5). Given a situation like this, using successive simple (L+q)–*bisimulation-preserving* moves of copying subtrees and re-attachment of nodes, one can *unravel* the original models **M** and **M'** to obtain the above situation with the following extra:

> the links between corresponding nodes in the distinguished branches are unique: these nodes do not attach to any others.

Then consider the model $\mathbf{N}^*$ which is **M'** with one difference: q is true only in t' . Clearly, our L-bisimulation is even an (L+q)–bisimulation between **M**, s and $\mathbf{N}^*$, s' . Then we can argue as follows. Since the modal formula $\phi(q)$ holds at **M**, s , it also holds at $\mathbf{N}^*$, s' . But then, by monotonicity, it also holds at **N'**, s' (whose denotation for q can only be larger). But this refutes the given failure of ϕ at **N'**, s' (which was unaffected by our (L+q)–bisimulation-preserving tree surgery). A contradiction. ■

Finally, from the syntactic description in the preservation theorem for continuity, one easily extracts the stated normal form for operations that are safe for bisimulation. ■

9 A Henkin Proof for Infinitary Generalized Interpolation

There exists a more traditional proof of generalized interpolation in infinitary logic which suggests a new ternary format for Gentzen sequents, keeping track of relevant 'transition vocabulary', that may work for logics lacking ordinary complete proof calculi.

Barwise & van Benthem 1996 propose a generalization of the Craig Interpolation Theorem which also applies to infinitary first-order logic, as well as other logical formalisms which lack the standard version of interpolation. (Examples where this works are finite-variable fragments of first-order logic.) Their general strategy is the replacement of ordinary consequence by a more general notion of consequence A|=B along an arbitrary model relation R : whenever **M** |= A and **M** R **N**, then **N** |= B. An important case has R as 'potential L-isomorphism': the existence of a family of finite partial L-isomorphisms between **M**, **N** with the usual back and forth properties. We state the main result here, and provide a new more traditional Henkin-style proof, derived from an earlier one in ELD, chapter 10, for the Modal Invariance Theorem. It avoids the Boundedness Theorem (while using notions from the cited paper). This proof is more laborious – but in return, it provides suggestive additional information.

<u>Theorem</u> For $L_{\infty\omega}$–formulas $\phi(x)$, $\psi(x)$, the following are equivalent:

(i) there is an $\alpha \in L_\phi \cap L_\psi$ such that $\phi \models \alpha \models \psi$

(ii) ϕ implies ψ along potential $L_\phi \cap L_\psi$–isomorphism.

<u>Proof</u> From (i) to (ii), this is an immediate consequence of the fact that potential isomorphism in a similarity type L preserves truth of the corresponding L–formulas. For the direction from (ii) to (i), assume that ϕ, ψ have no interpolant in $L = L_\phi \cap L_\psi$. We are going to construct a counterexample to (ii), using *'good triples'* (E, Σ, Δ), where the idea is that Σ describes a model for ϕ over some domain of constants A, Δ one for $\neg \psi$ over constants B, and E a potential L–isomorphism between A, B, all 'in statu nascendi'. We start with some preliminaries. First, set $\mu =_{def} \max(\aleph_0, |\text{subformulas}(\phi)|, |\text{subformulas}(\psi)|)$. Next, choose two disjoint sets of constants A, B of size μ^+, the first regular cardinal greater than μ. For convenience, in what follows, we shall be working with formulas in *normal form*, constructed from atoms and their negations using both quantifiers, as well as arbitrary set conjunctions and disjunctions. Moreover, throughout, formulas will only contain a finite number of constants.

<u>Definition</u> A *good triple* (E, Σ, Δ) satisfies the following requirements:

(1) E is a set of tuples $\mathbf{a}, \mathbf{b}$ $(\mathbf{a} \subseteq A, \mathbf{b} \subseteq B)$ with length $(\mathbf{a})$ = length $(\mathbf{b})$

(2) Σ is a set of subformulas of ϕ made into sentences by plugging in constants from A; and likewise for Δ w.r.t. subformulas of $\neg\psi$ and constants from B

(3) $|E|, |\Sigma|, |\Delta|$ are all smaller than μ^+

(4) Σ, Δ are *L-inseparable via* E . That is, there is no set $\mathbf{a}^i, \mathbf{b}^i$ of tuples in E , each with a corresponding L-formula $\beta(\mathbf{x}^i)$, such that for some infinitary $\vee, \wedge$–combination α of the formulas $\beta(\mathbf{x}^i)$, (i) $\Sigma \models \alpha\ [\mathbf{x}^i{:=}\mathbf{a}^i]$, while (ii) $\Delta \models \neg \alpha\ [\mathbf{x}^i{:=}\mathbf{b}^i]$

<u>Note</u> These $\vee, \wedge$–combinations genuinely extend $L_{\infty\omega}$, but they are still invariant under potential L–isomorphism, in an obvious sense. (Allowing existential quantifiers over infinite combinations, like in $\exists x \wedge_n Rxa_n$, would give problems with invariance.)

<u>Fact</u> Choose any starting tuple $\mathbf{a}, \mathbf{b}$ for the free variables of ϕ, $\neg \psi$.
Then $(\{\langle\mathbf{a}, \mathbf{b}\rangle\}, \{\phi(\mathbf{a})\}, \{\neg \psi(\mathbf{b})\})$ is a good triple.

<u>Proof</u> The only non-trivial property to be checked is Non-Separation. But the above strong formulation reduces to the usual inseparability given by the negation of clause (ii) in our Theorem, in this special case where we only have one tuple $\mathbf{a}, \mathbf{b}$ in E . ■

We check a bunch of extension principles for Σ (those for Δ are entirely similar), which are like the usual ones for 'consistency properties' in infinitary logic.

<u>Facts</u>

(i) If (E, Σ, Δ) is good, and $\wedge_i \phi_i \in \Sigma$, then $(E, \Sigma \cup \{\phi_i\}_i, \Delta)$ is good

(ii) If (E, Σ, Δ) is good, and $\vee_i \phi_i \in \Sigma$, then for some i, $(E, \Sigma \cup \{\phi_i\}, \Delta)$ is good

(iii) If (E, Σ, Δ) is good, and $\forall x\phi \in \Sigma$, then for all $a \in A$, $(E, \Sigma \cup \{\phi(a)\}, \Delta)$ is good

(iv) If (E, Σ, Δ) is good, and $\exists x\phi \in \Sigma$, then for any $a \in A$ that is *new* to Σ and E, $(E, \Sigma \cup \{\phi(a)\}, \Delta)$ is good

<u>Proof</u> (i) Adding all consequences ϕ_i of $\wedge_i \phi_i \in \Sigma$ does not affect (non-)separation. Moreover, the cardinality of the extended Σ stays below μ^+. (ii) Here we need the extended class of infinitary $\vee$, $\wedge$ –combinations. Suppose that all triples $(E, \Sigma \cup \{\phi_i\}, \Delta)$ *do* L–separate, say via extended formulas α_i. Then $\vee_i \alpha_i$ separates Σ, Δ via E : quod non. (To be completely precise, one needs to spell out some details about tuples of variables.) (iii) Again, adding the logical consequence $\phi(a)$ does not affect separation. (iv) Adding $\phi(a)$ with a new constant a does not yield new separations. For, this move does not trigger new tuples in E, and then we have the usual valid inference from $\Sigma \cup \{\phi(a_{new})\} \models \alpha$ to $\Sigma \cup \{\exists x\phi\} \models \alpha$. (Note that the new a does not occur in α). ■

The new feature, as compared with consistency properties, are extension principles for the component E , that will create the required features of a potential L–isomorphism.

<u>Facts (Continued, Symmetric Forms Suppressed)</u>

(v) If (E, Σ, Δ) is good, $\mathbf{a}, \mathbf{b} \in E$, $P \in L$, $P\mathbf{a} \in \Sigma$, then $(E, \Sigma, \Delta \cup (P\mathbf{b}\})$ is good

(vi) If (E, Σ, Δ) is good, $\mathbf{a}, \mathbf{b} \in E$, then for any $a \in A$ and any $b \in B$ that is *new* to Δ, E , $(E \cup (\mathbf{a}a, \mathbf{b}b\}, \Sigma, \Delta)$ is good

<u>Proof</u> (i) Suppose there were a separation, say by the formula α. Then we must have $\Sigma \models \alpha$ [*A–substitutions*] $\wedge$ P**a** , and Δ, P**b** $\models \neg\, \alpha$ [*B–substitutions*]. The latter implies $\Delta \models \neg\, (\alpha \wedge P\mathbf{x})$ [*B–substitutions*] . But this is a separation for Σ, Δ via E after all. (ii) Suppose that we get a separation via the new E–link. I.e., $\Sigma \models \alpha$ [*A–substitutions*], and $\Delta \models \neg\, \alpha$ [*B–substitutions*], where α is an extended L–formula as before, now also involving L–subformulas $\beta(\mathbf{x}, y)$ associated with **a**a (for Σ) and **b**b (for Δ) . This gives the following separation for the original case: $\Sigma \models \exists y\alpha\, [\mathbf{a}]$, $\Delta \models \neg\, \exists y\alpha\, [\mathbf{b}]$ (recall that b was *new*). We must show here that $\exists y\alpha$ is equivalent to an admissible extended formula. Using an infinitary distributive normal form for α , we first move the existential quantifier inside over disjunctions. Over the remaining conjunctions, we then move $\exists y$ inside until it only prefixes new subformulas $\beta(\mathbf{x}, y)$, using the valid equivalence $\exists y \wedge (\beta(\mathbf{x}, y) \wedge \gamma(\mathbf{x})) \leftrightarrow \wedge (\exists y\beta \wedge \gamma)$. The result of this procedure is an ordinary L-formula with respect to the old pair $(\mathbf{a}, \mathbf{b})$. ■

Now we construct our models. We list all good triples in a sequence of length μ^+, interspersed with all relevant formulas, and all constants. We make each item occur cofinally often, to ensure fair scheduling. This can be done, for cardinality reasons. Here is a construction sketch, via a (componentwise) growing sequence of good triples in an ordinal sequence $T_0, T_1, ..., T_\alpha, ...$ $(\alpha < \mu^+)$. Our steps follows the above decompositions, starting from the initial good triple $(\{<\mathbf{a}, \mathbf{b}>\}, \{\phi(\mathbf{a})\}, \{\neg \psi(\mathbf{b})\}))$. Whenever a formula is scheduled, we check if it triggers a possible extension as listed in the above Facts, and then perform that – and the same with constants and E–zigzags. At limit ordinals, we take the union of our efforts so far, and continue. In the standard manner, this gives us two models – one based on A for $\cup_i \Sigma^i$, one based on B for $\cup_i \Delta^i$, while $\cup_i E^i$ describes a potential L–isomorphism between these two. ■

Here is the surplus in this proof. The core of the argument are the construction rules. These may also be viewed as *tableau rules* for a calculus of 'joint consistency' along potential L–isomorphism. The rules deviate from standard ones in their *ternary* format

$$\Sigma \text{ cons}_E \ \Delta$$

where E codes the relevant vocabulary and object links. The intended interpretation validates equivalences like the following:

<u>Fact</u>

(i)	$\Sigma + P\mathbf{a}\ \text{cons}_{E + \mathbf{a}, \mathbf{b}}\ \Delta$	iff	$\Sigma + P\mathbf{a}\ \text{cons}_{E + \mathbf{a}, \mathbf{b}}\ \Delta + P\mathbf{b}$
(ii)	likewise for negated atoms $\neg P\mathbf{a}, \neg P\,\mathbf{b}$		
(iii)	$\Sigma + \wedge_i \phi_i\ \text{cons}_E\ \Delta$	iff	$\Sigma + \wedge_i \phi_i + \{\phi_i\}_i\ \text{cons}_E\ \Delta$
(iv)	$\Sigma + \vee_i \phi_i\ \text{cons}_E\ \Delta$	iff	*for some* i, $\Sigma + \phi_i\ \text{cons}_E\ \Delta$
(v)	$\Sigma + \exists x\, \phi\ \text{cons}_E\ \Delta$	iff	*for some new* a, $\Sigma + \phi(a)\ \text{cons}_E\ \Delta$
(vi)	$\Sigma\ \text{cons}_{E + \mathbf{a}, \mathbf{b}}\ \Delta$	only if	*for some new* b, $\Sigma\ \text{cons}_{E + \mathbf{a}, \mathbf{b} + \mathbf{a}a, \mathbf{b}b}\ \Delta$

<u>DIGRESSION</u> <u>A New Proof Format: Ternary Sequents for Interpolation Inferences</u>

The preceding analysis suggests an independent study of 'interpolation inferences'. We can recast the preceding principles as *inference rules* manipulating *ternary* sequents, with an additional argument recording relevant vocabulary:

$$\Sigma \Rightarrow_E \Delta$$

Historically, notions of inference keeping an explicit record of variable and fixed vocabulary occur as early as Bernard Bolzano's work (1837) on styles of consequence. Working with such sequents may change familiar features of logical consequence.

There are now three positions at which to formulate *structural rules*, and e.g., one can have Monotonicity or Additivity w.r.t. vocabulary. In this connection, recall that consequence along a model relation did not necessarily retain all usual structural rules. (In fact, what it does retain are strengthening and disjunction of antecedents, as well as weakening and conjunction of consequents.) We pursue these matters a little bit.

Indeed, a number of ternary inference notions occurred in the above. For convenience, disregard the complication of the *family of* links in E, with infinitary conjunctions and disjunctions over the associated formulas. The negation of $\Sigma\ \mathrm{cons}_E\ \Delta$ then states the existence of some separating L-formula γ with $\mathbf{a}E\mathbf{b}$, $\gamma(\mathbf{a})$ implied by Σ and $\gamma(\mathbf{b})$ refuted by Δ. If we turn this into a positive statement, using negations of the formulas in Δ for convenience, then we get the existence of some L-formula with $\gamma(\mathbf{a})$ implied by Σ and $\gamma(\mathbf{b})$ implying the disjunction of Δ, as usual. This notion of 'interpolation consequence' implies our initial one of 'consequence along potential L-isomorphism'. But the latter may also, of course, be studied in its own right. (By the analysis of Barwise & van Benthem 1986, it is RE for first-order logic, and many of its variants.) Consequence along potential isomorphism has some interesting features, as compared with ordinary sequent calculi. We already mentioned the structural rules. But also, this calculus does not obey all the usual *logical rules*. E.g., the usual *conditionalization rule* fails for conditionals. To see this, let the infinitary formula $\phi = \phi\ (D, <, =)$ define the ordinal ω_0 categorically, with D interpreted as the whole domain. Likewsie, let the formula $\psi = \psi\ (D', <', =)$ define the ordinal ω_1 categorically, with D' equal to the whole domain. Evidently, $\phi\ (D, <, =), \psi\ (D', <', =)\ \Rightarrow_{\{=\}} \bot$. But this does not imply $\phi\ (D, <, =)\ \Rightarrow_{\{=\}}\ \psi\ (D', <', =)\to\bot$, since any two infinite domains admit of a $\{=\}$-potential isomorphism. Conditionalization does hold when we modify the E-argument. For, if $\Sigma, A\ \Rightarrow_E\ D$, and $L(A)$ is the vocabulary of A, then $\Sigma\ \Rightarrow_{E \cup \{\mathbf{L(A)}\}} A \to D$.

We conjecture that this ternary rule format captures consequence, even for deviant languages like *finite-variable* fragments, where no Gentzen system can ever axiomatize ordinary validity (cf. Andréka, van Benthem & Németi 1996). E.g., consider the following counter-example to interpolation inside the two-variable fragment (with $=$):

$$|A| \leq 1, |-A| \leq 1 \Rightarrow \neg\ (\exists x\ (Bx \wedge Cx) \wedge \exists x\ (Bx \wedge \neg\ Cx) \wedge \exists x\ (\neg\ Bx \wedge Cx))$$

There is no pure identity interpolant in two variables. Such formulas cannot distinguish between domains with 2 objects (where the antecedent may hold) and domains with 3 objects (where the consequent can be refuted). With our ternary inference, we do obtain

$$|A| \leq 1, |-A| \leq 1, \exists x\ (Bx \wedge Cx), \exists x\ (Bx \wedge \neg\ Cx), \exists x\ (\neg\ Bx \wedge Cx)\ \Rightarrow\ \bot$$

$$|A| \leq 1, |-A| \leq 1 \Rightarrow_{\{=, B, C\}} \neg (\exists x\, (Bx \wedge Cx) \wedge \exists x\, (Bx \wedge \neg Cx) \wedge \exists x\, (\neg Bx \wedge Cx))$$

The enforced E–registration of cross-over blocks the sequent for a standard interpolant:

$$|A| \leq 1, |-A| \leq 1 \Rightarrow_{\{=\}} \neg (\exists x\, (Bx \wedge Cx) \wedge \exists x\, (Bx \wedge \neg Cx) \wedge \exists x\, (\neg Bx \wedge Cx))$$

This analysis can be pushed still further, to probe where the classical proof of the preceding sequent must employ principles beyond the two–variable Gentzen format.

10 Relevant Results by Other Authors

Here are a few relevant results obtained by others in the interval since ELD appeared.

D'Agostino and Hollenberg 1996, 1997 use the μ–automata techniques of Janin & Walukiewicz 1996 to generalize modal interpolation theorems and Los-Tarski preservation theorems to the μ–calculus (i.e., modal logic with arbitrary fixed points). Van Eyck 1996 gives a modal lambda calculus with update operators (in the tradition of Janssen and Muskens) that combines dynamic operations with type theory, with an explicit terminating decidable rewriting system. The semantics shares some features with modal CRS-style models for predicate logic in its use of 'admissible registers'. Gerbrandy & Groeneveld 1996, Gerbrandy 1996 propose a convincing account of collective epistemic updates over non-well-founded information models (i.e., LTSs for multi-S5 modulo bisimulation), which dynamifies the epistemic logic of Fagin, Halpern, Moses & Vardi 1995 – and axiomatize the resulting logic. Their update conditions may be cast as bisimulation-respecting process operations (ELD, chapter 10). For instance, updating with a's learning that ϕ amounts to one-step unrolling all a-links from the root and then updating that separate a-structure with ϕ (just once, or iteratively at all finite levels down), while tagging on the other b-links unchanged. Groenendijk 1997 shows how to combine dynamic predicate logic with Groenendijk & Stokhof's partition semantics for questions, to obtain information states that can deal with changing focus in discourse, and the attendant inferences. Hollenberg 1996 extends the modal safety theorem to the μ–calculus: the repertoire is the original modal one (as above) plus arbitrary fixed points. His forthcoming dissertation also contains a full proof of the generalized modal translation found at the end of the above Section 2. Kurtonina 1996 introduces a new kind of bisimulation between states and sets of states (cf. Concurrent PDL, ELD chapter 10) characterizing disjunction-free modal languages. Marx 1996 shows that Pair Arrow Logic is EXP-TIME complete, and indeed, polynomially equivalent with basic modal logic plus a universal modality. Abstract Arrow Logic remains PSPACE-complete. Further arrow axioms can easily lead to undecidability. (In private correspondence, Marx has also announced EXP-TIME complexity for the original Guarded Fragment, via a reduction to CRS over 'locally cube' models.) Marx and Venema 1996 is a systematic state-of-the-art presentation of many-dimensional modal logic, including bridges with algebraic logic, as well as many key techniques for dynamic logic, broadly conceived. Ter Meulen 1995 proposes a concise framework for temporal representation in natural language that may be viewed as an alternative dynamification of temporal logic, using an extra, intermediate level of representation. Successive formulas algorithmically generate successive 'dynamic aspect trees', for which there is a notion of 'succesful embedding' into standard temporal models. Valid inference can then be defined as verification of the conclusion by any succesful embedding for the DAT of the premise sequence. This alternative dynamic architecture, employing 'constructive states', needs to be compared with the dynamification strategy of ELD, chapter 2. Seligman & ter Meulen 1995 analyse further logical aspects of this framework. In response to the open question following the above analysis of GF, Németi & Kurucz (personal communication) have announced decidability of Arrow Logic with arbitrary polyadic compositions. Otto 1997 characterizes the bisimulation-invariant queries over finite models that are computable in polynomial time as being precisely those definable in k-variable fragments of propositional μ–calculus (with operators for smallest and greatest fixed points). Finally, Patterson 1996 gives a bisimulation-cum-heredity analysis of intuitionistic propositional logic, plus analogies with other modal results. In a different formulation, involving non-symmetric directed bisimulations, some of these results had been found independently by Rob van Glabbeek.

11 First Batch of Errata for ELD

Most items in the following list of Errata were kindly supplied by Eva Hoogland.

p. 18, +14	$[[\phi \wedge \psi]]\,(X) \;=\; [[\phi]]\,(X) \cap [[\psi]]\,(X)$
p. 18, –1	$(R_F)^{\#}$
p. 19, –14	... instructions (ii) will produce $\{\neg\, s\}$ ...
p. 20	The counterexample to Cut is defective, and should be replaced by $\Diamond p, \neg\Diamond p \models_{upd} q$ and $\neg p \models_{upd} \neg\Diamond p$ *but not* $\Diamond p, \neg p \models_{upd} q$. A valid Cut rule: $X \models_{upd} A$ and $A, Y \models_{upd} B$ imply $X, Y \models_{upd} B$
p. 29, + 1	... $(\neg\, \exists x\, Dx)$ **6**
p. 30, +8	... the De Morgan law *interchanging negation with disjunction* ...
p. 34, –21	$\lambda P \bullet$
p. 35, +16	shift P (presence of P) and $\neg$P (absence of p) to right under the nodes.

p. 65 <u>Fact</u> 'In the limit', Relational Set Algebra reduces to Boolean Algebra.

<u>Proof</u> Valid SRA identities remains BA-valid when read with Booleans unchanged, converses disregarded, conjunction for composition, taking Boolean **1** for identity. The reason is that such an identity must hold on the full set RA over a singleton point (an isomorph of the two-element Boolean set algebra) – and there, the given syntactic transformations are true. But now, one can use the fact that the identities valid on the two-element Boolean algebra are in fact valid on all Boolean algebras. ■

p. 65, –3 ... ($R^{\vee}$ o – ....

p. 73, +4 ... and **t** ≡ t' .

p. 91, +3 "From left to right, *this is a simple induction*."

p. 91, +5 "Success paths" are finite sequence of states running from the initial one, via successive atomic transitions, to some success state (where √ holds). "Succesful path formulas" are existental first-order formulas describing success paths, including truth/falsity of all relevant atoms at their nodes.

p. 96, +3 ... to Fragments 1 and **3** ...

p. 96 First picture: $\mathbf{y_1}$ instead of $\mathbf{x_1}$

p. 99, –5 ... is the unique subpath of π ...

p. 100, –6 Delete the sentence beginning with "We can be sure..."

p. 112, –16 ... and **t** ≡ t' .

p. 112 For finite models **M**, x and **N**, y, equivalence of their modal theories implies the existence of a bisimulation linking x and y, and vice versa. A similar result for fails for safety of operations on finite models.

<u>Counter-example</u> Consider the model

with the relation R = { <1, 2>, <4, 5>, <4, 6>}. This relation is safe: every bisimulation on the model w.r.t. the two actions a, b respects it. But R has no definition in the standard format of the Safety Theorem. ■

What we can show, however, is that all 'internally safe' relations must belong to the transitive closure of the union of all atomic actions. Passing from a model to its obvious 'bisimulation collapse', this does come close to an actual enumeration in the prescribed syntactic format which allows only uses of composition, union, and arbitrary modal tests. In general we are left with this *OpenQuestion:* "Find an internal version of the Safety Theorem on finite models".

p. 114, – 2 ... Φ_n (x_n) *and* Px_n , ...
p. 115, –16 .... modal *equivalence* between states ...
p. 116, +4 ... Start with the match w_1, $\mathbf{v_1}$ . ...
p. 117, In lines –14, –15 and –18, replace w by **v** .
p. 129, +4 ... $(\neg \exists x\ Dx)_{\mathbf{6}}$
p. 168, –1 replace first x by **s** .

12 References

H. Andréka, J. van Benthem & I. Németi, 1996, 'Modal Logics and Bounded First-Order Fragments'. To appear in the *Journal of Philosophical Logic*.

J. Barwise & L. Moss, 1996,*Vicious Circles*, CSLI Publications, Stanford.

J. Barwise & J. van Benthem, 1996, 'Interpolation, Preservation, and Pebble Games', Research Report ML-96-12, Institute for Logic, Language and Computation, University of Amsterdam.

J. van Benthem, 1976, *Modal Correspondence Theory*, dissertation, Mathematical Institute, University of Amsterdam.

J. van Benthem, 1983, *The Logic of Time*, Reidel, Dordrecht.

J. van Benthem, 1993, 'Programming Operations that are Safe for Bisimulation', Report 93-179, Center for the Study of Language and Information, Stanford University. To appear in Proc's Logic Colloquium. Clermont Ferrand 1994, *Studia Logica.*

J. van Benthem, 1995, 'Temporal Logic', in D. Gabbay, C. Hoggar & J. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 4, Oxford University Press, 241-350.

J. van Benthem, 1996, *Exploring Logical Dynamics*, CSLI Publications, Stanford. Distributed by Cambridge University Press, Cambridge.

P. Dekker, 1993, *Transsentential Meditations*, Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.

J. van Eyck, 1997, 'Typed Logics with States', CWI, Amsterdam. To appear in *Bulletin of the Interest Group for Pure and Applied Logic*, London.

J. Groenendijk, 1997, 'A Dynamic Alternative Semantics for Focus', colloquium talk, 'Dag der GRAMschap', Rijksuniversiteit Groningen.

J. Gerbrandy & W. Groeneveld, 1997, 'Reasoning about Information Change', to appear in *Journal of Logic, Language and Information*.

J. Gerbrandy, 1997, 'Dynamic Epistemic Logic', to appear in Proceedings STASS Workshop on Logic, Language and Computation, London.

M. Hollenberg, 1995, 'Finite Safety for Bisimulation', Institute for Philosophy, Rijksuniversiteit Utrecht.

M. Hollenberg, 1996A, 'Bisimulation Respecting First-Order Operations', Logic Group Preprint Series 156, Institute for Philosophy, Rijksuniversiteit Utrecht.

M. Hollenberg, 1996B, 'General Safety for Bisimulation', in P. Dekker & M. Stokhof, eds., *Proceedings 10th Amsterdam Colloquium*, Institute for Logic, Language and Computation, University of Amsterdam.

M. Hollenberg, 1997, *Process Invariance and Safety*, forthcoming Ph.D. dissertation, Institute for Philosophy, Rijksuniversiteit Utrecht.

D. Janin & I. Walukiewicz, 1996, 'On the Expressive Completeness of the Propositional μ–Calculus with Respect to Monadic Second-Order Logic', Department of Mathematics and Informatics, University of Bordeaux & Department of Computer Science, Aarhus Univ.

H. Kamp & U. Reyle, 1993, *From Discourse to Logic,* Kluwer, Dordrecht.

M. Marx, 1995, *Arrow Logic and Relativized Algebras of Relations*, dissertation, CCSOM and ILLC, University of Amsterdam.

M. Marx & Y. Venema, 1996, *Multi-Dimensional Modal Logic*, Kluwer, Dordrecht.

A. ter Meulen, 1995, *Representing Time in Natural Language*, Bradford Books / The MIT Press, Cambridge (Mass.).

I. Németi, 1991, 'Algebraizations of Quantifier Logics: An Introductory Overview', *Studia Logica* 50:3-4, 485-569. (There are continuous electronic updates.)

M. Otto, 1997, 'Capturing Bisimulation-Invariant PTime', Department of Informatics, RWTH Aachen.

E. Rosen, 1995, 'Modal Logic over Finite Structures', Report ML-95-08, ILLC, Univ. of Amsterdam. To appear in *Journal of Logic, Language and Information*.

M. de Rijke, 1993, *Extending Modal Logics*, Ph.D. dissertation, Institute for Logic, Language & Computation, University of Amsterdam.

J. Seligman & A. ter Meulen, 1995, 'Dynamic Aspect Trees', in L. Pólos & M. Masuch, eds., *Applied Logic: How, What and Why*, Kluwer, Dordrecht, 287–320.