

Modal and Temporal Logics

by

Colin Stirling

Modal and Temporal Logics

LFCS Report Series

ECS-LFCS-91-157

LFCS

May 1991

Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ

Copyright © 1991, LFCS

Modal and Temporal Logics

Colin Stirling

Contents

1	Introduction	2
1.1	Transition systems	2
1.2	Runs	5
1.3	Computational concerns	9
2	Propositional Modal Logics	11
2.1	Basics	11
2.2	Minimal modal logic	14
2.3	Correspondence and incompleteness	16
2.4	Dynamic logic	20
2.5	Modal algebras	22
2.6	Decision procedures	25
3	Propositional Temporal Logics	28
3.1	Between modal and temporal logics	28
3.2	Basics	31
3.3	Linear and branching time	34
3.4	Minimal temporal logics	39
3.5	Classes of models, automata, and correspondence	42
3.6	Families of runs and temporal properties	46
3.7	Decision procedures	48
4	Modal and Temporal Mu-Calculi	49
4.1	Modal and temporal equations	49
4.2	The mu-calculi	52
4.3	Monotonicity and continuity	55
4.4	Minimal mu-calculi	57
4.5	Decision procedures	58
5	Expressiveness	59
5.1	Expressive completeness	60
5.2	ω -regular expressions	64
5.3	Zig-zags, bisimulations, and histories	65
6	Sound and Complete Axiom Systems	69
6.1	Soundness	69

6.2	Canonical models	70
6.3	Points and schedulers	74

1 Introduction

1.1 Transition systems

In achieving their goals, programs manipulate data and thereby may change the state of the computer. Regarded as a dynamic system, the computer responds to program instructions and in so doing may change state. Activity and change are formalized by the operational semantics of programs and systems. Modal and temporal logics can directly express activity with their paradigm operators such as $[a]$ (*box a*) when a is an action, and F . Where Φ is a proposition, $[a]\Phi$ expresses *after every a action Φ holds* while $F\Phi$ expresses *eventually Φ holds*. The modal operator $[a]$ highlights the action that may provoke change of state whereas the temporal operator F focuses on the resulting changes. In both cases these operators may express properties of events that happen during a run of a system, events that may be crucial to understanding its overall behaviour. It is no accident that these logics can be used to express such notions. For their models comprehend structures that are encountered in operational semantics of programs and systems. These structures, (labelled) transition systems, are pivotal to this chapter.

1.1.1: Definition

A transition system is a pair $T = (S, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ where S is a non-empty set (of states or configurations), $\mathcal{L}$ is a non-empty set (of labels), and for each $a \in \mathcal{L}$, $\xrightarrow{a} \subseteq S \times S$.

Transition systems may be presented diagrammatically: figure 1 pictures the transition system with state set $\{s_0, s_1, s_2\}$, label set $\{a, b, c, d\}$, and where, for instance, $\xrightarrow{a} = \{(s_0, s_1), (s_0, s_2)\}$ (and $\xrightarrow{d} = \emptyset$). Usually we write $s \xrightarrow{a} s'$ instead of $(s, s') \in \xrightarrow{a}$.

Typically S is a set of states or configurations: states of a data base,

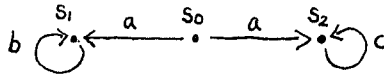


Fig. 1. A transition system where $\mathcal{L} = \{a, b, c, d\}$

program states, states of knowledge, configurations of a machine and so on. Each relation $\xrightarrow{a}$ is a *transition* relation: $s \xrightarrow{a} s'$ represents that state s may become s' due to, or under the influence of, a which could be an action, a period of time, a person, or whatever. A transition system with label set $\mathcal{L}$ is said to have *sort* $\mathcal{L}$. Transition systems may carry further structure, as we note below. A useful extension is the inclusion of an initial state, commonly called a *rooted* transition system.

Figure 2 embodies the behaviour of a very elementary vending machine V . In state V either a $2p$ or $1p$ coin can be accepted, resulting in one of the states V_1 or V_2 . Depending on which coin is accepted a *big* or *little* button may be depressed, and then an item may be collected. (Explicit representation of sort is omitted when it just consists of all the labels mentioned.)

Pictures of transition systems can become very large and unwieldy. It is therefore useful to have succinct presentations of them. A general technique for doing this is to have a language of expressions for states, together with rules for generating transitions. For instance, for the vending machine of Fig. 2 we could appeal to the language of CCS (Milner 1989):

$$\begin{array}{ll} V \stackrel{\text{def}}{=} 2p.V_1 + 1p.V_2 & V_1 \stackrel{\text{def}}{=} \text{big}.V_3 \\ V_2 \stackrel{\text{def}}{=} \text{little}.V_3 & V_3 \stackrel{\text{def}}{=} \text{collect}.V \end{array}$$

Employed here are some state constructors: $W \stackrel{\text{def}}{=} E$ declares the state constant W to be E ; $a.E$ names a state whenever $a \in \mathcal{L}$ and E is a state; and $+$ is a binary state combinator. Their meanings are given by the following rules for generating transitions:

$$\frac{}{a.E \xrightarrow{a} E} \quad \frac{E \xrightarrow{a} F}{W \xrightarrow{a} F} W \stackrel{\text{def}}{=} E \quad \frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'} \quad \frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'}$$

The definition rule, for instance, is to be understood as: if $E \xrightarrow{a} F$ and $W \stackrel{\text{def}}{=} E$ then also $W \xrightarrow{a} F$. These rules underpin the illustrated vending machine, justifying transitions such as $V \xrightarrow{2p} V_1$. A second example,

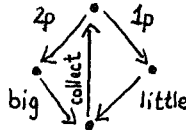


Fig. 2. A simple vending machine

presented in this expression and rule format, is an unbounded stack where $i \in \mathbb{N}$:

$$St_0 \stackrel{def}{=} \text{empty}.St_0 + \text{push}.St_1 \quad St_{i+1} \stackrel{def}{=} \text{pop}.St_i + \text{push}.St_{i+2}$$

The previous examples appeal to a language of states together with rules for transitions. An alternative approach is to have a language of labels, and transition rules. Consider the family of while programs, defined by:

$$p ::= c \mid p_1; p_2 \mid \text{if } b \text{ then } p_1 \text{ else } p_2 \mid \text{while } b \text{ do } p$$

where c ranges over members of a family of elementary programs (such as assignments) and b over a family of boolean expressions. The meaning of these programs can be defined as a transition system whose states are memories, finite associations of program variables with values. The family of labels consists of the boolean expressions and all the programs. The transition relations are defined by rules, assuming first that the relations $\xrightarrow{c}$ are given, and that $\xrightarrow{b} = \{(M, M) \mid b \text{ is true at memory } M\}$. For instance we would expect that $M \xrightarrow{x:=x+1} M'$ when M is the memory state $\{(x, 3), (y, 4)\}$ and M' is $\{(x, 4), (y, 4)\}$, and that $M' \xrightarrow{x=y} M'$ whereas $M \xrightarrow{x=y} M$. So $M \xrightarrow{c} M'$ represents that an execution of basic command c in memory state M may terminate in the (possibly new) state M' , and $M \xrightarrow{b} M$ that the boolean expression may evaluate successfully to true in the program state M . Rules for the remaining labels (programs) are structurally defined:

$$\frac{M \xrightarrow{p} M_1 \quad M_1 \xrightarrow{q} M_2}{M \xrightarrow{p;q} M_2}$$

$$\frac{M \xrightarrow{b} M \quad M \xrightarrow{p_1} M'}{M \xrightarrow{q} M'} \quad q = \text{if } b \text{ then } p_1 \text{ else } p_2 \quad \frac{M \xrightarrow{\neg b} M \quad M \xrightarrow{p_2} M'}{M \xrightarrow{q} M'}$$

$$\frac{M \xrightarrow{b} M \quad M \xrightarrow{p;q} M'}{M \xrightarrow{q} M'} \quad q = \text{while } b \text{ do } p \quad \frac{M \xrightarrow{\neg b} M}{M \xrightarrow{q} M}$$

These rules capture the intended meaning of the constructors when $M \xrightarrow{p} M'$ is understood as *an execution of program p in state M may terminate in state M'* .

Assume that the formulas Φ and Ψ express properties of memory states. The modal formula $[p]\Psi$ also describes a property: Ψ holds after every execution of p . This is true at state M if Ψ is true at each M' in the

set $\{M' \mid M \xrightarrow{p} M'\}$. More significantly, if $\Phi \rightarrow [p]\Psi$ (where $\rightarrow$ is implication) is true at every memory then program p is *partially correct* with respect to the precondition Φ and postcondition Ψ . This explicit association between the semantics of modal logic and the meaning of while programs was made by Pratt (1976). Related work includes *algorithmic logic* (Salwicki 1977), as well as Floyd-Hoare's *logic of partial correctness*, (Floyd 1967; Hoare 1969), and the *weakest precondition calculus* (Dijkstra 1976). Alternatively, if Φ expresses properties of states of the vending machine then the modal formula $[2p]\Phi$ expresses that Φ holds after every acceptance of $2p$: for instance, V_2 trivially has this property whereas V has it if V_1 has Φ . Modal formulas in this way can describe local capabilities of systems, a use of modal logic due to (Hennessy and Milner 1980,1985).

Transition systems generalize Kripke structures introduced in the late 1950s, (Kripke 1959, 1963), for interpreting modal, intuitionistic and later temporal logics. An important extension is *polymodal* logics, consisting of families of modalities $[a]$ for each $a \in \mathcal{L}$ instead of a single modality $[\]$. Polymodal logics are traditionally associated with logics of knowledge and belief where $[a]\Phi$ is understood as *a knows or believes Φ* (Hintikka 1962): in a transition system representing belief or knowledge $s \xrightarrow{a} t$ symbolizes that a 's knowledge or belief at state s is of t . The examples above suggest that polymodal logic can be applied naturally to understand (computational) activity and change. They also reveal the potential use of extra (computational) structure even at the level of propositional modal logics either in the set of labels, as an algebra of programs, or in the set of states, via a structured language of behaviours. In one case there is interest in the formal relationships between modalities such as $[p; q]$ and the modalities $[p]$ and $[q]$. In the other case there are questions about the relationship between modal properties of a structured state such as $E + F$ and the properties of its components, E and F .

1.2 Runs

For many crucial programs (including operating systems) termination is a catastrophe rather than an intended virtue. When p is such a program, the modality $[p]$ is ineffectual since there should not be memory states M and M' with the feature $M \xrightarrow{p} M'$. The vending machine provides another instance where termination is undesirable. More generally these are examples of *reactive* systems, whose behaviours may essentially depend on concurrent interaction with other systems (such as their users). Concurrency as a binary parallel operator $\parallel$ may be added to the language of states and also to the while language of labels. For instance, when E and F are states assume that $E \parallel F$ is their concurrent composition. Transition rules for $\parallel$ include the following interleaving rules:

$$\frac{E \xrightarrow{a} E'}{E \parallel F \xrightarrow{a} E' \parallel F} \quad \frac{F \xrightarrow{a} F'}{E \parallel F \xrightarrow{a} E \parallel F'}$$

There may also be synchronization rules. For instance, Milner (1989) associates with each label a a partner $\bar{a}$, and the performances of a and $\bar{a}$ together constitute a handshake synchronization.

The previous relational semantics of while programs as labels can not cope with either perpetual programs that are not intended to terminate or with parallel programs: the relation $\xrightarrow{p||q}$ is not definable solely from the relations $\xrightarrow{p}$ and $\xrightarrow{q}$. Knowledge of intermediate memory states is needed as the components may interact either by interfering or by cooperating with each other. As with the vending machine, a transition system semantics for these programs may be presented which only involves transitions for elementary actions instead of whole structured programs. By single stepping the transition relations in this way programs that are not intended to terminate are covered as is concurrent interaction.

Assume that $\parallel$ is added to the while language earlier. Again let c range over elementary programs which may now also include synchronization or communication primitives (such as await or channel input and output commands). The states of the transition system are now program and memory pairs (p, M) , as well as pairs (ϵ, M) where ϵ is a new symbol representing the empty (or terminated) program. Labels are restricted to booleans and elementary programs. Initially part of the relations $\xrightarrow{c}$ and $\xrightarrow{b}$ are given. The assumed part of $\xrightarrow{c}$ relates pairs (c, M) , (ϵ, M') : for instance we would expect that $(x := x + 1, M) \xrightarrow{x := x + 1} (\epsilon, M')$ whenever $(x, n) \in M$ and $(x, n + 1) \in M'$. The given part of $\xrightarrow{b}$ is for **if** and **while** programs: if $p = \text{if } b \text{ then } q_1 \text{ else } q_2$ and b is true at M then $(p, M) \xrightarrow{b} (q_1, M)$, otherwise $(p, M) \xrightarrow{b} (q_2, M)$; similarly if $p = \text{while } b \text{ do } q$ and b is true at M then $(p, M) \xrightarrow{b} (q; p, M)$, otherwise $(p, M) \xrightarrow{b} (\epsilon, M)$. Transitions from other composite programs and memories are defined by rules as follows assuming $\epsilon; p = p$ and $\epsilon \parallel p = p \parallel \epsilon$, and that a ranges over b or c labels:

$$\frac{(p, M) \xrightarrow{a} (p', M')}{(p; q, M) \xrightarrow{a} (p'; q, M')}$$

$$\frac{(p, M) \xrightarrow{a} (p', M')}{(p \parallel q, M) \xrightarrow{a} (p' \parallel q, M')} \quad \frac{(q, M) \xrightarrow{a} (q', M')}{(p \parallel q, M) \xrightarrow{a} (p \parallel q', M')}$$

Again there may be extra synchronization or communication rules for $\parallel$.

Modal formulas such as $[c]\Phi$ and $[2p]\Psi$ express local properties of program memory pairs, or of structured states such as those of the vending machine. Less local properties expressing features of eventual evolution such as that a program may eventually terminate (reach ϵ) when initially at a memory state are also salient. A similar example is the following *responsiveness* property of the vending machine that whenever a $2p$ is accepted then eventually *collect* must happen. Crucial for both the while language and the vending machine is a *run* of a program or system, as a maximal length path through its transition system. For instance, a run of the concurrent program $p \parallel q$ when p is **while** $y \neq 0$ **do** $x := x + 1$, and q is $y := 0$, and where (n, m) represents the memory $\{(x, n), (y, m)\}$ is:

$$(p \parallel q, (0, 1)) \xrightarrow{y \neq 0} \dots \xrightarrow{x := x + 1} (p \parallel q, (n, 1)) \xrightarrow{y := 0} (p, (n, 0)) \xrightarrow{\neg y \neq 0} (\epsilon, (n, 0))$$

And the following is a run of the vending machine: $V \xrightarrow{2p} V_1 \xrightarrow{big} V_3 \xrightarrow{collect} V \xrightarrow{2p} \dots$

Petri nets provide another kind of example. Associated with the net of figure 3 is a transition system whose state space is the set of markings, vectors of the form (k, m, n) representing that condition A has k tokens, B has m , and C has n . The transition relations (assuming that only one event happens at a time) are given by:

$$\begin{aligned} (k + 1, m, n) &\xrightarrow{a} (k + 1, m + 1, n) \\ (k + 1, m, n) &\xrightarrow{b} (k, m, n + 1) \\ (k, m + 1, n + 1) &\xrightarrow{c} (k, m, n + 1) \end{aligned}$$

A feature of any run from the initial state, the marking $(1, 0, 0)$, is that event c can only happen finitely often.

These properties of runs are expressible in temporal logics as developed by Prior (1957, 1967), thereby providing a formal framework for describing and reasoning about change through time. A variety of temporal operators (such as the eventually operator) specify how the truth of a formula at one time depends on the truth of subformulas at other times. Pnueli (1977) pioneered this use of temporal logics for describing and reasoning about the

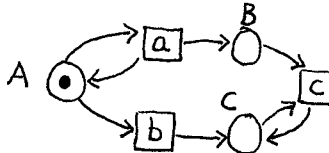


Fig. 3. A small Petri net

behaviour of concurrent, non-deterministic, or cyclic systems by linking the semantics of temporal logics and the semantics of such systems. Precursors include uses of transition systems such as (Keller 1976), the work on understanding cyclic programs, and applications of Büchi's sequential calculus.

Not every maximal length path through a transition system may count as a run. Fairness, or liveness assumptions, may preclude some. For instance, the following ω length run for the parallel program $p \parallel q$ defined earlier may be excluded on the ground that the component q does not contribute.

$$(p \parallel q, (0, 1)) \xrightarrow{y \neq 0} \dots \xrightarrow{x := x + 1} (p \parallel q, (n, 1)) \xrightarrow{y \neq 0} \dots$$

Focal structures for the temporal logics examined later are *extended* transition systems, containing an extra constituent which is a non-empty set of runs through its underlying transition system. For simplicity it is assumed that all paths in this constituent have ω length. This is guaranteed when the underlying transition system is *total*, in the sense that for every state s there is an s' and an a such that $s \xrightarrow{a} s'$. Loops with an imaginary label can be grafted onto states with no transitions emanating from them to make a transition system total such as that for the concurrent while language.

1.2.1: Definition

An *extended transition system* is a pair $(T, \mathcal{R})$ where T is a total transition system, and $\mathcal{R}$ is a non-empty set of runs through T with the property that every $s \in S_T$ occurs in some run.

Included here, for technical reasons, is that every state of the underlying transition system belongs to some run. More generally, the presupposition that a run has an initial state may be lifted allowing it to take the form $\dots s_i \xrightarrow{a_i} s_{i+1} \xrightarrow{a_{i+1}} \dots$ with indices drawn from the integers—useful for interpreting *reverse* temporal operators such as *previously*. Extended transition systems may be presented diagrammatically, with a statement about which are the admissible runs, as in figure 4.

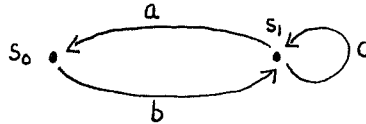


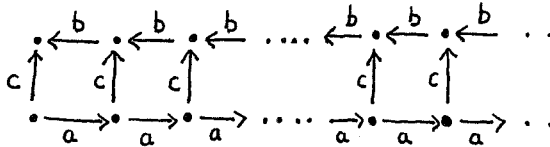
Fig. 4. Every run contains a finite number of c labels

1.3 Computational concerns

Transition systems model the behaviour of programs, and at the same time are models for modal and temporal logics. Showing that a program meets its specification expressed as a modal or temporal formula amounts to proving that the formula is true at a state (or set of states) or a set of runs in the transition system model of the program. One approach, advocated by Manna and Pnueli (1981), is proof-theoretic. They propose that the program via its model be coded as a theory (a set of temporal formulas) Γ . The required property should then follow from, be a temporal logic consequence of, Γ . A sound and complete axiomatization of valid formulas provides a framework wherein such deductions may be presented. This method views the transition system model of a program as secondary, to be dispensed with in favour of the theory Γ .

More direct methods of establishing relative truth, that a family of states or runs has a property, can be formalized using tableau techniques. Logically, these underpin *model checkers*, algorithms that automatically establish whether properties hold in finite state systems, as pioneered by Clarke, Emerson, and Sistla (1983), and Quielle and Sifakis (1981). Deriving a program from a modal or temporal specification is more difficult. Formally it amounts to producing a transition system model wherein the formula is true at a state or a set of runs and then extracting a program from it. This is intimately related to the *satisfiability* problem of deciding whether a formula has a model. This approach to program synthesis is discussed in (Clarke and Emerson 1981; Manna and Wolper 1984). An alternative, proposed in (Gabbay 1989; Moszkowski 1986), is to view temporal formulas not only as specifications but also as being executable. For instance the formula $F\Phi$ can be understood as the command *make Φ true in the future*.

Transition systems are also naturally rooted in language and automata theory. In this circumstance think of the relation $s \xrightarrow{a} t$ as representing that s *accepts* a (and becomes t). Any grammar determines a transition system. For instance, in figure 5 any finite length run from the state labelled X contains a string of labels $a^n cb^n$. The language accepted by X is the set of these strings. Extended transition systems are produce of au-



tomata on infinite strings or infinite trees. Their runs, either as sets or assembled instead into computation trees, fulfill the automata acceptance condition. Explicitly relating finite automata to modal and temporal logics has proved to be very fruitful, offering new insights into, and extensions of, these logics. As regular sequences of events are not first-order definable, and consequently not definable within traditional modal and temporal logics, this has provoked second-order propositional versions of these logics including extended temporal logic (Wolper 1983), and the modal μ -calculus (Kozen 1983). Automata-theoretic methods for satisfiability and model checking have been employed. These appeal to translations of formulas into automata which are then tested for non-emptiness (Streett 1982; Wolper, Vardi and Sistla 1983; Streett and Emerson 1989).

A key feature of modal and temporal logics is that their basic definition of truth is *relativized* to a state or a run (of a transition system model). Less local notions of truth are then abstractions: for example, that program p has the temporal property Φ can be captured by the stipulation that Φ is true of every admissible run starting from p (in the model for p). So questions of verification and program synthesis raise new issues about exposing structure in these models. A clear example is compositionality: can techniques be developed that, for instance, associate properties of $p \parallel q$ with properties of its components? Such questions take seriously the binary structures involved: program structure and logical structure.

Transition system models of programs are overly concrete. Useful abstractions are therefore of interest. Especially relevant here is the work on process algebras (Bergstra and Klop 1989; Hennessy 1988; Milner 1989) whose models are transition systems quotiented by equivalences (or pre-orders). Paramount is *bisimulation* equivalence (Park 1981) which is much finer than language equivalence: not only is it a congruence for a very large class of process combinators, but also it is decidable on general families of infinite state transition systems generated from context free grammars (Baeten *et al.* 1987). This equivalence also has a simple modal (and temporal) logic *characterization*: two states of an image finite¹ transition system are equivalent iff they have the same modal properties (Hennessy and Milner 1980).

Expressive power is a foundational logical concern bearing on practical issues. One immediate interest is whether a logic is able to (succinctly) capture a particular crucial property of a program. However notions of relevant program properties can themselves be guided by perspicuous formalisms. One very successful application of temporal logics is a general classification of program properties covering both concurrent and perpetual systems, see (Manna and Pnueli 1989) for a survey.

¹ A transition system is image finite, if for each state s and label a the set $\{t \mid s \xrightarrow{a} t\}$ is finite.

This Chapter presents a brief view of (classical) propositional modal and temporal logics (and their second-order versions with fixed point operators). I have concentrated on those logics that have been used to describe properties of computational systems presented as transition systems, and the results outlined are just a selection from this active research area. For a more general perspective see (van Benthem 1989, 1990), and for more details of the application of temporal logics to Computer Science see (Barringer and Kuiper 1991). Besides the editors, I would like to thank Johan van Benthem, Julian Bradfield, Ed Clarke, Rance Cleaveland, Mads Dam, Hans Hüttel, Kim Larsen, Angelika Mader, Pawel Paczkowski, Amir Pnueli, Michael Siegel, Bernhard Steffen, Martin Steffen, Terry Stroup, Dirk Taubner, Wolfgang Thomas, David Walker, and Jeff Zucker for comments on previous versions of this survey, or discussions about topics covered in this work.

2 Propositional Modal Logics

2.1 Basics

The basic propositional modal language in abstract syntax form is:

$$\Phi ::= Q \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid [a]\Phi$$

where Q ranges over members of a family of atomic sentences, and a over labels belonging to a set $\mathcal{L}$. This definition just means that a formula is either an atomic sentence Q ; a negated formula $\neg\Phi$; a conjunction of formulas $\Phi \wedge \Psi$; or a modalized formula $[a]\Phi$. Really the syntax specifies a family of languages parameterized by atomic sentences and labels. A modal language with labels drawn from $\mathcal{L}$ is said to have *sort* $\mathcal{L}$. This term is extended to formulas: the formula Φ has sort $\mathcal{L}$ if it belongs to a language of sort $\mathcal{L}$. Each modal language is an extension of classical propositional logic. Usual classical abbreviations for disjunction and implication apply: $\Phi \vee \Psi \stackrel{def}{=} \neg(\neg\Phi \wedge \neg\Psi)$; and $\Phi \rightarrow \Psi \stackrel{def}{=} \neg\Phi \vee \Psi$. Moreover, for each $a \in \mathcal{L}$ the dual of $[a]$ is the modal operator $\langle a \rangle$ (*diamond* a): that is, $\langle a \rangle \Phi \stackrel{def}{=} \neg[a]\neg\Phi$.

A modal language with sort $\mathcal{L}$ is interpreted on *transition systems* of sort $\mathcal{L}$. Recall from the Introduction, Definition 1.1.1, that these are structures $\mathcal{T}$ that are pairs $(\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ where $\mathcal{S}$ is a set of states, $\mathcal{L}$ (the sort of $\mathcal{T}$) a set of labels, and $\xrightarrow{a}$ for each $a \in \mathcal{L}$ is a binary relation on $\mathcal{S}$. As with propositional logic we also need to interpret the atomic sentences. So a *modal model* on the transition system $\mathcal{T}$ is a pair $\mathcal{M} = (\mathcal{T}, \mathcal{V})$ where $\mathcal{V}$ is a valuation which assigns to each atomic sentence Q a subset of the states of $\mathcal{T}$: the sentence Q is deemed to be true at each state $s \in \mathcal{V}(Q)$ under $\mathcal{M}$. Truth at state s of an arbitrary formula Φ under $\mathcal{M}$ is inductively defined

using the notation $s \models_{\mathcal{M}} \Phi$. When Φ is not true at s under $\mathcal{M}$ we write $s \not\models_{\mathcal{M}} \Phi$.

$$\begin{array}{ll}
 s \models_{\mathcal{M}} Q & \text{iff } s \in \mathcal{V}(Q) \\
 s \models_{\mathcal{M}} \neg\Phi & \text{iff } s \not\models_{\mathcal{M}} \Phi \\
 s \models_{\mathcal{M}} \Phi \wedge \Psi & \text{iff } s \models_{\mathcal{M}} \Phi \text{ and } s \models_{\mathcal{M}} \Psi \\
 s \models_{\mathcal{M}} [a]\Phi & \text{iff } \forall s'. \text{ if } s \xrightarrow{a} s' \text{ then } s' \models_{\mathcal{M}} \Phi
 \end{array}$$

In the final clause the quantifier $\forall s'$ ranges over $\mathcal{S}_{\mathcal{M}}$ the states of the model. (Commonly in the sequel this kind of implicit understanding is assumed to avoid burdensome notation.) The derived semantic clause for $\langle a \rangle$ is:

$$s \models_{\mathcal{M}} \langle a \rangle \Phi \text{ iff } \exists s'. s \xrightarrow{a} s' \text{ and } s' \models_{\mathcal{M}} \Phi$$

The major difference with traditional modal logic as presented in (Hughes and Cresswell 1973; Chellas 1980; and, Bull and Segerberg 1984) is the generalization to *families* of modalities, a feature exploited in some of the intended interpretations mentioned in the Introduction.

Example. Consider the transition system for the vending machine V presented in Section 1.1. Assume $\mathbf{tt}$ is an atomic formula expressing true, and $\mathcal{M}$ a modal model for V . Now $V \models_{\mathcal{M}} \langle 2p \rangle \mathbf{tt}$, that is V can accept $2p$. But a button can not yet be depressed (before money is deposited), so $V \models_{\mathcal{M}} [\mathbf{big}] \neg \mathbf{tt} \wedge [\mathbf{little}] \neg \mathbf{tt}$. V has more interesting properties too:

- $V \models_{\mathcal{M}} [2p]([\mathbf{little}] \neg \mathbf{tt} \wedge \langle \mathbf{big} \rangle \mathbf{tt})$: after $2p$ is deposited the *little* button can not be depressed whereas the *big* one can.
- $V \models_{\mathcal{M}} [1p][\mathbf{little}] \langle \mathbf{collect} \rangle \mathbf{tt}$: after $1p$ is entrusted, and then the *little* button is depressed, an item can be collected.

Similar properties could be defined for the stack of Section 1.1. ■

Example. The application of modal logic to the transition system for while programs in Section 1.1 has structured programs as labels. Transitions have the form $M \xrightarrow{p} M'$, representing that an execution of p in memory state M may terminate in state M' . The modal logic now contains modalities of the form $[p]$, $\langle p \rangle$. Consider a family of atomic formulas of the form $x \leq t$ and $x \geq t$ where x is a variable and t an arithmetic term. Let $\mathcal{V}$ be a valuation which assigns to each atomic formula the set of memories (finite associations of variables and numbers) at which it is true. For instance $\mathcal{V}(x \leq 3) = \{M \mid (x, n) \in M \text{ and } n \leq 3\}$. In this case $\mathcal{V}$ is the *intended* valuation, and the resulting modal model the *intended* model. An important kind of modal formula is $\Phi \rightarrow [p]\Psi$, called a *partial correctness* formula: M has this property (in $\mathcal{M}$) provided that if $M \models_{\mathcal{M}} \Phi$ and $M \xrightarrow{p} M'$ then $M' \models_{\mathcal{M}} \Psi$ ■

Verifying that the vending machine above has a modal property is a straightforward application of the inductive definition of truth at a state under a model. It is more difficult for partial correctness formulas where additional induction (on the data) may also be necessary: for instance, consider showing $M \models_{\mathcal{M}} z = n \wedge y = 1 \rightarrow [q]y = n!$, for any M when q is the program **while** $z \geq 0$ **do** ($y := z \times y; z := z - 1$).

Expressiveness of the basic modal language can be increased by additional modalities. Natural candidates are the *reverse* operators $\overline{[a]}$ (and their duals $\langle a \rangle$) for any label a .

$$s \models_{\mathcal{M}} \overline{[a]}\Phi \text{ iff } \forall s'. \text{ if } s' \xrightarrow{a} s \text{ then } s' \models_{\mathcal{M}} \Phi$$

In the case of a while program p the formula $\overline{[p]}\Phi$ expresses a property of running p backwards. Another useful extension when the label set $\mathcal{L}$ is infinite is to permit modalities $[J]$ for $J \subseteq \mathcal{L}$: $s \models_{\mathcal{M}} [J]\Phi$ iff for each s' and $a \in J$, if $s \xrightarrow{a} s'$ then $s' \models_{\mathcal{M}} \Phi$. In which case the unrelativized modality $[\]$, which pays no attention to the labels, abbreviates $[\mathcal{L}]$.

Some standard notation and terminology which abstracts from truth at a state under a model is:

2.1.1: Definition

- i. Φ is $\mathcal{M}$ -true, written $\models_{\mathcal{M}} \Phi$, if for all states s , $s \models_{\mathcal{M}} \Phi$,
- ii. Φ is $\mathcal{T}$ -valid, written $\models_{\mathcal{T}} \Phi$, if for all models $\mathcal{M}$ on $\mathcal{T}$, $\models_{\mathcal{M}} \Phi$,
- iii. Φ is $\mathcal{T}$ -satisfiable if for some $\mathcal{M}$ on $\mathcal{T}$ and state s , $s \models_{\mathcal{M}} \Phi$.

Implicit here is that formulas and transition systems have common sorts. There are numerous generalizations of these notions. One extension is the notion that Φ is $\mathcal{T}$ -valid at s : $s \models_{\mathcal{T}} \Phi$ if for all models $\mathcal{M}$ on $\mathcal{T}$, $s \models_{\mathcal{M}} \Phi$. Another is the extension to sets of formulas Γ . The set Γ is $\mathcal{M}$ -true, written either as $\models_{\mathcal{M}} \Gamma$ or $\mathcal{M} \models \Gamma$, if for every $\Phi \in \Gamma$, $\models_{\mathcal{M}} \Phi$. Similarly, Γ is $\mathcal{T}$ -valid, again written either as $\mathcal{T} \models \Gamma$ or $\models_{\mathcal{T}} \Gamma$, if every formula in Γ is $\mathcal{T}$ -valid. Moreover, we say that Γ is $\mathcal{T}$ -satisfiable if every formula in Γ is simultaneously true at a state under a model on $\mathcal{T}$. A third generalization is to sets of models, or sets of transition systems Λ (of the appropriate sort). For instance, Φ is Λ -true if for all $\mathcal{M} \in \Lambda$, $\mathcal{M} \models \Phi$. These extensions may be taken together. An example is: Γ is Λ -valid (which holds if every formula in Γ is $\mathcal{T}$ -valid for every $\mathcal{T} \in \Lambda$). Finally, when Λ is the family of all transition systems of an appropriate sort, it is dropped as an index from the terms Λ -valid and Λ -satisfiable: Φ of sort $\mathcal{L}$ is *valid* if it is $\mathcal{T}$ -valid for every $\mathcal{T}$ of sort $\mathcal{L}$.

Example. When $\mathcal{M}$ is the intended modal model for while programs, $\models_{\mathcal{M}} \Phi \rightarrow [p]\Psi$ expresses that p is partially correct with respect to Φ and Ψ . ■

Example. Suppose $\mathcal{T}$ is the transition system for the vending machine $\mathcal{V}$ and $\mathcal{M}$ is the model $(\mathcal{T}, \mathcal{V})$ with $\mathcal{V}(Q) = \{V, V_3\}$. Then

$$\begin{array}{ll} V \models_{\mathcal{M}} [2p][big]Q \wedge [1p](little)Q & \models_{\mathcal{M}} [big]Q \\ V \models_{\mathcal{T}} Q \rightarrow [1p](little)[collect]Q & \models_{\mathcal{T}} (collect)Q \rightarrow [collect]Q \end{array}$$

■

It is usual to define semantic consequence relations between sets of formulas Γ and individual formulas Φ as $\Gamma \models \Phi$ meaning Φ *follows from* Γ . Various definitions of $\models$ are possible. Especially important, when Λ is a family of models, is the *global* consequence relation $\models_{\Lambda}$ and its *local* version $\models_{\Lambda}$:

$$\begin{array}{ll} \Gamma \models_{\Lambda} \Phi & \text{if } \forall \mathcal{M} \in \Lambda. \text{ if } \mathcal{M} \models \Gamma \text{ then } \mathcal{M} \models \Phi \\ \Gamma \models_{\Lambda} \Phi & \text{if } \forall \mathcal{M} \in \Lambda. \forall s \in \mathcal{S}_{\mathcal{M}}. \text{ if } s \models_{\mathcal{M}} \Gamma \text{ then } s \models_{\mathcal{M}} \Phi \end{array}$$

(A similar pair can be defined when instead Λ is a family of transition systems.) Clearly, if $\Gamma \models_{\Lambda} \Phi$ then $\Gamma \models \Phi$. But the converse fails. For instance let Λ be the family of models on the vending machine example above. Then $Q \models_{\Lambda} [2p]Q$, but under the model $\mathcal{M}$ of the example $V \models_{\mathcal{M}} Q$ and $V \not\models_{\mathcal{M}} [2p]Q$.

2.2 Minimal modal logic

For each sort $\mathcal{L}$ the smallest modal logic, called $K_{\mathcal{L}}$, is the set of *valid* formulas (of sort $\mathcal{L}$):

$$K_{\mathcal{L}} \stackrel{def}{=} \{\Phi \mid \Phi \text{ has sort } \mathcal{L} \text{ and } \models \Phi\}$$

$K_{\mathcal{L}}$ consists of all those formulas that are true at every state under every model on every transition system (of sort $\mathcal{L}$). An alternative characterization of $K_{\mathcal{L}}$ is the set of theorems of the following axiom system where a ranges over members of $\mathcal{L}$ (and Φ, Ψ over modal formulas):

Axioms	1. Any tautology instance
	2. $[a](\Phi \rightarrow \Psi) \rightarrow ([a]\Phi \rightarrow [a]\Psi)$
Rules	MP if Φ and $\Phi \rightarrow \Psi$ then Ψ
	Nec if Φ then $[a]\Phi$

The rule Nec (necessitation) states that if Φ , the premise, then $[a]\Phi$, the conclusion. Φ is a *theorem* of this axiom system, written $\vdash \Phi$, if there is a *proof* of Φ , a finite sequence of formulas $\Phi_1, \dots, \Phi_n$ where $\Phi = \Phi_n$ and Φ_i for all $i : 1 \leq i \leq n$ is either an axiom instance or the result of an application of a rule instance to premises contained in the set $\{\Phi_1, \dots, \Phi_{i-1}\}$.

Example. $\vdash [2p](Q \wedge R) \rightarrow [2p]Q$

- | | |
|--|--------------------|
| 1. $Q \wedge R \rightarrow Q$ | tautology instance |
| 2. $[2p](Q \wedge R \rightarrow Q)$ | Nec applied to 1 |
| 3. $[2p](Q \wedge R \rightarrow Q) \rightarrow ([2p](Q \wedge R) \rightarrow [2p]Q)$ | axiom 2 instance |
| 4. $[2p](Q \wedge R) \rightarrow [2p]Q$ | MP applied to 2, 3 |

■

For each sort, this axiom system is both sound and complete:

Soundness : if $\vdash \Phi$ then $\models \Phi$
 Completeness : if $\models \Phi$ then $\vdash \Phi$

Soundness states that every provable formula is valid, and completeness that every valid formula is provable. So the set of theorems of this axiom system provide an alternative characterization of $K_{\mathcal{L}}$. Techniques for showing soundness and completeness of axiom systems are the topics of Section 6.

The axiom system for $K_{\mathcal{L}}$ supports local and global proof-theoretic consequence relations (correlates of $\vdash$ and $\models$). The formula Φ follows from the set of *assumptions* Γ , written $\Gamma \vdash \Phi$, if there is a finite sequence of formulas $\Phi_1, \dots, \Phi_n$ with $\Phi = \Phi_n$, and where each Φ_i is either an axiom instance; or a member of Γ ; or the result of MP applied to premises contained in $\{\Phi_1, \dots, \Phi_{i-1}\}$; or, finally, the result of an application of Nec to a *theorem* in $\{\Phi_1, \dots, \Phi_{i-1}\}$. Restricting the permissible application of Nec precludes derivations of $\Phi \vdash [a]\Phi$ (when Φ is not a theorem). This constraint on the application of Nec is relaxed in the definition of $\Gamma \Vdash \Phi$ which otherwise is the same. For $\Gamma \Vdash \Phi$ is to be understood differently as: Φ follows from the (extra) family of *axiom instances* Γ . As Φ is an extra axiom instance, $\Phi \Vdash [a]\Phi$. A consequence of this is that the deduction theorem holds for the local, but not the global, relation:

$$\Gamma, \Phi \vdash \Psi \text{ iff } \Gamma \vdash \Phi \rightarrow \Psi$$

Strong soundness and completeness of the axiom system for $K_{\mathcal{L}}$ is given by the next result, which is proved in Section 6.2.

2.2.1: Proposition

$\Gamma \vdash \Phi$ iff $\Gamma \models \Phi$

In general, this proposition fails for the global relations $\Vdash$ and $\models$, an example is presented later.

An alternative presentation of the local consequence relation for $K_{\mathcal{L}}$ is as a Gentzen system. Assume the usual Gentzen rules for classical propositional logic centred on the sequent $\Gamma \vdash \Phi$. There is just one extra rule for modalities:

$$\frac{\Gamma \vdash \Phi}{\{[a]\Psi \mid \Psi \in \Gamma\} \vdash [a]\Phi}$$

Surprisingly, there is no elimination rule for $[a]$. This is also a strongly sound and complete presentation of $K_{\mathcal{L}}$.

Wherever possible in the sequel the notion of sort is dropped, so we speak of the logic K rather than the family $K_{\mathcal{L}}$. Also we shall often index $\vdash$ with the name of a particular axiom system: $\vdash_J \Phi$ or $J \vdash \Phi$ abbreviates that Φ is a theorem of the system J .

Let $\overline{K}$ be the richer family of valid formulas drawn from the modal language which also includes the reverse modalities $\overline{[a]}$. A simple extension of the axiom system for K , employing the following extra axioms and rules completely characterizes this set:

$$\begin{array}{ll} \text{Axioms :} & \begin{array}{l} 3. \overline{[a]}(\Phi \rightarrow \Psi) \rightarrow (\overline{[a]}\Phi \rightarrow \overline{[a]}\Psi) \\ 4. \Phi \rightarrow [a]\overline{[a]}\Phi \\ 5. \Phi \rightarrow \overline{[a]}[a]\Phi \end{array} \\ \text{Rule} & \overline{\text{Nec}} \text{ if } \Phi \text{ then } \overline{[a]}\Phi \end{array}$$

The two schemas 4 and 5 relate the forwards and backwards modalities. The notion of a proof of Φ is suitably amended to take account of the additional axioms and rule.

2.3 Correspondence and incompleteness

Families of transition systems can be classified by the properties of their transition relations. A notable case is the transition system for while programs where, for instance, the relation $\xrightarrow{p;q}$ is defined from the pair $\xrightarrow{p}$ and $\xrightarrow{q}$. Other cases include (traditional) temporal structures whose transition relations are orderings. A simpler example is the following family: $\Lambda_0 \stackrel{\text{def}}{=} \{\mathcal{T} \mid \xrightarrow{2p} \text{ is transitive in } \mathcal{T}\}$. That is, each $\mathcal{T}$ in Λ_0 has the feature that if $s \xrightarrow{2p} s_1$ and $s_1 \xrightarrow{2p} s_2$ then also $s \xrightarrow{2p} s_2$. One task is to examine the resultant logics, the sets of Λ -valid formulas for such families Λ . These sets always contain K , minimal modal logic (of the appropriate sort). Sometimes these logics can also be characterized by the theorems of a finite axiom system. It is then usual to name them as $KA_1 \dots A_n$ where $A_1 \dots A_n$ are the additional axiom schemas.

Transition systems can also be categorized by the individual modal schemas that they validate. A straightforward instance is the family

$$\Lambda_1 \stackrel{\text{def}}{=} \{\mathcal{T} \mid \mathcal{T} \models [2p]\Phi \rightarrow [2p][2p]\Phi\}$$

So a transition system $\mathcal{T}$ belongs to Λ_1 if $[2p]\Phi \rightarrow [2p][2p]\Phi$ is $\mathcal{T}$ -valid (for every instance of Φ).

A pertinent inquiry is to discover how far these two classification methods coincide. For example, the two sets Λ_0 and Λ_1 are the *same* despite their very distinct definitions. That is, for any transition system $\mathcal{T}$ (whose sort includes $2p$):

$$\mathcal{T} \models [2p]\Phi \rightarrow [2p][2p]\Phi \text{ iff } \xrightarrow{2p} \text{ is transitive in } \mathcal{T}$$

A proof of this follows from the general *correspondence* theorem below relating validity of schemas and conditions on transition relations.

Derived transition relations labelled by words belonging to the language $\mathcal{L}^*$ can be simply defined. Assuming ε is the empty word and aw is the word whose first letter is a (in $\mathcal{L}$) and $w \in \mathcal{L}^*$ is the remainder then:

$$\begin{aligned} s &\xrightarrow{\varepsilon} s \\ s &\xrightarrow{aw} s' \text{ iff } \exists s_1. s \xrightarrow{a} s_1 \text{ and } s_1 \xrightarrow{w} s' \end{aligned}$$

Let $C(v, w', w, v')$ be the following condition on a transition system of sort $\mathcal{L}$ (where $v, w', w, v' \in \mathcal{L}^*$):

$$\text{if } s \xrightarrow{v} s_1 \text{ and } s \xrightarrow{w} s_2 \text{ then } \exists s'. s_1 \xrightarrow{w'} s' \text{ and } s_2 \xrightarrow{v'} s'$$

This is a convergence (or Church-Rosser) condition on transition systems. Here are four examples

- $C(\varepsilon, 2p, 2p2p, \varepsilon)$: $\xrightarrow{2p}$ is transitive
- $C(a, \varepsilon, a, \varepsilon)$: $\xrightarrow{a}$ is *monogenic*, if $s \xrightarrow{a} s_1$ and $s \xrightarrow{a} s_2$ then $s_1 = s_2$
- $C(\varepsilon, a, \varepsilon, \varepsilon)$: $\xrightarrow{a}$ is reflexive
- $C(a, a, \varepsilon, \varepsilon)$: $\xrightarrow{a}$ is symmetric

An interesting case arising in concurrency is $C(a, b, b, a)$, a diamond property when events a and b are causally independent. More complex conditions such as $C(a^k, b^l, c^m, d^n)$ for $k, l, m, n \in \mathbb{N}$ are covered too.

Derived modalities, indexed by words in $\mathcal{L}^*$, are definable as expected: $[\varepsilon]\Phi \stackrel{def}{=} \Phi$; $[aw]\Phi \stackrel{def}{=} [a][w]\Phi$; $\langle \varepsilon \rangle \Phi \stackrel{def}{=} \Phi$; and $\langle aw \rangle \Phi \stackrel{def}{=} \langle a \rangle \langle w \rangle \Phi$. Clearly, the modalities $[w]$, $\langle w \rangle$ receive their intended interpretation relative to the derived transition relations $\xrightarrow{w}$:

$$\begin{aligned} s \models_{\mathcal{M}} [w]\Phi &\text{ iff } \forall s'. \text{ if } s \xrightarrow{w} s' \text{ then } s' \models_{\mathcal{M}} \Phi \\ s \models_{\mathcal{M}} \langle w \rangle \Phi &\text{ iff } \exists s'. s \xrightarrow{w} s' \text{ and } s' \models_{\mathcal{M}} \Phi \end{aligned}$$

The following is a general correspondence result, based on (Lemmon 1977).

2.3.1: Theorem

$T \models \langle v \rangle [w'] \Phi \rightarrow [w] \langle v' \rangle \Phi$ iff T satisfies the condition $C(v, w', w, v')$.

Proof. Suppose $\langle v \rangle [w'] \Phi \rightarrow [w] \langle v' \rangle \Phi$ for every Φ is T -valid. Assume $s \xrightarrow{v} s_1$ and $s \xrightarrow{w} s_2$. So we need to show $s_1 \xrightarrow{w'} s'$ and $s_2 \xrightarrow{v'} s'$ for some s' . Let $\mathcal{M}$ be a modal model on T with valuation $\mathcal{V}$ having the feature $\mathcal{V}(Q) = \{t \mid s_1 \xrightarrow{w'} t\}$. We know $s \models_{\mathcal{M}} \langle v \rangle [w'] Q \rightarrow [w] \langle v' \rangle Q$. By definition of $\mathcal{V}$, $s_1 \models_{\mathcal{M}} [w'] Q$. Since $s \xrightarrow{v} s_1$, $s \models_{\mathcal{M}} \langle v \rangle [w'] Q$. Therefore s also has the property $[w] \langle v' \rangle Q$ under $\mathcal{M}$. But $s \xrightarrow{w} s_2$ and so $s_2 \models_{\mathcal{M}} \langle v' \rangle Q$. That is, for some s' , $s_2 \xrightarrow{v'} s'$ and $s' \models_{\mathcal{M}} Q$: but then by definition of $\mathcal{V}(Q)$, $s_1 \xrightarrow{w'} s'$. Suppose T has the property $C(v, w', w, v')$. To show $T \models \langle v \rangle [w'] \Phi \rightarrow [w] \langle v' \rangle \Phi$ consider any model $\mathcal{M}$ on T and any state s . Suppose $s \models_{\mathcal{M}} \langle v \rangle [w'] \Phi$, then we need to show $s \models_{\mathcal{M}} [w] \langle v' \rangle \Phi$. Assume $s \xrightarrow{w} s_2$. Because $s \models_{\mathcal{M}} \langle v \rangle [w'] \Phi$ for some s_1 , $s \xrightarrow{v} s_1$ and $s_1 \models_{\mathcal{M}} [w'] \Phi$. However by the condition $C(v, w', w, v')$ there is an s' with the features $s_1 \xrightarrow{w'} s'$ and $s_2 \xrightarrow{v'} s'$. As $s_1 \models_{\mathcal{M}} [w'] \Phi$ then $s' \models_{\mathcal{M}} \Phi$. Finally because $s_2 \xrightarrow{v'} s'$, $s_2 \models_{\mathcal{M}} \langle v' \rangle \Phi$. $\blacksquare$

Consider the four examples earlier:

- $[2p]\Phi \rightarrow [2p][2p]\Phi$ corresponds to transitivity of $\xrightarrow{2p}$
- $\langle a \rangle \Phi \rightarrow [a]\Phi$ corresponds to monogenicity of $\xrightarrow{a}$
- $[a]\Phi \rightarrow \Phi$ corresponds to reflexivity of $\xrightarrow{a}$
- $\langle a \rangle [a]\Phi \rightarrow \Phi$ corresponds to symmetry of $\xrightarrow{a}$

The concurrency case is given by $\langle a \rangle [b]\Phi \rightarrow [b]\langle a \rangle \Phi$.

There are many examples of correspondence results that are not covered by the previous theorem. The relation $\xrightarrow{a}$ is *connected* (in T) if whenever $s \xrightarrow{a} s_1$ and $s \xrightarrow{a} s_2$ then either $s_1 \xrightarrow{a} s_2$ or $s_2 \xrightarrow{a} s_1$. The modal schema that corresponds to this condition is:

$$[a]([a]\Phi \rightarrow \Psi) \vee [a]([a]\Psi \rightarrow \Phi)$$

A further example, this time of a property which is not definable within the first-order theory of transition systems, is (a relativized version of) Löb's axiom, $[a]([a]\Phi \rightarrow \Phi) \rightarrow [a]\Phi$. This corresponds to the condition that $\xrightarrow{a}$ is transitive and that there are no a^w paths, sequences of the form

$s_0 \xrightarrow{a} s_1 \xrightarrow{a} \dots \xrightarrow{a} s_i \xrightarrow{a} \dots$. Another interesting second-order case is that the relation $\xrightarrow{a}$ is the *reflexive* and *transitive* closure of $\xrightarrow{b}$:

$$s \xrightarrow{a} s' \text{ iff } \exists w \in \{b\}^*. s \xrightarrow{w} s'$$

which corresponds to the following schema Ψ :

$$([a]\Phi \rightarrow (\Phi \wedge [b][a]\Phi)) \wedge ([a](\Phi \rightarrow [b]\Phi) \rightarrow (\Phi \rightarrow [a]\Phi))$$

2.3.2: Lemma

$T \models \Psi$ iff $\xrightarrow{a}$ is the reflexive and transitive closure of $\xrightarrow{b}$ in T .

Proof. Suppose $T \models \Psi$ and $s \xrightarrow{a} s'$. Then we need to show $s \xrightarrow{w} s'$ for some $w \in \{b\}^*$. Let $\mathcal{M}$ be the model on T with $\mathcal{V}(Q) = \{t \mid s \xrightarrow{w} t \text{ and } w \in \{b\}^*\}$. By validity of Ψ , $s \models_{\mathcal{M}} [a](Q \rightarrow [b]Q) \rightarrow (Q \rightarrow [a]Q)$. A small argument shows $s \models_{\mathcal{M}} [a](Q \rightarrow [b]Q)$. Consequently $s \models_{\mathcal{M}} Q \rightarrow [a]Q$. Since $s \xrightarrow{a} s'$ it follows that $s \models_{\mathcal{M}} Q$, and so also $s \models_{\mathcal{M}} [a]Q$. Since $s \xrightarrow{a} s'$ this means $s' \models_{\mathcal{M}} Q$, but then $s \xrightarrow{w} s'$ for some $w \in \{b\}^*$ as required. Again suppose $T \models \Psi$ and now $s \xrightarrow{w} s'$ for some $w \in \{b\}^*$. Then we need to show $s \xrightarrow{a} s'$. Let $\mathcal{M}$ be the model on T with $\mathcal{V}(Q) = \{t \mid s \xrightarrow{a} t\}$. By definition $s \models_{\mathcal{M}} [a]Q$. From the validity of Ψ , $s \models_{\mathcal{M}} [a]Q \rightarrow (Q \wedge [b][a]Q)$. So $s \models_{\mathcal{M}} Q \wedge [b][a]Q$. A simple argument shows that if $s \xrightarrow{b} s_1 \xrightarrow{b} \dots \xrightarrow{b} s_n$ then for each $i : 1 \leq i \leq n$, $s_i \models_{\mathcal{M}} Q \wedge [b][a]Q$. But we know that $s \xrightarrow{w} s'$ for some $w \in \{b\}^*$. Hence $s' \models_{\mathcal{M}} Q$, and so $s \xrightarrow{a} s'$. The other half of the proof is left as an exercise. $\blacksquare$

This example has applications in epistemic logic when $[a]$ means *it is common knowledge that* and $[b]$ *it is known that*; in dynamic logic when the labels are programs and a is just b^* , in temporal logic when $[a]$ means *now or later (eventually)* and $[b]$ means *at the very next moment*. Many other cases of correspondence (for the unrelativized) modalities can be found in (van Benthem 1984).

Sound and complete axiom systems are also supported by Theorem 2.3.1. Let $G(v, w', w, v')$ be the schema $\langle v \rangle [w'] \Phi \rightarrow [w] \langle v' \rangle \Phi$. The axiom system $KG(v, w', w, v')$ characterizes the set of formulas that are $\Lambda(v, w', w, v')$ -valid when $\Lambda(v, w', w, v')$ is the family of transition systems with the property $C(v, w', w, v')$. This result, Proposition 2.3.3, is proved in Section 6.2.

2.3.3: Proposition

$\Lambda(v, w', w, v') \models \Phi$ iff $KG(v, w', w, v') \vdash \Phi$.

Strong soundness and completeness hold here too. A similar result (but not for strong completeness) is also supported by Lemma 2.3.2, as we note later when examining dynamic logic.

A sound and complete axiom system for Λ -validity does not imply that the family is modally definable. For instance, the axiom system K is also a sound and complete axiomatization of Λ -validity when Λ is restricted to the family of models with irreflexive transition relations, or when Λ is restricted to the set of image finite transition systems. This shows that irreflexivity and image finiteness are not by themselves modally definable: see (van Benthem 1983; 1984) for details of which conditions are so definable, and (Koymans 1989) for a slight extension of modal logic with a difference operator which permits irreflexivity (and other conditions) to be defined.

More surprising is that if Φ corresponds to the condition C then this does not guarantee that $K\Phi$ completely axiomatizes Λ -validity, when Λ consists of just the transition systems with the property C . An example from (van Benthem 1984) is provided by the following schema Ψ :

$$([a]\Phi \rightarrow \Phi) \wedge ([a]\langle a \rangle \Phi \rightarrow \langle a \rangle [a]\Phi) \wedge (((a)\Phi \wedge [a](\Phi \rightarrow [a]\Phi)) \rightarrow \Phi)$$

which defines the condition: $s \xrightarrow{a} s'$ iff $s = s'$. Let Λ be the family of transition systems having this property. At a first glance it would seem reasonable to conclude that the axiom system $K\Psi$ characterizes the set of Λ -valid formulas. But the formula $\Phi \leftrightarrow [a]\Phi$ which also corresponds to this condition is not derivable in $K\Psi$. The system $K\Psi$ is, therefore, *incomplete*. The problem is the second order nature of the global consequence relation, $\models$. For a sound and completeness axiomatization of Λ -validity must be able to deliver as theorems all formulas in the set $\{\Phi \mid \Gamma \models \Phi\}$ where Γ is the set of all instances of Ψ .

2.4 Dynamic logic

Dynamic logic, introduced by Pratt (1976) was the first explicitly developed modal logic of programs. Our interest is *propositional* dynamic logic due to Fischer and Ladner (1979). A *dynamic* transition system has as label set a Kleene algebra of the form $\mathcal{L} = (\mathcal{L}, ;, \cup, *)$. The labels are closed under the binary operations $;$ (sequential composition) and $\cup$ (choice), and under the unary operation $*$ (iteration). Their intended meanings are given by imposing closure conditions on the transition relations of a dynamic system. Sequentially composed labels obey the condition: $s \xrightarrow{a;b} t$ iff $s \xrightarrow{a} t$. Corresponding to this condition (in the sense of the previous section) is the schema D1 below. Labels combined by choice satisfy the following requirement: $s \xrightarrow{a \cup b} t$ iff $s \xrightarrow{a} t$ or $s \xrightarrow{b} t$ which corresponds to D2. Thirdly, $\xrightarrow{a^*}$ is the reflexive and transitive closure of $\xrightarrow{a}$, a condition we

have met before. We divide the corresponding schema (Lemma 2.3.2) into two D3 and D4. The second of these is an induction schema. Dynamic logic is usually presented with a further set of labels (programs), tests: programs of the form $\Phi?$ where Φ is a formula (which may be restricted so as not to involve modalities). The transition relation $\xrightarrow{\Phi?}$ is model dependent: $s \xrightarrow{\Phi?} s$ iff $s \models_{\mathcal{M}} \Phi$ (where $\mathcal{M}$ is the dynamic model). Nevertheless D5 corresponds to this condition.

- D1. $[a; b]\Phi \leftrightarrow [a][b]\Phi$
- D2. $[a \cup b]\Phi \leftrightarrow [a]\Phi \wedge [b]\Phi$
- D3. $[a^*]\Phi \rightarrow (\Phi \wedge [a][a^*]\Phi)$
- D4. $[a^*](\Phi \rightarrow [a]\Phi) \rightarrow (\Phi \rightarrow [a^*]\Phi)$
- D5. $[\Phi?]\Psi \leftrightarrow (\Phi \rightarrow \Psi)$

Propositional dynamic logic, PDL, is the set of formulas which are true in every dynamic model (of the appropriate sort). When arbitrary modal formulas can be tests then the notion of sort is defined by mutual recursion (on formulas and labels). Let Λ be the family of dynamic models. The axiom system $KD1 - D5$ is a sound and complete axiomatization of PDL.

2.4.1: Proposition

$KD1 - D5 \vdash \Phi$ iff $\Lambda \models \Phi$

A proof of this result, due to Kozen and Parikh (1981), is presented in Section 6.2. Strong completeness fails because of the second-order nature of the axioms D3 and D4. For instance, let Γ be the set of formulas $\{[a]^n Q \mid n \geq 0\}$. Although $\Gamma \models [a^*]Q$ it is not the case that $\Gamma \vdash [a^*]Q$.

The set of labels in a dynamic model can be thought of as the set of regular expressions generated by the atomic labels (and the tests $\Phi?$). They correspond to iterative or flowchart program schemes. For instance, the conditional and while schemes are expressed as:

$$\begin{aligned} \text{if } \Phi \text{ then } a \text{ else } b &\stackrel{\text{def}}{=} (\Phi?; a) \cup (\neg\Phi?; b) \\ \text{while } \Phi \text{ do } a &\stackrel{\text{def}}{=} (\Phi?; a)^*; \neg\Phi? \end{aligned}$$

One use is to show soundness of Hoare logic rules for while programs as the following are derived rules of dynamic logic:

$$\begin{aligned} &\text{if } \Phi \rightarrow [a]\Phi' \text{ and } \Phi' \rightarrow [b]\Psi \text{ then } \Phi \rightarrow [a; b]\Psi \\ &\text{if } \Phi \wedge \Psi \rightarrow [a]\Phi' \text{ and } \Phi \wedge \neg\Psi \rightarrow [b]\Phi' \text{ then } \Phi \rightarrow [\text{if } \Psi \text{ then } a \text{ else } b]\Phi' \\ &\text{if } \Phi \wedge \Psi \rightarrow [a]\Phi \text{ then } \Phi \rightarrow [\text{while } \Psi \text{ do } a](\Phi \wedge \neg\Psi) \end{aligned}$$

Using Theorem 2.3.1 further model conditions can be imposed. One example is deterministic dynamic logic, where each $\xrightarrow{a}$, when a is atomic,

is monogenic. This condition, as we saw, corresponds to the schema $\langle a \rangle \Phi \rightarrow [a] \Phi$ (with a atomic). Moreover adding it to $KD1 - D5$ results in an axiom system which characterizes validity with respect to deterministic dynamic systems (Ben-Ari *et al.* 1982).

Various enrichments of the label set have been studied. A curious example is the inclusion of the binary operator $\cap$ obeying the condition:

$$s \xrightarrow{a \cap b} t \text{ iff } s \xrightarrow{a} t \text{ and } s \xrightarrow{b} t$$

Now irreflexivity of $\xrightarrow{a}$ is definable by the formula $[a \cap \text{tt}] \text{ff}$. Moreover, the following formula

$$[a^*](\langle a \rangle \text{tt} \wedge [(a^*; a) \cap \text{tt}] \text{ff})$$

is only satisfiable in an infinite dynamic model—if there is an a^ω run passing through distinct states. This modal logic, therefore, fails to have the small model property (see Section 2.6).

Other enrichments of the label sets are known to yield undecidable dynamic logics. One such example is the addition of complement, $-$, fulfilling the requirement: $s \xrightarrow{-a} t$ iff $\text{not}(s \xrightarrow{a} t)$. Alternatively, the label set can be expanded to correspond to recursive procedures involving mutual calls of each other, given by the context-free languages over the atomic labels and tests. For a detailed survey of dynamic logics see (Harel 1984).

2.5 Modal algebras

Useful in the case of a while program p and a postcondition Ψ is the set of memories $\{M \mid M \models_{\mathcal{M}} [p]\Psi\}$ when $\mathcal{M}$ is the intended model. This consists of all those memories with respect to which if p is executed then it can not terminate at a memory failing Ψ . More generally, for any modal model $\mathcal{M}$ and formula Φ we can define the set $\{s \mid s \models_{\mathcal{M}} \Phi\}$ containing all the states which Φ is true at.

An alternative presentation of the meaning of modal formulas is to define the sets $\{s \mid s \models_{\mathcal{M}} \Phi\}$ directly by induction on the structure of Φ . This we now do, using the notation $\|\Phi\|_{\mathcal{V}}^T$ to denote this set when $(T, \mathcal{V})$ is the model. We drop the index T which we may assume is fixed:

$$\begin{aligned} \|Q\|_{\mathcal{V}} &= \mathcal{V}(Q) \\ \|\neg\Phi\|_{\mathcal{V}} &= \mathcal{S}_T - \|\Phi\|_{\mathcal{V}} \\ \|\Phi \wedge \Psi\|_{\mathcal{V}} &= \|\Phi\|_{\mathcal{V}} \cap \|\Psi\|_{\mathcal{V}} \\ \|[a]\Phi\|_{\mathcal{V}} &= \|[a]\|_{\mathcal{V}}^T \|\Phi\|_{\mathcal{V}} \end{aligned}$$

Here $\|[a]\|_{\mathcal{V}}^T$ transforms any set $S' \subseteq \mathcal{S}_T$ into the set $\|[a]\|_{\mathcal{V}}^T S' = \{s \mid \forall s' \in S'. \text{ if } s \xrightarrow{a} s' \text{ then } s' \in S'\}$, the *weakest liberal precondition* of S'

with respect to a . Derived operators have their expected meanings. In particular associated with $\langle a \rangle$ is the state transformer $\llbracket \langle a \rangle \rrbracket^T$ defined by $\llbracket \langle a \rangle \rrbracket^T S' = \{s \mid \exists s' \in S'. s \xrightarrow{a} s'\}$. Clearly, the two semantic definitions of the modal language agree, $s \models_{\mathcal{M}} \Phi$ iff $s \in \llbracket \Phi \rrbracket_{\mathcal{V}}^T$ when $\mathcal{M}$ is $(T, \mathcal{V})$. A simple consequence is that Φ is $\mathcal{M}$ -true when the set $\llbracket \Phi \rrbracket_{\mathcal{V}}^T$ is the full state space S_T .

A set of states $\llbracket \Phi \rrbracket_{\mathcal{V}}^T$ can be regarded as a *modal property*, a demarcation of the state space into two, those with and those without the property Φ . Modal properties form a boolean algebra, as they are closed under the operations of complementation and intersection, and obey the usual boolean axioms. The unit is the full state space while its complement, the zero, is the empty set. However, properties are also closed under the operations $\llbracket [a] \rrbracket^T$ which obey the healthiness conditions:

$$\llbracket [a] \rrbracket^T S_T = S_T \quad \llbracket [a] \rrbracket^T (S_1 \cap S_2) = \llbracket [a] \rrbracket^T S_1 \cap \llbracket [a] \rrbracket^T S_2$$

In turn $\llbracket \langle a \rangle \rrbracket^T$ complies with the dual conditions: $\llbracket \langle a \rangle \rrbracket^T \emptyset = \emptyset$ and $\llbracket \langle a \rangle \rrbracket^T (S_1 \cup S_2) = \llbracket \langle a \rangle \rrbracket^T S_1 \cup \llbracket \langle a \rangle \rrbracket^T S_2$.

Moreover, the state transformers $\llbracket [a] \rrbracket^T$ (and $\llbracket \langle a \rangle \rrbracket^T$) are *monotonic* with respect to the subset ordering on state sets: if $S_1 \subseteq S_2$ then $\llbracket [a] \rrbracket^T S_1 \subseteq \llbracket [a] \rrbracket^T S_2$. Also pertinent is under what conditions these transformers are continuous. An indexed non-empty family of state sets $\{S_i \mid i \in I\}$ is *directed* if for every $j, k \in I$ $S_j \cup S_k \subseteq S_i$ for some $i \in I$. Then $\llbracket [a] \rrbracket^T$ is *continuous* if for any directed family of subsets of S_T : $\llbracket [a] \rrbracket^T \bigcup S_i = \bigcup \llbracket [a] \rrbracket^T S_i$. Continuity of $\llbracket [a] \rrbracket^T$ corresponds to the condition that $\xrightarrow{a}$ is image finite in T .

2.5.1: Lemma

$\llbracket [a] \rrbracket^T$ is continuous iff $\xrightarrow{a}$ is image finite in T .

Proof. Suppose $\llbracket [a] \rrbracket^T$ is continuous but $\xrightarrow{a}$ is not image finite. Then for some state s the set $S' = \{s' \mid s \xrightarrow{a} s'\}$ is not finite. Let S'' be a countable subset of S' . Consider the family of sets $\{S_i^{fin} \mid S_i^{fin} = (S' - S'') \cup S_1 \text{ where } S_1 \text{ is a finite subset of } S''\}$. Clearly this family is directed. But for each member S_i^{fin} we have that $s \notin \llbracket [a] \rrbracket^T S_i^{fin}$ but $s \in \llbracket [a] \rrbracket^T S'$. Assume that $\xrightarrow{a}$ is image finite. Suppose $\{S_i \mid i \in I\}$ is a directed family of state sets. By monotonicity if $s \in \llbracket [a] \rrbracket^T S_j$ for $j \in I$ then $s \in \llbracket [a] \rrbracket^T \bigcup S_i$. If instead $s \in \llbracket [a] \rrbracket^T \bigcup S_i$ then because of image finiteness $s \in \llbracket [a] \rrbracket^T S'$ for some finite subset of $\bigcup S_i$. But then there is an S_j with $j \in I$ such that $s \in \llbracket [a] \rrbracket^T S_j$. ■

Notice that image finiteness holds when a ranges over standard while programs, but fails for constructs permitting countable nondeterminism such as random assignment.

Viewed as abstract objects subject to equational laws, rather than as sets of states, modal properties become *modal algebras*.

2.5.2: Definition

A modal algebra is a quintuple $(\mathcal{U}, 1, \sqcap, -, \{[a] \mid a \in \mathcal{L}\})$ where $(\mathcal{U}, 1, \sqcap, -)$ is a boolean algebra; and $[a] : \mathcal{U} \rightarrow \mathcal{U}$ for each $a \in \mathcal{L}$ satisfies $[a]1 = 1$ and $[a](x \sqcap y) = [a]x \sqcap [a]y$.

The unit of the boolean algebra is 1, its product is $\sqcap$, and $-$ is complementation. The sum, $\sqcup$, the dual of $\sqcap$ is defined as expected: $x \sqcup y = -(x \sqcap -y)$. Similarly, $\langle\langle a \rangle\rangle$ is the dual, $-[a]-$, of $[a]$. The sort of a modal algebra $\mathcal{U}$ is its label set $\mathcal{L}$.

Modal algebras were historically the first models of modal logics. For each modal formula Φ determines a modal equation $\Phi^+ = 1$. Suppose Φ contains at most the atomic sentences $Q_1, \dots, Q_n$ then Φ^+ is an n -ary polynomial with free variables $z_1, \dots, z_n$ which is defined inductively as follows: $Q_i^+ = z_i$; $(\neg\Phi)^+ = -\Phi^+$; $(\Phi \wedge \Psi)^+ = \Phi^+ \sqcap \Psi^+$; and $([a]\Phi)^+ = [a]\Phi^+$. A modal formula Φ of sort $\mathcal{L}$ is $\mathcal{U}$ -valid (where $\mathcal{U}$ has the same sort) when $\mathcal{U} \models \Phi^+ = 1$: that is, when all instances of the equation $\Phi^+ = 1$ are true in $\mathcal{U}$. Φ is *algebraically valid*, written $\models \Phi^+ = 1$, if it is $\mathcal{U}$ -valid for every modal algebra $\mathcal{U}$ of the appropriate sort. Algebraic validity coincides with validity (on transition systems).

2.5.3: Theorem

$\models \Phi$ iff $\models \Phi^+ = 1$.

Proof. Suppose $\models \Phi$ then Φ is a theorem of the axiom system for K (of Section 2.2). But the axioms of K are algebraically valid, and the rules preserve algebraic validity. Hence $\models \Phi^+ = 1$. For the other direction, suppose $\not\models \Phi$. Therefore $\mathcal{T} \not\models \Phi$ for some transition system $\mathcal{T}$. But each $\mathcal{T}$ determines a powerset modal algebra $\mathcal{U}(\mathcal{T}) = (2^{\mathcal{S}^{\mathcal{T}}}, \mathcal{S}_{\mathcal{T}}, \cap, -, \{\llbracket [a] \rrbracket^{\mathcal{T}} \mid a \in \mathcal{L}\})$. A small exercise shows that for every formula Ψ , $\mathcal{T} \models \Psi$ iff $\mathcal{U}(\mathcal{T}) \models \Psi^+ = 1$. ■

Associated with any transition system $\mathcal{T}$, as shown in the proof of the previous theorem, is its powerset modal algebra $\mathcal{U}(\mathcal{T})$ which validates the same modal formulas. Consequently, modal algebras provide a more general class of structures for interpreting modal languages than transition systems. As with transition systems, modal algebras can be classified in terms of conditions that they obey. Sometimes these requirements may be presented as sets of equations. A notable case is dynamic algebras which fulfill the equational versions of the axioms of dynamic logic of the previous section, (Pratt 1980a). A variant is to define the ω -continuous dynamic algebras which comply with the stronger requirement: $\langle\langle a^* \rangle\rangle x = \bigsqcup \{\langle\langle a^n \rangle\rangle x \mid n \geq 0\}$ (Kozen 1981).

Modal algebras also determine transition systems via a construction similar to the Stone representation theorem for boolean algebras. Think of elements of a modal algebra as modal properties or formulas. From these we need to construct a transition system. A state, therefore, will be an assembly of properties. As a strong indicator consider first the set $\|s\|_{\mathcal{V}}^T$ of modal properties true at state s in a model $(T, \mathcal{V})$. Clearly $\text{tt} \in \|s\|_{\mathcal{V}}^T$; $\neg\Phi \in \|s\|_{\mathcal{V}}^T$ iff $\Phi \notin \|s\|_{\mathcal{V}}^T$; and $\Phi \wedge \Psi \in \|s\|_{\mathcal{V}}^T$ iff $\Phi \in \|s\|_{\mathcal{V}}^T$ and $\Psi \in \|s\|_{\mathcal{V}}^T$. Consequently, we can identify as a state of a modal algebra $\mathcal{U}$ any set $X \subseteq \mathcal{U}$ obeying these conditions (when 1 replaces tt , $-$ replaces $\neg$, and $\sqcap$ replaces $\wedge$). Such a set is an *ultrafilter* in $\mathcal{U}$. For the transition relations again consider a modal model: if $s \xrightarrow{a} s'$ then whenever $[a]\Phi \in \|s\|_{\mathcal{V}}^T$, $\Phi \in \|s'\|_{\mathcal{V}}^T$. This is the criterion used for defining $\xrightarrow{a}$ between ultrafilters in $\mathcal{U}$ (when $[a]$ replaces $[a]$). Hence, a modal algebra $\mathcal{U}$ of sort $\mathcal{L}$ determines the transition system $T(\mathcal{U}) = (S(\mathcal{U}), \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ where $S(\mathcal{U})$ is the set of ultrafilters in $\mathcal{U}$. The Stone representation theorem for modal algebras states that $\mathcal{U}$ is isomorphic to a subalgebra of $\mathcal{U}(T(\mathcal{U}))$ (given by the morphism $\lambda x \in \mathcal{U}. \{\{X \in S(\mathcal{U}) \mid x \in X\}\}$).

But $\mathcal{U}$ and $T(\mathcal{U})$ need not validate the same formulas. Although from the Stone representation result (and the proof of Theorem 2.5.3) if $T(\mathcal{U}) \models \Phi$ then $\mathcal{U} \models \Phi^+ = 1$, the converse may fail. The problem is that $S(\mathcal{U})$ may contain non-standard states (given by non-principal ultrafilters). For instance, if $\mathcal{U}$ is a non-finite dynamic algebra then there are ultrafilters containing as subsets $\{[a^n]x \mid n \geq 0\} \cup \{-[a^*]x\}$. But in this circumstance $T(\mathcal{U})$ fails the dynamic axiom schema D3 $\wedge$ D4 (of the previous section). This mismatch between $\mathcal{U}$ and $T(\mathcal{U})$ prompted the introduction of descriptive frames (Goldblatt 1976) and dynamic spaces (Kozen 1981). Further restrictions provide dualities between modal algebras and spaces. For details see (Sambin and Vaccaro 1988). A general account of the significance of Stone duality to computation is contained in (Abramsky 1989).

2.6 Decision procedures

Besides having sound and complete axiom systems, the logics K , PDL , and $KG(v, w', w, v')$ are *decidable*: that is, there are algorithms for determining whether or not a formula is Λ -valid (or, what amounts to the same thing, Λ -satisfiable) when Λ is the appropriate set of models. There are various techniques available for showing this (besides exhibiting the particular algorithm) which we now briefly mention.

An important feature of minimal modal logic K is its possession of the *small model property*: each satisfiable formula Φ is satisfiable in a finite model whose size (the number of states) is bounded by the number of connectives in Φ . A simple proof of this follows from the modal collapse of a modal model with respect to a set of formulas. Suppose Γ is a non-

empty set of formulas and $\mathcal{M}$ a modal model (both of the same sort). Two states s and t of $\mathcal{S}_{\mathcal{M}}$ are Γ -equivalent in $\mathcal{M}$, written as $s \equiv_{\Gamma}^{\mathcal{M}} t$, if they have the same properties drawn from Γ : that is, if $\{\Phi \in \Gamma \mid s \models_{\mathcal{M}} \Phi\} = \{\Phi \in \Gamma \mid t \models_{\mathcal{M}} \Phi\}$. Let $|s|_{\Gamma}^{\mathcal{M}}$ be the equivalence class induced on $\mathcal{S}_{\mathcal{M}}$, the set $\{t \mid s \equiv_{\Gamma}^{\mathcal{M}} t\}$. The modal collapse is defined by filtering $\mathcal{M}$ through $\equiv_{\Gamma}^{\mathcal{M}}$.

2.6.1: Definition

The modal collapse of $\mathcal{M}$ with respect to Γ is the model $\mathcal{M}_{\Gamma}$ where

- i. $\mathcal{S}_{\Gamma} = \{|s|_{\Gamma}^{\mathcal{M}} \mid s \in \mathcal{S}_{\mathcal{M}}\}$,
- ii. $\mathcal{L}_{\Gamma} = \mathcal{L}_{\mathcal{M}} \cap \{a \mid a \text{ occurs in } \Gamma\}$,
- iii. $x \xrightarrow{a}_{\Gamma} y$ iff $\exists s \in x. \exists t \in y. s \xrightarrow{a} t$,
- iv. $x \in \mathcal{V}_{\Gamma}(Q)$ iff $\exists s \in x. s \in \mathcal{V}_{\mathcal{M}}(Q)$.

By definition $\mathcal{M}_{\Gamma}$ is a modal model. Moreover, irrespective of the size of $\mathcal{M}$, the model $\mathcal{M}_{\Gamma}$ only contains a finite number of states (and labels) whenever Γ is a finite set. If Γ is closed under subformulas then $\mathcal{M}_{\Gamma}$ inherits important semantic properties of $\mathcal{M}$. A set Γ is *closed under subformulas* if whenever $\neg\Phi \in \Gamma$ or $[a]\Phi \in \Gamma$ then also $\Phi \in \Gamma$, and if $\Phi \wedge \Psi \in \Gamma$ then also $\Phi \in \Gamma$ and $\Psi \in \Gamma$.

2.6.2: Lemma

If Γ is closed under subformulas and $\Phi \in \Gamma$ then, $s \models_{\mathcal{M}} \Phi$ iff $|s|_{\Gamma}^{\mathcal{M}} \models_{\mathcal{M}_{\Gamma}} \Phi$.

Proof. For $\Phi \in \Gamma$, $\Phi \in |s|_{\Gamma}^{\mathcal{M}}$ iff $\forall t \in |s|_{\Gamma}^{\mathcal{M}}. \Phi \in |t|_{\Gamma}^{\mathcal{M}}$. By definition if $\Phi \in \Gamma$ then $s \models_{\mathcal{M}} \Phi$ iff $\Phi \in |s|_{\Gamma}^{\mathcal{M}}$. So we need to show that $\Phi \in |s|_{\Gamma}^{\mathcal{M}}$ iff $|s|_{\Gamma}^{\mathcal{M}} \models_{\mathcal{M}_{\Gamma}} \Phi$ (for $\Phi \in \Gamma$). This follows by induction on the structure of Φ . The base case is by definition. The only interesting case in the induction step is $\Phi = [a]\Psi$. Suppose $[a]\Psi \in |s|_{\Gamma}^{\mathcal{M}}$. Consider any $|t|_{\Gamma}^{\mathcal{M}}$ such that $|s|_{\Gamma}^{\mathcal{M}} \xrightarrow{a}_{\Gamma} |t|_{\Gamma}^{\mathcal{M}}$. Clearly, as $s' \xrightarrow{a} t'$ for some $s' \in |s|_{\Gamma}^{\mathcal{M}}$ and $t' \in |t|_{\Gamma}^{\mathcal{M}}$ $\Psi \in |t|_{\Gamma}^{\mathcal{M}}$. The other direction is similar. $\blacksquare$

The small model property for K is now a consequence of this lemma. Suppose Φ is true at s in $\mathcal{M}$ then $|s|_{\Gamma}^{\mathcal{M}} \models_{\mathcal{M}_{\Gamma}} \Phi$ for any Γ closed under subformulas containing Φ . In particular, there is the smallest such Γ consisting of Φ and its subformulas. The size of $\mathcal{M}_{\Gamma}$ is at most 2^n where n is the number of connectives in Φ . This property guarantees that satisfiability (and hence validity) is decidable.

Lemma 2.6.2 does not imply that other modal logics have the small model property. Although $\mathcal{M}_{\Gamma}$ is always a modal model, there is no assurance, for instance in the case of *PDL* and *KG*(v, w', w, v'), that it obeys the requisite model conditions, the dynamic model conditions or the condition *C*(v, w', w, v'). Sometimes extra closure conditions on Γ suffice. Alternatively, it may be possible to show that $\mathcal{M}_{\Gamma}$ can be extended into a finite

model fulfilling the appropriate requirements. A mixture of these two is sufficient for dynamic logic (compare Section 6.2). However, as the formula $[a^*]((a)\text{tt} \wedge [(a^*; a) \cap \text{tt}] \text{ff})$, from Section 2.4, shows it is not assured that a logic does have the small model property. More powerful techniques for proving decidability include (Bull 1966), as well as reduction to a known decidability result (Gabbay 1976; Streett 1982).

Proofs of complete axiomatizations can also provide decidability results, as we illustrate in Section 6.2 for K and PDL . Simple algorithms for deciding Λ -satisfiability are given by tableau techniques, as described, for example, in (Hughes and Cresswell 1973; Fitting 1983), and for PDL by Pratt (1980b).

Another arena for decision procedures is proving properties *within* a given model and, in particular, of verifying modal properties of systems or programs structurally. For instance, suppose we are presented with a description of a transition system (such as the vending machine of Section 1.1) just in terms of the state constructors $\cdot$, $+$, and $\stackrel{def}{=}$: for simplicity it is assumed that all state constants on the right hand side of $\stackrel{def}{=}$ occur within the scope of a $\cdot$ constructor. Assume $\mathcal{M}$ is a modal model on this transition system whose valuation is $\mathcal{V}$. The following tableau system is an effective decision procedure for determining whether state E has the modal property Φ (under $\mathcal{M}$) which does not appeal to the transition relations, but instead to the state constructors. The rules, built from sequents of the form $E \vdash_{\mathcal{M}} \Phi$ (proof-theoretic analogues of $E \models_{\mathcal{M}} \Phi$), are goal directed: the premise sequent represents the goal to be achieved, the conclusions the subgoals.

$$\begin{array}{c}
 \frac{E \vdash_{\mathcal{M}} \neg\neg\Phi}{E \vdash_{\mathcal{M}} \Phi} \quad \frac{E \vdash_{\mathcal{M}} \Phi \wedge \Psi}{E \vdash_{\mathcal{M}} \Phi \quad E \vdash_{\mathcal{M}} \Psi} \quad \frac{E \vdash_{\mathcal{M}} \neg(\Phi \wedge \Psi)}{E \vdash_{\mathcal{M}} \neg\Phi} \\
 \\
 \frac{E \vdash_{\mathcal{M}} \neg(\Phi \wedge \Psi)}{E \vdash_{\mathcal{M}} \neg\Psi} \quad \frac{a.E \vdash_{\mathcal{M}} [a]\Phi}{E \vdash_{\mathcal{M}} \Phi} \quad \frac{a.E \vdash_{\mathcal{M}} \neg[a]\Phi}{E \vdash_{\mathcal{M}} \neg\Phi} \\
 \\
 \frac{E + F \vdash_{\mathcal{M}} [a]\Phi}{E \vdash_{\mathcal{M}} [a]\Phi \quad F \vdash_{\mathcal{M}} [a]\Phi} \quad \frac{U \vdash_{\mathcal{M}} \Phi \quad U \stackrel{def}{=} E}{E \vdash_{\mathcal{M}} \Phi} \\
 \\
 \frac{E + F \vdash_{\mathcal{M}} \neg[a]\Phi}{E \vdash_{\mathcal{M}} \neg[a]\Phi} \quad \frac{E + F \vdash_{\mathcal{M}} \neg[a]\Phi}{F \vdash_{\mathcal{M}} \neg[a]\Phi}
 \end{array}$$

A tableau for $E \vdash_{\mathcal{M}} \Phi$ is a maximal proof tree whose root is labelled with this sequent. Sequents labelling the immediate successors of a node are determined by an application of one of the rules. Maximality means

that no rules apply to a sequent labelling a leaf of a tableau. These rules are straightforward. A tableau is *successful* if all its leaves are successful. A leaf labelled $F \vdash_{\mathcal{M}} \Psi$ is *successful* if one of the following conditions apply: $\Psi = Q$ and $F \in \mathcal{V}(Q)$; $\Psi = \neg Q$ and $F \notin \mathcal{V}(Q)$; $\Psi = [b]\Psi'$ and $F = a.G$. A simple consequence is that $E \models_{\mathcal{M}} \Phi$ iff $E \vdash_{\mathcal{M}} \Phi$ has a successful tableau. This system provides an instance of *structured* model checking. For extensions to concurrency see (Winskel 1985; Stirling 1987).

3 Propositional Temporal Logics

3.1 Between modal and temporal logics

Transition systems, as emphasized in the Introduction, are suitable vehicles for defining the meanings of programs. They are also primary structures for interpreting modal logics, carrying through that fundamental idea of an action as a state transformer. However for many computational systems, especially when they involve concurrency, their on-going behaviour as sequences of actions or state changes is more important. Temporal logics are designed to reason about such behaviour. Initially, temporal logics were modal logics on unlabelled transition systems, whose transition relations are orderings (Prior 1957). Viewing these orderings as temporal, and the state set as consisting of times, the modality $(\)$ expresses a form of *eventually*, and $[\]$ expresses *invariantly*. A richer framework, for computational purposes, due to (Pratt 1976; Pnueli 1977; Abrahamson 1979; Lamport 1980; Emerson and Clarke 1980) amongst others, is to discern these temporal structures within transition systems themselves by extracting appropriate paths through them representing (partial) runs of a program, a net or whatever.

A *path* σ through a transition system is a finite or ω length sequence of the form $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots$ representing (part of) an ongoing behaviour from the state s_0 . The length of a path is its number of transitions (so a shortest path of zero length is a singleton state). For instance, $St_i \xrightarrow{push} St_{i+1} \xrightarrow{pop} St_i$ is a path through the transition system for the stack of Section 1.1.

Suppose $\sigma = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots$ has length at least n , then for $i \leq n$:

- $\sigma(i)$ is the i th state, s_i , of σ
- σ^i is the i th suffix of σ , the subpath $s_i \xrightarrow{a_{i+1}} \dots$ starting from s_i
- $\mathcal{L}(\sigma, i)$ is the i th label a_i of σ when $i > 0$

When σ has finite length let $\sigma(*)$ be its final state. Two paths can be concatenated if the final state of one is the initial state of the other. Let

$\sigma \cdot \pi$ represent the *concatenation* of σ and π (which is only defined when $\sigma(*) = \pi(0)$).

Modal formulas can be interpreted on paths by *unravelling* transition systems. If T is $(\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ then T^+ is the transition system $(\mathcal{S}^+, \{\xrightarrow{a}_+ \mid a \in \mathcal{L}\})$ where $\mathcal{S}^+$ is the set of all *finite* length paths through T and for any $a \in \mathcal{L}$, $\sigma \xrightarrow{a}_+ \pi$ iff $\pi = \sigma \cdot s \xrightarrow{a} t$. When $\mathcal{M}$ is the modal model $(T, \mathcal{V})$ let $\mathcal{M}^+$ be the model $(T^+, \mathcal{V}^+)$ whose valuation stipulates that $\mathcal{V}^+(Q)$ for any atomic Q is the set $\{\sigma \in \mathcal{S}^+ \mid \sigma(*) \in \mathcal{V}(Q)\}$. A straightforward consequence of unravelling models is the following preservation result (justified by a routine induction on modal formulas Φ): $s \models_{\mathcal{M}} \Phi$ iff $\forall \sigma$. if $s = \sigma(*)$ then $\sigma \models_{\mathcal{M}^+} \Phi$. When the label set has structure as in dynamic systems, it is more natural to assume that $\xrightarrow{a}_+$ is only defined as above in the case that a is non-composite. Such a label represents an *uninterruptable* state change, while composite labels name sequences of such changes. For instance, the operations of dynamic systems naturally extend to (finite or ω length) paths:

$$\begin{aligned} \sigma \xrightarrow{a;b}_+ \pi & \text{ iff } \exists \sigma_1. \sigma \xrightarrow{a}_+ \sigma_1 \text{ and } \sigma_1 \xrightarrow{b}_+ \pi \\ \sigma \xrightarrow{a \cup b}_+ \pi & \text{ iff } \sigma \xrightarrow{a}_+ \pi \text{ or } \sigma \xrightarrow{b}_+ \pi \\ \sigma \xrightarrow{a^*}_+ \pi & \text{ iff } \exists w \in \{a\}^*. \sigma \xrightarrow{w}_+ \pi \end{aligned}$$

A small exercise shows that the preservation result above is invariant under this redefinition of the transition relations for composite labels in dynamic systems.

Unravelled models are richer in structure than their modal counterparts. New operators, expressing path features, which are not naturally modally interpretable can be assigned meanings. An example is an extension of dynamic logic called Process Logic (Pratt 1979; Harel *et al.* 1982) which includes the extra path operators *f*, *chop* and *suf* whose meanings on unravelled dynamic models are:

$$\begin{aligned} \sigma \models_{\mathcal{M}^+} f(\Phi) & \text{ iff } \sigma(0) \models_{\mathcal{M}^+} \Phi \\ \sigma \models_{\mathcal{M}^+} \Phi \text{ chop } \Psi & \text{ iff } \exists \sigma_1. \exists \sigma_2. \sigma = \sigma_1 \cdot \sigma_2, \sigma_1 \models_{\mathcal{M}^+} \Phi \text{ and } \sigma_2 \models_{\mathcal{M}^+} \Psi \\ \sigma \models_{\mathcal{M}^+} \Phi \text{ suf } \Psi & \text{ iff } \exists i > 0. \sigma^i \models_{\mathcal{M}^+} \Psi \text{ and } \forall k : 0 < k < i. \sigma^k \models_{\mathcal{M}^+} \Phi \end{aligned}$$

So *eventually* Φ can be expressed in this logic by the formula $\Phi \vee (\text{tsuf} \Phi)$: for σ has this property if some suffix σ^i has the property Φ .

Runs have a special status amongst paths, as they represent complete computations. It is therefore common to interpret temporal logics on this distinguished set. Modal logics can also be interpreted on these maximal length paths. When T is the transition system $(\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ let $T^\times$ be the system $(\mathcal{S}^\times, \{\xrightarrow{a}_\times \mid a \in \mathcal{L}\})$ whose state set consists of *all* runs through T and where $\sigma \xrightarrow{a}_\times \pi$ iff $\sigma(0) \xrightarrow{a} \pi(0)$. When $\mathcal{M}$ is the modal model

$(\mathcal{T}, \mathcal{V})$ let $\mathcal{M}^\times$ be the model $(\mathcal{T}^\times, \mathcal{V}^\times)$ where $\mathcal{V}^\times(Q)$ for any atomic Q is the set $\{\sigma \mid \sigma(0) \in V(Q)\}$. A consequence of this unravelling is a similar preservation result (again proved by a routine induction on Φ): $s \models_{\mathcal{M}} \Phi$ iff $\forall \sigma$. if $\sigma(0) = s$ then $\sigma \models_{\mathcal{M}^\times} \Phi$. As before, operators expressing path properties can be naturally interpreted on unravelled systems. A sample collection is:

$$\begin{aligned} \sigma \models_{\mathcal{M}^\times} (a)\Phi & \text{ iff } \sigma^1 \models_{\mathcal{M}^\times} \Phi \text{ and } \mathcal{L}(\sigma, 1) = a \\ \sigma \models_{\mathcal{M}^\times} \Phi U' \Psi & \text{ iff } \exists j > 0. \sigma^j \models_{\mathcal{M}^\times} \Psi \text{ and } \forall k : 0 < k < j. \sigma^k \models_{\mathcal{M}^\times} \Phi \\ \sigma \models_{\mathcal{M}^\times} \forall F \Phi & \text{ iff } \forall \pi. \text{ if } \pi(0) = \sigma(0) \text{ then } \exists i \geq 0. \pi^i \models_{\mathcal{M}^\times} \Phi \\ \sigma \models_{\mathcal{M}^\times} E_a \Phi & \text{ iff } \exists \pi. \mathcal{L}(\pi, 1) = a \text{ and } \pi^1 \models_{\mathcal{M}^\times} \Phi \\ \sigma \models_{\mathcal{M}^\times} G_f \Phi & \text{ iff } \forall j \geq 0. \sigma^{f(j)} \models_{\mathcal{M}^\times} \Phi \end{aligned}$$

The first example is a *relativized next* operator expressing that the initial action is a and after it Φ holds. The second is an *until* operator: $\Phi U' \Psi$ holds if eventually Ψ holds and until then Φ is true. $\forall F$ expresses a *strong eventuality*. $E_a \Phi$ expresses *somewhen* a and after it Φ . The final example defines a family of operators indexed by functions $f : \mathbb{N} \rightarrow \mathbb{N}$. For instance, $G_{\lambda n.2n}$ expresses *at every even point* Φ is true, and $G_{\lambda n.n}$ expresses *now and always*.

Example. Consider the transition system which is the union of the vending machine and the stack of Section 1.1. Let $\mathcal{M}$ be any model on this transition system and let σ and π be the following cyclic runs:

$$\begin{aligned} \sigma &= (V \xrightarrow{2p} V_1 \xrightarrow{big} V_3 \xrightarrow{collect} V)^\omega \\ \pi &= (St_{i+1} \xrightarrow{pop} \dots \xrightarrow{pop} St_0 \xrightarrow{empty} St_0 \xrightarrow{push} \dots \xrightarrow{push} St_{i+1})^\omega \end{aligned}$$

The following temporal properties are true of these runs:

$$\begin{aligned} \sigma \models_{\mathcal{M}^\times} (2p)tt & \quad \pi \models_{\mathcal{M}^\times} (pop)tt U' (empty)tt \\ \sigma \models_{\mathcal{M}^\times} \forall F(collect)tt & \quad \pi \not\models_{\mathcal{M}^\times} \forall F(pop)tt \\ \sigma \models_{\mathcal{M}^\times} G_{\lambda n.3n}(2p)tt & \quad \pi \models_{\mathcal{M}^\times} E_{2p}tt \end{aligned}$$

Reverse operators are common to temporal logics. A more general notion of path through a transition system does not assume an initial state: a path may now have the form $\dots \xrightarrow{a_{-1}} s_{-1} \xrightarrow{a_0} s_0 \xrightarrow{a_1} \dots$. The notion of run is also generalized to those paths that are maximal in both forwards and backwards directions. Instead of employing runs and their suffixes an alternative semantics appeals to pairs (σ, i) where σ is a run and $i \in \mathbb{N}$ (or $\mathbb{Z}$) is an index. The pair σ, i (we drop the brackets) should be read as σ at its i th point. This is different from $\sigma(i)$, the i th state of σ , and σ^i , its i th

suffix. Both forward and reverse operators can now be interpreted in the same framework, as the following examples illustrate:

$$\begin{aligned} \sigma, i \models_{\mathcal{M}} \overline{(a)}\Phi & \text{ iff } \mathcal{L}(\sigma, i) = a \text{ and } \sigma, i-1 \models_{\mathcal{M}} \Phi \\ \sigma, i \models_{\mathcal{M}} \forall G\Phi & \text{ iff } \forall \pi. \text{ if } \pi(j) = \sigma(i) \text{ then } \forall k \leq j. \pi, k \models_{\mathcal{M}} \Phi \\ \sigma, i \models_{\mathcal{M}} \Phi U' \Psi & \text{ iff } \exists j > i. \sigma, j \models_{\mathcal{M}} \Psi \text{ and } \forall k : i < k < j. \sigma, k \models_{\mathcal{M}} \Phi \end{aligned}$$

3.2 Basics

Unravelling focuses on paths or runs through a transition system. More explicitly, they are an ingredient of *temporal* structures, extended transition systems, Definition 1.2.1. These are pairs $(\mathcal{T}, \mathcal{R})$ where $\mathcal{T}$ is a total transition system while $\mathcal{R}$ is a non-empty set of runs through $\mathcal{T}$. We let $\mathcal{T}$ range over extended transition systems as well as transition systems. The *sort* of an extended transition system is its label set. The definition allows $\mathcal{R}$ to consist of an arbitrary set of runs. In applications it could consist of the admissible computations relative to some fairness assumptions. Such a set could for instance be generated by an automaton (see Section 3.5).

A propositional temporal language has the form:

$$\Phi ::= Q \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid Op(\Phi_1, \dots, \Phi_n)$$

where Q ranges over members of a family of atomic formulas and Op over n -ary temporal operators. The sort of a temporal language is given by the labels that the meanings of temporal operators may depend on. A temporal model for a language with sort $\mathcal{L}$ is a pair $\mathcal{M} = (\mathcal{T}, \mathcal{V})$ where $\mathcal{T}$ is an extended transition system with sort $\mathcal{L}$ and $\mathcal{V}$, as in the modal case, assigns to each atomic formula a subset of $\mathcal{S}_{\mathcal{T}}$. The satisfaction relation representing truth of a formula Φ at the index i of a run $\sigma \in \mathcal{R}$ is defined inductively using the notation $\sigma, i \models_{\mathcal{M}} \Phi$. When Φ is not true at i of σ we write $\sigma, i \not\models_{\mathcal{M}} \Phi$. We assume the general case here when there may be reverse operators. The clauses are:

$$\begin{aligned} \sigma, i \models_{\mathcal{M}} Q & \text{ iff } \sigma(i) \in \mathcal{V}(Q) \\ \sigma, i \models_{\mathcal{M}} \neg\Phi & \text{ iff } \sigma, i \not\models_{\mathcal{M}} \Phi \\ \sigma, i \models_{\mathcal{M}} \Phi \wedge \Psi & \text{ iff } \sigma, i \models_{\mathcal{M}} \Phi \text{ and } \sigma, i \models_{\mathcal{M}} \Psi \\ \sigma, i \models_{\mathcal{M}} Op(\Phi_1, \dots, \Phi_n) & \text{ iff } \dots \end{aligned}$$

Example temporal operators were given in the previous section.

In the circumstance that a temporal language does not contain reverse operators the semantic clauses can be simplified by employing suffixes σ^i of a run σ instead of the pair σ, i . Hence we pick out as a special class of structures *future time extended transition systems* whose sets of runs $\mathcal{R}$ have the two features that first every member run has an initial state, and

secondly that they are *suffix closed*: whenever $\sigma \in \mathcal{R}$ then the i th suffix $\sigma^i \in \mathcal{R}$ for each i . Now $\models_{\mathcal{M}}$ can be defined directly between runs and formulas.

$$\begin{array}{ll} \sigma \models_{\mathcal{M}} Q & \text{iff } \sigma(0) \in \mathcal{V}(Q) \\ \sigma \models_{\mathcal{M}} \neg \Phi & \text{iff } \sigma \not\models_{\mathcal{M}} \Phi \\ \sigma \models_{\mathcal{M}} \Phi \wedge \Psi & \text{iff } \sigma \models_{\mathcal{M}} \Phi \text{ and } \sigma \models_{\mathcal{M}} \Psi \\ \sigma \models_{\mathcal{M}} Op(\Phi_1, \dots, \Phi_n) & \text{iff } \dots \end{array}$$

Notation for truth and validity follows that for modal logic, Definition 2.1.1.

3.2.1: Definition

- i. Φ is $\mathcal{M}$ -true, $\models_{\mathcal{M}} \Phi$, if for all pairs σ, i , $\sigma, i \models_{\mathcal{M}} \Phi$,
- ii. Φ is T -valid, $\models_T \Phi$, if for all models $\mathcal{M}$ on T , $\models_{\mathcal{M}} \Phi$,
- iii. Φ is T -satisfiable if for some $\mathcal{M}$ on T and pair σ, i , $\sigma, i \models_{\mathcal{M}} \Phi$.

Implicit here is that the sorts of formulas and extended transition systems correspond and that quantification of runs is over the set $\mathcal{R}_{\mathcal{M}}$. As with the modal case (see Section 2.1) there are numerous extensions of these notions. First is the generalization to sets of formulas: for instance, Γ is T -satisfiable if every formula in Γ is simultaneously true at an index of a run in some model on T . Second is the extension to sets of models, or extended transition systems: for instance, Φ is Λ -valid if for every $T \in \Lambda$ Φ is T -valid. Moreover, semantic consequence relations between sets of formulas Γ and individual formulas Φ can also be defined. As in Section 2.1, these divide into *global* and *local* relations. For instance, when Λ is a family of models:

$$\begin{array}{ll} \Gamma \models_{\Lambda} \Phi & \text{iff } \forall \mathcal{M} \in \Lambda. \text{ if } \mathcal{M} \models \Gamma \text{ then } \mathcal{M} \models \Phi \\ \Gamma \models_{\Lambda} \Phi & \text{iff } \forall \mathcal{M} \in \Lambda. \forall \sigma \in \mathcal{R}_{\mathcal{M}}. \forall i. \text{ if } \sigma, i \models_{\mathcal{M}} \Gamma \text{ then } \sigma, i \models_{\mathcal{M}} \Phi \end{array}$$

Similar but simpler definitions can be given for truth, validity and consequence with respect to future time structures and their models. When Λ is the family of all (possibly future) extended transition systems of the appropriate sort it is dropped as an index from the terms Λ -valid and Λ -satisfiable: Φ of sort $\mathcal{L}$ is *valid* if it is T -valid for every (future) T of sort $\mathcal{L}$.

A useful additional generalization is that of a temporal formula being *true at a state s* under a model $\mathcal{M}$:

$$s \models_{\mathcal{M}} \Phi \text{ iff } \forall \sigma \in \mathcal{R}_{\mathcal{M}}. \forall i. \text{ if } \sigma(i) = s \text{ then } \sigma, i \models_{\mathcal{M}} \Phi$$

When $\mathcal{M}$ is future time the right hand side can be simplified to: $\forall \sigma. \text{ if } \sigma(0) = s \text{ then } \sigma \models_{\mathcal{M}} \Phi$. This abstraction permits contrasts between modal and temporal logics. As every state in $\mathcal{S}_{\mathcal{M}}$ appears in some run in $\mathcal{R}_{\mathcal{M}}$, Φ

is $\mathcal{M}$ -true just in case Φ is true at every state under $\mathcal{M}$. (In the case of a rooted extended transition system, validity may be defined in terms of truth at the root state.)

Example. Consider the stack again defined by the equations $St_0 \stackrel{def}{=} \text{empty}.St_0 + \text{push}.St_1$; $St_{i+1} \stackrel{def}{=} \text{pop}.St_i + \text{push}.St_{i+2}$. Let $\mathcal{T}$ be the future time extended transition system, representing an *active* stack, whose set of runs consist of those where *pop* happens infinitely often. Assume $\mathcal{M}$ is a model on $\mathcal{T}$ where $\mathcal{V}(Q) = \{St_{2i} \mid i \geq 0\}$. Finally, let Λ be the family of all models on this transition system.

$$\begin{array}{ll} St_{2i} \models_{\mathcal{M}} \neg(\text{pop})Q & \models_{\mathcal{M}} \forall FQ \\ St_0 \models_{\mathcal{T}} \neg(\text{pop})Q & \models_{\mathcal{T}} \forall F(\text{pop})\text{tt} \\ \{Q\} \models_{\Lambda} G_{\lambda n.n}Q & \{Q\} \not\models_{\Lambda} G_{\lambda n.n}Q \end{array}$$

Propositional temporal languages are identified by their temporal operators. The expressive power of formulas and languages (of the same sort) can therefore be compared over families of models. The following definition encompasses standard stipulations.

3.2.2: Definition

- i. $\Phi \equiv_{\mathcal{M}} \Psi$ iff $\forall \sigma. \forall i. \sigma, i \models_{\mathcal{M}} \Phi$ iff $\sigma, i \models_{\mathcal{M}} \Psi$,
- ii. $\Phi \equiv_{\Lambda} \Psi$ iff $\forall \mathcal{M} \in \Lambda. \Phi \equiv_{\mathcal{M}} \Psi$,
- iii. $TL \leq_{\Lambda} TL'$ iff $\forall \Phi \in TL. \exists \Psi \in TL'. \Phi \equiv_{\Lambda} \Psi$.

First, is the definition of when two temporal formulas are *equi-expressive* with respect to a model. A variant of this is to define $\equiv_{\mathcal{M}}$ with respect to *states* instead of run index pairs. Part ii. of the definition generalizes equi-expressiveness to families of models. The final stipulation defines when one temporal language is no more Λ -expressive than another: it is assumed that TL and TL' have the same sort and share atomic sentences. When Λ is the set of all models of the appropriate sort it is dropped as an index from $\equiv_{\Lambda}$ and $\leq_{\Lambda}$. Moreover, if TL and TL' are no more Λ -expressive than each other we write $TL =_{\Lambda} TL'$. And, finally, if $TL \leq_{\Lambda} TL'$ but $TL' \not\leq_{\Lambda} TL$ then TL is strictly less Λ -expressive than TL' , and we denote this by $TL <_{\Lambda} TL'$.

Example. Consider a very simple clock $Cl \stackrel{def}{=} \text{tick}.Cl$ and let Λ be the family of models on this transition system. Then $G_{\lambda n.n}\Phi \equiv_{\Lambda} G_{\lambda n.2n}\Phi$; as temporal distinctions collapse on such a simple structure. Assume TL contains as sole operator $G_{\lambda n.n}$, while TL' has the single until operator U' . Then $TL \leq TL'$ as $G_{\lambda n.n}\Phi \equiv \Phi \wedge \neg(\text{tt}U'\neg\Phi)$. In fact, $TL < TL'$ as is shown below.

3.3 Linear and branching time

The meaning of the modality $[a]$ in an unravelled modal model based on runs is:

$$\sigma \models_{\mathcal{M} \times} [a]\Phi \text{ iff } \forall \pi. \text{ if } \pi(0) = \sigma(0) \text{ and } \mathcal{L}(\pi, 1) = a \text{ then } \pi^1 \models_{\mathcal{M} \times} \Phi$$

Two general concepts are intertwined here: first is the idea that a run *branches* into all those runs with the same initial state and second is the notion of the subsequent *point* of a run. These aspects can be separated into the following two operators (generalizing to arbitrary temporal models):

$$\begin{aligned} \sigma, i \models_{\mathcal{M}} \forall \Phi & \text{ iff } \forall \pi. \forall j. \text{ if } \pi(j) = \sigma(i) \text{ then } \pi, j \models_{\mathcal{M}} \Phi \\ \sigma, i \models_{\mathcal{M}} (a)\Phi & \text{ iff } \mathcal{L}(\sigma, i+1) = a \text{ and } \sigma, i+1 \models_{\mathcal{M}} \Phi \end{aligned}$$

So $[a]$ can be dissected into the combination $\forall \neg(a)\neg$ and $\langle a \rangle$, its dual, into $\exists(a)$ where $\exists$ is $\neg\forall\neg$, the dual of $\forall$. Similarly the reverse modality $\overline{[a]}$ becomes the combination $\forall\neg(\overline{a})\neg$ where $\overline{a}$ is the reverse of (a) . Under this analysis modal logic can be viewed as a *relativized next* temporal logic (where there are two modes of next).

The operator $\forall$ (Emerson and Clarke 1980; Ben-Ari *et al.* 1983) expresses constrained path switching. It is a quantifier over all paths passing through a particular state. In contrast (a) does not involve path switching. This contrast is the basis for an important division of temporal logics used for describing program properties into *linear* and *branching* time. Although a traditional division of temporal logics (Prior 1967; Rescher and Urquhart 1971), its relevance to computation was primarily due to Lamport (1980). When $\mathcal{R}$ captures the potential computational behaviour of a program, linear time temporal logics reason about it in terms of the properties of these runs. However, runs in $\mathcal{R}$ can be organized into computation trees or graphs with choice points at a state s representing the different ways that computation may proceed from s . Branching time logics reason about the program in terms of the properties of these trees (with their choice points). Logically the difference between a linear and a branching time operator resides with the possibility of path switching: the semantic clause for a branching time operator may involve the constrained path switching that $\forall$ expresses.

Example. Consider the difference between the two vending machines V_1 and V_2 : $V_1 \stackrel{def}{=} 2p.big.collect.V_1 + 2p.little.collect.little.collect.V_1$ and $V_2 \stackrel{def}{=} 2p.(big.collect.V_2 + little.collect.little.collect.V_2)$. Although their runs involve the same sequences of actions, V_2 , unlike V_1 , has the property $\forall(2p)\exists(little)\mathbf{tt}$. ■

A more syntactic criterion for linearity appeals to the following infinitary language L_∞ :

$$\Phi ::= Q \mid \neg\Phi \mid \bigwedge \{\Phi_j \mid j \in J\} \mid (a)\Phi \mid \overline{(a)}\Phi$$

where J is a possibly infinite (indexed) set. The semantic clause for conjunction is as expected: $\sigma, i \models_{\mathcal{M}} \bigwedge \{\Phi_j \mid j \in J\}$ iff $\forall k \in J. \sigma, i \models_{\mathcal{M}} \Phi_k$. We say that a temporal language TL is *linear* if $TL \leq L_\infty$.

Example. The temporal language TL whose operators are $G_{\lambda n.2n}$ and O , the unrelativized next operator, is less expressive than L_∞ : $O\Phi \equiv \bigvee \{(a)\Phi \mid a \in \mathcal{L}\}$ (where $\mathcal{L}$ is the sort) while $G_{\lambda n.2n}\Phi \equiv \bigwedge \{O^{2^i}\Phi \mid i \in \mathbb{N}\}$ (where $O^0\Psi = \Psi$ and $O^{n+1}\Psi = O^n\Psi$). On the other hand if TL has as sole operator $\exists(a)$ then $TL \not\leq L_\infty$. ■

Moreover, TL is said to be a *future (past)* linear time language if it is less expressive than the sublanguage of L_∞ without the reverse operators $\overline{(a)}$ (without the (a) operators). Not every linear language is either future or past: an example is the language whose sole operator expresses the future perfect, *will have been*, $F\bar{F}$:

$$\sigma, i \models_{\mathcal{M}} F\bar{F}\Phi \quad \text{iff} \quad \exists j \geq i. \exists k \leq j. \sigma, k \models_{\mathcal{M}} \Phi$$

We use the notation $L(Op_1, \dots, Op_n)$ to name the linear time temporal language whose temporal operators are $Op_1, \dots, Op_n$. An important future instance is $L(O, U)$ where O is the unrelativized *next* operator and U is a binary *until* operator:

$$\begin{aligned} \sigma \models_{\mathcal{M}} O\Phi & \quad \text{iff} \quad \sigma^1 \models_{\mathcal{M}} \Phi \\ \sigma \models_{\mathcal{M}} \Phi U \Psi & \quad \text{iff} \quad \exists j \geq 0. \sigma^j \models_{\mathcal{M}} \Psi \text{ and } \forall k : 0 \leq k < j. \sigma^k \models_{\mathcal{M}} \Phi \end{aligned}$$

A useful derived operator is F expressing *now or eventually*: $F\Phi \stackrel{\text{def}}{=} ttU\Phi$. Its dual $(\neg F\neg)$, written as G , expresses *now and always*. Unlike U' the operator U includes the moment of evaluation as a possible point for Ψ to be true at. The logic $L(U')$ is a more succinct version of $L(O, U)$, as they are equi-expressive: for instance, $O\Phi$ is just $ttU'\Phi$. A reason for choosing U rather than U' is that the operator O is not definable in terms of U (as shown by the following lemma) a virtue according to Lamport (1983) who argues against the presence of O in temporal logics for programs.

3.3.1: Lemma

$L(U) < L(O, U)$ and $L(O, F) < L(O, U)$.

Proof. Let T be the transition system for W given by: $W \stackrel{def}{=} \text{down}.D$; $D \stackrel{def}{=} \text{down}.R$; $R \stackrel{def}{=} \text{round}.R$. Assume $\mathcal{M}$ is a model whose valuation stipulates that $\mathcal{V}(Q) = \{W, D\}$. Consider the run $\sigma = W \xrightarrow{\text{down}} D \xrightarrow{\text{down}} (R \xrightarrow{\text{round}} R)^\omega$ and its initial suffix σ^1 . A small induction on formulas $\Phi \in L(U)$ shows $\sigma \models_{\mathcal{M}} \Phi$ iff $\sigma^1 \models_{\mathcal{M}} \Phi$. But $\sigma \models_{\mathcal{M}} OQ$ and $\sigma^1 \not\models_{\mathcal{M}} OQ$. For the second half assume $Q_1 U Q_2 \equiv \Psi$ for $\Psi \in L(O, F)$. Suppose Ψ contains less than m occurrences of the operator O . Let T be the transition system generated from the following: $T_i \stackrel{def}{=} \text{up}.T_{i+1}$ for $i : 0 \leq i \leq 4m-1$; and $T_{4m} \stackrel{def}{=} \text{down}.T_{2m}$. Let $\mathcal{M}$ be a model that stipulates that $\mathcal{V}(Q_1) = \{T_i \mid i \neq 3m\}$ and $\mathcal{V}(Q_2) = \{T_{2m-1}, T_{4m}\}$. Now consider the paths $\sigma = (T_{2m} \xrightarrow{\text{up}} \dots \xrightarrow{\text{up}} T_{4m} \xrightarrow{\text{down}} T_{2m})^\omega$ and $\pi = T_0 \xrightarrow{\text{up}} \dots \xrightarrow{\text{up}} T_{2m} \cdot \sigma$. When $\Phi \in L(O, F)$ contains less than m occurrences of O , $\sigma \models_{\mathcal{M}} \Phi$ iff $\pi \models_{\mathcal{M}} \Phi$. But $\pi \models_{\mathcal{M}} Q_1 U Q_2$ even though $\sigma \not\models_{\mathcal{M}} Q_1 U Q_2$. ■

The reverses of the operators O and U are $\bar{O}$ and $\bar{U}$:

$$\begin{aligned} \sigma, i \models_{\mathcal{M}} \bar{O}\Phi & \quad \text{iff} \quad \sigma, i-1 \models_{\mathcal{M}} \Phi \\ \sigma, i \models_{\mathcal{M}} \bar{U}\Psi & \quad \text{iff} \quad \exists j \leq i. \sigma, j \models_{\mathcal{M}} \Psi \text{ and } \forall k : j < k \leq i. \sigma, k \models_{\mathcal{M}} \Phi \end{aligned}$$

$\bar{O}$ expresses *at the previous moment* while $\bar{U}\Psi$ expresses Φ *has been true since* Ψ . The reverse of F *now or previously* is $\bar{F}$ defined by $\text{tt}\bar{U}Q_1$. The previous lemma holds when these reverse operators are substituted for their forward versions. Reverse operators are a hallmark of traditional temporal logics where $\bar{O}\bar{F}$ is written as P and $\bar{U}$ as S , *since*, (Prior 1967).

As testified by the previous lemma, showing that an operator is *not* definable in a temporal language is generally much harder than showing that it is. General techniques for showing non-definability spring from Section 5 where relationships between temporal languages and other formalisms are considered. One reason that $L(O, U)$ is an interesting logic is because it is *expressively complete*. A natural question to ask is whether there are other future linear time operators (whose meanings are independent of labels) that are not definable in $L(O, U)$. Gabbay *et al.* (1980) extending (Kamp 1968) showed that $L(O, U)$ is as expressive as the first-order language the semantics these temporal formulas is couched in—a result discussed in Section 5. However, as pointed out by Wolper (1983) if attention is shifted from the metalanguage of $L(O, U)$ to ω -regular expressions as expressions of path properties then there are natural future linear time operators that are not definable in $L(O, U)$. This turns out to be equivalent, as noted in Section 5.2, to considering operators whose semantics are given in a second-order metalanguage. Wolper proved the following result.

3.3.2: Proposition

$$L(O, U) < L(O, U, G_{\lambda n, 2n}).$$

Instead Wolper proposed generating (unrelativized) future linear time operators from grammars. Transition systems with finite label sets would do instead. Suppose T is a transition system whose label set is of size n . Each state X of T determines an n -ary future linear time operator X_T as follows, where σ ranges over all the runs through T , and π over those runs whose initial state is X :

$$\sigma \models_{\mathcal{M}} X_T(\Phi_1, \dots, \Phi_n) \text{ iff } \exists \pi. \forall i > 0. \text{ if } \mathcal{L}(\pi, i) = a_k \text{ then } \sigma^{i-1} \models_{\mathcal{M}} \Phi_k$$

In general, satisfiability of formulas of the resulting temporal logics is undecidable. However, Wolper just considered a decidable subfamily of such logics, generated from regular grammars, or in this case finite state transition systems.

Example. Assume that 0 is a state which has no transitions emanating from it. Consider the transition system T given by: $X \stackrel{def}{=} a_1.a_2.0$; $Y \stackrel{def}{=} a_1.Y$; and $Z \stackrel{def}{=} a_1.Z + a_2.0$. Then $O\Phi$ is $X_T(\text{tt}, \Phi)$; $G\Phi$ is given by $Y_T(\Phi, \text{tt})$ and $\Phi U \Psi$ is $F\Psi \wedge Z_T(\Phi, \Psi)$. If instead $X \stackrel{def}{=} a_1.a_2.X$ then $X_T(\Phi, \text{tt}) \equiv G_{\lambda n.2n}\Phi$. ■

Let EL , for *extended* temporal logic (Wolper 1983), be the future time linear temporal logic $L(Op_1, \dots, Op_n, \dots)$ where $Op_1, \dots$ consists of all the regular grammar definable temporal operators. We will return to EL in Section 4.2 in a much more succinct form, as a finitary temporal language, a fixed point extension of $L(O)$. There are, of course, future linear time operators (definable in L_∞) that are not definable in EL . An example is $G_{\lambda n.2^n}$, as can be shown by the pumping lemma.

Associated with any linear time operator Op is the pair of *pure* branching time operators $\forall Op$ and $\exists Op$. For instance $\forall F$ is a strong eventually operator in contrast to $\exists F$.

$$\begin{aligned} \sigma \models_{\mathcal{M}} \forall F\Phi & \text{ iff } \forall \pi. \text{ if } \pi(0) = \sigma(0) \text{ then } \exists k \geq 0. \pi^k \models_{\mathcal{M}} \Phi \\ \sigma \models_{\mathcal{M}} \exists F\Phi & \text{ iff } \exists \pi. \pi(0) = \sigma(0) \text{ and } \exists k \geq 0. \pi^k \models_{\mathcal{M}} \Phi \end{aligned}$$

When $L(Op_1, \dots, Op_n)$ is a linear time language then $B(Op_1, \dots, Op_n)$ is its *pure* branching time version containing the temporal operators $\forall Op_i$ and $\exists Op_i$. With respect to models on unravelled total transition systems, modal logic is merely a sublogic of $B((a), \neg(a)\neg)$, a branching time next logic. Pure branching time formulas cannot distinguish paths passing through the same state at those points. That is, any such formula Φ obeys the following condition: if $\sigma(i) = \pi(j)$ then $\sigma, i \models_{\mathcal{M}} \Phi$ iff $\pi, j \models_{\mathcal{M}} \Phi$. This property generally has the consequence that a linear time logic and its branching time version are incomparable. For instance the following result is proved by Emerson and Halpern (1986).

3.3.3: Proposition

$L(O, U) \not\leq B(O, U)$, and $L(O, U) \not\leq B(O, U)$.

The future branching time language $B(O, U)$ introduced by Clarke and Emerson (1981) is called CTL (standing for *Computation Tree Logic*—as it naturally expresses tree properties). There are numerous pure branching time extensions of CTL. A very general extension is the language EB which is the pure branching version of EL . That Op is a derived operator of a linear time language does not mean that $\exists Op$ or $\forall Op$ are derivable from its pure branching time version. As pointed out by Emerson and Halpern (1986) the operator $\overset{\infty}{F}$ is definable in $L(O, U)$ as FGQ_1 (meaning *almost always*) but $B(O, U, \overset{\infty}{F})$ is strictly more expressive than $B(O, U)$.

A more general temporal logic encompassing $L(Op_1, \dots, Op_n)$ and its pure branching time version is its *full* branching time version $\forall L(Op_1, \dots, Op_n)$. This temporal logic contains the linear operators together with the branching operator $\forall$. Such logics are very expressive, allowing arbitrary embedding of linear time and branching operators. For instance, $\forall L(O, U)$ includes formulas such as $\exists F G \Phi$ as well as $\exists (F \Phi \wedge O \forall (\Phi_1 U \Psi))$: the reader is invited to work out the intuitive meaning of the latter formula. The language $\forall L(O, U)$ is called CTL* and was introduced by Clarke, Emerson, Halpern and Sistla in various papers. Standard presentations of CTL* involve a slightly unwieldy syntactic and semantics definition, distinguishing between path and state formulas—a state formula is a boolean combination of formulas of the form $\forall \Phi$ and $\exists \Phi$. The important semantic property enjoyed by a state formula was given above: a state formula cannot distinguish paths passing through the same state at those points. This property could be used instead to provide a more general distinction between these two kinds of formulas. More expressive than $\forall L(O, U)$ is the language $\forall EL$, the full branching time version of EL —this is examined in a more succinct form in Section 4.2. All full branching time languages are sublanguages of $\forall L_{\infty}$. There are various branching time logics between $B(O, U)$ and $\forall L(O, U)$, see (Emerson and Halpern 1986). These authors also show expressiveness of various temporal languages—however, their definition of a temporal model is more restricted than here as they assume that sets of paths are both fusion and limit closed, terms explained in Section 3.5.

Starting with a linear time language there is a systematic method of generating pure branching and full branching time languages. The merits of using these language for specification and verification have been widely discussed by a number of authors, initiated by Lamport (1980). Increasingly, technical criteria are adduced for preferring one kind of language over another. These include expressiveness of the particular system properties that are of interest, as well as the complexity of model checking. The consequence has been a rich set of results, some of which we will encounter

later. Finally, it should be pointed out that there are temporal languages that are neither linear nor branching. An example is the language whose sole operator is the *somewhen a* operator, E_a , defined earlier.

3.3.4: Proposition

If TL contains the operator E_a then $TL \not\subseteq \forall L_\infty$.

The proof of this is left as an exercise for the reader.

3.4 Minimal temporal logics

For each temporal language its associated minimal temporal logic is its set of *valid* formulas, consisting of those formulas that are true at every point in every run in every model (of the appropriate sort). We use the same notation for naming minimal temporal logics as for naming temporal languages. So $L(Op_1, \dots, Op_n)$ now names the minimal linear time temporal logic whose operators are $Op_1, \dots, Op_n$. Similarly, $B(Op_1, \dots, Op_n)$ and $\forall L(Op_1, \dots, Op_n)$ name the minimal pure and full branching time logics associated with $L(Op_1, \dots, Op_n)$.

A characterization of $L(O, U)$ is the set of theorems of the following axiom system:

Axioms	L1. Any tautology instance
	L2. $G(\Phi \rightarrow \Psi) \rightarrow (G\Phi \rightarrow G\Psi)$
	L3. $O\neg\Phi \leftrightarrow \neg O\Phi$
	L4. $O(\Phi \rightarrow \Psi) \rightarrow (O\Phi \rightarrow O\Psi)$
	L5. $G\Phi \rightarrow \Phi \wedge OG\Phi$
	L6. $G(\Phi \rightarrow O\Phi) \rightarrow (\Phi \rightarrow G\Phi)$
	L7. $\Phi U \Psi \rightarrow F\Psi$
	L8. $\Phi U \Psi \leftrightarrow \Psi \vee (\Phi \wedge O(\Phi U \Psi))$
Rules	MP if Φ and $\Phi \rightarrow \Psi$ then Ψ
	RG if Φ then $G\Phi$

As in Section 2.2 Φ is a *theorem* of this axiom system, written $\vdash \Phi$, if there is a *proof* of Φ . As with modal logics a proof of Φ is a finite sequence of formulas $\Phi_1, \dots, \Phi_n$ with $\Phi = \Phi_n$ and where each Φ_i is either an axiom instance or the result of an application of a rule instance to premisses belonging to the set $\{\Phi_1, \dots, \Phi_{i-1}\}$. The rule RG is reminiscent of the modal rule Nec. Moreover the relationship between G and O is similar to that between $[a^*]$ and $[a]$ in dynamic logic: hence the correspondence schemas L5 and L6 (the induction schema). We call this axiom system LT. It is a sound and complete characterization of $L(O, U)$, a result due to Gabbay *et al.* (1980), which is also proved in Section 6.3.

3.4.1: Proposition

$LT \vdash \Phi$ iff $\models \Phi$

LT also supports local and global proof-theoretic consequence relations. The formula Φ follows from the set of *assumptions* Γ , written $\Gamma \vdash \Phi$, if there is a finite sequence of formulas $\Phi_1, \dots, \Phi_n$ with $\Phi = \Phi_n$, and where each Φ_i is an instance of an axiom; or the result of an application of MP to premises in $\{\Phi_1, \dots, \Phi_{i-1}\}$; or the result of an application of RG to a *theorem* in $\{\Phi_1, \dots, \Phi_{i-1}\}$. As with the similar restriction on the application of the rule Nec, derivations of the form $\Phi \vdash G\Phi$ are precluded except when Φ is a theorem. The same relaxation of the application of RG distinguishes the local from the global relation as $\Phi \Vdash G\Phi$. Again the deduction theorem holds for the local but not the global relation (which as before is to be understood as Φ follows from the additional axioms Γ). Although LT is strongly sound (whenever $\Gamma \vdash \Phi$ then also $\Gamma \models \Phi$) the converse, strong completeness, fails for the same reason as dynamic logic. A simple counterexample is that although $\Gamma \models GQ$ when $\Gamma = \{O^n Q \mid n \geq 0\}$, we do not have that $\Gamma \vdash GQ$: for any purported proof, being finite, can only depend on a finite subset of Γ , and clearly no such subset could deliver GQ as a consequence without being inconsistent.

A characterization of the sublogic $L(O, F)$ is given by the theorems of the axiom system LT without the two axioms L7 and L8 for until. And the logic $L(O)$ is completely axiomatized by L1, L3, L4, MP and the rule RO:

$$\text{RO} \quad \text{if } \Phi \text{ then } O\Phi$$

This axiom system is also strongly sound and complete.

The reverse logics can also be characterized by axiom systems. But there is a question as to their models. If runs do not have initial states then the axiom system for $L(\overline{O}, \overline{U})$ is just the *reverse* of LT: where each temporal operator in each axiom and rule is replaced by its reverse. And similarly for sublogics. However, if runs have initial states then changes are necessary: only one half of the reverse of L3 remains, namely $\overline{O}\neg\Phi \rightarrow \neg\overline{O}\Phi$; and in the reverses of L5 and L6 the operator $\overline{O}$ should be replaced by its dual $\neg\overline{O}\neg$.

In the case of $L(O, U, \overline{O}, \overline{U})$ with respect to models with runs lacking an initial state instead of adding reverse axioms and rules we can just add a reverse, or a *mirror image*, rule. Let $\overline{\Phi}$ be the formula Φ when all its temporal operators are reversed (assuming that $\overline{\overline{O}}$ is O and $\overline{\overline{U}}$ is U). The reverse rule is:

$$\text{R}^- \quad \text{if } \Phi \text{ then } \overline{\Phi}$$

Also needed are axioms which interrelate temporal operators and their reverses:

$$\text{L9. } \Phi \leftrightarrow O\overline{O}\Phi$$

$$\text{L10. } \Phi \rightarrow F\overline{F}\Phi$$

By R^- we also have: $\Phi \leftrightarrow \overline{O}O\Phi$ and $\Phi \rightarrow \overline{F}F\Phi$. A complete axiom system for $L(O, U, \overline{O}, \overline{U})$ is given by LT together with L9, L10, and R^- . Sublogics are determined by the appropriate subsystems.

A richer logic incorporates the relativized next operators (a) . An axiomatic characterization of $L(O, U, \{(a) \mid a \in \mathcal{L}\})$ includes the following extra axiom schemas:

- L11. $(a)(\Phi \rightarrow \Psi) \rightarrow ((a)\Phi \rightarrow (a)\Psi)$
- L12. $(a)\neg\Phi \rightarrow \neg(a)\Phi$
- L13. $(a)\Phi \rightarrow O\Phi$

(A complete axiomatization depends on further axioms relating O and (a) : in the case of a finite label set $\mathcal{L}$ the extra axiom is $O\Phi \rightarrow \bigvee \{(a)\Phi \mid a \in \mathcal{L}\}$).

All the full branching time logics for any of the linear time systems above are completely axiomatized by adding the following axioms and rules to their linear time axiom system:

- | | |
|--------|--|
| Axioms | B1. $Q \rightarrow \forall Q$ Q atomic |
| | B2. $\exists Q \rightarrow Q$ Q atomic |
| | B3. $\forall\Phi \rightarrow \Phi$ |
| | B4. $\forall(\Phi \rightarrow \Psi) \rightarrow (\forall\Phi \rightarrow \forall\Psi)$ |
| | B5. $\forall\Phi \rightarrow \forall\forall\Phi$ |
| | B6. $\exists\Phi \rightarrow \forall\exists\Phi$ |
| Rule | R $\forall$ if Φ then $\forall\Phi$ |

In particular, the full branching time logic $\forall L(O, U)$ is completely characterized by LT together with these additional axioms and rule. We call this logic $\forall LT$. The following result (Stirling 1989) is proved in Section 6.3.

3.4.2: Proposition

$\forall LT \vdash \Phi$ iff $\models \Phi$.

The pure branching time logic $B(O, U)$ is $\forall LT$ restricted to pure branching time formulas.

The axiom system consisting of the axioms L1, B1-B6 and the two rules MP, R $\forall$ can be viewed as a *modal* logic with the operator $\forall$ as an unrelativized modality []. Let Λ consist of all modal models (of any sort) $\mathcal{M} = (T, \mathcal{V})$ obeying the conditions:

- i. $\mathcal{L}$ is a singleton set.
- ii. $\xrightarrow{a}$ is an equivalence relation.
- iii. if $s \in \mathcal{V}(Q)$ and $s \xrightarrow{a} s'$ then $s' \in \mathcal{V}(Q)$

The family of theorems of this modal logic then coincides with the set of Λ -true formulas. Hence the logics $\forall L(Op_1, \dots, Op_n)$ can be thought of as

modalized linear time temporal logics with $\forall$ expressing *must* and $\exists$ *may*, see (Thomason 1984) for a different perspective on such logics, and (Zanardo 1991) for an axiom system. No special axioms are needed to interrelate the $\forall$ operator and the linear time operators: we can completely separate out their independent inferential contributions. However, this is no longer true when we restrict attention to families of models satisfying various conditions.

3.5 Classes of models, automata, and correspondence

We now examine canonical representatives for future time structures. Initially attention is restricted to future time logics without temporal operators whose meanings depend on the transition labels. Let L_∞^+ be the following sublogic of L_∞ :

$$\Phi ::= Q \mid \neg\Phi \mid \bigwedge \{\Phi_j \mid j \in J\} \mid O\Phi$$

A general future time structure for L_∞^+ is the extended transition system $T_0 = (\{0\}^*, \{\xrightarrow{0}\}, \mathcal{R})$ whose set of states is the set of words on $\{0\}^*$, and where $w \xrightarrow{0} w0$ for any state w , and whose set of runs $\mathcal{R}$ comprises all runs of the form: $w \xrightarrow{0} w0 \xrightarrow{0} w00 \xrightarrow{0} \dots$. Consequently $\mathcal{R}$ is suffix closed, and so T_0 is a future time structure (and is just the unravelling of its underlying transition system). The next result shows that it is a *canonical* future time structure for L_∞^+ :

3.5.1: Lemma

If $\Phi \in L_\infty^+$ and $T_0 \models \Phi$ then $\models \Phi$.

Proof. Suppose $\not\models \Phi$. So there is a model $\mathcal{M} = (T, \mathcal{V})$ and a run $\sigma \in \mathcal{R}_T$ and an index i such that $\sigma, i \not\models_{\mathcal{M}} \Phi$. Let $\mathcal{M}' = (T_0, \mathcal{V}')$ whose valuation has the property that $0^n \in \mathcal{V}'(Q)$ iff $\sigma(n) \in \mathcal{V}(Q)$. Let π be the run $\varepsilon \xrightarrow{0} 0 \xrightarrow{0} 00 \xrightarrow{0} \dots$. A small induction on the structure of $\Psi \in L_\infty^+$ shows that $\pi^j \models_{\mathcal{M}'} \Psi$ iff $\sigma, j \models_{\mathcal{M}} \Psi$. $\blacksquare$

A small consequence of this result is that the minimal logics $L(O, U)$ and EL are characterized either as the sets of valid formulas or as the sets of formulas that are valid on all future time structures, thus justifying the dual semantics of previous sections which appeal at one time to pairs, runs and indices, and at other times to runs and their suffixes.

A natural generalization for branching time logics subsumed by $\forall L_\infty^+$ is the structure $T_\omega = (\mathbb{N}^*, \{\xrightarrow{i} \mid i \in \mathbb{N}\}, \mathcal{R})$ whose states are finite sequences of numerals and where $w \xrightarrow{i} wi$, and $\mathcal{R}$ consists of all ω length sequences of

the form: $w \xrightarrow{i_1} wi_1 \xrightarrow{i_2} wi_1i_2 \xrightarrow{i_3} \dots$. But Lemma 3.5.1 fails to generalize to $\forall L_\infty^+$ formulas. The problem is that $\mathcal{T}_\omega$ -validity is too permissive. For example the following two formulas, interrelating branching and linear time operators, are $\mathcal{T}_\omega$ -valid but not valid.

$$\begin{aligned} \forall O\Phi &\rightarrow O\forall\Phi \\ \forall G(\forall\Phi \rightarrow \exists O\forall\Phi) &\rightarrow (\forall\Phi \rightarrow \exists G\forall\Phi) \end{aligned}$$

$\mathcal{T}_\omega$ contains every run through its underlying transition system $(\mathbf{N}^*, \{\xrightarrow{i} \mid i \in \mathbf{N}\})$. A future time structure with this feature is said to be *R-generable* (as R commonly names the transition relation of a Kripke structure). Such structures are just the unravelled transition systems described in Section 3.1. Let Λ be the family of all *R-generable* structures. $\mathcal{T}_\omega$ is a canonical representative of Λ for branching time formulas.

3.5.2: Theorem

If $\Phi \in \forall L_\omega^+$ and $\mathcal{T}_\omega \models \Phi$ then $\Lambda \models \Phi$.

Proof. By similar techniques to those used in Lemma 3.5.1. ■

This result can be refined. For any $n \in \mathbf{N}$, consider $\mathcal{T}_n = (\{0, \dots, n\}^*, \{\xrightarrow{i} \mid i \leq n\}, \mathcal{R})$, mediating between $\mathcal{T}_0$ and $\mathcal{T}_\omega$, where $w \xrightarrow{i} wi$ and again $\mathcal{R}$ is all the generable runs. Let the branching degree of a formula $\Phi \in \forall L_\infty^+$ be the number of occurrences of $\forall$ in Φ . Now $\mathcal{T}_n$ is canonical with respect to Λ for any formula whose degree is no more than n .

Extended transition systems can be classified in terms of conditions on their label sets and their transition relations. But they can also be catalogued according to the properties of their run sets. The *R-generable* structures constitute one family. In fact this family Λ can also be characterized in terms of closure conditions. A little extra notation: the i th prefix, for $i \geq 0$ of a run $\sigma = s_0 \xrightarrow{a_1} s_1 \dots$ is the finite length path $\sigma[i] = s_0 \xrightarrow{a_1} \dots \xrightarrow{a_i} s_i$. We say:

- $\mathcal{R}$ is *fusion closed* if $\sigma(i) = \pi(j)$ (for σ and $\pi \in \mathcal{R}$) then $\sigma[i] \cdot \pi^j \in \mathcal{R}$
- $\mathcal{R}$ is *limit closed* if $\forall i \geq 0. \exists \pi_i$. such that $\sigma[i] \cdot \pi_i \in \mathcal{R}$ then $\sigma \in \mathcal{R}$.

Limit closure states that if every finite prefix of σ is a prefix of some run in $\mathcal{R}$ then σ is in $\mathcal{R}$. The next result is due to Emerson (1983).

3.5.3: Lemma

- i. If $T \in \Lambda$ then $\mathcal{R}_T$ is *suffix, fusion and limit closed*,
- ii. If $\mathcal{R}$ is *suffix, fusion and limit closed* then $\exists T \in \Lambda. \mathcal{R} = \mathcal{R}_T$.

Proof. Part i is straightforward. For ii suppose $\mathcal{R}$ is suffix, fusion and limit closed. Let $\mathcal{S} = \{\sigma[i] \mid \sigma \in \mathcal{R}\}$ and let $\mathcal{L}$ be the set of labels appearing in $\mathcal{R}$. For $a \in \mathcal{L}$ let $w \xrightarrow{a} w'$ when $w' = w \cdot (s \xrightarrow{a} s')$ for $w, w' \in \mathcal{S}$ and where $s \xrightarrow{a} s'$ appears within a run in $\mathcal{R}$. Clearly, $\mathcal{R}$ consists of all runs through $(\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$. ■

Notice, however, if $\mathcal{R}_T$ is suffix, fusion, and limit closed this does not guarantee that T itself is in Λ . For instance, let $T = (\{s, s_1, s_2\}, \{\xrightarrow{a}\}, \mathcal{R})$ with $s \xrightarrow{a} s_1$, $s \xrightarrow{a} s_2$, $s_1 \xrightarrow{a} s$, and $s_2 \xrightarrow{a} s$, and where $\mathcal{R}$ consists of the suffixes of $s_1 \xrightarrow{a} (s \xrightarrow{a} s_2 \xrightarrow{a} s)^\omega$. Clearly, $\mathcal{R}$ is also fusion and limit closed, but does not contain the run $(s \xrightarrow{a} s_1 \xrightarrow{a} s)^\omega$. The set $\mathcal{R}_T$ should also be *prefix* closed: if $s \xrightarrow{a} t$ and $\sigma \in \mathcal{R}_T$ with $\sigma(0) = t$ then $(s \xrightarrow{a} t) \cdot \sigma \in \mathcal{R}_T$.

Consequently, we can define families of future time extended transition systems Σ_X denoting the family fulfilling conditions X where F is fusion closure, L limit closure, and P prefix closure as in figure 6. So Σ_{FLP} is Λ and Σ the family of all future time extended transition systems. One enterprise is to axiomatize Σ' -validity for particular future temporal logics. Some results are collected in Proposition 3.5.4 below. First, the condition P makes no difference: so any axiomatization of Σ_X -validity also characterizes Σ_{XP} -validity (for formulas definable in $\forall L_\infty^+$). Second, the different closure conditions are not reflected within linear time logics (and so they can not contain formulas that correspond to them, in the sense of Section 2.3). One immediate consequence of this is that the axiom system LT is sound and complete for Λ -validity. For branching time logics the additional axiom for fusion closure is FC :

$$FC \quad \forall O\Phi \rightarrow O\forall\Phi$$

The axiom system $\forall LT + FC$ characterizes Σ_F -validity of $\forall L(O, U)$ formulas. Part iv. states that limit closure without fusion closure is uninteresting. Finally, part v offers an axiomatic characterization of Λ -validity of the pure branching time logic $B(O, U)$ employing the added axiom LC :

$$LC \quad \forall G(\forall\Phi \rightarrow \exists O\forall\Phi) \rightarrow (\forall\Phi \rightarrow \exists G\forall\Phi)$$

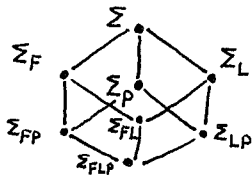


Fig. 6. Families of extended transition systems

A somewhat different axiomatization is given in (Emerson and Halpern 1982.) Most of these results are proved later in Section 6.3. Missing, however, as an open problem, is the crucial axiomatization of Λ -validity of the full branching language $\forall L(O, U)$ formulas, known as CTL^* . For instance, the following extra induction schema is also valid:

$$\forall G(\forall \Phi \rightarrow \exists OF \forall \Phi) \rightarrow (\forall \Phi \rightarrow \exists GF \forall \Phi)$$

Notice that $\forall G(\Phi \rightarrow \exists O \Phi) \rightarrow (\Phi \rightarrow \exists G \Phi)$ is *not* Λ -valid.

3.5.4: Proposition

- i. If $\Phi \in \forall L_{\infty}^+$ then $\Sigma_X \models \Phi$ iff $\Sigma_{XP} \models \Phi$,
- ii. If $\Phi \in L_{\infty}^+$ then $\Sigma \models \Phi$ iff $\Sigma_X \models \Phi$,
- iii. If $\Phi \in \forall L(O, U)$ then $\forall LT + FC \vdash \Phi$ iff $\Sigma_F \models \Phi$,
- iv. If $\Phi \in \forall L(O, U)$ then $\forall LT \vdash \Phi$ iff $\Sigma_L \models \Phi$,
- v. If $\Phi \in B(O, U)$ then $\forall LT + FC + LC \vdash \Phi$ iff $\Sigma_{FL} \models \Phi$.

Computationally, the family Σ_{FP} (or Σ_F) is important, as members naturally model computations relative to fairness assumptions. Not every run through the underlying transition system may count as admissible. Standard definitions of fairness produce fusion closed families of admissible computations. A small amendment to the definition of T_{ω} -validity, see (Courcoubetis *et al.* 1986) transforms T_{ω} into a canonical structure for Σ_{FP} -validity. A *leftist* path σ has the property that almost all its labels are 0: $\exists i \geq 0. \mathcal{L}(\sigma^i) = 0^{\omega}$. If the meaning of $\forall$ on models $\mathcal{M} = (T_{\omega}, V)$ is reinterpreted as

$$\sigma \models_{\mathcal{M}} \forall \Phi \text{ iff } \forall \text{ leftist } \sigma'. \text{ if } \sigma'(0) = \sigma(0) \text{ then } \sigma' \models_{\mathcal{M}} \Phi$$

and the definition of $\mathcal{M}$ -true is similarly changed, $\mathcal{M} \models \Phi$ if for all leftist $\sigma. \sigma \models_{\mathcal{M}} \Phi$, then T_{ω} is canonical for Σ_{FP} . This result can also be refined by introducing the meditors T_n , as in (Courcoubetis *et al.* 1986).

Automata generate future time temporal structures from transition systems. Consider them as pairs $(T, \mathcal{A})$ where T is a total transition system, and $\mathcal{A}$ an *acceptance* condition which an admissible run must fulfill. As with Büchi and Rabin acceptance, $\mathcal{A}$ could constrain the states that runs may pass through, or instead it could limit their associated labels. Another possibility is that $\mathcal{A}$ could be a formula of some logic (such as a temporal formula) that admissible runs must satisfy. In all these cases one can ask if the condition $\mathcal{A}$ corresponds, in the sense of Section 2.3, to a formula of the temporal logic $(T, \mathcal{A})$ is a structure for. More specifically, correspondence can be relativized to states of S_T . Let $\mathcal{R}(s)$ be the set of all generable runs through T starting at s , and $\mathcal{R}_{\mathcal{A}}(s)$ the admissible subset of $\mathcal{R}(s)$ according to $\mathcal{A}$. The condition $\mathcal{A}$ *corresponds* to the temporal formula Φ at state $s \in S_T$ if $\forall \sigma \in \mathcal{R}(s), \sigma \models_T \Phi$ iff $\sigma \in \mathcal{R}_{\mathcal{A}}(s)$.

A concrete case is Büchi automata where the acceptance condition $\mathcal{A} \subseteq \mathcal{S}_T$ and where $\mathcal{S}_T$ is finite. Let $\sigma(*)$ be the set of states that occur infinitely often in σ . Büchi acceptance requires that any admissible run σ has the feature that $\sigma(*) \cap \mathcal{A} \neq \emptyset$. An appropriate linear time logic for such structures is $L(\{(a) \mid a \in \mathcal{L}\}, U)$ with sole atomic sentence **tt**. This logic can be directly interpreted on Büchi automata without recourse to valuations.

Example. Consider the Büchi automaton $(T, \{X\})$ where T is given by: $X \stackrel{def}{=} c.X + b.X + a.Y$ and $Y \stackrel{def}{=} a.Y + c.Y + b.X$. This acceptance condition corresponds to the temporal formula $G((a)\mathbf{tt} \rightarrow F(b)\mathbf{tt})$ at state X or Y . ■

An alternative notion of correspondence for Büchi automata is based on the ω -language, the set of ω length words on $\mathcal{L}$, recognized by a state s in $(T, \mathcal{R})$. For each $s \in \mathcal{S}_T$ let $\mathcal{L}(s)$ be the language $\{\mathcal{L}(\sigma) \mid \sigma \in \mathcal{R}_\mathcal{A}(s)\}$, where $\mathcal{L}(\sigma)$ is the ω length sequence of labels in σ .

Example. Consider the example above (T, X) . The languages $\mathcal{L}(X)$ and $\mathcal{L}(Y)$ consist of ω length words on $\{a, b, c\}$ with the property that after any occurrence of a there is some occurrence of b which corresponds to the formula: $G((a)\mathbf{tt} \vee (b)\mathbf{tt} \vee (c)\mathbf{tt}) \wedge G((a)\mathbf{tt} \rightarrow F(b)\mathbf{tt})$. ■

However Proposition 3.3.2 shows that there are sets $\mathcal{L}(s)$ that do not correspond to $L(\{(a) \mid a \in \mathcal{L}\}, U)$ formulas. More will be said on this topic in Section 5.

3.6 Families of runs and temporal properties

An alternative presentation of the meaning of temporal formulas is to directly define the set of runs (or of pairs σ, i) a formula is true at in a model—compare Section 2.5 for the appropriate abstraction in the case of modal formulas. We use the notation $\|\Phi\|_{\mathcal{V}}^T$ to denote this set when $(T, \mathcal{V})$ is the model. For instance, the meanings of formulas drawn from $\forall L(O, U)$ with respect to future time models are defined as follows (where the fixed index T is omitted from $\|\Phi\|_{\mathcal{V}}^T$):

$$\begin{aligned}
 \|Q\|_{\mathcal{V}} &= \{\sigma \mid \sigma(0) \in \mathcal{V}(Q)\} \\
 \|\neg\Phi\|_{\mathcal{V}} &= \mathcal{R}_T - \|\Phi\|_{\mathcal{V}} \\
 \|\Phi \wedge \Psi\|_{\mathcal{V}} &= \|\Phi\|_{\mathcal{V}} \cap \|\Psi\|_{\mathcal{V}} \\
 \|O\Phi\|_{\mathcal{V}} &= \|O\|^T \|\Phi\|_{\mathcal{V}} \\
 \|\Phi U \Psi\|_{\mathcal{V}} &= \|U\|^T (\|\Phi\|_{\mathcal{V}}, \|\Psi\|_{\mathcal{V}}) \\
 \|\forall\Phi\|_{\mathcal{V}} &= \|\forall\|^T \|\Phi\|_{\mathcal{V}}
 \end{aligned}$$

Here, $|O|^T$, $|U|^T$, and $|V|^T$ are the following expected run transformers where $\mathcal{R}_1, \mathcal{R}_2 \subseteq \mathcal{R}_T$:

$$\begin{aligned} |O|^T \mathcal{R}_1 &= \{\sigma \in \mathcal{R}_T \mid \sigma^1 \in \mathcal{R}_1\} \\ |U|^T (\mathcal{R}_1, \mathcal{R}_2) &= \{\sigma \in \mathcal{R}_T \mid \exists i \geq 0. \sigma^i \in \mathcal{R}_2 \text{ and } \forall j : 0 \leq j < i. \sigma^j \in \mathcal{R}_1\} \\ |V|^T \mathcal{R}_1 &= \{\sigma \in \mathcal{R}_T \mid \forall \pi \in \mathcal{R}_T. \text{ if } \pi(0) = \sigma(0) \text{ then } \pi \in \mathcal{R}_1\} \end{aligned}$$

Derived operators induce corresponding transformers: for example, $|G|^T \mathcal{R}_1 = \{\sigma \in \mathcal{R}_T \mid \forall i \geq 0. \sigma^i \in \mathcal{R}_1\}$. Clearly the two semantic definitions of $\forall L(O, U)$ formulas agree: $\sigma \models_{\mathcal{M}} \Phi$ iff $\sigma \in |\Phi|_{\mathcal{V}}^T$ when $\mathcal{M} = (T, \mathcal{V})$. Consequently, Φ is $\mathcal{M}$ -true if it determines the admissible run space $\mathcal{R}_T$.

The run transformers are monotonic with respect to the subset ordering on run sets: if $\mathcal{R} \subseteq \mathcal{R}'$ then, for instance, $|V|^T \mathcal{R} \subseteq |V|^T \mathcal{R}'$, and $|U|^T (\mathcal{R}, \mathcal{R}_1) \subseteq |U|^T (\mathcal{R}', \mathcal{R}_1)$. Continuity with respect to an extended transition system T is a further issue. A monadic temporal operator Op is *continuous* if $|Op|^T \bigcup \mathcal{R}_i = \bigcup |Op|^T \mathcal{R}_i$ for any directed family $\bigcup \mathcal{R}_i$ of subsets of $\mathcal{R}_T$ (see Section 2.5): for polyadic operators continuity is defined componentwise. The next lemma relates continuity to other notions: $\mathcal{R}_T$ is finitary branching if for all $\sigma \in \mathcal{R}_T$ the set $\{\pi \in \mathcal{R}_T \mid \pi(0) = \sigma(0)\}$ is finite, and $\sigma \in \mathcal{R}_T$ is *eventually cyclic* if the set $\{\sigma^i \mid i \geq 0\}$ is finite.

3.6.1: Lemma

- i. $|O|^T$ is continuous,
- ii. $|V|^T$ is continuous iff $\mathcal{R}_T$ is finitary branching,
- iii. $|G|^T$ is continuous iff every $\sigma \in \mathcal{R}_T$ is eventually cyclic.

Proof. Similar to the proof of Lemma 2.5.1. ■

Just as we viewed a set of states $|\Phi|_{\mathcal{V}}^T$ as a modal property when Φ is a modal formula, we can regard a set of runs (or pairs) $|\Phi|_{\mathcal{V}}^T$ as a temporal property here. Temporal properties also form a boolean algebra whose unit is $\mathcal{R}_T$. Moreover they are closed under the operations $|Op|^T$ when Op is a temporal operator. These obey healthiness conditions corresponding to the axioms of minimal temporal logics (such as $|O|^T - \mathcal{R} = - |O|^T \mathcal{R}$ where $-$ is complementation). From these ingredients, linear and branching time temporal algebras can be defined when temporal properties are viewed as abstract objects subject to equational laws. The reader is invited to fill out details such as the relevant Stone representation theorems.

An attractive application of temporal logics is the classification of program properties. Lamport proposed a distinction between *safety* and *liveness* properties: intuitively, a safety property expresses that something bad *never* happens (expressed as a formula $\forall G \neg \Phi$) while a liveness property states that something good does *eventually* happen (expressed as a formula $\forall F \Phi$). There are various more abstract accounts of these properties as sets of runs (Alpern and Schneider 1985; Sistla 1985) as well as an alternative classification (Manna and Pnueli 1989).

3.7 Decision procedures

The linear and branching time logics $L(O, U)$, EL , $B(O, U)$, and $\forall L(O, U)$, even when restricted to R-generable or fusion closed (future time) models are decidable. However techniques for proving decidability are more intricate than for similar results for modal logics as outlined in Section 2.6.

A useful criterion for indicating that a modal logic is decidable is its possession of the small model property, given by a function $f : \mathbb{N} \rightarrow \mathbb{N}$, and a proof that any Λ -satisfiable formula of size n is true somewhere in a Λ -model whose size is no more than $f(n)$. An immediate question in the temporal case, because of the extra run component, is what counts as the size of a model. The simplest proposal is to identify it with the size of its underlying transition system. Definition 2.6.1 shows how an arbitrary modal model may be collapsed by filtering it through a set of formulas Γ : when Γ is finite so is the filtered model. Similarly, we can define the *temporal collapse* of a model.

Assume that $\mathcal{M}$ is a future time model, and Γ is a non-empty set of temporal formulas. Two states s and t of $\mathcal{S}_{\mathcal{M}}$ are Γ -equivalent in $\mathcal{M}$, $s \equiv_{\Gamma}^{\mathcal{M}} t$, if they have the same properties drawn from Γ : that is, if $\{\Phi \in \Gamma \mid s \models_{\mathcal{M}} \Phi\} = \{\Phi \in \Gamma \mid t \models_{\mathcal{M}} \Phi\}$. Let $|s|_{\Gamma}^{\mathcal{M}}$ be the equivalence class induced on $\mathcal{S}_{\mathcal{M}}$ by s with respect to $\equiv_{\Gamma}^{\mathcal{M}}$. As with the modal case, the temporal collapse of $\mathcal{M}$ is defined by filtering it through $\equiv_{\Gamma}^{\mathcal{M}}$.

3.7.1: Definition

The temporal collapse of $\mathcal{M}$ with respect to Γ is the model $\mathcal{M}_{\Gamma}$ where (assuming the index Γ abbreviates $\mathcal{M}_{\Gamma}$)

- i. $\mathcal{S}_{\Gamma} = \{|s|_{\Gamma}^{\mathcal{M}} \mid s \in \mathcal{S}_{\mathcal{M}}\}$,
- ii. $\mathcal{L}_{\Gamma} = \mathcal{L}_{\mathcal{M}} \cap \{a \mid a \text{ occurs in } \Gamma\}$,
- iii. $x \xrightarrow{a}_{\Gamma} y$ iff $\exists s \in x. \exists t \in y. s \xrightarrow{a} t$,
- iv. $\mathcal{R}_{\Gamma} = \{x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} \dots \mid \exists \sigma \in \mathcal{R}_{\mathcal{M}}. \forall i \geq 0. \sigma(i) \in x_i \text{ and } L(\sigma, i) = a_i\}$,
- v. $x \in \mathcal{V}_{\Gamma}(Q)$ iff $\exists s \in x. s \in \mathcal{V}_{\mathcal{M}}(Q)$.

By definition $\mathcal{M}_{\Gamma}$ is a future time model (as $\mathcal{R}_{\Gamma}$ is suffix closed). Moreover, whenever Γ is finite so is the underlying transition system of $\mathcal{M}_{\Gamma}$. The essential difference between this definition and that of the modal collapse is condition iv, which stipulates that every run in $\mathcal{R}_{\Gamma}$ is represented in $\mathcal{R}_{\mathcal{M}}$, and vice-versa. As with the modal case, if Γ is closed under subformulas then $\mathcal{M}_{\Gamma}$ inherits important semantic properties of $\mathcal{M}$. When σ is a run of $\mathcal{R}_{\mathcal{M}}$ let $|\sigma|_{\Gamma}$ be its representative in $\mathcal{M}_{\Gamma}$. The following result holds for future formulas of $\forall L_{\infty}$.

3.7.2: Lemma

If Γ is closed under subformulas and $\Phi \in \Gamma$ then, $\sigma \models_{\mathcal{M}} \Phi$ iff $|\sigma|_{\Gamma} \models_{\mathcal{M}_{\Gamma}} \Phi$.

Proof. By induction on $\Phi \in \Gamma$ we show the stronger: $\sigma^i \models_{\mathcal{M}} \Phi$ iff

$|\sigma|_{\Gamma}^i \models_{\mathcal{M}_{\Gamma}} \Phi$. The only interesting case is the $\forall$ operator which depends on the definition of $\mathcal{R}_{\Gamma}$. ■

The role of condition iv in Definition 3.7.1 is illustrated in the following example.

Example. Let $\mathcal{T}$ be the transition system determined by $S_i \stackrel{def}{=} up.S_{i+1}$ for $i \geq 0$, and assume that $\mathcal{R}$ is the set of all runs through $\mathcal{T}$. Moreover, let $\mathcal{M}_j$ be the model $(\mathcal{T}, \mathcal{V}_j)$ where $\mathcal{V}_j(Q) = \{S_j\}$. Clearly, for any j $S_0 \models_{\mathcal{M}_j} \forall FQ$. But consider a temporal collapse of $\mathcal{M}_j$ with respect to a finite set Γ containing $\forall FQ$ closed under subformulas. For sufficiently large j , $|S_k|_{\Gamma}^{\mathcal{M}_j} = |S_n|_{\Gamma}^{\mathcal{M}_j}$ for some k and n such that $k < n < j$. However, the run $(|S_k|_{\Gamma}^{\mathcal{M}_j} \xrightarrow{up} \dots \xrightarrow{up} |S_n|_{\Gamma}^{\mathcal{M}_j})^{\omega}$ does not belong to $\mathcal{R}_{\Gamma}$. ■

In this example the temporal collapse of an R-generable model is not itself R-generable. Lemma 3.7.2 shows, for instance, that $\forall EL$ and its sublogics (such as $\forall LT$) have the small model property (assuming that the size of a model is the size of its underlying transition system). But it does not imply that $\forall LTF$ or CTL^* (valid $\forall L(O, U)$ formulas with respect to R-generable models) also has this property. Again extra closure conditions on the set Γ may suffice (as it does in the case of $\forall LTF$ — this follows from Section 6.3). Alternatively, it may be possible to show that $\mathcal{M}_{\Gamma}$ can be extended into a finite model fulfilling the required model conditions.

Tableau methods may be used instead to prove decidability of a temporal logic. Emerson and Sistla (1984) use automata to guide the construction of tableaux in their proof that CTL^* is decidable. Reduction to SnS ($S1S$ or Büchi automata in the case of linear time logics) provides another technique for establishing decidability of temporal logics (Gabbay 1976; Harel *et al.* 1982; Sistla *et al.* 1987). Also see (Gurevitch and Shelah 1985).

Another important arena for decision procedures is proving that a temporal formula is true somewhere in a given finite (and usually R-generable model). Algorithms that decide this are known as *model checkers*: see (Clarke *et al.* 1983) for a model checker for $B(O, U)$ formulas, and (Lichtenstein and Pnueli 1984) for $L(O, U)$ formulas.

4 Modal and Temporal Mu-Calculi

4.1 Modal and temporal equations

Consider appealing to definitional equality, $\stackrel{def}{=}$, when defining modal or temporal properties. For instance, the modal equation $Z \stackrel{def}{=} \langle tick \rangle tt$ stipulates that Z expresses the same property, of having a *tick* transition, as

$\langle tick \rangle tt$. In any modal model $(\mathcal{T}, \mathcal{V})$, whose sort includes *tick*, this property is given by the set $\llbracket \langle tick \rangle tt \rrbracket^{\mathcal{T}}$, see Section 2.5 for notation. Suppose recursive equations are also permitted such as $Z \stackrel{def}{=} \langle tick \rangle Z$. Now Z can be viewed as expressing (various) properties on a model, each of which is a subset $\mathcal{S}'$ of the state set $\mathcal{S}_{\mathcal{T}}$ with the feature:

$$\mathcal{S}' = \llbracket \langle tick \rangle \rrbracket^{\mathcal{T}} \mathcal{S}' = \{s \in \mathcal{S}_{\mathcal{T}} \mid \exists s' \in \mathcal{S}'. s \xrightarrow{tick} s'\}$$

Any such set $\mathcal{S}'$ is a *fixed point* of the function $f = \lambda \mathcal{S}_1 \subseteq \mathcal{S}_{\mathcal{T}}. \llbracket \langle tick \rangle \rrbracket^{\mathcal{T}} \mathcal{S}_1$, as $f(\mathcal{S}') = \mathcal{S}'$.

Example. Both subsets, $\emptyset$ and $\{Cl\}$, of any modal model on the simple transition system generated by $Cl \stackrel{def}{=} tick.Cl$ solve $Z \stackrel{def}{=} \langle tick \rangle Z$. In the case of $Cl' \stackrel{def}{=} tick.tock.Cl'$ there is just one solution, $\emptyset$. If instead we consider this modal equation on any model for the following, $Co_0 \stackrel{def}{=} tick.Co_0 + up.Co_1$; $Co_{2i+1} \stackrel{def}{=} up.Co_{2i+2} + down.Co_{2i}$; $Co_{2i+2} \stackrel{def}{=} tick.Co_{2i+2} + up.Co_{2i+3} + down.Co_{2i+1}$ then every subset of $\{Co_{2i} \mid i \in \mathbb{N}\}$ is a solution. ■

On any modal model (of the appropriate sort) $Z \stackrel{def}{=} \langle tick \rangle Z$ not only has solutions, but also has a *smallest* and a *largest* with respect to the subset ordering on state sets (which may coincide as for Cl' above) suggesting that we can select these as the prominent solutions. The general theorem that guarantees this is due to Tarski.

4.1.1: Proposition

Let $\mathcal{P}$ be any set and assume $f : 2^{\mathcal{P}} \rightarrow 2^{\mathcal{P}}$ is monotonic (with respect to $\subseteq$). Then f has

- i. a least fixed point given by $\bigcap \{\mathcal{P}' \subseteq \mathcal{P} \mid f(\mathcal{P}') \subseteq \mathcal{P}'\}$,
- ii. a greatest fixed point given by $\bigcup \{\mathcal{P}' \subseteq \mathcal{P} \mid \mathcal{P}' \subseteq f(\mathcal{P}')\}$.

This proposition applies to $Z \stackrel{def}{=} \langle tick \rangle Z$ on any modal model as $\llbracket \langle tick \rangle \rrbracket^{\mathcal{T}}$ is a monotonic state transformer (see Section 2.5). But what properties are expressed by these two solutions? The least case is of little import as it expresses the same property as **ff**: irrespective of the state set $\mathcal{S}$, clearly $\emptyset = \{s \in \mathcal{S} \mid \exists t \in \emptyset. s \xrightarrow{tick} t\}$. More stimulating is the maximal solution, as it expresses an ability to endlessly perform the *tick* transition. A state $s_0 \in \mathcal{S}$ has this property if there is an ω length run from s_0 of the form $s_0 \xrightarrow{tick} s_1 \xrightarrow{tick} \dots$. Consequently, allowing recursive modal equations blurs the distinction between modal and temporal logics. For the maximal solution here on any modal model $\mathcal{M}$ expresses the same property as the temporal formula $\exists G(\langle tick \rangle tt)$ on the R-generable or unravelled future time model associated with $\mathcal{M}$.

A slightly more composite equation schema (where Φ does not contain Z free) is $Z \stackrel{def}{=} \Phi \vee \langle a \rangle Z$. As before the function $\lambda S' \subseteq \mathcal{S}_T. \llbracket \Phi \rrbracket_V^T \cup \llbracket \langle a \rangle \rrbracket_1^T S'$ is monotonic. On any dynamic model the least solution expresses the property $\langle a^* \rangle \Phi$ and, mixing formalisms, the maximal solution expresses $\langle a^* \rangle \Phi \vee \exists G(a)tt$.

Not every modal equation has a solution. For instance, $Z \stackrel{def}{=} \Phi \wedge [a] \neg Z$ may not, depending on the model: the problem is that the function $\lambda S' \subseteq \mathcal{S}_T. \llbracket \Phi \rrbracket_V^T \cap \llbracket [a] \rrbracket^T (\mathcal{S}_T - S')$ is not in general monotonic, and so Proposition 4.1.1 does not apply. Such equations can be ruled out, by only permitting those of the form $Z \stackrel{def}{=} \Psi$ where all occurrences of Z in Ψ lie within an even number of negations.

Proposition 4.1.1 also applies to recursive temporal equations, when $\mathcal{P}$ is the set of runs of a temporal model. For instance, the equation $Z \stackrel{def}{=} \Phi \wedge OZ$ (when Φ does not contain Z free) determines the monotonic function $\lambda \mathcal{R} \subseteq \mathcal{R}_T. \llbracket \Phi \rrbracket_V^T \cap \llbracket O \rrbracket^T \mathcal{R}$ on any future time model (T, ν) —see Section 3.5 for notation. Its maximal solution expresses $G\Phi$. Evidence for this is the *LT* theorem $\vdash G\Phi \leftrightarrow \Phi \wedge OG\Phi$. Another example is that the least solution to $Z \stackrel{def}{=} Q \vee (R \wedge \exists OZ)$ is just $\exists(RUQ)$ when R and Q are atomic. Moreover such equations can yield operators which are not definable in $\forall L(O, U)$: a simple case is that the maximal solution to $Z \stackrel{def}{=} Q \wedge OOOZ$ on any future time model defines the operator $G_{\lambda n.3n}$ (compare Proposition 3.3.2). Equations can also be used to capture families of runs in terms of their label sequences. An example is the maximal solution to $Z \stackrel{def}{=} (a)(b)(c)Z$ which defines the set of runs whose sequence of labels is $(abc)^\omega$.

Combinations of $L(O, U)$ definable operators are not given by individual temporal equations. Instead, families of equations are needed. For instance, $F(Q_1 U Q_2)$ is determined by the pair $Z_1 \stackrel{def}{=} Z_2 \vee OZ_1$ and $Z_2 \stackrel{def}{=} Q_2 \vee (Q_1 \vee OZ_2)$, both of which must be solved as least solutions. But other combinations depend on families of equations whose solutions are mixtures of least and greatest: a simple case is FGQ defined by the pair $Z_1 \stackrel{def}{=} Z_2 \vee OZ_1$ and $Z_2 \stackrel{def}{=} Q \wedge OZ_2$ where the second is to be the greatest, while the first is the least, solution. Quickly the equational format becomes cumbersome. Instead modal and temporal languages can be extended with fixed point operators μZ . (and their duals νZ .) which bind free occurrences of Z in Φ in the formula $\mu Z. \Phi$. It expresses the minimal solution to the equation $Z \stackrel{def}{=} \Phi$ where Φ may contain further fixed point operators. For instance, $F(Q_1 U Q_2)$ is given by the single formula $\mu Z. (\mu Y. Q_2 \vee (Q_1 \wedge OY)) \vee OZ$. Semantics of such languages are now examined.

4.2 The mu-calculi

Formulas of the modal and temporal mu-calculi are defined by the following abstract syntax:

$$\Phi ::= Q \mid Z \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid Op\Phi \mid \mu Z.\Phi$$

where Q ranges over atomic sentences; Z over a denumerable family of propositional variables; and Op over a family of monotonic one place modal or temporal operators². One syntactic restriction on $\mu Z.\Phi$, motivated earlier through Proposition 4.1.1, is that all free occurrences of Z in Φ lie within the scope of an even number of negations. As with $\exists Z$ and λZ , the operator $\mu Z.$ is a binder. Consequently usual terminology applies, such as that of a *closed formula* containing no free propositional variables. The dual binder $\nu Z.$ is definable:

$$\nu Z.\Phi \stackrel{def}{=} \neg\mu Z.\neg(\Phi[Z := \neg Z])$$

where $\Phi[Z := \neg Z]$ is the result of replacing all free occurrences of Z in Φ with $\neg Z$. For instance, $\nu Z.Q \wedge OZ$ abbreviates $\neg\mu Z.\neg(Q \wedge O\neg Z)$ which is equivalent to $\neg\mu Z.\neg Q \vee OZ$: the former defines GQ while the latter defines its equivalent $\neg F\neg Q$. Depending on the range of Op , various mu-calculi are determined, each of which is identified by its family of *closed* formulas:

$$\begin{aligned} \text{modal:} & \quad \{[a] \mid a \in \mathcal{L}\} \\ \text{linear time:} & \quad \{(a), O \mid a \in \mathcal{L}\} \\ \text{pure branching time:} & \quad \{\exists O, \exists(a), \forall(a) \mid a \in \mathcal{L}\} \\ \text{full branching time:} & \quad \{O, (a), \forall \mid a \in \mathcal{L}\} \end{aligned}$$

Often the relativized operators are omitted from the temporal mu-calculi. The modal mu-calculus was introduced by Kozen (1983), and in a different form by Pratt (1982) who interprets $\mu Z.$ as a least root, as in recursive function theory, instead of as a least fixed point. The temporal mu-calculi are due to Emerson and Clarke (1980), and Barringer, Kuiper, and Pnueli (1984). Mention should also be made of the mu-calculus, first-order logic extended with fixed point operators, studied by de Bakker, de Roeper, and Park. The notion of sort of a modal or temporal mu-calculus is, as previously, the label set $\mathcal{L}$ appealed to in its definition.

A model for both kinds of mu-calculi is a pair $\mathcal{M} = (T, \mathcal{V})$ where T is a transition system or a future time structure. In the case of a modal model the valuation $\mathcal{V}$ assigns to each atomic formula Q and propositional variable Z a set of states (a subset of $\mathcal{S}_T$). But when the model is temporal

²The restriction to single place operators could be relaxed—but in the sequel we only appeal to mu-calculi built from such operators.

$\mathcal{V}$ assigns a set of runs (a subset of $\mathcal{R}_T$) to each propositional variable (and, as before, sets of states to atomic sentences). The standard updating notation $\mathcal{V}[\mathcal{W}/Z]$ is assumed where $\mathcal{W}$ is either a set of runs or a set of states: $\mathcal{V}[\mathcal{W}/Z]$ is the valuation $\mathcal{V}'$ which agrees with $\mathcal{V}$ everywhere except on Z when $\mathcal{V}'(Z) = \mathcal{W}$. The set of runs (states) Φ is true at in the temporal (modal) model $(T, \mathcal{V})$ is defined inductively as the set $\models \Phi \rceil_{\mathcal{V}}$. First for atomic sentences:

$$\begin{aligned} \text{modal : } \quad \models Q \rceil_{\mathcal{V}}^T &= \mathcal{V}(Q) \\ \text{temporal : } \models Q \rceil_{\mathcal{V}}^T &= \{\sigma \in \mathcal{R}_T \mid \sigma(0) \in \mathcal{V}(Q)\} \end{aligned}$$

The rest of the clauses are given jointly, assuming that $\mathcal{W}_T$ is $\mathcal{S}_T$ in the modal case and $\mathcal{R}_T$ in the temporal case.

$$\begin{aligned} \models Z \rceil_{\mathcal{V}}^T &= \mathcal{V}(Z) \\ \models \neg \Phi \rceil_{\mathcal{V}}^T &= \mathcal{W}_T - \models \Phi \rceil_{\mathcal{V}}^T \\ \models \Phi \wedge \Psi \rceil_{\mathcal{V}}^T &= \models \Phi \rceil_{\mathcal{V}}^T \cap \models \Psi \rceil_{\mathcal{V}}^T \\ \models Op\Phi \rceil_{\mathcal{V}}^T &= \models Op \rceil_{\mathcal{V}}^T \models \Phi \rceil_{\mathcal{V}}^T \\ \models \mu Z. \Phi \rceil_{\mathcal{V}}^T &= \bigcap \{\mathcal{W}' \subseteq \mathcal{W}_T \mid \models \Phi \rceil_{\mathcal{V}[\mathcal{W}'/Z]}^T \subseteq \mathcal{W}'\} \end{aligned}$$

The clause for the least fixed point operator is directly determined from Proposition 4.1.1 as is the derived clause for νZ .

$$\models \nu Z. \Phi \rceil_{\mathcal{V}}^T = \bigcup \{\mathcal{W}' \subseteq \mathcal{W}_T \mid \mathcal{W}' \subseteq \models \Phi \rceil_{\mathcal{V}[\mathcal{W}'/Z]}^T\}$$

Notice that the meaning of a formula in a model may depend on the meanings of its subformulas in other models.

The derived notion of satisfaction, that Φ is true at a state or a run w in a model $\mathcal{M} = (T, \mathcal{V})$ is defined as expected: $w \models_{\mathcal{M}} \Phi$ if $w \in \models \Phi \rceil_{\mathcal{V}}^T$. A more direct definition of $\models_{\mathcal{M}}$ is possible using ordinal approximants. Let $\mu^0 Z. \Phi = \mathbf{ff}$ and for any ordinal κ let $\mu^{\kappa+1} Z. \Phi = \Phi[Z := \mu^{\kappa} Z. \Phi]$, and for any limit ordinal ξ let $\mu^{\xi} Z. \Phi = \bigvee \{\mu^{\kappa} Z. \Phi \mid \kappa < \xi\}$. Now it follows that $w \in \models \mu Z. \Phi \rceil_{\mathcal{V}}^T$ iff $w \models_{\mathcal{M}} \bigvee \{\mu^{\kappa} Z. \Phi \mid \kappa \in \alpha\}$ when α is the family of ordinals whose cardinality is that of $\mathcal{S}_T$ or $\mathcal{R}_T$ according to whether $\mathcal{M}$ is modal or temporal. Consequently, this shows that the linear time mu-calculus is a future time linear logic in the sense of Section 3.3 as every formula of it is equivalent to a formula of the future fragment of L_{∞} (of the same sort).

The usual semantic abstractions, Definition 2.1.1 and Definition 3.2.1, of truth and validity apply to the mu-calculi. An example of a valid formula is an unrolling of a fixed point, $\mu Z. \Phi \rightarrow \Phi[Z := \mu Z. \Phi]$. Global and local consequence relations can also be defined, as before. Richer mu-calculi employing reverse operators $\overline{[a]}$, $\overline{O}$, and $\overline{(a)}$ can also be introduced. Then in the temporal case $\models \Phi \rceil_{\mathcal{V}}^T$ is defined as a set of run index pairs, instead of as future runs.

Definition 3.2.2 stipulated when one temporal language is as expressive as another relative to a family of models. A similar definition also applies to modal languages. First Φ and Ψ are equally expressive with respect to Λ , $\Phi \equiv_{\Lambda} \Psi$, if for every model $(T, \mathcal{V}) \in \Lambda$. $\|\Phi\|_{\mathcal{V}}^T = \|\Psi\|_{\mathcal{V}}^T$. So $M_1 \leq_{\Lambda} M_2$ when $\forall \Phi \in M_1. \exists \Psi \in M_2$. such that $\Phi \equiv_{\Lambda} \Psi$. Suppose M is the standard modal language of Section 2.1 and μM the modal mu-calculus (both of the appropriate sort). The next lemma from (Kozen 1983) states that μM is more expressive than M , and that it is more expressive than PDL (see Section 2.4) relative to the family Λ of dynamic models (of the appropriate sort).

4.2.1: Lemma

$M < \mu M$, and $PDL <_{\Lambda} \mu M$.

Proof. Clearly every $\Phi \in M$ also belongs to μM . Below we show that $\mu Z.[a]Z$ is not equivalent to any modal formula. For the second part, every formula $\Phi \in PDL$ can be translated into an equivalent μM formula Φ^t as follows: $Q^t = Q$; $Z^t = Z$; $(\neg \Phi)^t = \neg \Phi^t$; $(\Phi \wedge \Psi)^t = \Phi^t \wedge \Psi^t$; if a is atomic then $([a]\Phi)^t = [a]\Phi^t$; $([a \cup b]\Phi)^t = ([a]\Phi^t) \wedge ([b]\Phi^t)$; $([\Psi?]\Phi)^t = \Psi^t \rightarrow \Phi^t$; $([a;b]\Phi)^t = ([a][b]\Phi)^t$; and the final important case $([a^*]\Phi)^t = \nu Z. \Phi^t \wedge ([a]Z)^t$ where Z is a new propositional variable. For all models $(T, \mathcal{V}) \in \Lambda$. $\|\Phi\|_{\mathcal{V}}^T = \|\Phi^t\|_{\mathcal{V}}^T$. So $PDL \leq_{\Lambda} \mu M$. To show μM is strictly more expressive than PDL (and M) consider the formula $\mu Z.[a]Z$ when a is an atomic label. This formula is true at a state s if there are no a^{ω} runs emanating from s . But consider any (dynamic) model $\mathcal{M}$ generable from the following transition system where 0 is a state with no transitions emanating from it: $St_0 \stackrel{def}{=} 0$; $St_{i+1} \stackrel{def}{=} a.St_i$. Suppose $\Phi \in PDL$ (or in M) is equivalent to $\mu Z.[a]Z$. Let Γ be the smallest set of formulas containing Φ which is closed under subformulas. Now consider the modal collapse of $\mathcal{M}$ with respect to Γ , the model $\mathcal{M}_{\Gamma}$. By Lemma 2.6.2 $St_i \models_{\mathcal{M}} \Phi$ iff $|St_i|_{\Gamma}^{\mathcal{M}} \models_{\mathcal{M}_{\Gamma}} \Phi$. But as Γ is finite, $|St_i|_{\Gamma}^{\mathcal{M}} = |St_j|_{\Gamma}^{\mathcal{M}}$ for some j and $i > j$. But then $|St_i|_{\Gamma}^{\mathcal{M}} \not\models_{\mathcal{M}_{\Gamma}} \mu Z.[a]Z$ as $|St_i|_{\Gamma}^{\mathcal{M}} \xrightarrow{a} \dots \xrightarrow{a} |St_j|_{\Gamma}^{\mathcal{M}}$. ■

Let $\mu L(O)$, and $\mu \forall L(O)$ be the future linear and full branching time mu-calculi without the relativized next operators. The next lemma states expressibility results for these mu-calculi. The most noticeable is that $\mu L(O)$ and Wolper's extended temporal logic EL , Section 3.3, are equally expressive, despite the appeal to an infinite family of temporal operators in EL . Thus $\mu L(O)$ is a very succinct version of EL .

4.2.2: Lemma

- i. $L(O, U) < \mu L(O) = EL$,
- ii. $\forall L(O, U) < \mu \forall L(O) = \forall EL$.

Sketch Proof. Part ii follows from i. First $EL \leq \mu L(O)$ is straightforward to establish as every regular grammar temporal operator is definable in $\mu L(O)$. To show $\mu L(O) \leq EL$ is much more delicate. The proof proceeds by establishing that every formula of $\mu L(O)$ is equivalent to an $S1S$ formula with one free variable (see Section 5.1). But every such $S1S$ formula is equivalent to an EL formula. ■

Elegant expressibility results relating the mu-calculi to automata and language theory can also be given. When $\mathcal{L}$ is a finite set of labels consider the following mu-calculi of sort $\mathcal{L}$: the closed formulas of μM , $\mu L((a))$, and $\mu \forall L((a))$, built from the empty set of atomic sentences. These logics can be directly interpreted on ω -words of, and ω -trees on, $\mathcal{L}$. For instance the meaning of $\mu L((a))$ formulas is given as follows where $\mathcal{V}$ assigns to each variable a subset of $\mathcal{L}^\omega$.

$$\begin{aligned} \|Z\|_{\mathcal{V}} &= \mathcal{V}(Z) \\ \|\neg\Phi\|_{\mathcal{V}} &= \mathcal{L}^\omega - \|\Phi\|_{\mathcal{V}} \\ \|\Phi \wedge \Psi\|_{\mathcal{V}} &= \|\Phi\|_{\mathcal{V}} \cap \|\Psi\|_{\mathcal{V}} \\ \|(a)\Phi\|_{\mathcal{V}} &= \{aw \mid w \in \|\Phi\|_{\mathcal{V}}\} \\ \|\mu Z. \Phi\|_{\mathcal{V}} &= \bigcap \{\mathcal{L}' \subseteq \mathcal{L}^\omega \mid \|\Phi\|_{\mathcal{V}[\mathcal{L}'/Z]} \subseteq \mathcal{L}'\} \end{aligned}$$

For any closed formula let $\|\Phi\|$ be the language $\|\Phi\|_{\mathcal{V}}$ for some $\mathcal{V}$. Each formula determines an ω -language on $\mathcal{L}$. An interesting result is that each such language is ω -regular (see Section 5.2), and that every ω -regular language is determined by some formula. Less is known about the tree languages determined by $\mu \forall L((a))$ and its sublanguages.

4.3 Monotonicity and continuity

Every closed formula of a modal or temporal mu-calculus from the previous section is equivalent to a formula in *positive* form. Positive forms are defined as follows:

$$\Phi ::= Q \mid \neg Q \mid Z \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid Op\Phi \mid \mu Z. \Phi \mid \nu Z. \Phi$$

where Op ranges over appropriate subsets and combinations of the following: $\{[a], \langle a \rangle, \forall, \exists, O, (a), \neg(a)\neg\}$. The dual of each connective is included in this stipulation: for instance, the dual of $\wedge$ is $\vee$; the dual of $[a]$ is $\langle a \rangle$; and the dual of O is itself. The equivalence follows easily by moving negations inwards and invoking dual connectives.

As in the previous section, the interpretation of a positive form Φ in a model $(T, \mathcal{V})$ is the set $\|\Phi\|_{\mathcal{V}}^T$ where dual operators receive their intended meanings. Valuations $\mathcal{V}$ form a partial order under the ordering $\sqsubseteq$:

- $\mathcal{V} \sqsubseteq \mathcal{V}'$ iff
- i. for all atomic Q . $\mathcal{V}(Q) = \mathcal{V}'(Q)$ and
 - ii. for all variables Z . $\mathcal{V}(Z) \subseteq \mathcal{V}'(Z)$

A benefit of positive forms is the following monotonicity property, showing that syntactic restrictions on $\mu Z. \Phi$ and $\nu Z. \Phi$ are no longer necessary for Proposition 4.1.1 to apply.

4.3.1: Lemma

If $\mathcal{V} \sqsubseteq \mathcal{V}'$ then $\|\Phi\|_{\mathcal{V}}^{\mathcal{T}} \subseteq \|\Phi\|_{\mathcal{V}'}^{\mathcal{T}}$.

Proof. By induction on positive forms Φ . Just one case is examined, $\Phi = \mu Z. \Psi$. Assume that $w \in \|\Phi\|_{\mathcal{V}}^{\mathcal{T}}$ but $w \notin \|\Phi\|_{\mathcal{V}'}^{\mathcal{T}}$. But then there is a set $\mathcal{W}$ where $w \notin \mathcal{W}$ and $\|\Psi\|_{\mathcal{V}'[\mathcal{W}/Z]}^{\mathcal{T}} \subseteq \mathcal{W}$. But by the induction hypothesis $\|\Psi\|_{\mathcal{V}[\mathcal{W}/Z]}^{\mathcal{T}} \subseteq \|\Psi\|_{\mathcal{V}'[\mathcal{W}/Z]}^{\mathcal{T}}$ which contradicts that $w \in \|\Phi\|_{\mathcal{V}}^{\mathcal{T}}$. ■

A non-empty set of valuations $\{\mathcal{V}_i \mid i \in I\}$ is directed if for every $\mathcal{V}_j, \mathcal{V}_k$ ($j, k \in I$) there is a $\mathcal{V}_m, m \in I$, such that $\mathcal{V}_j \sqsubseteq \mathcal{V}_m$ and $\mathcal{V}_k \sqsubseteq \mathcal{V}_m$. Every valuation in a directed family agrees on atomic sentences, but may differ on propositional variables. Clearly, any directed set $X = \{\mathcal{V}_i \mid i \in I\}$ has a least upper bound (relative to $\sqsubseteq$), the valuation $\bigsqcup X$ defined as: $\bigsqcup X(Q) = \mathcal{V}(Q)$ for any $\mathcal{V} \in X$; and $\bigsqcup X(Z) = \bigcup \{\mathcal{V}(Z) \mid \mathcal{V} \in X\}$. Given this, we can stipulate that a modal or temporal mu-calculus of positive forms, μC , is *continuous* on a structure $\mathcal{T}$ if for all non-empty directed families X and $\Phi \in \mu C$. $\|\Phi\|_{\bigsqcup X}^{\mathcal{T}} = \bigcup \{\|\Phi\|_{\mathcal{V}}^{\mathcal{T}} \mid \mathcal{V} \in X\}$.

Lemma 2.5.1 and Lemma 3.6.1 provide criteria for when modal and temporal operators are continuous. However, the presence of fixed points presents further possibilities for failing continuity as the next example illustrates.

Example. Let $\mathcal{T}$ be the structure determined by $St_i \xrightarrow{a} St_{i+1}$ for $i \in \mathbb{N}$. When viewed as a temporal structure let $\sigma_i = St_i \xrightarrow{a} \dots$ and $\mathcal{R} = \{\sigma_i \mid i \geq 0\}$. For the modal case let $\mathcal{V}_i$ be the valuation with the property $\mathcal{V}_i(Y) = \{St_0, \dots, St_i\}$, and in the temporal case let $\mathcal{V}_i(Y) = \{\sigma_0, \dots, \sigma_i\}$, and assume $\mathcal{V}_i = \mathcal{V}_j$ otherwise. Clearly $X = \{\mathcal{V}_i \mid i \geq 0\}$ is directed. When Φ is the formula $\nu Z. Y \wedge OZ$ or $\nu Z. Y \wedge [a]Z$ continuity fails (even though $\mathcal{T}$ is image finite and O is continuous) as $\|\Phi\|_{\bigsqcup X}^{\mathcal{T}} \neq \emptyset$, whereas $\|\Phi\|_{\mathcal{V}}^{\mathcal{T}} = \emptyset$ for each $\mathcal{V} \in X$. ■

The next result provides a *compactness* criterion for when μC is continuous on $\mathcal{T}$, which appeals to finite valuations: $\mathcal{V}$ is *finite* if for each variable Z the set $\mathcal{V}(Z)$ is finite.

4.3.2: Lemma

μC is continuous on T iff for all $\Phi \in \mu C$ if $w \in \llbracket \Phi \rrbracket_V^T$ then $w \in \llbracket \Phi \rrbracket_{V'}^T$ for some finite $V' \subseteq V$.

Proof. Suppose μC is continuous on T , but $w \in \llbracket \Phi \rrbracket_V^T$ and $w \notin \llbracket \Phi \rrbracket_{V'}^T$ for all finite $V' \subseteq V$. Consider $X = \{V' \mid V' \subseteq V \text{ and } V' \text{ finite}\}$. Clearly, $V = \bigsqcup X$ which contradicts continuity. Suppose μC is not continuous on T . Then for some directed X and Φ , $\llbracket \Phi \rrbracket_{\bigsqcup X}^T \neq \bigcup \{\llbracket \Phi \rrbracket_V^T \mid V \in X\}$. But this can only happen if for some w , $w \in \llbracket \Phi \rrbracket_{\bigsqcup X}^T$ and $w \notin \llbracket \Phi \rrbracket_V^T$ for each $V \in X$. Let $\{Z_1, \dots, Z_k\}$ be the free variables of Φ . Then it follows that $\bigsqcup X$ must associate a non-finite set to at least one Z_j , otherwise $w \in \llbracket \Phi \rrbracket_V^T$ for some finite $V \subseteq \bigsqcup X$. So the result follows. $\blacksquare$

When μC is continuous on T , the meaning of a fixed point formula in a model (T, V) can be computed iteratively via the approximants as then the only limit ordinal required is ω . In the previous section a syntactic approach was described. Instead, the iterations could be done semantically as follows: $\llbracket \mu Z. \Phi \rrbracket_V^T = \bigcup \{\mathcal{W}_n \mid n \geq 0\}$ where $\mathcal{W}_0 = \emptyset$ and $\mathcal{W}_{n+1} = \llbracket \Phi \rrbracket_{V[\mathcal{W}_n/Z]}^T$.

4.4 Minimal mu-calculi

Minimal modal and temporal mu-calculi consist of their valid formulas, those that are true at every state (run) in every model (of the appropriate sort). We call the minimal modal mu-calculus μK , an extension of the logic K of Section 2.2. Similarly, $\mu L(O)$ is minimal linear time logic, an extension of $L(O, U)$ of Section 3.4 (when U is defined as $\mu Y. Q_2 \vee (Q_1 \wedge OY)$). Associated with $\mu L(O)$ are its pure and full branching time minimal logics $\mu B(O)$ and $\mu \forall L(O)$. Although some of these logics are known to be decidable, the techniques for showing this which appeal to automata have not yielded axiomatic characterizations.

However Kozen (1983) produced a sound and complete axiomatization of a sublogic of μK (when formulas are aconjunctive, a technical notion we will not expand on) using a tableau method. From this a clear indication of sound and complete axiom systems may be gleaned:

- | | |
|--------|--|
| Axioms | 1. All tautology instances
2. Axioms for the basis operators ($[a], O, \forall$)
3. $\Phi[Z := \mu Z. \Phi] \rightarrow \mu Z. \Phi$ |
| Rules | MP if Φ and $\Phi \rightarrow \Psi$ then Ψ
ROp if Φ then $Op\Phi$ (for $Op \in \{[a], O, \forall\}$)
R μ if $\Phi[Z := \Psi] \rightarrow \Psi$ then $(\mu Z. \Phi) \rightarrow \Psi$ |

The axioms for the basis operators are those provided in Section 2.2 and Section 3.4 such as the K axiom $[a](\Phi \rightarrow \Psi) \rightarrow ([a]\Phi \rightarrow [a]\Psi)$. The third

axiom is the fixed point unfolding (whose converse is a derived theorem). The rules include Modus Ponens, MP, and those for the basis operators. Finally there is a fixed point induction rule $R\mu$. Soundness of the respective systems is straightforward. But it is not known if all the resources necessary to prove valid fixed point formulas are provided here.

4.5 Decision procedures

The logics μK and $\mu L(O)$ are decidable as they can be embedded within Rabin's SnS. But this reduction does not yield proofs of their elementary decidability. Streett and Emerson (1989) appeal to a tableau method which is guided by an automaton for showing elementary decidability of μK . A tableau technique for a similar result for $\mu L(O)$ is employed in (Banieqbal and Barringer 1989).

Another concern is proving properties within a given model. Consider, for example, the question of how to extend the model checker of Section 2.6 for modal formulas to μK formulas. The tableau system presented there verified modal properties of states structurally by appealing to the state constructors $\cdot$, $\stackrel{def}{=}$, and $+$, instead of the transition relations. The goal directed rules were built from sequents of the form $E \vdash_{\mathcal{M}} \Phi$, analogues of $E \models_{\mathcal{M}} \Phi$, where E is a state term. A tableau for $E \vdash_{\mathcal{M}} \Phi$ is a maximal proof tree whose root is labelled with this sequent. Sequents labelling the immediate successors of a node are determined by an application of one of the rules. A tableau was said to be successful if all its leaves are successful.

Assume all the rules from Section 2.6. The only question is how to deal with fixed point formulas (and their negations). One proposal, inspired by Larsen (1988), is to invoke fixed point induction. However, for this to work in the presence of alternating fixed points we need an auxiliary family of propositional constants, ranged over by A . The two rules for fixed point formulas introduce new constants:

$$\frac{E \vdash_{\mathcal{M}} \mu Z. \Phi}{E \vdash_{\mathcal{M}} A} A \text{ new} \quad \frac{E \vdash_{\mathcal{M}} \neg \mu Z. \Phi}{E \vdash_{\mathcal{M}} A} A \text{ new}$$

Consequently, extra rules are now needed for these constants. Assume that $A \stackrel{def}{=} \mu Z. \Phi$ ($A \stackrel{def}{=} \neg \mu Z. \Phi$) represents that the constant A was introduced for the fixed point formula $\mu Z. \Phi$ ($\neg \mu Z. \Phi$). Then the rules for constants are, in effect, fixed point unrolling rules:

$$\frac{E \vdash_{\mathcal{M}} A}{E \vdash_{\mathcal{M}} \Phi[Z := A]} A \stackrel{def}{=} \mu Z. \Phi \quad \frac{E \vdash_{\mathcal{M}} A}{E \vdash_{\mathcal{M}} \neg \Phi[Z := \neg A]} A \stackrel{def}{=} \neg \mu Z. \Phi$$

To test if E has the property Φ in the model $\mathcal{M}$, one tries to achieve the goal $E \vdash_{\mathcal{M}} \Phi$ by building a successful tableau, a maximal proof tree

containing only successful leaves and whose root is labelled with this sequent. An extension of the modal case here is to count as a leaf any node labelled with a sequent $F \vdash_{\mathcal{M}} A$ if the very same sequent appears above it in the proof tree: thus, the constant rules above only apply to nodes that are not leaves. A leaf labelled $F \vdash_{\mathcal{M}} \Psi$ is *successful* if one of the following conditions apply: $\Psi = Q$ and $F \in \mathcal{V}_{\mathcal{M}}(Q)$; $\Psi = \neg Q$ and $F \notin \mathcal{V}_{\mathcal{M}}(Q)$; $\Psi = A$ and $A \stackrel{def}{=} \neg\mu Z. \Phi$; and finally $\Psi = [b]\Phi$ and $F = a.E$ (with $a \neq b$).

Therefore, a leaf of the form $F \vdash_{\mathcal{M}} A$ when $A \stackrel{def}{=} \mu Z. \Phi$ is unsuccessful.

The next result, a consequence of (Stirling and Walker 1989), affirms the soundness, completeness, and decidability of this tableau method with respect to finite models.

4.5.1: Proposition

If E (built from $\cdot$, $\stackrel{def}{=}$, and $+$)³ generates the finite transition system T , and $\mathcal{M} = (T, \mathcal{V})$, then for each $\Phi \in \mu K$

- i. every tableau for $E \vdash_{\mathcal{M}} \Phi$ is finite,
- ii $E \vdash_{\mathcal{M}} \Phi$ has a successful tableau iff $E \in \llbracket \Phi \rrbracket_{\mathcal{V}}^T$.

The propositional constants are essential to this method: if instead, fixed point formulas are syntactically unfolded, and a node labelled $E \vdash_{\mathcal{M}} \neg\mu Z. \Phi$ ($E \vdash_{\mathcal{M}} \mu Z. \Phi$) counts as a successful (unsuccessful) leaf when the very same sequent appears above it in the proof tree, then the resulting method would be neither sound nor complete. A semantic analysis explaining this can be found in (Cleaveland 1990; Winskel 1989). Moreover, this local model checking technique can be extended to arbitrary infinite state transition systems at the expense of losing decidability, (Bradfield and Stirling 1990).

An alternative approach to model checking fixed point formulas is to employ finite approximations. But, although this can be done for μK with respect to finite models, unlike the fixed point method, it may fail whenever the mu-logic is not continuous with respect to the model such as in the case of infinite state models and for $\mu L(O)$ formulas with respect to finite models, as the next example shows.

Example. Let T be the extended transition system determined by $U \stackrel{def}{=} a.U + a.V$ and $V \stackrel{def}{=} a.V$ where $\mathcal{R}_T$ is the set of runs passing through V almost always. Let $\mathcal{V}(Q) = \{V\}$. Although U has the property FQ , given by $\mu Z. Q \vee OZ$, it fails to have the properties $\mu^n Z. Q \vee OZ$ for all $n \in \mathbb{N}$. ■

³ As in Section 2.6 we assume that all state occurrences of variables on the right hand side of $\stackrel{def}{=}$ occur within the scope of the $\cdot$ operator.

5 Expressiveness

5.1 Expressive completeness

Temporal logics are identified by their families of operators. In previous sections special attention has been accorded to the linear time logics $L(O, U)$ and $\mu L(O)$ possibly augmented with the relativized operators (a) and reverse operators, as well as their branching time versions. There are good reasons for this. First, these logics are able to express properties of elements (states and runs) of transition systems, properties such as liveness and safety that are crucial to computational systems. Secondly, they support verification methods for showing that systems have these properties either via decision procedures for, or axiomatizations of, validity or via effective model checkers. A further question about expressiveness is: are there families of temporal operators that are *completely expressive*, the addition of further operators adds nothing to their expressive power? Complete truth functional expressiveness and combinator completeness provide two relevant analogies. However, unlike these two there is not a sufficiently rounded, or agreed upon, notion of a temporal operator. Instead we can consider expressive completeness relative to particular formal prescriptions of what a temporal operator is. Natural formalisms for describing such operators are the metalanguages within which the semantics of temporal languages are couched. An appropriate starting point is to examine canonical models.

Recall the canonical structures $\mathcal{T}_0 = (\{0\}^*, \{\overset{0}{\longrightarrow}\}, \mathcal{R})$ from Section 3.6 for future linear time logics. Members of $\mathcal{R}$ can be represented as natural numbers: n symbolizing the run starting from 0^n . With this understanding $>$ on $\mathbf{N}$ is the suffix ordering. Consequently, the ω -structure $(\mathbf{N}, >)$ is also a canonical structure for future linear time logics. As our interest is with sets of models individual formulas define, let Q range over a fixed finite family of atomic sentences $\{Q_1, \dots, Q_n\}$. A temporal model on $(\mathbf{N}, >)$, called an ω -model, includes the extra information as to which elements of $\mathbf{N}$ each Q_i is true at. An ω -model can itself be viewed as a structure $\Omega = (\mathbf{N}, >, q_1, \dots, q_n)$ where each q_i , the interpretation of Q_i , is a subset of $\mathbf{N}$. However to interpret $\mu L(O)$ subformulas with free variables valuation extensions of ω -models are required. These are pairs $(\Omega, \mathcal{V})$ where $\mathcal{V}$ maps each propositional variable into a subset of $\mathbf{N}$. For uniformity, redundant valuation extensions are also assumed when interpreting $L(O, U)$ formulas. Each formula of these future time logics determines a subset of $\mathbf{N}$ in any model $(\Omega, \mathcal{V})$: we write $\models \Phi \parallel_{\mathcal{V}}^{\Omega}$ for this set. Some sample clauses of the expected inductive definition are:

$$\begin{aligned}
 |Q_j|_{\mathcal{V}}^{\Omega} &= q_j \\
 |\neg\Phi|_{\mathcal{V}}^{\Omega} &= \mathbb{N} - |\Phi|_{\mathcal{V}}^{\Omega} \\
 |O\Phi|_{\mathcal{V}}^{\Omega} &= \{n \mid n+1 \in |\Phi|_{\mathcal{V}}^{\Omega}\} \\
 |\mu Z. \Phi|_{\mathcal{V}}^{\Omega} &= \bigcap \{I \subseteq \mathbb{N} \mid |\Phi|_{\mathcal{V}[U/Z]}^{\Omega} \subseteq I\}
 \end{aligned}$$

Clearly, if $|\Phi|_{\mathcal{V}}^{\Omega} = \mathbb{N}$ for every valuation $\mathcal{V}$ then Φ is valid. Moreover, since we are dealing with future time logics, this result may be sharpened:

5.1.1: Lemma

If $0 \in |\Phi|_{\mathcal{V}}^{\Omega}$ for every $\mathcal{V}$ then Φ is valid.

Proof. Suppose $0 \in |\Phi|_{\mathcal{V}}^{\Omega}$ for every $\mathcal{V}$, but Φ is not valid. Then there is a model $(\Omega, \mathcal{V})$ and an n such that $n \notin |\Phi|_{\mathcal{V}}^{\Omega}$, as ω -models are canonical. But then consider the model $(\Omega', \mathcal{V}')$ where Ω' is $(\mathbb{N}, >, q'_1, \dots, q'_n)$ with $q'_i = \{m - n \mid m \geq n \text{ and } m \in q_i\}$ (where q_i is in Ω). Similarly for $\mathcal{V}'$. An easy induction shows that for $m \geq n$, $m \in |\Phi|_{\mathcal{V}}^{\Omega}$ iff $m - n \in |\Phi|_{\mathcal{V}'}^{\Omega'}$. ■

Assume that the set of ω -models defined by a closed temporal formula is $\{\Omega \mid \text{for some } \mathcal{V}. 0 \in |\Phi|_{\mathcal{V}}^{\Omega}\}$, and that this set is denoted by $|\Phi|$. For instance $|FQ_i| = \{\Omega \mid q_i \neq \emptyset\}$ and $|\nu Z. Q_i \wedge OOZ| = \{\Omega \mid \{2m \mid m \geq 0\} \subseteq \{q_i\}\}$.

The expressive power of individual formulas can be revealed by comparing and contrasting the sets of models they define as in Section 3.3. But we can also examine expressiveness by identifying the ω -models formulas define. Assume that α ranges over sets of ω -models. When L is either $L(O, U)$ or $\mu L(O)$, we say that α is *L-definable* if for some $\Phi \in L$, $|\Phi| = \alpha$. But are there independent characterizations of these *L-definable* sets? To explore this, further notations for defining ω -models are needed.

The interpretation of $L(O, U)$ formulas in an ω -model is described using the following first-order language L_1 with identity:

$$\Phi ::= q(x) \mid x_1 > x_2 \mid x_1 = x_2 \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \forall x. \Phi$$

Here q ranges over the family $\{q_1, \dots, q_n\}$ viewed now as monadic predicates; x ranges over individual variables; and $>$ is a binary relation. The interpretation of $\mu L(O)$ formulas goes beyond the resources of L_1 , requiring quantification over subsets of $\mathbb{N}$ to describe the meaning of $\mu Z. \Phi$. The natural language for describing the meaning of such formulas is L_2 , the monadic second-order extension of L_1 . This involves the following two extra constructors:

$$\Phi ::= Z(x) \mid \dots \mid \forall Z. \Phi$$

where Z ranges over monadic predicate variables⁴. Formulas of L_1 and L_2 can be interpreted on ω -models. As before we need valuation extensions to interpret variables. Now let $(\Omega, \mathcal{V})$ be a model where $\mathcal{V}$ maps individual variables into $\mathbb{N}$, and predicate variables into subsets of $\mathbb{N}$. We use the notation $(\Omega, \mathcal{V}) \models \Phi$ to denote that Φ is true in the model $(\Omega, \mathcal{V})$, which is inductively defined in the expected way. Some sample clauses are:

$$\begin{aligned} (\Omega, \mathcal{V}) \models q_j(x) & \text{ iff } \mathcal{V}(x) \in q_j^\Omega \\ (\Omega, \mathcal{V}) \models Z(x) & \text{ iff } \mathcal{V}(x) \in \mathcal{V}(Z) \\ (\Omega, \mathcal{V}) \models x > y & \text{ iff } \mathcal{V}(x) >_\Omega \mathcal{V}(y) \\ (\Omega, \mathcal{V}) \models \forall Z. \Phi & \text{ iff } \forall I \subseteq \mathbb{N}. (\Omega, \mathcal{V}[I/Z]) \models \Phi \end{aligned}$$

Assume that $\Phi(x)$ represents an L_1 or L_2 formula containing one free variable x . Let the set of ω -models defined by this formula be $\llbracket \Phi(x) \rrbracket = \{(\Omega, \mathcal{V}) \mid \text{for some } \mathcal{V}. (\Omega, \mathcal{V}[0/x]) \models \Phi(x)\}$. As with the temporal case, 0 receives special treatment because we are dealing with future time logics: consequently, the formula $\exists y. x \geq y \wedge q_i(y)$ picks out the same models as $q_i(x)$ instead of defining $\overline{F}q_i$. We say that a set of ω -models α is L_1 -definable (L_2 -definable) if for some $\Phi(x) \in L_1$ (L_2) $\llbracket \Phi(x) \rrbracket = \alpha$. Clearly, this invites comparisons between $L(O, U)$ and L_1 , and $\mu L(O)$ and L_2 -definability. First

5.1.2: Lemma

- i. If α is $L(O, U)$ -definable then α is L_1 -definable,
- ii. If α is $\mu L(O)$ -definable then α is L_2 -definable.

Proof. For both parts there is a straightforward translation from closed temporal formulas into L_1 and L_2 formulas with one free variable, preserving definability. For each closed temporal formula Φ and individual variable y , define (Φ, y) inductively as follows: $(Q_i, y) = q_i(y)$; $(\neg\Phi, y) = \neg(\Phi, y)$; $(\Phi \wedge \Psi, y) = (\Phi, y) \wedge (\Psi, y)$; $(O\Phi, y) = \exists z > y. \forall u > y. u \geq z \wedge (\Phi, z)$; $(\Phi U \Psi, y) = \exists z \geq y. (\Psi, z) \wedge \forall u. (z > u \wedge u \geq y) \rightarrow (\Phi, u)$; $(\mu Z. \Phi, y) = \forall Z. (\forall u. (\Phi, u) \rightarrow Z(u)) \rightarrow Z(y)$. Clearly, $\llbracket \Phi \rrbracket = \llbracket (\Phi, x) \rrbracket$. ■

This result is hardly surprising, given that the semantics of temporal formulas is defined in L_1 and L_2 .

What is surprising is that the converses of Lemma 5.1.2 also hold.

5.1.3: Proposition

- i. If α is L_1 -definable then α is $L(O, U)$ -definable,
- ii. If α is L_2 -definable then α is $\mu L(O)$ -definable.

⁴ L_2 is equivalent to the second-order theory of one successor, $S1S_q$, with predicate symbols in q .

The proofs of these are extremely difficult, and depend on providing suitable normal forms for L_1 and L_2 formulas $\Phi(x)$. The proof of i. is due to (Gabbay *et al.* 1980) where details of the normal form are provided. The proof of ii. is more indirect. First L_2 formulas $\Phi(x)$ can be reduced to a normal form via Büchi automata on the alphabet 2^q , (Büchi 1962). And, secondly, Büchi automata define the ω -regular languages (see the next section), which turn out to be equivalent to Wolper's EL (Wolper *et al.* 1983), which is a sublogic of $\mu L(O)$.

A consequence of Proposition 5.1.3 is that the temporal languages $L(O, U)$ and $\mu L(O)$ are expressively complete relative to L_1 and L_2 : adding additional future time operators whose meaning is definable as formulas $\Phi(x)$ from L_1 or L_2 does not increase their expressiveness. Extensions to L_1 and L_2 are needed if similar results are to be obtained for temporal languages employing the relativized operators (α). Decidable extensions of L_2 permit more expressive temporal logics such as $\mu L(O, G_{\lambda n.2^*})$.

Part i. of Proposition 5.1.3 is a particularization of a more general result due to Kamp (1968), that the temporal logic $L(U', \bar{U}')$ (where U' is the until operator of Section 3.1) is expressively complete with respect to the first-order logic of complete linear orders. A linear order $(C, >)$ is complete if every non-empty bounded subset of C has both a least upper bound and a greatest lower bound with respect to $>$: so $(\mathbf{N}, >)$ and $(\mathbf{R}, >)$ are complete, unlike $(\mathbf{Q}, >)$ where $\mathbf{Q}$ is the rationals. Given a structure $(C, >)$ each L_1 formula $\Phi(x)$ defines an n -ary temporal operator:

$$[Op_{\Phi}^n] = \lambda q_1 \dots \lambda q_n \subseteq C. \{c_i \in C \mid (C, >, q_1, \dots, q_n, \forall [c_i/x]) \models \Phi(x)\}$$

Kamp showed that each such operator is definable in $L(U', \bar{U}')$. Stavi, in unpublished work, extended this result to all linear orders by introducing a further until operator and its inverse, see (Gabbay 1981) for their description. A more general technique for proving expressive completeness is given in (Gabbay 1981). Moreover, Amir and Gabbay (1987) have extended Proposition 5.1.3 i. to multi-dimensional linear time logics whose temporal formulas may depend on more than one time index for their truth value—such temporal formulas are associated with L_1 formulas containing more than one free variable.

Generalizing Proposition 5.1.3 to branching time logics is problematic. One issue is the choice of a suitable metalanguage. A natural candidate at the second-order level is SkS_Q , the second-order theory of k successors (Rabin 1969). This is given by the monadic second-order logic over structures constructed from T_k of the form:

$$\Omega_k = (\{0, \dots, k\}^*, \{\overset{i}{\longrightarrow} \mid i \in \{0, \dots, k\}\}, >, q_1, \dots, q_n)$$

where again $>$ is the suffix ordering. Runs are definable as appropriate sets of elements. However, this language is too rich for expressive completeness

of $\mu\forall L(O)$ because incomparable elements can be counted. For instance, the formula $\exists x_1, \dots, x_i. \bigwedge (x \not\leq x_i \wedge x_i \not\leq x)$ has no branching time correlate. A natural candidate temporal logic for expressive completeness here is $\mu B(\{(i) \mid 0 \leq i \leq k\})$ when (i) have their expected meaning, see (Hüttel 1990). At the first-order level the notion of run is not definable making it impossible to relate it to $\forall L(O, U)$. For more details together with positive results for theories of the binary tree see (Amir 1985; Thomas 1987; Hafer and Thomas 1987; Schlingoff 1990).

5.2 ω -regular expressions

Language theory offers another framework for defining families of ω -models. Assume, as in the previous section, a fixed finite set of atomic sentences $\{Q_1, \dots, Q_n\}$. The alphabet of interest consists of all subsets of these atomic sentences. Consequently, a word is a finite sequence $u_0 u_1 \dots u_m$ where each $u_i \subseteq \{Q_1, \dots, Q_n\}$ such as $\{Q_1\} \emptyset \{Q_2, Q_n\}$. An ω -word is an ω -length sequence $u_0 u_1 \dots u_m \dots$, and an ω -language is a set of such words.

Each ω -model $\Omega = (\mathbf{N}, >, q_1, \dots, q_n)$ determines a unique ω -word, which we denote by $w(\Omega)$, defined as the sequence $u_0 u_1 \dots u_m \dots$ where the sets u_i consist of those atomic sentences that are true at i in Ω : $u_i = \{Q_j \mid i \in q_j\}$. Conversely, associated with any ω -word σ is the unique model Ω with the property that $\sigma = w(\Omega)$. These notions can be generalized to languages and sets of models. Any family of ω -models uniquely determines an ω -language. And conversely, any ω -language L defines a unique set of ω -models: we let $\llbracket L \rrbracket$ denote this set:

$$\llbracket L \rrbracket \stackrel{def}{=} \{\Omega \mid w(\Omega) \in L\}$$

This offers the potential for a further classification of ω -models: a set α of ω -models is language definable just in case $\alpha = \llbracket L \rrbracket$ for some ω -language L . By restricting to particular notations for defining languages, we can obtain a result analogous to Proposition 5.1.3 and its converse.

Regular expressions have proved to be extremely useful for defining languages. They define languages whose words have finite length. The *star-free* regular languages on an alphabet ranged over by u is defined as follows:

$$Sf ::= \{u\} \mid Sf_1 \cup Sf_2 \mid -Sf \mid Sf_1; Sf_2$$

Here $\cup$ is language union; $-$ is complementation, so $-Sf$ defines the set of finite length words which are not in Sf ; and $Sf_1; Sf_2$ is the set $\{w_1 w_2 \mid w_1 \in Sf_1 \text{ and } w_2 \in Sf_2\}$. The regular languages are similarly defined:

$$Re ::= \{u\} \mid Re_1 \cup Re_2 \mid -Re \mid Re_1; Re_2 \mid Re^*$$

Included is the extra $*$ operator: the language Re^* is $\{w_0 \dots w_m \mid m \geq 0 \wedge w_i \in Re\}$. A regular language L is said to be concatenation closed if $L; L \subseteq L$.

These definitions are now extended to ω -languages. A simple enhancement is to permit the operation $^\omega$. However, assume that it is only defined for concatenation closed languages containing at least one word other than the empty word ε : L^ω is then the ω -language $\{w_0 \dots w_n \dots \mid \varepsilon \neq w_i \text{ and } w_i \in L\}$. The star-free ω -regular languages, ranged over by S , have the following syntax:

$$S ::= \emptyset \mid Sf_1; Sf_2^\omega \mid S_1 \cup S_2$$

and the ω -regular languages, ranged over by R , are defined similarly:

$$R ::= \emptyset \mid Re_1; Re_2^\omega \mid R_1 \cup R_2$$

A set of ω -models α is R -definable (S -definable) if there is a (star-free) ω -regular language L with the feature that $\alpha = \models L$. The main result is that these language definable models coincide with those definable in L_2 and L_1 , of the previous section.

5.2.1: Proposition

- i. α is S -definable iff α is L_1 -definable,
- ii. α is R -definable iff α is L_2 -definable.

For discussion and details of these results see (Thomas 1989).

5.3 Zig-zags, bisimulations, and histories

Various expressibility issues stem from modal and temporal logics. At the micro-level there is interest in the particular way that individual formulas express properties and delimit models. Pertinent to this, as we have seen, is the extent to which the metalanguage itself is reflected in the underlying modal or temporal logic. Instead, our concern here is at the macro-level, with the totality of modal or temporal formulas of a logic. We examine when two states of a model satisfy exactly the same formulas. This is of special interest in the context of program or process equivalence, as transition systems are widely used as models of processes: $s \xrightarrow{a} s'$ is to be understood as process s may perform action a and become the process s' . Usually as noted in the Introduction, the state set is structured algebraically. A variety of behavioural equivalences between processes have been proposed, see (De Nicola 1987) for a discussion. Often these are also congruences with respect to the algebraic operators. Some of these equivalences appeal to the path behaviour of processes, how they perform under sequences of actions. Modal and temporal logics offer another kind of equivalence: two

processes are equivalent if they are not distinguishable by formulas of the logical language under investigation, (Hennessy and Milner 1980; Brookes and Rounds 1983).

Given a modal (or modal mu-calculus) model $\mathcal{M} = (T, V)$ when do two states in $\mathcal{S}_T$ satisfy the same formulas? Equally, the question could be asked of two states from different models. The answer appeals to zig-zag relations as introduced in the 1960s, see the survey (van Benthem 1984).

5.3.1: Definition

A zig-zag on $\mathcal{M}$ is a binary relation Z on $\mathcal{S}_\mathcal{M}$ such that if sZs' then for all labels a and atomic Q

- i. $s \in V(Q)$ iff $s' \in V(Q)$,
- ii. $\forall s_1$. if $s \xrightarrow{a} s_1$ then $\exists s'_1$. $s' \xrightarrow{a} s'_1$ and $s_1 Z s'_1$,
- iii. $\forall s'_1$ if $s' \xrightarrow{a} s'_1$ then $\exists s_1$. $s \xrightarrow{a} s_1$ and $s_1 Z s'_1$.

Notice that a zig-zag is just a relation with the hereditary property in the final two clauses and only relating states satisfying the same atomic formulas. Without clause i. the definition is of a bisimulation relation due to Park (1981), and used by Hennessy and Milner (1985). Alternatively, a bisimulation is a zig-zag for modal logics whose sole atomic sentence is **tt**. Zig-zags are closed under various operations: if Z_1 and Z_2 are zig-zags on $\mathcal{M}$ then so are $Z_1 \cup Z_2$, and $Z_1 \circ Z_2$. When there is a zig-zag on $\mathcal{M}$ relating s and s' , we say s and s' are *zig-zag equivalent* in $\mathcal{M}$ and we write $s \equiv_{\mathcal{M}} s'$.

Clearly, $\equiv_{\mathcal{M}}$ is an equivalence relation on $\mathcal{M}$ (and is itself a zig-zag, being the union of all zig-zags on $\mathcal{M}$). When the set of atomic sentences, and the model are finite then there is a natural decision procedure for deciding which states of $\mathcal{S}_\mathcal{M}$ are zig-zag equivalent (Kanellakis and Smolka 1990). Let $\|s\|^\mathcal{M}$ be the set of modal formulas or (closed) modal mu-calculus formulas that are true at the state s .

5.3.2: Theorem

If $s \equiv_{\mathcal{M}} s'$ then $\|s\|^\mathcal{M} = \|s'\|^\mathcal{M}$.

Proof. By routine structural induction on modal formulas and in the case of the modal mu-calculus by induction on formulas of infinitary modal propositional logic. ■

Note that this theorem fails if the modal logics include the reverse operators $[a]$. However, by extending the definition of zig-zag to include inverse transitions then a corresponding theorem holds: for an interesting application see (De Nicola and Vaandrager 1990).

The converse of Theorem 5.3.2 holds if infinitary conjunction is allowed in the modal languages. Alternatively, it holds for modal logic and the

modal mu-calculus if the model $\mathcal{M}$ is image finite (Hennessy and Milner 1985).

5.3.3: Theorem

If $\mathcal{M}$ is image finite and $\llbracket s \rrbracket^{\mathcal{M}} = \llbracket s' \rrbracket^{\mathcal{M}}$ then $s \equiv_{\mathcal{M}} s'$.

Proof. It suffices to show that the relation Z defined by $s_1 Z s_2$ whenever $\llbracket s \rrbracket^{\mathcal{M}} = \llbracket s' \rrbracket^{\mathcal{M}}$ is a zig-zag. Suppose not. If $s_1 \in V(Q)$ and $s_2 \notin V(Q)$ then this contradicts $\llbracket s \rrbracket^{\mathcal{M}} = \llbracket s' \rrbracket^{\mathcal{M}}$. Otherwise, suppose $s_1 \xrightarrow{a} s'_1$ and for all s'_2 , $s_2 \xrightarrow{a} s'_2$, $\llbracket s'_1 \rrbracket^{\mathcal{M}} \neq \llbracket s'_2 \rrbracket^{\mathcal{M}}$. Let $\{t_1, \dots, t_n\} = \{s'_2 \mid s_2 \xrightarrow{a} s'_2\}$. If this set is empty then $s_2 \models_{\mathcal{M}} [a] \text{ff}$ unlike s_1 which contradicts that $s_1 Z s_2$. Hence, there are formulas Φ_i , $1 \leq i \leq n$, such that $s'_1 \models_{\mathcal{M}} \Phi_i$ and $t_i \models_{\mathcal{M}} \neg \Phi_i$. Let $\Psi = \Phi_1 \wedge \dots \wedge \Phi_n$. But $s_1 \models_{\mathcal{M}} \langle a \rangle \Psi$ whereas $s_2 \models_{\mathcal{M}} \neg \langle a \rangle \Psi$ which contradicts that $s_1 Z s_2$. ■

Let $\mathcal{M}$ now be a future time temporal model. We are interested when two states satisfy the same future linear or branching time formulas (including the mu-calculi). First, we define a history equivalence relation $\simeq_{\mathcal{M}}$ on the sets of runs $\Sigma_{\mathcal{M}}$ where $\mathcal{L}(\sigma)$ is the ω sequence of labels of σ .

5.3.4: Definition

Two runs σ and π of $\mathcal{R}_{\mathcal{M}}$ are history equivalent, written $\sigma \simeq_{\mathcal{M}} \pi$ iff

- i. $\mathcal{L}(\sigma) = \mathcal{L}(\pi)$,
- ii. for all i , and atomic Q , $\sigma(i) \in V(Q)$ iff $\pi(i) \in V(Q)$.

Condition i. is dropped when the temporal logic doesn't involve indexed temporal operators like (a) . When the only atomic sentence is tt then condition ii. is redundant. A general lemma for any future linear time logic (where $\llbracket \sigma \rrbracket^{\mathcal{M}}$ is the set $\{\Phi \mid \sigma \models_{\mathcal{M}} \Phi\}$).

5.3.5: Lemma

If $\sigma \simeq_{\mathcal{M}} \pi$ then $\llbracket \sigma \rrbracket^{\mathcal{M}} = \llbracket \pi \rrbracket^{\mathcal{M}}$.

Proof. By routine structural induction on formulas Φ from the future fragment of the infinitary linear time language L_{∞} of Section 3.3. ■

A suitable amendment to Definition 5.3.4 caters for linear time logic with reverse operators. For future linear logics which don't include the next operator there is a notion of a stuttering history equivalence (Lamport 1983).

The converse of Lemma 5.3.5 is guaranteed when the linear time logic can express the relative next operators (a) . We now generalize the history equivalence to states, the linear time version of zig-zags. Let $\mathcal{R}_{\mathcal{M}}(s)$ be the set of runs in $\mathcal{R}_{\mathcal{M}}$ whose initial state is s .

5.3.6: Definition

Two states s, s' of $\mathcal{S}_M$ are history equivalent, written $s \downarrow_M s'$, iff

- i. $\forall \sigma \in \mathcal{R}_M(s). \exists \sigma' \in \mathcal{R}_M(s'). \sigma \simeq_M \sigma'$,
- ii. $\forall \sigma' \in \mathcal{R}_M(s'). \exists \sigma \in \mathcal{R}_M(s). \sigma \simeq_M \sigma'$.

The following is almost a corollary of Lemma 5.3.5, where $|s|^\mathcal{M}$ is now the family of linear time formulas true at s : if $s \downarrow_M s'$ then $|s|^\mathcal{M} = |s'|^\mathcal{M}$. The converse of this holds if infinitary conjunction is allowed in the linear time logic (or, for finitary logics, if one of the sets $\mathcal{R}_M(s)$ or $\mathcal{R}_M(s')$ is finite) and if it expresses the operators (a). In the particular case that these relative operators are expressible, and $\mathbf{tt}$ is the only atomic sentence, then we can think of $\downarrow_M$ as ω -language equivalence: if $s \downarrow_M s'$ then these states (in the machine M) accept the same ω -words on the alphabet $\mathcal{L}$. But the associated branching time logics distinguishes states which are not bisimulation equivalent as the modalities $[a]$ are expressible. So there is an enormous chasm in state distinguishability between, for instance, the linear time mu-calculus with sole atomic sentence $\mathbf{tt}$ and its associated branching time versions.

The notion of zig-zag needs to be extended when we examine states satisfying the same sets of future branching time formulas. First, we extend zig-zags to be relations on runs. Assume M is a future time model.

5.3.7: Definition

A run zig-zag on M is a binary relation Z on $\mathcal{R}_M$ such that if $\sigma Z \pi$ then for all $i \geq 0$

- i. $\sigma \simeq_M \pi$,
- ii. $\forall \sigma_1 \in \mathcal{R}(\sigma(i)). \exists \pi_1 \in \mathcal{R}(\pi(i)). \sigma_1 Z \pi_1$,
- iii. $\forall \pi_1 \in \mathcal{R}(\pi(i)). \exists \sigma_1 \in \mathcal{R}(\sigma(i)). \sigma_1 Z \pi_1$.

Two runs σ and π are zig-zag equivalent on M , written as $\sigma \rightleftharpoons_M \pi$, if there is a run zig-zag on M between them. The correlate of Lemma 5.3.5 for future branching time logics is: if $\sigma \rightleftharpoons_M \pi$ then $|\sigma|^\mathcal{M} = |\pi|^\mathcal{M}$. Run zig-zags can be extended to states.

5.3.8: Definition

An extended zig-zag on M is a binary relation Z on $\mathcal{S}_M$ such that if $s Z s'$ then $\exists \sigma \in \mathcal{R}(s). \exists \pi \in \mathcal{R}(s'). \sigma \rightleftharpoons_M \pi$.

We write $s \rightleftharpoons_M^+ s'$ if s and s' are related by an extended zig-zag. The correlate for branching future time logics is: if $s \rightleftharpoons_M^+ s'$ then $|s|^\mathcal{M} = |s'|^\mathcal{M}$. Extended zig-zag equivalence is associated with branching time temporal logics. If these logics don't contain the next operator then there is a stuttering variant (Browne *et al.* 1986). In the case of the branching time mu-calculus with $\mathbf{tt}$ as sole atomic formula the associated equivalence is extended bisimulation equivalence (Hennessy 1984). Generally, extended zig-zag equivalence is finer than zig-zag equivalence. However, in the par-

ticular case that $\mathcal{R}_M$ consists of all runs through the underlying transition system then they coincide—a slight variant of this result is proved in (Hennessy and Stirling 1985).

6 Sound and Complete Axiom Systems

6.1 Soundness

In previous sections a variety of modal and temporal logics have been introduced. Typically these are sets of Λ -valid formulas when Λ is a general class of models. In some cases axiom systems were also provided, yielding an alternative characterization of Λ -validity (given by their sets of theorems). Examples include the axiom systems for K (Section 2.2) and PDL (Section 2.4), and the systems $KG(v, w, w', v')$ (Section 2.3); LT , $\forall LT$ (Section 3.4); and $\forall LTF$ (Section 3.5). An axiom system J consists of a finite set of axiom schemas, and a finite set of rules. A proof of Φ in J (written $J \vdash \Phi$ or $\vdash_J \Phi$) is a finite sequence of formulas $\Phi_1, \dots, \Phi_n$, with $\Phi = \Phi_n$ and where each Φ_i , $1 \leq i \leq n$, is either an axiom instance, or the result of an application of a rule to formulas in the set $\{\Phi_1, \dots, \Phi_{i-1}\}$.

One half of the axiomatic characterization of validity is *soundness*: if $J \vdash \Phi$ then $\Lambda \models \Phi$. The usual technique for showing this is straightforward: first prove that every axiom instance of J is Λ -valid, and then show that the rules of J preserve Λ -validity. Similarly, proving that J is *not* sound is also clear, by exhibiting a counterexample to an axiom instance or an application of a rule. The following lemma offers samples from axiom systems scattered throughout previous sections.

6.1.1: Lemma

- i. If Λ is the family of dynamic models then $\Lambda \models (\Phi \wedge [a^*](\Phi \rightarrow [a]\Phi)) \rightarrow [a^*]\Phi$,
- ii. If Λ is all temporal models then $\Lambda \models \Phi U \Psi \leftrightarrow \Psi \vee (\Phi \wedge O(\Phi U \Psi))$,
- iii. If Λ is all fusion closed future models then $\Lambda \models \forall O\Phi \rightarrow O\forall\Phi$,
- iv. If Λ is all modal models (with a in its sort) and $\Lambda \models \Phi$ then $\Lambda \models [a]\Phi$.

Proof. i. Suppose $\mathcal{M}$ is a dynamic model and $s \models_{\mathcal{M}} \Phi \wedge [a^*](\Phi \rightarrow [a]\Phi)$. We show that for all $w \in \{a\}^*$ if $s \xrightarrow{w} s'$ then $s' \models_{\mathcal{M}} \Phi$ by induction on the length of w . The base case is given as $s \models_{\mathcal{M}} \Phi$. Assume $s \xrightarrow{w} s_1 \xrightarrow{a} s'$. Since $s \xrightarrow{w} s_1$, $s_1 \models_{\mathcal{M}} \Phi \rightarrow [a]\Phi$. And by the induction hypothesis $s_1 \models_{\mathcal{M}} \Phi$. Consequently $s' \models_{\mathcal{M}} \Phi$ as $s_1 \models_{\mathcal{M}} [a]\Phi$. ii. Suppose $\mathcal{M}$ is a temporal model and $\sigma, i \models_{\mathcal{M}} \Phi U \Psi$. Then $\exists j \geq i$. $\sigma, j \models_{\mathcal{M}} \Psi$ and $\forall k : i \leq k < j$. $\sigma, k \models_{\mathcal{M}} \Phi$. If $j = i$ then $\sigma, i \models_{\mathcal{M}} \Psi$. Otherwise $j > i$ and clearly $\sigma, i \models_{\mathcal{M}} \Phi \wedge O(\Phi U \Psi)$. The other direction is similar. iii. Suppose $\mathcal{M}$ is a fusion closed future time model with $\sigma \models_{\mathcal{M}} \forall O\Phi$. So $\forall \pi$.

if $\pi(0) = \sigma(0)$ then $\pi^1 \models_{\mathcal{M}} \Phi$. On the other hand if $\sigma \not\models_{\mathcal{M}} O\forall\Phi$ then $\sigma^1 \not\models_{\mathcal{M}} \forall\Phi$. But then there is a run π_1 with $\pi_1(0) = \sigma^1(0)$ and $\pi_1 \not\models_{\mathcal{M}} \Phi$. However, by fusion closure there is the run $(\sigma(0) \xrightarrow{a} \pi_1(0)) \cdot \pi_1$ (for some label a). But then $\sigma \not\models_{\mathcal{M}} \forall O\Phi$. iv. Suppose $\Lambda \models \Phi$. Then for any modal model $\mathcal{M}$ whose sort includes a and state $s \in \mathcal{S}_{\mathcal{M}}$, we have that $s \models_{\mathcal{M}} \Phi$. But then also $s \models_{\mathcal{M}} [a]\Phi$. So $\Lambda \models [a]\Phi$. ■

Moreover the axiom systems mentioned above are *strongly* sound with respect to validity: if $\Gamma \vdash_J \Phi$ then $\Gamma \models_{\Lambda} \Phi$ when $\models_{\Lambda}$ is the *local* consequence relation (and $\Gamma \vdash_J \Phi$ means that there is a proof of Φ in J from the set of assumptions Γ as in Section 2.2). This follows from the deduction theorem for these logics: $\Gamma, \Phi \vdash_J \Psi$ iff $\Gamma \vdash_J \Phi \rightarrow \Psi$, and from the observation that $\Gamma \vdash_J \Phi$ iff $\Gamma_0 \vdash_J \Phi$ for some finite $\Gamma_0 \subseteq \Gamma$.

6.2 Canonical models

The other half of the axiomatic characterization of validity is *completeness*: if $\Lambda \models \Phi$ then $J \vdash \Phi$. Techniques for a direct proof of this property are not known, as sets of valid formulas can not be described in terms of simple inductive features. Instead the standard approach for proving completeness (called the *Henkin method*) is indirect. First, as every formula can be rewritten as a negated formula, by contraposition completeness is equivalent to: if $J \not\vdash \neg\Phi$ then $\Lambda \not\models \neg\Phi$. Clearly, $\neg\Phi$ is *not* Λ -valid when Φ is Λ -satisfiable. A formula Φ is said to be *J-consistent* if $J \not\vdash \neg\Phi$, which is equivalent to the statement that there is not a proof of $\mathbf{ff}$ in J from the assumption Φ . Consequently, an alternative statement of completeness is: *if Φ is J-consistent then Φ is Λ -satisfiable*. The notion of *J-consistency* can be extended to arbitrary sets of formulas: Γ is *J-consistent* if there is not a proof of $\mathbf{ff}$ in J from the set of assumptions Γ , that is if $\Gamma \not\vdash_J \mathbf{ff}$. *Strong* completeness of J (with respect to Λ -validity) is therefore equivalent to: if Γ is *J-consistent* then Γ is Λ -satisfiable.

The Henkin method for showing an axiom system is complete, as presented for instance in (Hughes and Cresswell 1973), involves building a modal or temporal model $(\mathcal{T}, \mathcal{V})$ with $\mathcal{T} \in \Lambda$ from sets of *J-consistent* formulas. The simplest technique, when it applies, is to build the *canonical model* with respect to a set of formulas. This approach is now illustrated for modal axiom systems.

A set of formulas $\Gamma \subseteq \alpha$ is a *maximal J-consistent* set of α formulas if Γ is *J-consistent* and $\Gamma \vdash_J \neg\Phi$ for any $\Phi \in \Gamma - \alpha$. No additional formulas belonging to α can be added to Γ without incurring inconsistency. Let Jmc_{α} be the set of all maximal *J-consistent* sets of α formulas. Every *J-consistent* subset of α formulas can be extended to a maximal *J-consistent* set of α formulas.

6.2.1: Lemma

If $\beta \subseteq \alpha$ is J -consistent then $\exists \Gamma \in Jmc_\alpha. \beta \subseteq \Gamma$.

Proof. By construction. Let $\alpha - \beta$ be the set $\{\Phi_1, \dots, \Phi_n, \dots\}$. Let Γ_0 be the set β . For each $i > 0$ let $\Gamma_i = \Gamma_{i-1}$ if $\Gamma_{i-1} \vdash_J \neg\Phi_i$, otherwise let $\Gamma_i = \Gamma_{i-1} \cup \{\Phi_i\}$. A small exercise shows that $\bigcup\{\Gamma_i \mid i \geq 0\} \in Jmc_\alpha$. ■

In the sequel attention is restricted to non-empty sets of formulas α subject to closure conditions. The set α is closed under *subformulas* and *negation* if whenever $\Phi \in \alpha$ and Ψ is a subformula of Φ then $\Psi \in \alpha$; and, whenever $\Phi \in \alpha$ and $\Phi \neq \neg\Psi$ for some Ψ then $\neg\Phi \in \alpha$. The second condition ensures that there are finite closed sets of formulas. A finite set of formulas $\beta = \{\Phi_1, \dots, \Phi_n\}$ is J -consistent just in case the single conjunction $\Phi_1 \wedge \dots \wedge \Phi_n$ is J -consistent. Hence in the following, we often let a finite set such as β stand for the single formula $\Phi_1 \wedge \dots \wedge \Phi_n$.

Assume that J is a modal axiom system (which, at least, contains the axioms and rules for minimal modal logic of Section 2.2), and let α be a non-empty set of modal formulas closed under subformulas and negation. Uniquely associated with α and J is its canonical modal model.

6.2.2: Definition

The canonical modal model with respect to α and J is $\mathcal{M}_J^\alpha = (T, \mathcal{V})$ where:

- i. $\mathcal{S}_T = Jmc_\alpha$,
- ii. $\mathcal{L}_T = \{a \mid a \text{ occurs in some formula in } \alpha\}$,
- iii. $s \xrightarrow{a} s'$ iff $\forall$ finite $\beta \subseteq s'. s \cup \{a\}\beta$ is J -consistent,
- iv. $\mathcal{V}(Q) = \{s \mid Q \in s\}$.

By definition $\mathcal{M}_J^\alpha$ is a modal model. Moreover because of the closure conditions on α and because J contains those axioms and rules presented for minimal modal logic, this model has the following crucial property:

6.2.3: Theorem

For all $\Phi \in \alpha. s \models_{\mathcal{M}_J^\alpha} \Phi$ iff $\Phi \in s$.

Proof. By induction on the structure of Φ . The base case when Φ is atomic follows from the definition of the valuation $\mathcal{V}$. The only interesting case in the inductive step is $\Phi = [a]\Psi$. Suppose first $[a]\Psi \in s$. Then $\Psi \in s'$ for all s' such that $s \xrightarrow{a} s'$. So via the induction hypothesis $s \models_{\mathcal{M}_J^\alpha} [a]\Psi$. Instead suppose $s \models_{\mathcal{M}_J^\alpha} [a]\Psi$ but $[a]\Psi \notin s$. We show that if $[a]\Psi \notin s$ then $\Psi \notin s'$ for some s' such that $s \xrightarrow{a} s'$ which contradicts $s \models_{\mathcal{M}_J^\alpha} [a]\Psi$. This we do by construction. Let $\Gamma_0 = \{\neg\Psi\} \cup \Delta$ where $\Delta = \{\Phi \mid s \vdash_J [a]\Phi\}$ where these formulas need not all be in α . First Γ_0 is J -consistent. For otherwise for some finite $\beta \subseteq \Delta$ we would have $J \vdash \beta \rightarrow \Psi$ and so by the axioms and the rule Nec $J \vdash [a]\beta \rightarrow [a]\Psi$. But $\beta \in \Delta$ (as this set is closed under $\wedge$). So $s \vdash_J [a]\Psi$ which contradicts J -consistency of s . By

Lemma 6.2.1 there is a $\Gamma \supseteq \Gamma_0$ such that $\Gamma \in \text{Jmc}_{\alpha'}$ where α' is the full modal language (of the appropriate sort). Now consider $s' = \Gamma \cap \alpha$. Clearly $s' \in \text{Jmc}_{\alpha}$ and $s \xrightarrow{a} s'$. But $\Psi \notin s'$. ■

Let J be the axiom system for minimal modal logic. Strong completeness of J follows from Lemma 6.2.1 and Theorem 6.2.3. Any J -consistent set Γ is satisfiable, as witnessed by any model $\mathcal{M}_J^\alpha$ with $\Gamma \subseteq \alpha$. Two important models stand out for each Γ . First is the minimal model when α is the smallest closed set containing Γ . If Γ is finite then so is this model. Second is the largest model when α is the set of all modal formulas (of the appropriate sort). This model, abbreviated to $\mathcal{M}_J$, is known as the *canonical J-model*, as every J -consistent set Γ is satisfiable within it.

Theorem 6.2.3 does not deliver, by itself, completeness of those extensions of the axiom system for K such as the systems $KG(v, w', w, v)$ and the axiom system for PDL . A further condition is required in these cases, that the resulting model $\mathcal{M}_J^\alpha$ is of the appropriate kind. In the case of the system for $KG(v, w', w, v')$ the model $\mathcal{M}_J^\alpha$ should obey the condition $C(v, w', w, v')$:

$$\text{if } s \xrightarrow{v} s_1 \text{ and } s \xrightarrow{w} s_2 \text{ then } \exists s'. s_1 \xrightarrow{w'} s' \text{ and } s_2 \xrightarrow{v'} s'$$

Strong completeness of this family of axiom systems follows from the next result which states that the canonical model has this property.

6.2.4: Theorem

If J is $KG(v, w', w, v')$ then $\mathcal{M}_J$ satisfies the model condition $C(v, w', w, v')$.

Proof. Suppose $s \xrightarrow{v} s_1$ and $s \xrightarrow{w} s_2$ within $\mathcal{M}_J$. Let $\beta = \beta_1 \cup \beta_2$ where $\beta_1 = \{\Phi \mid [w']\Phi \in s_1\}$ and $\beta_2 = \{\Psi \mid [v']\Psi \in s_2\}$. These sets are not empty as they at least contain tt . We show that β is J -consistent. Suppose not. Then $\vdash_J \Phi \rightarrow \neg\Psi$ (where $\Phi \in \beta_1$ and $\Psi \in \beta_2$, as these sets are closed under $\wedge$). So $\vdash_J [w']\Phi \rightarrow [w']\neg\Psi$ via axioms and rules of J . Consequently, $[w']\neg\Psi \in s_1$. Because $s \xrightarrow{v} s_1$, $\langle v \rangle[w']\neg\Psi \in s$. Therefore, by the $G(v, w', w, v')$ schema $[w](v')\neg\Psi \in s$ too. But then $\langle v' \rangle\neg\Psi \in s_2$ as $s \xrightarrow{w} s_2$. But this is a contradiction because $[v']\Psi \in s_2$. The proof is completed by showing that any maximal J -consistent set $\Gamma \supseteq \beta$ has the property $s_1 \xrightarrow{w'} \Gamma$ and $s_2 \xrightarrow{v'} \Gamma$ which is clear from the definition of β . ■

In general the minimal modal model $\mathcal{M}_J^\alpha$, when J is $KG(v, w', w, v')$, does not obey the condition $C(v, w', w, v')$. Imposing extra closure conditions on α rectifies this but at the expense of its finiteness. However, for particular instances of v, w', w and v' , when Γ is finite, there is a finite subset β of α (containing Γ) with the property that $\forall \Phi \in \alpha. \exists \Psi \in \beta. \vdash_J \Phi \leftrightarrow \Psi$. In which case the finite model $\mathcal{M}_J^\beta$ is sufficient for proving completeness.

Unlike the axiom system $KG(v, w', w, v')$, the axiom system J for PDL (of Section 2.4) does not have a canonical model. The problem, as noted in Section 2.3, is the second-order condition that $\xrightarrow{a^*}$ is the reflexive and transitive closure of $\xrightarrow{a}$. The model $\mathcal{M}_J$ contains strange states. For instance the set $\Gamma = \{[w]Q \mid w \in \{a\}^*\} \cup \{\neg[a^*]Q\}$ is J -consistent (because every finite subset of it is). So there is a state s in $\mathcal{M}_J$ with $\Gamma \subseteq s$. By Theorem 6.2.4 there is an s' with $\neg Q \in s'$ and where $s \xrightarrow{a^*} s'$. But for all $w \in \{a\}^*$ not $s \xrightarrow{w} s'$. Moreover this shows that the axiom system for PDL is not strongly complete because although Γ is J -consistent it is not satisfiable in a dynamic model.

At the other extreme the minimal model $\mathcal{M}_J^\alpha$ is also not a dynamic model (where α is the smallest closed set containing the J -consistent Φ). However by imposing further closure conditions on α the model $\mathcal{M}_J^\alpha$ becomes *almost* a dynamic model. A set α is *Fischer-Ladner closed*, after (Fischer and Ladner 1979) if it is closed under subformulas, negation and the following: if $[a \cup b]\Psi \in \alpha$ then $[a]\Psi, [b]\Psi \in \alpha$; if $[a; b]\Psi \in \alpha$ then $[a][b]\Psi \in \alpha$; if $[\Psi?]\Psi' \in \alpha$ then $\Psi \in \alpha$; and, if $[a^*]\Psi \in \alpha$ then $[a][a^*]\Psi \in \alpha$. The smallest closed set α containing Φ is finite. For such an α the model $\mathcal{M}_J^\alpha$ is not quite a dynamic model because its label set is not a Kleene algebra. Shadowing $\mathcal{M}_J^\alpha$ is the dynamic model $\mathcal{N}_J^\alpha$ whose state set and valuation is the same as $\mathcal{M}_J^\alpha$. But its label set $\mathcal{L}'$ and its transition relations $\xrightarrow{a}$ differ from $\mathcal{L}$ and $\xrightarrow{a}$ of $\mathcal{M}_J^\alpha$. Let $\mathcal{L}_0 \subseteq \mathcal{L}$ be the atomic labels of $\mathcal{L}$. For $a \in \mathcal{L}_0$ let $\xrightarrow{a} = \xrightarrow{a}$. Now assume $\mathcal{L}'$ is the full Kleene algebra generated from $\mathcal{L}_0$ (and arbitrary tests). The transition relations for $\xrightarrow{a} \in \mathcal{L}' - \mathcal{L}_0$ are defined as for dynamic models (with $s \xrightarrow{\Phi} s$ just in case $s \wedge \Phi$ is J -consistent). By definition $\mathcal{N}_J^\alpha$ is a dynamic model. So completeness of the axiom system for PDL follows if Theorem 6.2.3 holds for this model. This is guaranteed by the next result as the theorem does hold for $\mathcal{M}_J^\alpha$.

6.2.5: Theorem

- i. For $a \in \mathcal{L}$. $\xrightarrow{a} \subseteq \xrightarrow{a}$,
- ii. If $[a]\Phi \in s$ and $s \xrightarrow{a} s'$ then $\Phi \in s'$.

Proof. Both results follow by induction on the structure of a . We just consider i. The base case when a is atomic or a test holds by definition. Let $S = \{s_1, \dots, s_n\}$ be the finite state set of $\mathcal{M}_J^\alpha$ and $\mathcal{N}_J^\alpha$. Because α is closed under subformulas and negation then $J \vdash s_1 \vee \dots \vee s_n$. Assume that $a = b; c$ and $s \xrightarrow{a} s'$. So $s \wedge \langle b; c \rangle s'$ is J -consistent. Consequently, so are $s \wedge \langle b \rangle \langle c \rangle s'$ and $s \wedge (\bigvee S \wedge \langle c \rangle s')$ via J axioms and rules. So for some s_i , $s \wedge \langle b \rangle (s_i \wedge \langle c \rangle s')$ is J -consistent. Hence $s \xrightarrow{b} s_i \xrightarrow{c} s'$. So by the induction hypothesis $s \xrightarrow{b} s_i \xrightarrow{c} s'$, and therefore $s \xrightarrow{b; c} s'$. Suppose $a = b \cup c$ and $s \xrightarrow{a} s'$. Then $s \wedge \langle b \cup c \rangle s'$ is J -consistent. But then either

$s \wedge \langle b \rangle s'$ or $s \wedge \langle c \rangle s'$ is J -consistent. Now the result follows that $s \xrightarrow{a} s'$. Finally, assume $a = b^*$ and $s \xrightarrow{a} s'$. Hence $s \wedge \langle a \rangle s'$ is J -consistent. Let $S' \subseteq S$ be the smallest set containing s such that if $t \in S'$ and $t \wedge \langle b \rangle t'$ is J -consistent then $t' \in S'$. If $s' \in S'$ then via the induction hypothesis $s \xrightarrow{a} s'$. So instead suppose $s' \notin S'$. Assume $S' = \{s_1, \dots, s_k\}$. For all $t \in S - S'$, $J \vdash \bigvee S' \rightarrow [b] \neg t$. Therefore $J \vdash \bigvee S' \rightarrow [b] \bigvee S'$. By the Nec rule and the induction axiom $J \vdash \bigvee S' \rightarrow [a] \bigvee S'$. But $J \vdash s \rightarrow \bigvee S'$ and $J \vdash \bigvee S' \rightarrow \neg s'$. So if $s' \notin S'$ then $J \vdash s \rightarrow [a] \neg s'$ which contradicts that $s \xrightarrow{a} s'$. $\blacksquare$

A decision procedure for determining whether a *PDL* formula is satisfiable can be extracted from this completeness result by appealing to tableau methods.

6.3 Points and schedulers

The building blocks of completeness proofs are maximal J -consistent sets, as we have seen in the case of modal logics. Members of Jmc_α were regarded as states, and transition system models were constructed from them. Here we prove parts of Propositions 3.4.1, 3.4.2, and 3.5.4; that the temporal axiom systems *LT*, *VL**T*, and *VL**TFC* are complete. The additional complication, not present in the modal case, is that we need to build both runs and states from members of Jmc_α , when J is a temporal axiom system. We now call members of Jmc_α *points*.

A future run will just be an ω length sequence of points. Typically we use σ and π to range over such sequences of sets: $\pi = \Gamma_0 \Gamma_1 \dots$. Here the transition labels are omitted as the temporal operators of these logics do not appeal to them. We use the notation $\sigma(i)$ to represent the i th point in σ , and σ^i to be the i th suffix. First, we isolate an acceptable subset of ω length sequences of points.

6.3.1: Definition

An ω length sequence σ of Jmc_α points is admissible if for all $i \geq 0$:

- i. if $O\Phi \in \alpha$ then $O\Phi \in \sigma(i)$ iff $\Phi \in \sigma(i+1)$,
- ii. if $F\Phi \in \alpha$ then $F\Phi \in \sigma(i)$ iff $\exists j \geq i. \Phi \in \sigma(j)$,
- iii. if $\Phi U \Psi \in \alpha$ then $\Phi U \Psi \in \sigma(i)$ iff $\exists j \geq i. \Psi \in \sigma(j)$ and $\forall k : i \leq k < j. \Phi \in \sigma(k)$.

Notice that σ is admissible iff every suffix σ^i is also admissible.

A first attempt at building admissible sequences is to define a next (or suffix) relation $\mathcal{X}$ on Jmc_α :

$$\Gamma \mathcal{X} \Delta \text{ iff } \Gamma \cup \{O\Psi \mid \Psi \in \Delta\} \text{ is } J\text{-consistent}$$

with the intention that if $\Gamma_0 \mathcal{X} \Gamma_1 \mathcal{X} \dots$ then the sequence $\Gamma_0 \Gamma_1 \dots$ is admissible. Certainly, as the next lemma shows, each $\Gamma_0 \in Jmc_\alpha$ gives rise to at least one ω length sequence for any logic J which extends the axiom system for $L(O)$ of Section 3.4.

6.3.2: Lemma

$\forall \Gamma \in Jmc_\alpha. \exists \Gamma' \in Jmc_\alpha. \Gamma \mathcal{X} \Gamma'.$

Proof. Let β be the full temporal language under consideration so $\alpha \subseteq \beta$. Suppose $\Gamma \in Jmc_\alpha$. Then by Lemma 6.2.1 there is a $\Delta \in Jmc_\beta$ such that $\Gamma \subseteq \Delta$. Let $\Delta^1 = \{\Phi | O\Phi \in \Delta\}$. By the axioms and rules for O it follows that $\Delta \in Jmc_\beta$ too, and by definition $\Delta \mathcal{X} \Delta^1$. Let Γ' be $\Delta^1 \cap \alpha$. It follows that $\Gamma \mathcal{X} \Gamma'$. ■

If O is the sole linear time operator of J then any ω length sequence generated by $\mathcal{X}$ is admissible. But this is not true for temporal logics involving F (or U) as well. Here we face a major problem. If the set α is closed under the operator O (so $O\Phi \in \alpha$ whenever $\Phi \in \alpha$) then inadmissible sequences can be explicitly generated. Consider the J -consistent set $\beta = \{O^n Q \mid n \geq 0\} \cup \{F \neg Q\}$. So $\beta \subseteq \Gamma \in Jmc_\alpha$ for some Γ . But any ω sequence from Γ must be inadmissible. This is similar to the non-existence of the canonical model for PDL , which here shows that LT , $\forall LT$, and $\forall LTF$ are not strongly complete. The problem was overcome in the case of PDL by restricting to finite sets of formulas. But if α is finite (and, therefore, not closed under O) then the relation $\mathcal{X}$ on Jmc_α is not functional. This means that, in general, inadmissible sequences can also be generated by $\mathcal{X}$, merely by making inappropriate choices of $\mathcal{X}$ successors which do not fulfill formulas of the form $F\Phi$ and $\Phi U \Psi$. To overcome this problem we add one additional closure condition on α , that if $\Phi U \Psi \in \alpha$ then $F\Psi \in \alpha$ (or $\text{tt} U \Psi$ when F is not a primitive operator), and we also introduce some extra structure.

The extra structure comes in the guise of simple schedulers which control generation of runs under $\mathcal{X}$. Suppose α is finite. For each $\Gamma \in Jmc_\alpha$ let Γ^+ be the possible $\mathcal{X}$ successors of Γ , the set $\{\Delta \mid \Gamma \mathcal{X} \Delta\}$. Lemma 6.3.2 guarantees that Γ^+ is not empty. A permutation of Γ^+ is any sequence $\Gamma_0 \dots \Gamma_n$ where each Γ_i is distinct, and $\{\Gamma_0, \dots, \Gamma_n\} = \Gamma^+$. A scheduler for Jmc_α is a function which associates a permutation of Γ^+ with each $\Gamma \in Jmc_\alpha$. We let f_Γ be the sequence associated with Γ : so $f_\Gamma(0)$ is its initial member and f_Γ^1 its initial suffix. Moreover, if f is a scheduler and σ a permutation of Γ^+ then $f[\sigma/\Gamma]$ is the scheduler like f except that it associates σ with Γ . An *extended* point is a pair (Γ, f) where $\Gamma \in Jmc_\alpha$ and f is a scheduler on Jmc_α . The $\mathcal{X}$ relation is redefined for extended points: $(\Gamma, f) \mathcal{X} (\Gamma', g)$ iff $f[\sigma/\Gamma](0) = \Gamma'$ and $f[\sigma/\Gamma]^1 = g$. The relation $\mathcal{X}$ is now functional since f determines the $\mathcal{X}$ successor for Γ (which is then placed at the end of the

permutation). Now $\mathcal{X}$ only generates admissible sequences on extended points when J contains the axiom system for LT of Section 3.4.

6.3.3: Lemma

If $(\Gamma_0, f_0)\mathcal{X}(\Gamma_1, f_1)\mathcal{X} \dots$ then $\Gamma_0\Gamma_1 \dots$ is admissible.

Proof. Let $(\Gamma_0, f_0)\mathcal{X}(\Gamma_1, f_1) \dots$, and let $\sigma = \Gamma_0\Gamma_1 \dots$. By induction on Φ we show that σ is admissible. If Φ is $O\Psi$ then this follows as $\Gamma_i\mathcal{X}\Gamma_{i+1}$ for all i . Suppose $F\Psi \in \alpha$ and $\Psi \in \Gamma_j$ then $F\Psi \in \Gamma_j$. And as $J \vdash OF\Psi \rightarrow F\Psi$ then also $F\Psi \in \Gamma_k$ for any $k < j$. Suppose $F\Psi \in \Gamma_i$. We need to show that $\Psi \in \Gamma_j$ for some $j \geq i$. Suppose not. Then for all $j \geq i$, $\neg\Psi \in \Gamma_j$. But as $J \vdash F\Psi \rightarrow \Psi \vee OF\Psi$ consequently $F\Psi \in \Gamma_j$ for all $j \geq i$. Consider any suffix of σ^i such that each Γ in it occurs infinitely often. There is such a suffix σ^k as Jmc_α is finite. Let $\{\Gamma_1, \dots, \Gamma_m\}$ be all the points occurring in σ^k . By definition of the scheduler, for each i , $\Gamma_i^+ \subseteq \{\Gamma_1, \dots, \Gamma_m\}$. Let $\Phi' = \Gamma_1 \vee \dots \vee \Gamma_m$. It therefore follows that $J \vdash \Gamma_i \rightarrow O\Phi'$. So $J \vdash \Phi' \rightarrow O\Phi'$. By the rule RG and the induction axiom it follows that $J \vdash \Phi' \rightarrow G\Phi'$. But $J \vdash \Phi' \rightarrow F\Psi \wedge \neg\Psi$. So we get the contradiction $J \vdash \Phi' \rightarrow G(F\Psi \wedge \neg\Psi)$. A similar argument holds for $\Phi U \Psi$ given the extra closure condition on α . ■

Lemma 6.3.3 is the crucial step in the proof of completeness of the axiom system LT . Suppose J is this axiom system and α is a *finite* closed set. We now construct a canonical finite model $\mathcal{M}_J^\alpha$. Its set of states is the family of extended points (Γ, f) where $\Gamma \in Jmc_\alpha$ and f is a scheduler for Jmc_α . The transition relation $\xrightarrow{\alpha}$ is just $\mathcal{X}$, and the set of runs $\mathcal{R}$ contains *all* future runs through the underlying transition system: so $\mathcal{R}$ is suffix (fusion, limit, and prefix) closed. Finally, the valuation $\mathcal{V}$ decrees that $\mathcal{V}(Q) = \{(\Gamma, f) \mid Q \in \Gamma\}$. A little piece of notation: we write $\Psi \in (\Gamma, f)$ just in case $\Psi \in \Gamma$. An important consequence of the definition of $\mathcal{M}_J^\alpha$ is the following.

6.3.4: Theorem

If $\Psi \in \alpha$ and $\sigma \in \mathcal{R}_{\mathcal{M}_J^\alpha}$ then for all $i \geq 0$. $\sigma^i \models_{\mathcal{M}_J^\alpha} \Psi$ iff $\Psi \in \sigma(i)$.

Proof. By induction on Ψ , utilizing Lemma 6.3.3 for formulas whose main connective is a temporal operator. ■

This theorem proves completeness of LT . Suppose Φ is LT -consistent, then let α be the smallest closed set containing Φ . By Lemma 6.2.1 there is a point Γ containing Φ . Consider any state (Γ, f) of $\mathcal{M}_{LT}^\alpha$ and unique run from it: by Theorem 6.3.4 Φ is true of σ .

The proofs of completeness of the axiom systems $\forall LT$ and $\forall LTFC$ require us to distinguish between points and states, a separation that is

essential for interpreting the path modality $\forall$. One idea is to construct states from equivalence classes of points. A formula Φ is a *state* formula iff $J \vdash \Phi \leftrightarrow \forall \Phi$ or $J \vdash \Phi \leftrightarrow \exists \Phi$ (compare Section 3.3). Notice that Φ is a state formula provided that $\neg\Phi$ is too. Let ST_α be the state formulas of α . Two points in Jmc_α are *state equivalent*, denoted by $\equiv$, if they contain the same state formulas: $\Gamma \equiv \Delta$ iff $\Gamma \cap ST_\alpha = \Delta \cap ST_\alpha$. The axioms and rules for $\forall$ of $\forall LT$ guarantee the next property.

6.3.5: Lemma

If $\exists \Phi \in \Gamma$ then for some $\Delta \equiv \Gamma$. $\Phi \in \Delta$.

Proof. It suffices to show that the set $\Delta_0 = (\Gamma \cap ST_\alpha) \cup \{\Phi\}$ is J-consistent. If it is then by Lemma 6.2.1 Δ_0 can be extended into a point $\Delta \in Jmc_\alpha$, and for any such Δ , $\Delta \equiv \Gamma$. Suppose Δ_0 is not consistent. Then for some $\Phi_1, \dots, \Phi_n \in \Gamma \cap ST_\alpha$, $J \vdash \Phi_1 \wedge \dots \wedge \Phi_n \rightarrow \neg\Phi$. Since $J \vdash \Phi_i \leftrightarrow C_i\Phi_i$ where C_i is either $\forall$ or $\exists$, also $J \vdash C_1\Phi_1 \wedge \dots \wedge C_n\Phi_n \rightarrow \neg\Phi$. By the rules and axioms for $\forall$ and $\exists$ we can then derive $J \vdash C_1\Phi_1 \wedge \dots \wedge C_n\Phi_n \rightarrow \forall\neg\Phi$. But then $\forall\neg\Phi \in \Gamma$ which contradicts that $\exists\Phi \in \Gamma$. ■

For each $\Gamma \in Jmc_\alpha$ let $|\Gamma|$ be the set $\{\Delta \mid \Gamma \equiv \Delta\}$. We now construct a finite canonical transition system $\mathcal{T}_J^\alpha$ when J is any of the branching time logics, and α is finite. Its set of states consists of pairs of the form $(|\Gamma|, f)$ where $\Gamma \in Jmc_\alpha$ and f is a scheduler, and the transition relation $\xrightarrow{a}$ is defined as follows: $(|\Gamma|, f) \xrightarrow{a} (|\Delta|, g)$ iff $\exists \Gamma' \in |\Gamma|$ and $\Delta' \in |\Delta|$. $(\Gamma', f) \xrightarrow{a} (\Delta', g)$. Moreover, for any of the models constructed below we assume the expected valuation $\mathcal{V}$ which stipulates that $\mathcal{V}(Q) = \{(|\Gamma|, f) \mid Q \in \Gamma\}$, which is well defined as Q is a state formula.

Missing are the definitions of the families of runs. These will consist of a subset of all the possible runs through $\mathcal{T}_J^\alpha$ (which is total by Lemma 6.3.2). The required runs should generate admissible sequences of Jmc_α points. The following useful result shows that there can be at most one such sequence of points associated with any run.

6.3.6: Lemma

If $(|\Gamma_0|, f_0) \xrightarrow{a} (|\Gamma_1|, f_1) \xrightarrow{a} \dots$ is a run through $\mathcal{T}_J^\alpha$ then there is at most one admissible sequence of points $\Gamma'_0\Gamma'_1 \dots$ with $\Gamma'_j \in |\Gamma_j|$, for all $j \geq 0$.

Proof. A simple induction shows that if $\Gamma'_0\Gamma'_1 \dots$ and $\Delta'_0\Delta'_1 \dots$ are both admissible when Γ'_j and Δ'_j belong to $|\Gamma_j|$ then $\Gamma'_j = \Delta'_j$, for any $j \geq 0$. ■

In the case of $\forall LT$ we assume that $(|\Gamma_0|, f_0) \xrightarrow{a} (|\Gamma_1|, f_1) \xrightarrow{a} \dots$ belongs to the set of runs $\mathcal{R}$ provided

$$\exists \Gamma'_0 \in |\Gamma_0|. (\Gamma'_0, f_0) \mathcal{X} (\Gamma'_1, f_1) \mathcal{X} \dots \text{ and } \forall j > 0. \Gamma'_j \in |\Gamma_j|$$

Clearly, $\mathcal{R}$ is suffix closed (but need not be fusion closed). Moreover, it is also limit closed. By definition there is at least one admissible sequence of points associated with any such run σ , and therefore by the previous lemma there is exactly one: we let $Ad(\sigma)$ be this unique sequence of points. Let $\mathcal{M}_f^\alpha$ be the future time model built on $\mathcal{T}_f^\alpha$ with the run set $\mathcal{R}$, and the valuation $\mathcal{V}$ described above. The following result amounts to completeness of $\forall L T$ both with respect to arbitrary models and limit closed models.

6.3.7: Theorem

If $\Psi \in \alpha$ and $\sigma \in \mathcal{R}$ then for all $i \geq 0$. $\sigma^i \models_{\mathcal{M}_f^\alpha} \Psi$ iff $\Psi \in (Ad(\sigma))(i)$.

Proof. This merely extends Theorem 6.3.4. There is only one new case when Ψ is $\forall \Psi'$, which follows from Lemma 6.3.5 and the definition of $Ad(\sigma)$. ■

The proof of completeness of $\forall L T F C$ is similar. However extra closure conditions need to be imposed on the set α : if $\Phi, \Psi \in \alpha$ then $\Phi \wedge \Psi \in \alpha$; and if $\Phi \in \alpha$ then $\forall \Phi \in \alpha$. Now the smallest closed set containing Φ is not finite. However, there is a finite subset $\beta \subseteq \alpha$ with the property: $\forall \Psi \in \alpha. \exists \Psi' \in \beta. \forall L T F C \vdash \Psi \leftrightarrow \Psi'$. Assume that α now ranges over such finite representatives. The following extra result is needed for showing that the set of runs below is fusion closed.

6.3.8: Lemma

If $\Gamma \mathcal{X} \Delta$ then $\forall \Delta' \in |\Delta|. \exists \Gamma' \in |\Gamma|. \Gamma' \mathcal{X} \Delta'$.

Proof. Suppose $\Gamma \mathcal{X} \Delta$ and $\Delta' \in |\Delta|$. We show $\Gamma' \mathcal{X} \Delta'$ for some $\Gamma' \in |\Gamma|$. It suffices to prove that the set $\Gamma'_0 = (\Gamma \cap ST_\alpha) \cup \{O\Psi \mid \Psi \in \Delta'\}$ is $\forall L T F C$ -consistent. Suppose not. Then $(\Gamma \cap ST_\alpha) \vdash \neg(O\Psi_1 \wedge \dots \wedge O\Psi_k)$ for some $\Psi_1, \dots, \Psi_k \in \Delta'$. But then by O and $\forall$ axioms and rule, $(\Gamma \cap ST_\alpha) \vdash \forall O \neg(\Psi_1 \wedge \dots \wedge \Psi_k)$. Consequently, by the axiom FC and closure conditions above $\Delta \vdash \forall \neg(\Psi_1 \wedge \dots \wedge \Psi_k)$ which is impossible as $\Delta' \equiv \Delta$. ■

Now $(|\Gamma_0|, f_0) \xrightarrow{a} (|\Gamma_1|, f_1) \xrightarrow{a} \dots$ counts as a run in $\mathcal{R}$ through $\mathcal{T}_f^\alpha$ if the following holds:

$$\exists i \geq 0. \exists \Gamma'_i \in |\Gamma_i|. (\Gamma'_i, f_i) \mathcal{X} (\Gamma'_{i+1}, f_{i+1}) \mathcal{X} \dots \text{ and } \forall j > i. \Gamma'_j \in |\Gamma_j| \}$$

$\mathcal{R}$ is both suffix and fusion closed: the latter follows from Lemma 6.3.8. Moreover there is exactly one admissible sequence of points associated with any run $\sigma \in \mathcal{R}$. Theorem 6.3.7 also applies to the model $\mathcal{M}_f^\alpha$ built on $\mathcal{T}_f^\alpha$ with this new run set and the valuation as before, and thereby shows completeness of $\forall L T F C$ with respect to fusion closed models.

References

- [1] Abrahamson, K. (1979). Modal logic of concurrent nondeterministic programs. *Lect. Notes in Comp. Sci.* **70**, 21-33.
- [2] Abramsky, S. (1989). Domain theory in logical form. To appear in *Annals of Pure and Applied Logic*.
- [3] Alpern, A. and Schneider, F. (1985). Defining liveness, *Inform. Process. Lett.* **21**, 181-185.
- [4] Amir, A. (1985). Separation in nonlinear time models. *Inf. and Con.* **66**, 177-203.
- [5] Amir, A. and Gabbay, D. (1987). Preservation of expressive completeness in temporal models. *Inf. and Con.* **72**, 66-83.
- [6] Baeten, J., Bergstra, J., and Klop, J. (1987). Decidability of bisimulation equivalence for processes generating context-free languages. *Lect. Notes in Comp. Science*, **259**, 94-113.
- [7] Banieqbal, B. and Barringer, H. (1989). Temporal logic with fixed points. *Lect. Notes in Comp. Sci.*, **398**, 62-74.
- [8] Barringer, H. and Kuiper, R. (1991). Applications of temporal logics. To appear in this Handbook.
- [9] Barringer, H., Kuiper, R. and Pnueli, A. (1984). Now you may compose temporal logic specifications. *Proc. 16th ACM Symp. on Theory of Computing*, 51-63.
- [10] Bergstra, J. and Klop, J. (1989). Process theory based on bisimulation semantics. *Lect. Notes in Comp. Science*, **354**, 50-122.
- [11] Ben-Ari, M., Halpern, J. and Pnueli, A. (1982). Deterministic propositional dynamic logic: finite models, complexity, and completeness. *J. of Comp. and Syst. Sci.* **25**, 402-417.
- [12] Ben-Ari, M., Manna, Z. and Pnueli, A. (1983). The temporal logic of branching time. *Acta Inform.* **20**, 207-226.
- [13] van Benthem, J. (1983). *Modal Logic and Classical Logic*, Bibliopolis.
- [14] van Benthem, J. (1984). Correspondence theory. In *Handbook of Philosophical Logic Vol. II*, ed. Gabbay, D. and Guenther, F., Reidel.
- [15] van Benthem, J. (1989). Modal logic as a theory of information. *Report LP-89-05, Institute for Language, Logic and Information, University of Amsterdam*.

- [16] van Benthem, J. (1990). Temporal Logic. Draft chapter for *Handbook of Logic in Artificial Intelligence and Logic Programming*, ed. Gabbay. D, Hogger. C, and Robinson. J., Oxford University Press.
- [17] Bradfield, J. and Stirling, C. (1990). Verifying temporal properties of processes. *Lect. Notes in Comput. Science*, **458**, 115-125.
- [18] Brookes, S. and Rounds, W. Behavioural equivalence relations induced by programming logics. *Lect. Notes in Comp. Sci.*, **154**, 97-108.
- [19] Browne, M., Clarke, E., and Grumberg, O. (1988). Characterizing finite Kripke structures in propositional temporal logic. *Theoret. Comp. Science.*, **59**, 115-131.
- [20] Bull, R. That all normal extensions of S4.3 have the finite model property. *Zeit Math. Logik. Grund.*, **12**, 341-344.
- [21] Bull, R. and Segerberg, K. (1984). Basic modal logic. In *Handbook of Philosophical Logic Vol. II*, edited Gabbay. D, and Guenther, F., Reidel, 1-88.
- [22] Büchi, J. (1962). On a decision method in restricted second order arithmetic. In *Logic, Methodology, and Philosophy of Science: Proc. 1960 Int. Congress*, Stanford Uni. Press, 1-11.
- [23] Chellas, B. (1980). *Modal Logic*. Cambridge University Press.
- [24] Clarke, E. and Emerson, E. (1981). Design and synthesis of synchronization skeletons using branching time temporal logic. *Lect. Notes in Comp. Sci.* **131**, 52-71.
- [25] Clarke, E., Emerson, E., and Sistla, A. (1983). Automatic verification of finite state concurrent systems using temporal logic specifications: a practical approach. *Proc. 10th ACM Symp. on Principles of Programming Languages*, 117-126.
- [26] Cleaveland, R. (1990). Tableau-based model checking in the propositional mu-calculus. *Acta Informat.*, **27**, 725-747.
- [27] Courcoubetis, C., Vardi, M., and Wolper, P. (1986). Reasoning about fair concurrent programs. *Proc. of 18th ACM Symp. on Theory of Computing*, 283-294.
- [28] De Nicola, R. (1987). Extensional equivalences for transition systems. *Acta Informatica*, **24**, 211-237.
- [29] De Nicola, R. and Vaandrager, V. (1990). Three logics for branching bisimulation. *Proc. 5th IEEE Symp. on Logic in Comput. Sci.*, 118-129.
- [30] Dijkstra, E. (1976). *A Discipline of Programming*. Prentice-Hall.

- [31] Emerson, E. (1983). Alternative semantics for temporal logics. *Theoret. Comput. Sci.*, **26**, 121-130.
- [32] Emerson, E. and Clarke, E. (1980). Characterizing correctness properties of parallel programs using fixpoints. *Lect. Notes in Comput. Sci.* **85**, 169-181.
- [33] Emerson, E. and Halpern, J. (1982). Decision procedures and expressiveness in the temporal logic of branching time. *Proc. of 14th ACM Symp. on Theory of Computing*, 169-180.
- [34] Emerson, E. and Halpern, J. (1986). "Sometimes" and "not never" revisited: on branching versus linear time. *J. Assoc. Comput. Mach.*, **33**, 151-178.
- [35] Emerson, E. and Sistla, A. (1984). Deciding full branching time logic. *Inform. and Contr.*, **61**, 175-201.
- [36] Fischer, M. and Ladner, R. (1979). Propositional dynamic logic of regular programs. *J. Comput. System Sci.* **18**, 194-211.
- [37] Fitting, M. (1983). *Proof Methods for Modal and Intuitionistic Logics*. Reidel.
- [38] Floyd, R. (1967). Assigning meanings to programs. In *Mathematical Aspects of Computer Science*. Edited Schwartz, J. American Math. Soc., 19-32.
- [39] Gabbay, D. (1976). *Investigations In Modal And Tense Logics With Applications To Problems In Philosophy And Linguistics*, Reidel.
- [40] Gabbay, D. (1981). Expressive functional completeness in tense logic. In *Aspects Of Philosophical Logic*. Edited Monnich, U. Reidel, 91-117.
- [41] Gabbay, D. (1989). The declarative past and imperative future: executable temporal logic for interactive systems. *Lect. Notes in Comput. Science*, **398**, 431-448.
- [42] Gabbay, D., Pnueli, A., Shelah, S., and Stavi, J. (1980). On the temporal analysis of fairness. *Proc. 7th ACM Symp. on Principles of Programming Languages*, 163-173.
- [43] Goldblatt, R. (1976). Metamathematics of modal logic. *Reports on Mathematical Logic* **6**, 41-78 and **7**, 21-52.
- [44] Gurevitch, Y. and Shelah, S. (1985). The decision problem for branching time logic. *The Journ. of Symb. Logic*, **50**, 668-681.

- [45] Hafer, T. and Thomas, W. (1987). Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. *Lect. Notes in Comput. Science*, **267**, 269-279.
- [46] Harel, D. (1984). Dynamic logic. In *Handbook of Philosophical Logic Vol II*, edited Gabbay, D., and Guenther, F., Reidel, 497-604.
- [47] Harel, D., Kozen, D., and Parikh, R. (1982). Process logic: expressiveness, decidability, completeness. *J. Comput. System Sci.*, **25**, 144-170.
- [48] Hennessy, M. (1984). Axiomatizing finite delay operators. *Acta Inform.*, **21**, 61-88.
- [49] Hennessy, M. (1988). *An Algebraic Theory of Processes*. MIT Press.
- [50] Hennessy, M. and Milner, R. (1980). On observing nondeterminism and concurrency. *Lect. Notes in Comp. Sci.* **85**, 295-309.
- [51] Hennessy, M. and Milner, R. (1985). Algebraic laws for nondeterminism and concurrency. *J. Ass. of Comput. Mach.* **32**, 137-162.
- [52] Hennessy, M. and Stirling, C. (1985). The power of the future perfect in program logics. *Inf. and Control*, **67**, 23-52.
- [53] Hintikka, J. (1962). *Knowledge And Belief*. Cornell Univ. Press.
- [54] Hoare, C. (1969). An axiomatic basis for computer programming. *Comm. of the ACM*, **12**, 576-580.
- [55] Hughes, G. and Cresswell, M. (1973). *An Introduction To Modal Logic*. Methuen.
- [56] Hüttel, H. (1990). SnS can be modally characterized. *Theoret. Comput. Sci.*, **74**, 239-248.
- [57] Kamp, H. (1968). *Tense Logic and the Theory of Linear Order*. Ph.D thesis, Univ. of California.
- [58] Kannellakis, P. and Smolka, S. (1990). CCS expressions, finite state processes, and three problems of equivalence. *Inf. and Comp.*, **86**, 43-68.
- [59] Keller, R. (1976). Formal verification of parallel programs. *Comm. of Assoc. of Comp. Mach.*, **19**, 561-572.
- [60] Koymans, R. (1989). *Specifying Message Passing and Time-Critical Systems with Temporal Logic*. Ph.D thesis, Technical University of Eindhoven.
- [61] Kozen, D. (1981). On the duality of dynamic algebras and Kripke models. *Lect. Notes in Comput. Sci.* **125**, 1-11.

- [62] Kozen, D. (1983). Results on the propositional μ -calculus. *Theoret. Comput. Sci* **27**, 333-354.
- [63] Kozen, D. and Parikh, R. (1981). An elementary proof of the completeness of PDL. *Theoret. Comput. Sci.*, **14**, 113-118.
- [64] Kripke, S. (1959). A completeness proof in modal logic. *J. Symbolic Logic*, **24**, 1-14.
- [65] Kripke, S. (1963). Semantical analysis of modal logic I, normal propositional calculi. *Zeit. für Math. Logik und Grund. der Math.*, **9**, 67-96.
- [66] Lamport, L. (1980). "Sometime" is sometimes "not never"—on the temporal logic of programs. *Proc. 7th Annual ACM Symp. on Prin. of Prog. Lang.*, 174-185.
- [67] Lamport, L. (1983). What good is temporal logic? *Proc. IFIP Congress*, North-Holland, 657-668.
- [68] Larsen, K. (1988). Proof systems for Hennessy-Milner logic with recursion. *Lect. Notes in Comput. Science*, **299**,
- [69] Lemmon, E. (1977). *An Introduction to Modal Logic*. Am. Phil. Quart. Mon. **11**, Oxford.
- [70] Lichtenstein, O. and Pnueli, A. (1984). Checking that finite state concurrent programs satisfy their linear specification. *Proc. 10th ACM Symp. on Princ. of Prog. Lang.*, 97-107.
- [71] Manna, Z. and Pnueli, A. (1981). Verification of concurrent programs: the temporal framework. In *The Correctness Problem in Computer Science*, edited Boyer, R. and Moore, J. Academic Press, 215-273.
- [72] Manna, Z. and Pnueli, A. (1989). The anchored version of the temporal framework. *Lect. Notes in Comput. Science*, **354**, 201-284.
- [73] Manna, Z. and Wolper, P. (1984). Synthesis of communicating processes from temporal logic specifications. *ACM Trans. on Prog. Lang. and Syst.*, **6**, 68-93.
- [74] Milner, R. (1989). *Communication and Concurrency*. Prentice Hall.
- [75] Moszkowski, B. (1986). *Executing Temporal Logic Programs*. Cambridge Univ. Press.
- [76] Park, D. (1981). Concurrency and automata on infinite sequences. *Lect. Notes in Comput. Sci.*, **154**, 561-572.
- [77] Pnueli, A. (1977). The temporal logic of programs. *Proc. of 18th Annual Symp. on Foundations of Computer Science*, 46-57.

- [78] Pratt, V. (1976). Semantical considerations on Floyd-Hoare logic. *Proc. 17th IEEE Symp. on Foundations of Comput. Sci.*, 109-121.
- [79] Pratt, V. (1979). Process logic. *Proc. 6th ACM Symp. on Principles of Programming Languages*, 93-100.
- [80] Pratt, V. (1980a). Dynamic algebras and the nature of induction. *Proc. 12th ACM Symp. Theory of Comput.*, 22-28.
- [81] Pratt, V. (1980b). A near optimal method for reasoning about action. *J. Comput. Systm. Sci.*, 20, 231-254.
- [82] Pratt, V. (1982). A decidable mu-calculus, *22nd IEEE Symp. on Found. of Comp. Sci.*, 421-427.
- [83] Prior, A. (1957). *Time and Modality*. Oxford University Press.
- [84] Prior, A. (1967). *Past, Present, and Future*. Oxford University Press.
- [85] Quielle, J. and Sifakis, J. (1981) Specification and verification of concurrent systems in CESAR. *Lect. Notes in Comput. Science*, 137, 337-350.
- [86] Rabin, M. (1969). Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141, 1-35.
- [87] Rescher, N. and Urquhart, A. (1971). *Temporal Logic*. Springer-Verlag.
- [88] Salwicki, A. (1977). Algorithmic logic: a tool for investigations of programs. In Butts and Hintikka (eds.) *Logic, Foundations of Mathematics, and Computability Theory*. Reidel, 281-295.
- [89] Sambin, G. and Vaccaro, V. (1988). Topology and duality in modal logic. *Annals of Pure and Applied Logic*, 37, 249-296.
- [90] Schlingloff, B-H. (1990). Expressive completeness of temporal logic over trees. To appear in *Journal of Applied Non Classical Logics*.
- [91] Sistla, A. (1985). On characterisation of safety and liveness properties in temporal logic. *Proc. 4th ACM Symp. Princ. of Dist. Comp.*, 39-48.
- [92] Sistla, A. and Clarke, E. (1985). The complexity of propositional linear temporal logic. *J. Assoc. of Comp. Mach.*, 32, 733-749.
- [93] Sistla, A., Vardi, M., and Wolper, P. (1987). The complementation problem for Büchi automata with applications to temporal logic. *Theoret. Comput. Science*, 49, 217-237.
- [94] Stirling, C. (1987). Modal logics for communicating systems. *Theoret. Comp. Science*, 49, 311-347.

- [95] Stirling, C. (1989). Comparing linear and branching time temporal logics. *Lect. Notes in Comput. Science*, **398**, 1-20.
- [96] Stirling, C. and Walker, D. (1989). Local model checking in the propositional mu-calculus. *Lect. Notes in Comput. Science*, **351**, 369-383.
- [97] Streett, R. (1982) Propositional dynamic logic of looping and converse is elementarily decidable. *Inf. and Control*, **54**, 121-141.
- [98] Streett, R. and Emerson, E. (1989). An automata theoretic decision procedure for the propositional mu-calculus. *Inf. and Comp.*, **81**, 249-264.
- [99] Thomas, W. (1987) On chain logic, path logic, and first-order logic over infinite trees. *Proc. 2nd IEEE Symp. on Logic in Comp. Sci.*, 245-256.
- [100] Thomas, W. (1989) Automata on infinite objects. To appear in *Handbook of Theoret. Comput. Science*, edited von Leeuwen, J., North-Holland.
- [101] Thomason, S. (1984). Combinations of tense and modality. In *Handbook of Philosophical Logic Vol II*, ed. Gabbay, D. and Guentner, F., Reidel, 135-165.
- [102] Vardi, M. and Wolper, P. (1986). Automata-theoretic techniques for modal logic of programs. *J. Comput. and System Sciences*, **32**, 183-221.
- [103] Winskel, G. (1985). A complete proof system for SCCS with modal assertions. *Lect. Notes in Comput. Science*, **206**, 392-410.
- [104] Winskel, G. (1989). Model checking the modal nu-calculus. *Lect. Notes in Comput. Science*, **372**.
- [105] Wolper, P. (1983). Temporal logic can be more expressive. *Inf. and Control*, **56**, 72-99.
- [106] Wolper, P., Vardi, M., and Sistla, A. (1983). Reasoning about infinite computation paths. *Proc. of 24th Annual Symp. on Foundations of Comp. Science*, 185-194.
- [107] Zanardo, A. (1991). A complete deductive-system for since-until branching-time logic. *Journ. of Philos. Logic*, **20**, 131-148.

**Copyright © 1991, Laboratory for Foundations of Computer Science,
University of Edinburgh. All rights reserved.**

**Reproduction of all or part of this work
is permitted for educational or research use
on condition that this copyright notice is
included in any copy.**