Technical Report TR-ARP-15-95

Automated Reasoning Project
Research School of Information Sciences and Engineering
and Centre for Information Science Research
Australian National University

May 5, 1996

Rajeev Goré

Abstract: This document is a complete draft of a chapter by Rajeev Goré on "Tableau Methods for Modal and Temporal Logics" which is part of the "Handbook of Tableau Methods", edited by M. D'Agostino, D. Gabbay, R. Hähnle and J. Posegga, to be published in 1996 by Kluwer, Dordrecht.

Any comments and corrections are highly welcome. Please email me at rpg@arp.anu.edu.au

The latest version of this document can be obtained via my WWW home page: `http://arp.anu.edu.au/`

Tableau Methods for Modal and Temporal Logics

Rajeev Goré

Contents

1 Introduction

Modal and temporal logics are finding new and varied applications in Computer Science in fields as diverse as Artificial Intelligence [MST91], Models for Concurrency [Sti92] and Hardware Verification [NFKT87]. Often the eventual use of these logics boils down to the task of deducing whether a certain formula of a logic is a logical consequence of a set of other formula of the same logic. The method of semantic tableaux is now well established in the field of Automated Deduction [OS88, BHE95, BP95] as a viable alternative to the more traditional methods based on resolution [CL73]. In this chapter we give a systematic and unified introduction to tableau methods for automating deduction in modal and temporal logics. We concentrate on the propositional fragments restricted to a two-valued (classical) basis and assume some prior knowledge of modal and temporal logic, but give a brief overview of the associated Kripke semantics to keep the chapter self-contained.

One of the best accounts of proof methods for modal logics is the book by Melvin Fitting [Fit83]. To obtain generality, Fitting uses Smullyan's idea of abstract consistency properties and the associated maximal consistent set approach for proving completeness. As Fitting notes, maximal consistent sets can also be used to determine decidability, but in general, they do not give information about the efficacy of the associated tableau method. Effectiveness however is of primary importance for automated deduction, and a more constructive approach using finite sets, due to Hintikka, is more appropriate. We therefore base our work on a method due to Hintikka [Hin55] and Rautenberg [Rau83].

In Section 2 we give the syntax and (Kripke) semantics for propositional modal logics, the traditional axiomatic methods for defining modal logics and the correspondences between axioms and certain conditions on frames.

In Section 3 we give a brief overview of the history of modal tableau systems.

Section 4 is the main part of the chapter and it can be split into two parts.

In Section 4.1 we motivate our study of modal tableau systems. In Section 4.2 we cover the syntax of modal tableau systems, explain tableau constructions and tableau closure. Section 4.3 covers the (Kripke) semantics of modal tableau systems and the notions of soundness and completeness with respect to these semantics. Sections 4.4-4.6 relate our tableau systems with the well-known systems of Fitting and Smullyan, and then cover proof theoretic issues like structural rules, admissible rules and derivable rules. Section 4.7 covers decidability issues like the subformula property, the analytic superformula property, and finiteness of proof search. Sections 4.8-4.10 explain the technical machinery we need to prove the soundness and completeness results, and their connections with decidability. The first half of Section 4 concludes with a summary of the techniques covered so far and sets up the specific examples of tableau systems covered in the second half.

The second half of Section 4 covers tableau systems for: the basic systems; modal logics with epistemic interpretations; modal logics with "provability" interpretations and mono-modal logics with temporal interpretations. Sections 4.16-4.17 cover proof-theoretic issues again by highlighting some deficiencies of the tableau methods of Section 4. Section 4.18 closes the loop on the Kripke semantics by highlighting the finer characterisation results that are immediate from our constructive proofs of tableau completeness. Finally, Section 4.19 covers the connection between modal tableau systems and modal sequent systems, and the admissibility of the cut rule.

Section 5 is a very brief guide to tableau methods for multimodal logics, particularly linear and branching time logics over discrete frames with operators like "next", "until" and "since".

Section 6 gives a brief overview of labelled modal tableau systems where labels attached to formulae are used to explicitly keep track of the possible worlds in the tableau constructions.

2 Preliminaries

2.1 Syntax and Notational Conventions

The sentences of modal logics are built from a denumerable non-empty set of primitive propositions $\mathcal{P} = \{p_1, p_2, \cdots\}$, the parentheses) and (, together with the classical connectives $\wedge$ ("and"), $\vee$ ("inclusive or"), $\neg$ ("not"), $\rightarrow$ ("implies"), and the non-classical unary modal connectives $\Box$ ("box") and $\Diamond$ ("diamond").

A well-formed formula, hereafter simply called a **formula**, is any sequence of these symbols obtained from the following rules: any $p_i \in \mathcal{P}$ is a formula and is usually called an **atomic formula**; and if A and B are formulae then so are $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(\Box A)$ and $(\Diamond A)$.

For convenience we use $\bot$ to denote a constant false formula $(p_1 \wedge \neg p_1)$ (say) and then use $\top = (\neg\bot)$ to define a constant true formula.

Lower case letters like p and q denote members of $\mathcal{P}$. Upper case letters from the beginning of the alphabet like A and B together with P and Q (all possibly annotated) denote formulae. Upper case letters from the end of the alphabet like X, Y, Z (possibly annotated) denote *finite* (possibly empty) sets of formulae.

The symbols $\neg, \wedge, \vee$ and $\rightarrow$ respectively stand for logical negation, logical conjunction, logical disjunction and logical (material) implication. To enable us to omit parentheses, we adopt the convention that the connectives $\neg, \Box, \Diamond$ are of equal binding strength but bind tighter than $\wedge$ which binds tighter than $\vee$ which binds tighter than $\rightarrow$. So $\neg A \vee B \wedge C \rightarrow D$ should be read as $((\neg A) \vee (B \wedge C)) \rightarrow D$. The symbols $\Box$ and $\Diamond$ can take various meanings but traditionally stand for "necessity" and "possibility". In the context of temporal logic, they stand for "always" and "eventually" so that $\Box A$ is read as "A is always true" and $\Diamond A$ is read as "A is eventually true".

2.2 Axiomatics of Modal Logics

The logics we shall study are all normal extensions of the basic modal logic **K** and are traditionally axiomatised by taking the rule of necessitation **RN** (if A is a theorem then so is $\Box A$) and modus ponens **MP** (if A and $A \rightarrow B$ are theorems then so is B) as inference rules, and by taking the appropriate formulae from Figure 1 as axiom *schemas*. Thus the rule of uniform substitution **US** is built in so that any substitutional instance of an axiom schema or theorem, is also a theorem.

If a logic is axiomatised by adding axioms $A_1, A_2, \cdots, A_n$ to **K** then its name is written as $\mathbf{KA_1A_2} \cdots \mathbf{A_n}$. Sometimes however, the logic is so well known in the literature by another name that we revert to the traditional name. The logic **KT4**, for example, is usually known as **S4**.

For an introduction to these notions see the introductory texts by Hughes and Cresswell [HC68, HC84] or Chellas [Che80], or the article by Fitting [Fit93].

We write $\vdash_{\mathbf{L}} A$ to denote that A is a theorem of an axiomatically formulated logic **L**. As with classical logic, the notion of theoremhood can be extended to the notion of "deducibility" where we write $X \vdash_{\mathbf{L}} A$ to mean "there is a deduction of A from the set of formulae X". However, some care is needed when extending this notion to modal logics if we want to preserve the "deduction theorem": $X \ \vdash_{\mathbf{L}} \ (A \rightarrow B)$ iff $X \cup \{A\} \ \vdash_{\mathbf{L}} \ B$, since it is well known that the deduction theorem fails if we use the notion of deducibility from classical Hilbert system formulations (due to the rule of necessitation). Fitting [Fit93] shows how to set up the notions of "deducibility" so that the deduction theorem holds, but since axiomatics are of a secondary nature here, we omit details. The important point is that

Axiom	Defining Formula
K	$\Box(A \to B) \to (\Box A \to \Box B)$
T	$\Box A \to A$
D	$\Box A \to \Diamond A$
4	$\Box A \to \Box\Box A$
5	$\Diamond A \to \Box\Diamond A$
B	$A \to \Box\Diamond A$
2	$\Diamond\Box A \to \Box\Diamond A$
M	$\Box\Diamond A \to \Diamond\Box A$
L	$\Box((A \wedge \Box A) \to B) \vee \Box((B \wedge \Box B) \to A)$
3	$\Box(\Box A \to B) \vee \Box(\Box B \to A)$
X	$\Box\Box A \to \Box A$
F	$\Box(\Box A \to B) \vee (\Diamond\Box B \to A)$
R	$\Diamond\Box A \to (A \to \Box A)$
G	$\Box(\Box A \to A) \to \Box A$
Grz	$\Box(\Box(A \to \Box A) \to A) \to A$
Go	$\Box(\Box(A \to \Box A) \to A) \to \Box A$
Z	$\Box(\Box A \to A) \to (\Diamond\Box A \to \Box A)$
Zbr	$\Box(\Box A \to A) \to (\Box\Diamond\Box A \to \Box A)$
Zem	$\Box\Diamond\Box A \to (A \to \Box A)$
Dum	$\Box(\Box(A \to \Box A) \to A) \to (\Diamond\Box A \to \Box A)$
Dbr	$\Box(\Box(A \to \Box A) \to A) \to (\Box\Diamond\Box A \to \Box A)$

Fig. 1. Axiom names and defining formulae.

the notion of theoremhood $\vdash_{\mathbf{L}} A$ remains the same since it corresponds to "deducibility" of A from the empty set viz: $\{\} \vdash_{\mathbf{L}} A$. We return to this point in Section 4.3.

2.3 Kripke Semantics For Modal Logics

A Kripke **frame** is a pair $\langle W, R\rangle$ where W is a non-empty set (of possible worlds) and R is a binary relation on W. We write wRw' iff $(w, w') \in R$ and we say that world w' is **accessible from** world w, or that w' is **reachable from** w, or w' is a **successor** of w, or even that w **sees** w'. We also write $w\not R w'$ to mean $(w, w') \notin R$.

A Kripke **model** is a triple $\langle W, R, V\rangle$ where V is a mapping from primitive propositions to sets of worlds; that is, $V : \mathcal{P} \mapsto 2^W$. Thus $V(p)$ is the set of worlds at which p is "true" under the valuation V.

Given some model $\langle W, R, V\rangle$, and some $w \in W$, we write $w \models p$ iff $w \in V(p)$, and say that w satisfies p or p is true at w. We also write $w \not\models p$ to mean $w \notin V(p)$. This satisfaction relation $\models$ is then extended to more complex formulae according to the primary connective as below:

Axiom	Condition	First Order Formula
T	Reflexive	$\forall w : R(w, w)$
D	Serial	$\forall w \, \exists w' : R(w, w')$
4	Transitive	$\forall s, t, u : (R(s, t) \wedge R(t, u)) \rightarrow R(s, u)$
5	Euclidean	$\forall s, t, u : (R(s, t) \wedge R(s, u)) \rightarrow R(t, u)$
B	Symmetric	$\forall w, w' : R(w, w') \rightarrow R(w', w)$
2	Weakly-directed	$\forall s, t, u \, \exists v :$ $(R(s, t) \wedge R(s, u)) \rightarrow (R(t, v) \wedge R(u, v))$
L	Weakly-connected	$\forall s, t, u :$ $(R(s, t) \wedge R(s, u)) \rightarrow (R(t, u) \vee t = u \vee R(u, t))$
X	Dense	$\forall u, v \, \exists w : R(u, v) \rightarrow (R(u, w) \wedge R(w, v))$

Fig. 2. Axioms and corresponding first order conditions on R.

$w \models p$	iff	$w \in V(p)$;
$w \models \neg A$	iff	$w \not\models V(A)$;
$w \models A \wedge B$	iff	$w \models A$ and $w \models B$;
$w \models A \vee B$	iff	$w \models A$ or $w \models B$;
$w \models A \rightarrow B$	iff	$w \not\models A$ or $w \models B$;
$w \models \Box A$	iff	for all $v \in W$, $w \not R v$ or $v \models A$;
$w \models \Diamond A$	iff	there exists some $v \in W$, with wRv and $v \models A$.

We say that w **satisfies** A iff $w \models A$ where the valuation is left as understood. If $w \models A$ we sometimes also say that A **is true at** w, or that w **makes** A **true**.

A formula A is **satisfiable in a model** $\langle W, R, V \rangle$ iff there exists some $w \in W$ such that $w \models A$. A formula A is **satisfiable on a frame** $\langle W, R \rangle$, iff there exists some valuation V and some world $w \in W$ such that $w \models A$. A formula A is **valid in a model** $\langle W, R, V \rangle$, written as $\langle W, R, V \rangle \models A$, iff it is true at every world in W. A formula A is **valid in a frame** $\langle W, R \rangle$, written as $\langle W, R \rangle \models A$, iff it is valid in all models $\langle W, R, V \rangle$ (based on that frame). An axiom (schema) is valid in a frame iff all instances of that axiom (schema) are valid in all models based on that frame.

Given a class of frames $\mathcal{C}$, an axiomatically formulated logic $\mathbf{L}$ is **sound** with respect to $\mathcal{C}$ if for all frames $\mathcal{F} \in \mathcal{C}$ and all formulae A we have $\vdash_{\mathbf{L}} A$ implies $\mathcal{F} \models A$. Logic $\mathbf{L}$ is **complete** with respect to $\mathcal{C}$ if for all frames $\mathcal{F} \in \mathcal{C}$ and all formulae A we have $\mathcal{F} \models A$ implies $\vdash_{\mathbf{L}} A$. A logic $\mathbf{L}$ is **characterised** by a class of frames $\mathcal{C}$ iff $\mathbf{L}$ is sound and complete with respect to $\mathcal{C}$.

2.4 Known Correspondence and Completeness Results

The logics we study are known to be characterised by certain classes of frames because it is known that particular axioms correspond to particular restrictions on the reachability relation R. That is, suppose $\langle W, R\rangle$ is a frame, then a certain axiom A_1 will be valid on $\langle W, R\rangle$ *if and only if* the reachability relation R meets a certain condition. Many of the restrictions are definable as formulae of first order logic where the binary predicate $R(x, y)$ represents the reachability relation, as shown in Figure 2, where the correspondences between certain axioms and certain conditions are also summarised. Some interesting properties of frames which cannot be captured by any one axiom are given in Figure 3; see [Gol87]. But some quite bizarre axioms, whose corresponding conditions cannot be expressed in first-order logic [vB84, vB83] are of particular interest precisely because of this “higher order” nature. Some of these “higher order” conditions are explained next.

Given a frame $\langle W, R\rangle$, an R-**chain** is a sequence of (not necessarily distinct) points from W with $w_1Rw_2Rw_3R\cdots Rw_n$. An ∞-R-**chain** is an R-chain where n can be chosen arbitrarily large. A **proper** R-**chain** is an R-chain where the points are distinct. For example, a single reflexive point gives an (improper) ∞-R-chain: $wRwRwRw\cdots$.

Transitive frames are of particular interest when R is viewed as a flow of time. Informally, if $\langle W, R\rangle$ is a frame where R is transitive, then a **cluster** C is a maximal subset of W such that for all *distinct* worlds w and w' in C we have wRw' and $w'Rw$. A cluster is **degenerate** if it is a single irreflexive world, otherwise it is **nondegenerate**. A nondegenerate cluster is **proper** if it consists of two or more worlds. A nondegenerate cluster is **simple** if it consists of a single reflexive world. Note that in a nondegenerate cluster, R is reflexive, transitive *and symmetric*.

Because clusters are maximal we can order them with respect to R and speak of a cluster preceding another one. Similarly, a cluster C is **final** if no other cluster succeeds it and a cluster is **last** if *every* other cluster precedes it. For an introduction to Kripke frames, Kripke models and the notion of clusters see Goldblatt [Gol87] or Hughes and Cresswell [HC84].

Figure 4 encapsulates the known characterisation results for each of our logics by listing the conditions on some class of frames that characterises each logic. The breaks in Figure 4 correspond to the grouping of the tableau systems for these logics under Sections 4.12-4.15. Thus we define a frame

Property Name	Property of R
Irreflexive	$\forall w : \neg R(w, w)$
Intransitive	$\forall s, r, t : (R(s, t) \wedge R(t, r)) \rightarrow \neg R(s, r)$
Antisymmetric	$\forall s, t : (R(s, t) \wedge R(t, s)) \rightarrow (s = t)$
Asymmetric	$\forall w_1, w_2 : R(w_1, w_2) \rightarrow \neg R(w_2, w_1)$
Strict-order	$\forall w_1, w_2 : (w_1 \neq w_2) \rightarrow (R(w_1, w_2) \text{ exor } R(w_2, w_1))$

Fig. 3. Names of some non-axiomatisable conditions on R.

to be an **L-frame** iff it meets the restrictions of Figure 4. Then, a model $\langle W, R, V \rangle$ is an **L-model** iff $\langle W, R \rangle$ is an **L**-frame. A formula A is **L-valid** iff it is true in every world of every **L**-model. An **L**-model $\langle W, R, V \rangle$ is **an L-model for a finite set** X of formulae iff there exists some $w_0 \in W$ such that for all $A \in X$, $w_0 \models A$. A set X is **L-satisfiable** iff there is an **L**-model for X.

An axiomatically formulated logic **L** has the **finite model property** if every nontheorem A of **L** can be falsified at some world in some *finite* **L**-model. That is, if $\not\vdash_{\mathbf{L}} A$ implies that $\{\neg A\}$ has a *finite* **L**-model.

2.5 Logical Consequence

Suppose we are given some finite set of formulae Y, some formula A, and assume that the logic of interest is **L**. We say that the formula A is a **local logical consequence** of the set Y iff: for every **L**-model $\langle W, R, V \rangle$ and for every $w \in W$, if $w \models Y$ then $w \models A$. We write $Y \models_{\mathbf{L}} A$ whenever A is a local logical consequence of Y in logic **L**; thus the subscript is for the logic, not for the word "local".

Since both Y and A are evaluated at the same world w in this definition, it is straightforward to show that $Y \models_{\mathbf{L}} A$ iff $\{\} \models_{\mathbf{L}} \widehat{Y} \rightarrow A$ where $\{\}$ is the empty set, and $\widehat{Y}$ is just the conjunction of the members of Y. Furthermore, a semantic verison of the usual deduction theorem holds for local logical consequence viz: $Y, A \models_{\mathbf{L}} B$ iff $Y \models_{\mathbf{L}} A \rightarrow B$ where we write Y, A to mean $Y \cup \{A\}$.

As we saw in Section 2.2, the traditional axiomatically formulated logics obey the deduction theorem only if deducibility is defined in a special way. Fitting [Fit83] shows that a stronger version of logical consequence called global logical consequence corresponds to this notion of deducibility. Fitting also gives tableau systems that cater to both notions of logical consequence. We concentrate only on the local notion since Fitting's techniques can be used to extend our systems to cater for the global notion.

2.6 Summary

The semantic notion of validity $\models A$ and the axiomatic notion of theoremhood $\vdash A$ are tied to each other via the notions of soundness and completeness of the axiomatic deducibility relation $\vdash$ with respect to some class of Kripke frames. These notions can be generalised respectively to logical consequence $Y \models A$ and $Y \vdash A$. By careful definition we can maintain the soundness and completeness results intact for these generalisations. Unfortunately, axiomatic systems are notoriously bad for proof *search* because they give no guidance on how to look for a proof. Tableau systems also give rise to a syntactic notion of theoremhood but have the added benefit that they facilitate proof search in a straightforward way. Such systems are the subject of the rest of this chapter.

L	Axiomatic Basis	**L**-frame restriction
K	K	no restriction
KT	KT	reflexive
KD	KD	serial
K4	$K4$	transitive
K5	$K5$	euclidean
KB	KB	symmetric
KDB	KDB	symmetric and serial
B	KTB	reflexive and symmetric
KD4	$KD4$	serial and transitive
K45	$K45$	transitive and euclidean
KD5	$KD5$	serial and euclidean
KD45	$KD45$	serial, transitive and euclidean
S4	$KT4$	reflexive and transitive
KB4	$KB4$	symmetric and transitive
S5	$KT5$	reflexive, transitive and symmetric
S4R	$KT4R$	reflexive, transitive and $\forall x, y, z$: $(x \neq z \wedge R(x, z)) \rightarrow (R(x, y) \rightarrow R(y, z))$
S4F	$KT4F$	reflexive, transitive and $\forall x, y, z$: $(R(x, z) \wedge \neg R(z, x)) \rightarrow (R(x, y) \rightarrow R(y, z))$
S4.2	$KT4.2$	reflexive, transitive and weakly-directed
S4.3	$KT4.3$	reflexive, transitive and weakly-connected
S4.3.1	$KT4.3Dum$	reflexive, transitive, weakly-connected and no nonfinal proper clusters
S4Dbr	$KT4Dbr$	reflexive, transitive and no nonfinal proper clusters
K4DL	$KD4L$	serial, transitive and weakly-connected
K4DLX	$KD4LX$	serial, transitive, weakly-connected and dense
K4DLZ	$KD4LZ$	serial, transitive, weakly-connected and no nonfinal non-degenerate clusters
K4DZbr	$K4DZbr$	serial, transitive and no nonfinal nondegenerate clusters
G	KG	transitive and no ∞-R-chains (irreflexive)
Grz	$KGrz$	reflexive, transitive, no proper clusters and no proper ∞-R-chains
K4Go	$K4Go$	transitive, no proper clusters and no proper ∞-R-chains
GL	KGL	transitive, weakly-connected, no proper clusters and no ∞-R-chains (irreflexive)

Fig. 4. Axiomatic Bases and **L**-frames

3 History of Modal Tableau Systems

The history of modal tableau systems can be traced back through two routes, one semantic and one syntactic.

The syntactic route began with the work of Gerhard Gentzen [Gen35] and the numerous attempts to extend Gentzen's results to modal logics. Curry [Cur52] appears to be the first to seek Gentzen systems for modal logics, soon followed by Ohnishi and Matsumoto [OM57b, OM59, OM57a]. Kanger [Kan57] is the first to use extra-logical devices to obtain Gentzen systems and is the precursor of what are now known as prefixed or labelled tableau systems. Once the basic method was worked out other authors tried to find similar systems for other logics, turning modal Gentzen systems into an industry for almost twenty years.

Not surprisingly, modal Gentzen systems involve a cut-elimination theorem. In many respects this early work on modal Gentzen systems was very difficult because these authors had no semantic intuitions to guide them and had to work quite hard to obtain a syntactic cut-elimination theorem. As we shall see, the task is much easier when we use the associated Kripke semantics.

The semantic route began with the work of Beth for classical propositional logic [Bet55, Bet53] but lay dormant for modal logics for almost twenty years until the advent of Kripke semantics [Kri59]. From then on, modal tableau systems, and in general modal logic, witnessed a resurgence.

The two routes began to meet in the late sixties when it was realised that classical semantic tableau systems and classical Gentzen systems were essentially the same thing. Zeman [Zem73] appears to be the first to give an account of both traditions simultaneously, although he is sometimes unable to relate his tableau systems to his Gentzen systems (c.f. his tableau system for **S4.3** is cut-free, yet his sequent system for **S4.3** is not). Rautenberg [Rau79] gives a rigorous account and covers many logics but has not received much attention as his book is written in German. Fitting's book [Fit83] is the most widely known and covers most of the basic logics.

During the eighties the two traditions were seen as two sides of the same coin, but more recently, the semantic tradition has assumed prominence in the field of automated deduction, while the syntactic tradition has gained prominence in the field of type theory [Mas93], [Bor93]. In automated deduction, the primary emphasis is on finding a proof, whereas in type theory, the primary emphasis is on the ability to distinguish different proofs so as to put a computational interpretation on proofs.

Regardless of this historical basis, there are essentially two types of tableau systems which we shall call **explicit systems** and **implicit systems**. Recall that tableau systems are essentially semantic in nature, hence the reachability relation R plays a crucial part. In explicit systems, the reachability relation is represented *explicitly* by some device, and we are allowed to reason directly about the known properties of R, such as transitivity or reflexivity. In implicit systems, there is no explicit representation of the reachability relation, and these properties must be *built into the rules* in some way since we are not allowed to reason explicitly about R. We shall see that in some sense the two types of systems are dual in nature since implicit systems can be turned into explicit systems by giving a systematic method or strategy for the application of the implicit tableau rules.

Here is an outline of what follows. In the first few sections we introduce the syntax of implicit modal tableau systems by defining the form of the rules and tableau systems. These are all purely syntactic aspects of modal tableau systems allowing us to associate a syntactic deducibility relation with modal tableau systems. In the second part we introduce the semantics of modal tableau rules, and systems, and define the notions of soundness and completeness of modal tableau systems with respect to these semantics. In the last part we introduce the mathematical structures that we shall need to prove the soundness and completeness of the given tableau systems.

We then give tableau systems in decreasing detail for: the basic modal logics; the monotonic modal logics used to define nonmonotonic modal logics of knowledge and belief; modal logics with "provability interpretations"; monomodal logics of linear and branching time; and multimodal logics of linear and branching time.

In the later sections of this chapter we introduce explicit tableau systems since they are an extension of implicit tableau systems. The extra power of explicit tableau systems comes from the labels which carry very specific semantic information about the (counter-)model under construction. Consequently we see that explicit tableau systems are better for the symmetric logics.

For the sake of brevity we do not consider quantified modal logics, but see Fitting [Fit83] for a treatement of quantified modal tableau systems.

4 Modal Tableau Systems With Implicit Accessibility

4.1 Purpose of Modal Tableau Systems

As stated in the introduction, we concentrate on the use of modal tableau systems for performing deduction. In this context, modal tableau systems can be seen as refutation procedures that decompose a given set of formulae into a network of sets with each set representing a possible world in the associated Kripke model. Thus, our modal tableau systems are anchored to the semantics of the modal logic although they can be used in sequent form to obtain metamathetical results like interpolation theorems as well; see [Fit83] and [Rau83, Rau85].

The main features of semantic tableau systems carry over from classical propositional logic in that a set of formulae X is deemed consistent if and only if no tableau for X closes. Furthermore, from these open tableaux, we can construct a model to demonstrate that X is indeed satisfiable, thus tying the syntactic notion of consistency to semantic notion of satisfiability.

Now, assume we are given some finite set of formulae $Y = \{A_1, \cdots, A_k\}$, and some formula A. Let $\widehat{Y} = (A_1 \wedge A_2 \wedge \cdots \wedge A_k)$ with $\widehat{Y} = \bot$ when $k = 0$. By definition, if the set $Y \cup \{\neg A\}$ is *not* **L**-satisfiable, then, in every **L**-model, each world that makes each member of Y true, must also make A true. That is, if the set $Y \cup \{\neg A\}$ is *not* **L**-satisfiable, then, the formula $\widehat{Y} \rightarrow A$ must be **L**-valid. Modal tableau systems give us a purely syntactic method of determining whether or not some given formula is **L**-valid. Thus, they give us a method of determining whether A is a *local logical consequence* of a set of formulae Y.

4.2 Syntax of Modal Tableau Systems

The most popular tableau formulation is due to Smullyan as expounded by Fitting [Fit83]. Following Hintikka [Hin55] and Rautenberg [Rau83, Rau85], we use a slightly different formulation where formulae are carried from one tableau node to its child because the direct correspondence between sequent systems and tableau systems is easier to see using this formulation. To minimise the number of rules, we work with primitive notation, taking $\Box, \neg$ and $\wedge$ as primitives and defining all other connectives from these. Thus, for example, there are no explicit rules for $\vee$ and $\rightarrow$ but these can be obtained by rewriting $A \vee B$ as $\neg(\neg A \wedge \neg B)$ and $A \rightarrow B$ as $\neg(A \wedge \neg B)$. All our tableau systems work with *finite* sets of formulae.

We use the following notational conventions:

- $\bot$ denotes a constant false proposition and $\emptyset$ denotes the empty set;
- p, q denote primitive (atomic) propositions from $\mathcal{P}$;
- P, Q, Q_i and P_i denote (well formed) formulae;
- X, Y, Z denote *finite* (possibly empty) sets of (well formed) formulae;
- $(X; Y)$ stands for $X \cup Y$ and $(X; P)$ stands for $X \cup \{P\}$;
- $\Box X$ stands for $\{\Box P \mid P \in X\}$;
- $\neg\Box X$ stands for $\{\neg\Box P \mid P \in X\}$.

We use P and Q as formulae in the tableau rules and use A and B in the axioms to try and separate the two notions. Note that $(X; P; P) = (X; P)$ and also that $(X; P; Q) = (X; Q; P)$ so that the number of copies of the formulae and their order is immaterial as far as the notation is concerned.

A **tableau rule** ρ consists of a **numerator** $\mathcal{N}$ above the line and a (finite) list of **denominators** $\mathcal{D}_1$, $\mathcal{D}_2$,... ,$\mathcal{D}_k$ (below the line) separated by vertical bars:

$$(\rho)\ \frac{\mathcal{N}}{\mathcal{D}_1 \mid \mathcal{D}_2 \mid \cdots \mid \mathcal{D}_k}$$

$$(\wedge)\ \frac{X; P \wedge Q}{X; P; Q} \qquad (\bot)\ \frac{X; P; \neg P}{\bot} \qquad (\vee)\ \frac{X; \neg(P \wedge Q)}{X; \neg P \ \mid\ X; \neg Q}$$

$$(\neg)\ \frac{X; \neg\neg P}{X; P} \qquad (\theta)\ \frac{X; Y}{X} \qquad (K)\ \frac{\Box X; \neg\Box P}{X; \neg P}$$

Fig. 5. Tableau rules for $\mathcal{C}\mathbf{K}$ where X, Y are sets and P, Q are formulae.

The numerator is a finite set of formulae and so is each denominator. We use the terms numerator and denominator rather than premiss and conclusion to avoid confusion with the sequent terminology. As we shall see later, each tableau rule is read downwards as "if the numerator is **L**-satisfiable, then so is one of the denominators".

The numerator of each tableau rule contains one or more distinguished formulae called the **principal formulae**. Each denominator usually contains one or more distinguished formulae called the **side formulae**. Each tableau rule is labelled with a name which usually consists of the main connective of the principal formula, in parentheses, but may consist of a more complex name. The rule name appears at the left when the rule is being defined, and appears at the right when we use a particular instance of the rule.

For example, below at right is a tableau rule with:

1. a rule name $(\vee)$;
2. a numerator $X; \neg(P \wedge Q)$ with a principal formula $\neg(P \wedge Q)$; and
3. two denominators $X; \neg P$ and $X; \neg Q$ with respective side formulae $\neg P$ and $\neg Q$.

$$(\vee)\ \frac{X; \neg(P \wedge Q)}{X; \neg P \mid X; \neg Q}$$

A **tableau system** (or calculus) $\mathcal{C}\mathbf{L}$ is a finite collection of tableau rules $\rho_1, \rho_2, \cdots, \rho_m$ identified with the set of its rule names; thus $\mathcal{C}\mathbf{L} = \{\rho_1, \rho_2, \cdots, \rho_m\}$. Figure 5 contains some tableau rules which we shall later prove are those that capture the basic normal modal logic **K**; thus $\mathcal{C}\mathbf{K} = \{(\bot), (\wedge), (\vee), (\neg), (K), (\theta)\}$.

A **$\mathcal{C}$L-tableau for** X is a finite tree with root X whose nodes carry *finite formula sets.* A tableau rule with numerator $\mathcal{N}$ is applicable to a node carrying set Y if Y is an instance of $\mathcal{N}$. The steps for extending the tableau are:

- choose a leaf node n carrying Y where n is not an end node, and choose a rule ρ which is applicable to n;

- if ρ has k denominators then create k successor nodes for n, with successor i carrying an appropriate instantiation of denominator D_i;

- all with the proviso that if a successor s carries a set Z and Z has already appeared on the branch from the root to s then s is an end node.

A branch in a tableau is **closed** if its end node is $\{\perp\}$; otherwise it is **open**. A tableau is **closed** if all its branches are closed; otherwise it is **open**.

The rule $(\perp)$ is really a check for inconsistency, therefore, we say that a set X is **$\mathcal{C}$L-consistent** if no $\mathcal{C}\mathbf{L}$-tableau for X is closed. Conversely we say that a formula A is a **theorem** of $\mathcal{C}\mathbf{L}$ iff there is a closed tableau for the set $\{\neg A\}$. We write $\vdash_{\mathcal{C}\mathbf{L}} A$ if A is a theorem of $\mathcal{C}\mathbf{L}$ and write $Y \vdash_{\mathcal{C}\mathbf{L}} A$ if $Y \cup \{\neg A\}$ is $\mathcal{C}\mathbf{L}$-inconsistent.

Example 4.2.1. The formula $\Box(p \to q) \to (\Box p \to \Box q)$ is an instance of the axiom **K**. Its negation can be written in primitive notation and simplified to $\Box(\neg(p \wedge \neg q)) \wedge \Box p \wedge \neg\Box q$. Below at left is a closed $\mathcal{C}\mathbf{K}$-tableau for the (singleton) set $X = \{\Box(\neg(p \wedge \neg q)) \wedge \Box p \wedge \neg\Box q\}$ where each node is labelled at the right by the rule that produces its successor(s). Below at right is a more succinct version of the same $\mathcal{C}\mathbf{K}$-tableau. Hence $\Box(p \to q) \to (\Box p \to q)$ is a theorem of $\mathcal{C}\mathbf{K}$.

$$
\begin{array}{c}
\{\Box\neg(p \wedge \neg q) \wedge \Box p \wedge \neg\Box q\}\ (\wedge) \\
| \\
\{\Box\neg(p \wedge \neg q) \wedge \Box p, \neg\Box q\}\ (\wedge) \\
| \\
\{\Box\neg(p \wedge \neg q), \Box p, \neg\Box q\}\ (K) \\
| \\
\{\neg(p \wedge \neg q), p, \neg q\}\ (\vee) \\
\swarrow \qquad \searrow \\
\begin{array}{cc}
\{\neg p, p, \neg q\}\ (\bot) & \{\neg\neg q, p, \neg q\}\ (\bot) \\
| & | \\
\bot & \bot
\end{array}
\end{array}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\Box\neg(p \wedge \neg q) \wedge \Box p \wedge \neg\Box q}{\Box\neg(p \wedge \neg q) \wedge \Box p; \neg\Box q}\,(\wedge)}{\Box\neg(p \wedge \neg q); \Box p; \neg\Box q}\,(\wedge)}{\neg(p \wedge \neg q); p; \neg q}\,(K)}{\neg p; p; \neg q\ (\bot) \quad | \quad \neg\neg q; p; \neg q\ (\bot)}\,(\vee)
$$

4.3 Soundness and Completeness

Tableau systems give us a syntactic way to define consistency, and hence theoremhood. As with the axiomatic versions of these notions, the notions of soundness and completeness relate these syntactic notions to the semantic notions of satisfiability and validity as follows.

Soundness: We say that $\mathcal{C}\mathbf{L}$ is **sound with respect to L-frames** (the Kripke semantics of $\mathbf{L}$) if: $Y \ \vdash_{\mathcal{C}\mathbf{L}} A$ implies $Y \ \models_{\mathbf{L}} A$. In words, if there is a closed $\mathcal{C}\mathbf{L}$-tableau for $Y \cup \{\neg A\}$ then any $\mathbf{L}$-model that makes Y true at world w must make A true at world w.

Completeness: We say that $\mathcal{C}\mathbf{L}$ is **complete with respect to L-frames** (the Kripke semantics of $\mathbf{L}$) if: $Y \ \models_{\mathbf{L}} A$ implies $Y \ \vdash_{\mathcal{C}\mathbf{L}} A$. In words, if every $\mathbf{L}$-model that makes Y true at world w also make A true at world w then some $\mathcal{C}\mathbf{L}$-tableau for $Y \cup \{\neg A\}$ must close.

We already know that axiomatically formulated $\mathbf{L}$ is also sound and complete with respect to $\mathbf{L}$-frames. If we can show that $\mathcal{C}\mathbf{L}$ is also sound and complete with respect to **L**-frames then we can complete the link between $\mathcal{C}\mathbf{L}$ and $\mathbf{L}$ via: $Y \vdash_{\mathcal{C}\mathbf{L}} A$ iff $Y \models_{\mathbf{L}} A$ iff $\models_{\mathbf{L}} \widehat{Y} \rightarrow A$ iff $\vdash_{\mathbf{L}} \widehat{Y} \rightarrow A$. Thus our tableau systems, as given, capture axiomatically formulated theoremhood only. As stated previously, they can be easily extended to handle the stronger notion of "deducibility" using techniques for handling global logical consequence from Fitting [Fit83].

4.4 Relationship to Smullyan Tableau Systems

Tableau systems are often presented using trees where each node is labelled by a single (possibly signed) formula [Fit83]. The associated tableau rules then allow us to choose some formula on the current branch as the principal formula of the rule, and then to extend all branches below this formula by adding other formulae onto the end of these branches. For modal logics, some of the tableau rules demand the deletion of formulae from the current branch, as well as the addition of new formulae. In fact, the tableau rules are often summarised using set notation by collecting into a numerator all the formulae on the branch prior to a tableau rule application, and collecting into one or more denominators all the formulae that remain after the tableau rule application. Such summarised rules correspond exactly to the tableau rules we use. In particular, the thinning rule (θ) allows us to capture the desired deletion rules.

4.5 Structural Rules

Tableau systems are closely related to Gentzen systems and both often contain three rules known as structural rules; so called because they do not affect a particular formula in the numerator but the whole of the numerator itself.

4.5.1 Exchange

Since we use sets of formulae, the order of the formulae in the set is immaterial. Thus a commonly used rule called the "exchange" rule that simply swaps the order of formulae is implicit in our formulation.

4.5.2 Contraction

The $(\wedge)$ rule is shown below left. Consider the two applications of the $(\wedge)$ rule shown at right:

$$(\wedge)\,\frac{X; P \wedge Q}{X; P; Q} \qquad\qquad \frac{p \wedge q}{p; q}\,(\wedge) \qquad\qquad \frac{p \wedge q}{p \wedge q; p; q}\,(\wedge)$$

The left hand application is intuitive, corresponding to putting $X = \emptyset$, $P = p$, and $Q = q$ giving a numerator

$$\mathcal{N} = (X; P \wedge Q) = (\emptyset; p \wedge q) = \{p \wedge q\}$$

and hence obtaining the denominator

$$\mathcal{D} = (X; P; Q) = (\emptyset; p; q) = \{p, q\}.$$

However, the right hand derivation is also legal since we can put $X = \{p \wedge q\}$, $P = p$, and $Q = q$ to give the numerator

$$\mathcal{N} = (X; P \wedge Q) = (p \wedge q; p \wedge q) = \{p \wedge q\}$$

and hence obtain the denominator

$$\mathcal{D} = (X; P; Q) = (p \wedge q; p; q) = \{p \wedge q, p, q\}.$$

Thus, although our tableau rules seem to delete the principal formulae in a rule application, they also allow us to carry that formula into the denominator if we so choose.

Now, in classical propositional logic, it can be shown that the deletion of the principal formula does no harm. However, in certain modal logics, the deletion of the principal formula leads to incompleteness. That is, a tableau for X may not close if we always delete the principal formula, and yet, a similar tableau for X may close if we carry a copy of the principal formula into the denominator. For an example, see Example 4.12.1 on page 29.

Completeness is essential if our tableau systems are to be used as decision procedures, thus we need a way to duplicate formulae. It is tempting to add a rule called the contraction rule (ctn) as shown below left. And below at right is an application of it where we duplicate the formula $\Box p$ in $\mathcal{N} = \{p \wedge q, \Box p\}$:

$$(\mathrm{ctn})\ \frac{X; P}{X; P; P} \qquad\qquad \frac{p \wedge q; \Box p}{p \wedge q; \Box p; \Box p}\ (\mathrm{ctn})$$

But now we have a problem, for the definition of a tableau is in terms of nodes carrying *sets* and the two nodes of the right hand tableau carry identical sets since $(p \wedge q; \Box p) = (p \wedge q; \Box p; \Box p) = \{p \wedge q, \Box p\}$. Thus, any explicit application of the contraction rule immediately gives a cycle and stops the tableau construction. An explicit contraction rule is not what we want.

In order to avoid these complexities we shall omit an explicit contraction rule from our tableau systems and make no assumptions about the deletion or copying of formulae when moving from the numerator to the denominator. However, when we wish to copy the principal formula into the denominator we shall explicitly show it in the denominator. So for example, the rule below at left explicitly stipulates that a copy of the principal formula $P \wedge Q$ must be carried into the denominator, whereas the rule below at right allows us to choose for ourselves:

$$(\wedge') \frac{X; P \wedge Q}{X; P \wedge Q; P; Q} \qquad\qquad (\wedge) \frac{X; P \wedge Q}{X; P; Q}$$

4.5.3 Thinning

The thinning rule (θ) allows us to convert any tableau for a given set Y into a tableau for a bigger set $(X; Y)$ simply by adding $(X; Y)$ as a new root node. It encodes the monotonicity of a logic since it encodes the principle that if $Y \vdash_{\mathcal{C}\mathbf{L}} A$ then $X \cup Y \vdash_{\mathcal{C}\mathbf{L}} A$. In tableau systems for classical logic it can be built into the basic consistency check by using a base rule like our $(\perp)$ (shown below right) since all formulae that are not necessary to obtain closure can be stashed in the set X. Alternatively it becomes necessary if we use a base rule like the one shown at below left:

$$\frac{P; \neg P}{\perp} \qquad\qquad \frac{X; P; \neg P}{\perp} (\perp)$$

Consequently, our tableau system $\mathcal{C}\mathbf{K}$ is complete for classical propositional logic without (θ) and the thinning rule is required only for the modal aspects. The thinning rule can also be built into the modal rules as we shall show, but we choose to make it explicit because it helps to keep the modal rules simpler.

4.5.4 Cut

The cut rule shown below encodes the law of the excluded middle but suffers the disadvantage that the new formulae P and $\neg P$ are totally arbitrary, bearing no relationship to the numerator X. To use the (cut) rule we have to guess the correct P (although note that modal tableau systems based on Mondadori's system **KE** [DM94] can use cut sensibly):

$$(cut) \frac{X}{X; P \mid X; \neg P}$$

The redundancy of the cut rule is therefore very desirable and can be proved in two ways. The first is to allow the cut rule and show syntactically that whenever there is a closed $\mathcal{C}\mathbf{L}$-tableau for X containing uses of the cut rule, there is another closed $\mathcal{C}\mathbf{L}$-tableau for X containing no uses of the cut rule. This is the cut-elimination theorem of Gentzen. The alternative is to omit the cut rule from the beginning and show that the cut-free tableau system $\mathcal{C}\mathbf{L}$ is nevertheless sound and complete with respect to the semantics of the logic $\mathbf{L}$. For most of our systems, we follow this latter route.

A more practical version of the cut rule, known as **analytical cut**, is one where P is a subformula of some formula in X. Thus the formulae that appear in the denominator are not totally arbitrary. Some of our systems require such an analytic cut rule for completeness. The use of analytic cut is not as bad as it may seem since it can lead to exponentially shorter proofs.

4.6 Derived Rules and Admissible Rules

Our rules are couched in terms of (set) variables like X, which denote sets of formulae, and formulae variables like $\neg\Box P$ which denote formulae with a particular structure. Thus our rules are really rule schemata which we instantiate by instantiating X to a set of formulae, and instantiating $\neg\Box P$ to a particular formula like $\neg\Box q$ say. And up till now, we have always applied the rules to sets of formulae. But if a sequence of rule applications is used often then it is worth defining a new rule as a macro or derived rule. And in defining a macro, we apply rules to set variables and to formula variables, not to actual sets of formulae.

More formally, a rule (ρ) with numerator $\mathcal{N}$ and denominators $\mathcal{D}_1$, $\mathcal{D}_2$, $\cdots$,$\mathcal{D}_k$ is **derivable in $\mathcal{C}\mathbf{L}$** iff there is a finite $\mathcal{C}\mathbf{L}$-tableau that begins with the schema $\mathcal{N}$ itself, and has leaves labelled with the schemata $\mathcal{D}_1$, $\mathcal{D}_2$, $\cdots$,$\mathcal{D}_k$, but where the rules are applied to schema rather than to actual sets of formulae. The addition of derived rules does not affect soundness and completeness of $\mathcal{C}\mathbf{L}$ since their applications can be replaced by the macro-expansion.

For example, in order to apply the (K) rule, the numerator (schema) $\Box X; \neg\Box P$ is not allowed to contain nonmodal formulae like $p \wedge q$. Before applying the (K) rule, these undesirable elements have to be "thinned out" via the set Y as shown below left. But notice that here we have applied the (θ) rule, not to a set of formulae, but to a schema which represents a set of formulae. And similarly, the subsequent application of the (K) rule is also applied to a schema rather than an actual set of formulae. Since such an application of (θ) may be necessary before every application of (K) it may be worth defining a "derived rule" $(K\theta)$ which builds in this thinning step as shown below right. In fact, if we replace (K) by $(K\theta)$ in $\mathcal{C}\mathbf{K}$ then (θ) becomes superfluous since these are the only necessary applications of (θ).

$$\dfrac{\dfrac{Y; \Box X; \neg\Box P}{\Box X; \neg\Box P}(\theta)}{X; \neg P}(K) \qquad (K\theta)\,\dfrac{Y; \Box X; \neg\Box P}{X; \neg P}$$

On the other hand, it is often possible (and useful) to add extra rules even though these rules are not derivable. For example, the cut rule is not derivable in $\mathcal{C}\mathbf{K}$ since the denominators of each rule of $\mathcal{C}\mathbf{K}$ are always related to the numerator of that rule, whereas (cut) breaks this property since the P in the denominator is arbitrary.

We can ensure that the new rules do not add to the deductive power of the system as follows. Let (ρ) be an arbitrary tableau rule with a numerator $\mathcal{N}$ and n denominators $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_n$ and let $\mathcal{C}\mathbf{L}\rho$ be the tableau system $\mathcal{C}\mathbf{L} \cup \{(\rho)\}$. Then the rule (ρ) is said to be **admissible** in $\mathcal{C}\mathbf{L}$ if: X is $\mathcal{C}\mathbf{L}$-consistent iff X is $\mathcal{C}\mathbf{L}\rho$-consistent. That is, if: a $\mathcal{C}\mathbf{L}$-tableau for X is closed iff a $\mathcal{C}\mathbf{L}\rho$-tableau for X is closed.

Lemma 4.6.1. *If $\mathcal{C}\mathbf{L}$ is sound and complete with respect to $\mathbf{L}$-frames and (ρ) is sound with respect to $\mathbf{L}$-frames then (ρ) is admissible in $\mathcal{C}\mathbf{L}$.*

Proof: Since $\mathcal{C}\mathbf{L} \subseteq \mathcal{C}\mathbf{L}\rho$ we know that if X is $\mathcal{C}\mathbf{L}\rho$-consistent then X is $\mathcal{C}\mathbf{L}$-consistent. To prove the other direction suppose that $\mathcal{C}\mathbf{L}$ is sound and complete with respect to $\mathbf{L}$-frames, that (ρ) is sound with respect to $\mathbf{L}$-frames, and that X is $\mathcal{C}\mathbf{L}$-consistent. By the completeness of $\mathcal{C}\mathbf{L}$, the set X must be $\mathbf{L}$-satisfiable. Since (ρ) is sound with respect to $\mathbf{L}$-frames, so is $\mathcal{C}\mathbf{L}\rho$. Suppose X is not $\mathcal{C}\mathbf{L}\rho$-consistent. Then there is a closed $\mathcal{C}\mathbf{L}\rho$-tableau for X which must utilise the rule (ρ) since this is the only difference between $\mathcal{C}\mathbf{L}$ and $\mathcal{C}\mathbf{L}\rho$. But, by the soundness of $\mathcal{C}\mathbf{L}\rho$ this implies that X must be $\mathbf{L}$-unsatisfiable; contradiction. Hence X must be $\mathcal{C}\mathbf{L}\rho$-consistent. ∎

For example, there is no rule for $A \to B$ in our tableau system since we always use primitive notation and rewrite $A \to B$ as $\neg(A \wedge \neg B)$. But the following rules are clearly sound with respect to the semantics of classical logic, and hence are admissible for $\mathcal{C}\mathbf{PC}$ (the calculus $\mathcal{C}\mathbf{K}$ minus the (K) rule) since $\mathcal{C}\mathbf{PC}$ is sound and complete with respect to the same semantics:

$$(\to)\,\dfrac{X; P \to Q}{X; \neg P \mid X; Q} \qquad (\neg\to)\,\dfrac{X; \neg(P \to Q)}{X; P; \neg Q}$$

4.7 Subformula Property and Analytic Superformula Property

For a formula A, the **degree** $deg(A)$ counts the maximum depth of nesting while the **modal degree** $mdeg(A)$ counts the maximum depth of modal nesting. Their definitions are:

$deg(A) = 0$ when A is atomic
$deg(\neg A) = 1 + deg(A)$
$deg(A \wedge B) = 1 + max(deg(A), deg(B))$
$deg(\Box A) = 1 + deg(A)$

$mdeg(A) = 0$ when A is atomic
$mdeg(\neg A) = mdeg(A)$
$mdeg(A \wedge B) = max(mdeg(A), mdeg(B))$
$mdeg(\Box A) = 1 + mdeg(A)$

For a finite set X:

$deg(X) = \max\{deg(A) \mid A \in X\}$
$mdeg(X) = \max\{mdeg(A) \mid A \in X\}$

The set of all **subformulae** of a formula, or of a set of formulae, is used extensively. For a formula A, the *finite* set of all subformulae $Sf(A)$ is defined inductively as [Gol87]:

$Sf(p) = \{p\}$ where p is an atomic formula	$Sf(\neg A) = \{\neg A\} \cup Sf(A)$
$Sf(A \wedge B) = \{A \wedge B\} \cup Sf(A) \cup Sf(B)$	$Sf(\Box A) = \{\Box A\} \cup Sf(A)$

Note that under this definition, a formula must be in primitive notation to obtain its subformulae; for example:

$$\begin{array}{lllll} Sf(p \vee q) & = & Sf(\neg(\neg p \wedge \neg q)) & = & \{\neg(\neg p \wedge \neg q), \neg p \wedge \neg q, \neg p, \neg q, p, q\} \\ Sf(\Diamond p) & = & Sf(\neg\Box\neg p) & = & \{\neg\Box\neg p, \Box\neg p, \neg p, p\} \end{array}$$

For a finite set of formulae X, the set of all subformulae $Sf(\mathrm{X})$ consists of all subformulae of all members of X; that is, $Sf(X) = \bigcup_{A \in X} Sf(A)$. The set of **strict subformulae** of A is $Sf(A) \setminus \{A\}$.

A tableau rule has the **subformula property** iff every formula in the denominators is a subformula of some formula in the numerator. A tableau system $\mathcal{C}\mathbf{L}$ has the subformula property iff every rule in $\mathcal{C}\mathbf{L}$ has it.

If $\mathcal{C}\mathbf{L}$ has the subformula property then each rule can be seen to "break down" its principal formula(e) into its subformulae. Furthermore, if the principal formula is not copied into the numerator, then termination is guaranteed without cycles since every rule application is guaranteed to give a denominator of lower degree, eventually leading to a node with degree zero.

Notice that the rules of $\mathcal{C}\mathbf{K}$ do not have the subformula property, for both the $(\vee)$ and (K) rule denominators contain formulae which are *negations* of a subformula of the principal formula. But clearly this is not a disaster since the degree is not actually *increased*, but may remain the same.

The modal tableau rules for more complex logics, however, introduce quite complex "superformulae" into their denominators, thereby *increasing* both the degree and the modal degree. Nevertheless, all is not lost, for every tableau will be guaranteed to terminate, possibly with a cycle.

In order to prove this claim we need to introduce the idea of an **analytic superformula**. The intuition is simple: rules will be allowed to "build up" formulae so long as the rules cannot conspire to give an infinite chain of "building up" operations.

A tableau system $\mathcal{C}\mathbf{L}$ has the **analytical superformula** property iff to every finite set X we can assign, *a priori*, a *finite* set $X^*_{\mathcal{C}\mathbf{L}}$ such that $X^*_{\mathcal{C}\mathbf{L}}$ contains all formulae that may appear in *any* tableau for X.

Lemma 4.7.1. *If $\mathcal{C}\mathbf{L}$ has the analytic superformula property then there are (only) a finite number of $\mathcal{C}\mathbf{L}$-tableaux for the finite set X.*

Proof: Since $\mathcal{C}\mathbf{L}$ has the analytical superformula property the only $\mathcal{C}\mathbf{L}$-tableaux we need consider are those whose nodes carry subsets of the set $X^*_{\mathcal{C}\mathbf{L}}$. Since $X^*_{\mathcal{C}\mathbf{L}}$ is finite there are (only) a finite number of subsets of $X^*_{\mathcal{C}\mathbf{L}}$. ■

For example, the calculus $\mathcal{C}\mathbf{K}$ has the analytic superformula property because for any given finite X we can put $X^*_{\mathcal{C}\mathbf{K}} = Sf(X) \cup \neg Sf(X) \cup \{\bot\}$.

4.8 Proving Soundness

By definition, a tableau system $\mathcal{C}\mathbf{L}$ is sound with respect to $\mathbf{L}$-frames if $Y \vdash_{\mathcal{C}\mathbf{L}} A$ implies $Y \models_{\mathbf{L}} A$.

Proof Outline: To prove this claim we assume that $Y \vdash_{\mathcal{C}\mathbf{L}} A$; that is, that we have a closed $\mathcal{C}\mathbf{L}$-tableau for $X = (Y; \neg A)$. Then we use induction on the structure of this tableau to show that X is $\mathbf{L}$-unsatisfiable; that is, that $Y \models_{\mathbf{L}} A$.

The base case is when the tableau consists of just one application of the $(\bot)$ rule. In this case, the set X must contain some P and also $\neg P$ and is clearly $\mathbf{L}$-unsatisfiable (since our valuations are always classical two-valued ones).

Now suppose that the (closed) $\mathcal{C}\mathbf{L}$-tableau is some finite but arbitrary tree. We know that all leaves of this (closed) tableau end in $\{\bot\}$. So all we have to show is that for each $\mathcal{C}\mathbf{L}$-tableau rule: if all the denominators are $\mathbf{L}$-unsatisfiable then the numerator is $\mathbf{L}$-unsatisfiable. This would allow us to lift the $\mathbf{L}$-unsatisfiability of the leaves up the tree to conclude that the root X is $\mathbf{L}$-unsatisfiable. Instead, we show the contrapositive; that is, for each $\mathcal{C}\mathbf{L}$-tableau rule we show that if the numerator is $\mathbf{L}$-satisfiable then at least one of the denominators is $\mathbf{L}$-satisfiable. ■

Thus proving the soundness of a tableau system is possible on a rule by rule basis. For example, the $(\wedge)$ rule is sound with respect to $\mathbf{K}$-models because if we are given some $\mathbf{K}$-model $\langle W, R, V\rangle$ with some $w \in W$ such that $w \models X; p \wedge q$, then we can always find a $\mathbf{K}$-model $\langle W', R', V'\rangle$ with some $w' \in W'$ such that $w' \models X; p; q$ by simply putting $\langle W, R, V\rangle = \langle W', R', V'\rangle$ and putting $w = w'$.

As another example the (K) rule is sound with respect to **K**-models because if we are given some **K**-model $\langle W, R, V\rangle$ with some $w \in W$ such that $w \models \Box X; \neg\Box P$ then we know that w has some successor $w' \in W$ such that wRw' and $w' \models \neg P$ (by the definition of $w \models \neg\Box P$). Furthermore, since $w \models \Box X$ and wRw' we know that $w' \models X$ (by the definition of $w \models \Box P$). Thus we can find a $w' \in W$ such that $w' \models X; \neg P$. In this case, although the underlying model has remained the same, the world w' may be different from w.

The previous two examples show that, in general, the numerator and denominators of a tableau rule either represent the same world in the same model as in the $(\wedge)$ example, or they represent different worlds in the same model as in the (K) example. We therefore categorise each rule as either a **static rule** or as a **transitional rule**.

The intuition behind this sorting is that in the static rules, the numerator and denominator represent the same world (in the same model), whereas in the transitional rules, the numerator and denominator represent different worlds (in the same model).

For example, the tableau rules for $\mathcal{C}\mathbf{K}$ are categorised as follows:

$\mathcal{C}\mathbf{L}$	Static Rules	Transitional Rules
$\mathcal{C}\mathbf{K}$	(θ), $(\bot), (\neg), (\wedge), (\vee)$	(K)

4.9 Proving Completeness Via Model-Graphs

By definition, $\mathcal{C}\mathbf{L}$ is complete with respect to **L**-frames iff: $Y \models_{\mathbf{L}} A$ implies $Y \vdash_{\mathcal{C}\mathbf{L}} A$.

Proof Outline: We prove the contrapositive. That is, we assume $Y \not\vdash_{\mathcal{C}\mathbf{L}} A$ by assuming that *no* $\mathcal{C}\mathbf{L}$-tableau for $X = (Y; \neg A)$ is closed. Then we pick and choose sets with certain special properties from *possibly different* open tableaux for X, and use them as possible worlds to construct an **L**-model $\mathcal{M}$ for X, safe in the knowledge that each of these sets is $\mathcal{C}\mathbf{L}$-consistent. The model $\mathcal{M}$ is deliberately constructed so as to contain a world w_0 such that $w_0 \models Y$ and $w_0 \models \neg A$, Hence we demonstrate by construction that $Y \not\models A$. The basic idea is due to Hintikka [Hin55]. ∎

In order to do so we first need some technical machinery.

4.9.1 Downward Saturated Sets

A set X is **closed with respect to a tableau rule** if, whenever (an instantiation of) the numerator of the rule is in X, so is (a corresponding instantiation of) at least one of the denominators of the rule. A set X is $\mathcal{C}\mathbf{L}$**-saturated** if it is $\mathcal{C}\mathbf{L}$-consistent and closed with respect to the static rules of $\mathcal{C}\mathbf{L}$.

Lemma 4.9.1. *For each $\mathcal{C}\mathbf{L}$ with the analytic superformula property and each finite $\mathcal{C}\mathbf{L}$-consistent X there is an effective procedure to construct some finite $\mathcal{C}\mathbf{L}$-saturated (and $\mathcal{C}\mathbf{L}$-consistent) X^s with $X \subseteq X^s \subseteq X^*_{\mathcal{C}\mathbf{L}}$.*

Proof: Since X is $\mathcal{C}\mathbf{L}$-consistent, we know that no $\mathcal{C}\mathbf{L}$-tableau for X closes and hence that the $(\perp)$ rule is not applicable. In particular, we know that a tableau where (θ) is never used, and where the principal formula is always copied into the denominator will also not close.

So put $X_0 = X$ and repeatedly apply the static rules except (θ) in this fashion to give a sequence of $\mathcal{C}\mathbf{L}$-consistent sets $X_0, X_1, X_2, \cdots$ with the proviso that if applying some (static) rule to X_i gives a previous member of the sequence then we backtrack and choose a different (static) rule to apply to X_i. For a static rule with more than one denominator we choose some denominator that is $\mathcal{C}\mathbf{L}$-consistent as the next set in the sequence. This procedure will terminate with some final X_n either because all (static) rule applications lead to a cycle or because no (static) rule is applicable to X_n.

Put $X^s = X_n$ ensuring that X^s is $\mathcal{C}\mathbf{L}$-consistent. Because we always copy the principal formula into the denominator, we have $X_0 \subseteq X_1 \subseteq X_2 \subseteq \cdots \subseteq X_n$ hence $X \subseteq X^s$. Since each rule carries subsets of $X^*_{\mathcal{C}\mathbf{L}}$ to subsets of $X^*_{\mathcal{C}\mathbf{L}}$ and we start with $X \subseteq X^*_{\mathcal{C}\mathbf{L}}$ we have $X \subseteq X^s \subseteq X^*_{\mathcal{C}\mathbf{L}}$. ∎

In classic logic, such sets are called downward saturated sets and form the basis of Hintikka's [Hin55] method for proving completeness of classical tableau systems. In the next section we introduce the necessary technical machinery to extend this method to modal logics.

4.9.2 Model Graphs and Satisfiability Lemma

The following definition from Rautenberg [Rau83] is central for the model constructions. A **model graph** for some finite fixed set of formulae X is a finite $\mathbf{L}$-frame $\langle W, R\rangle$ such that all $w \in W$ are $\mathcal{C}\mathbf{L}$-saturated sets with $w \subseteq X^*_{\mathcal{C}\mathbf{L}}$ and

(i) $X \subseteq w_0$ for some $w_0 \in W$;

(ii) if $\neg\Box P \in w$ then there exists some $w' \in W$ with wRw' and $\neg P \in w'$;

(iii) if wRw' and $\Box P \in w$ then $P \in w'$.

Lemma 4.9.2. *If $\langle W, R\rangle$ is a model graph for X then there exists an* $\mathbf{L}$-*model for X [Rau83].*

Proof: For every $p \in \mathcal{P}$, let $\vartheta(p) = \{w \in W : p \in w\}$. Using simultaneous induction on the degree of an arbitrary formula $A \in w$, it is easy to show that (a) $A \in w$ implies $w \models A$; and (b) $\neg A \in w$ implies $w \not\models A$. By (a), $w_0 \models X$ hence the model $\mathcal{M} = \langle W, R, \vartheta\rangle$ is an $\mathbf{L}$-model for X [Rau83]. ∎

This model graph construction is similar to the subordinate frames construction of Hughes and Cresswell [HC84] except that Hughes and Cresswell use maximal consistent sets and do not consider cycles, giving infinite models rather than finite models.

4.10 Finite Model Property and Decidability

In the above procedure, if $\mathcal{M}$ can be chosen finite (for finite X) then the logic $\mathbf{L}$ has the finite model property (fmp). It is known that a finitely axiomatisable normal modal logic with the finite model property must be decidable; see Hughes and Cresswell [HC84, page 154]. Hence $\mathcal{C}\mathbf{L}$ provides a decision procedure for determining whether $Y \models_{\mathbf{L}} A$.

4.11 Summary

In the rest of this section we present tableau systems for many propositional normal modal logics based on the work of Rautenberg [Rau83], Fitting [Fit83], Shvarts [Shv89], Hanson [Han66], Goré [Gor92, Gor91, Gor94] and Amerbauer [Ame93]. Most of the systems are cut-free but even those that are not use only an analytical cut rule. Each tableau system immediately gives an analogous (cut-free) sequent system. The presentation is based on the basis laid down in the previous subsections and is therefore rather repetitive. The procedure for each tableau system $\mathcal{C}\mathbf{L}$ is:

1. define the tableau rules for $\mathcal{C}\mathbf{L}$;
2. define $X^*_{\mathcal{C}\mathbf{L}}$ for a given fixed X;
3. prove that the $\mathcal{C}\mathbf{L}$ rules are sound with respect to $\mathbf{L}$-frames;
4. prove that each $\mathcal{C}\mathbf{L}$-consistent X can be extended (effectively) to a $\mathcal{C}\mathbf{L}$-saturated X^s with $X \subseteq X^s \subseteq X^*_{\mathcal{C}\mathbf{L}}$;
5. prove that the $\mathcal{C}\mathbf{L}$ rules are complete with respect to $\mathbf{L}$-frames by giving a procedure to construct a *finite* $\mathbf{L}$-model for any finite $\mathcal{C}\mathbf{L}$-consistent X and hence prove that $\mathbf{L}$ has the finite model property, that $\mathbf{L}$ is decidable and that $\mathcal{C}\mathbf{L}$ is a decision procedure for deciding local logical consequence ($Y \models_{\mathbf{L}} A$) in $\mathbf{L}$.

4.12 The Basic Normal Systems

In this section we study the tableau systems which capture the basic normal modal logics obtained from various combinations of the five basic axioms of reflexivity, transitivity, seriality, euclideaness, and symmetry. We shall see that implicit tableau systems can handle certain combinations of the first four properties with ease, but require an analytic cut rule to handle symmetry. In each case, we give the tableau calculi and prove them sound and complete with respect to the appropriate semantics. We shall also see that some of the basic logics have no known implicit tableau systems, leaving an avenue for further work.

The following notational conventions are useful for defining $X^*_{\mathcal{C}\mathbf{L}}$ for each X and each $\mathcal{C}\mathbf{L}$. For any finite set X :

- let $Sf(X)$ denote the set of all subformulae of all formulae in X ;
- let $\neg Sf(X)$ denote $\{\neg P \mid P \in Sf(X)\}$;
- let $\widetilde{X}$ denote the set $Sf(X) \cup \neg Sf(X) \cup \{\bot\}$;
- let $\Box(\widetilde{X} \to \Box\widetilde{X})$ denote the set $\{\Box(P \to \Box P) \mid P \in \widetilde{X}\}$.

We sometimes write SfX instead of $Sf(X)$ whence $\widetilde{X} = (Sf\neg SfX) \cup \{\bot\}$.

4.12.1 Tableau Calculi

All the tableau calculi contain the rules of $\mathcal{C}\mathbf{PC}$ and one or more logical rules from Figure 6 on page 27. The tableau systems are shown in Figure 7 on page 28 and the only structural rule is (θ). The calculi marked with a superscript † require analytic cut whilst the others are cut-free. The entries marked by question-marks are open questions.

The rules are categorised into two sorts, static rules and transitional rules as explained on page 23. This sorting should become even clearer once we prove soundness.

The semantic and sometimes axiomatic intuitions behind these rules are as follows.

Intuitions for (K) **:** if the numerator represents a world w where $\Box X$ and $\neg\Box P$ are true, then since $\neg\Box P = \Diamond\neg P$, there must be a world w' reachable from w such that w' makes P false and makes all the formulae in X true. The denominator of the (K) rule represents w'.

Intuitions for (T) **:** if the numerator represents a world w where X and $\Box P$ are true, then every successor of w must make P true. By reflexivity of R the world w itself must be one of these successors.

Intuitions for (D) **:** if the numerator represents a world w where X and $\Box P$ are true, then by seriality of R there must exist some w' such that wRw'. Then the definition of $\Box P$ forces P to be true at w'. Hence $\neg\Box\neg P$, that is $\Diamond P$, must be true at w *itself*. Note that (D) is a static rule since its numerator and denominator represent the same world, and also that (D) creates a superformula $\neg\Box\neg P$.

Intuitions for (KD) **:** if the numerator represents a world w where $\Box X$ is true, then the seriality of R guarantees a successor w' for this world, and the definition of $\Box X$ forces X to be true at w'. So we can apply the (KD) rule even when the numerator contains no formulae of the form $\neg\Box P$. Of course, if such a formulae is present then the intuitions for the (K) rule suffice. Note that (KD) is a transitional rule since the numerator and denominator represent different worlds, and also that it has the subformula property.

Intuitions for $(K4)$ **:** if the numerator represents a world w where $\Box X$ and $\neg\Box P$ are true, there must be a world w' representing the denominator, with wRw', such that w' makes X true and makes P false. Then by

$$(K)\ \frac{\Box X; \neg\Box P}{X; \neg P} \qquad (T)\ \frac{X; \Box P}{X; \Box P; P} \qquad (D)\ \frac{X; \Box P}{X; \Box P; \neg\Box\neg P}$$

$$(KD)\ \frac{\Box X; \neg\Box P}{X; \neg P} \text{ where } \{\neg\Box P, \neg P\} \text{ may be empty}$$

$$(K4)\ \frac{\Box X; \neg\Box P}{X; \Box X; \neg P} \qquad (S4)\ \frac{\Box X; \neg\Box P}{\Box X; \neg P}$$

$$(45)\ \frac{\Box X; \neg\Box Y; \neg\Box P}{X; \Box X; \neg\Box Y; \neg\Box P; \neg P}$$

$$(45D)\ \frac{\Box X; \neg\Box Y; \neg\Box P}{X; \Box X; \neg\Box Y; \neg\Box P; \neg P} \text{ where } Y \cup \{P\} \cup \{\neg P\} \text{ may be empty}$$

$$(B)\ \frac{X; \neg\Box P}{X; \neg\Box P; P \mid X; \neg\Box P; \neg P; \Box\neg\Box P} \qquad (T_{\Diamond})\ \frac{X; \Box P; \neg\Box Q}{X; \Box P; P; \neg\Box Q}$$

$$(5)\ \frac{X; \neg\Box P}{X; \neg\Box P; \Box\neg\Box P} \qquad (S5)\ \frac{\Box X; \neg\Box Y; \neg\Box P}{\Box X; \neg\Box Y; \neg\Box P; \neg P}$$

$$(sfc\Box)\frac{X; \Box P}{X; \Box P; P \mid X; \Box P; \neg P}$$

$$(sfc\Diamond)\frac{X; \neg\Box P}{X; \neg\Box P; P \mid X; \neg\Box P; \neg P}$$

$$(sfc\vee)\frac{X; \neg(P \wedge Q)}{X; \neg P; \neg Q \mid X; \neg P; Q \mid X; P; \neg Q}$$

$$(sfc) = \{(sfc\Box), (sfc\Diamond), (sfc\vee)\} \qquad (sfcT) = \{(sfc\vee), (sfc\Diamond)\}$$

Fig. 6. Tableau Rules for Basic Systems

$\mathcal{C}\mathbf{L}$	Static Rules	Transitional Rules	$X^*_{\mathcal{C}\mathbf{L}}$
$\mathcal{C}\mathbf{PC}$	(θ), $(\bot), (\neg), (\wedge), (\vee)$	—	$\widetilde{X}$
$\mathcal{C}\mathbf{K}$	$\mathcal{C}\mathbf{PC}$	(K)	$\widetilde{X}$
$\mathcal{C}\mathbf{T}$	$\mathcal{C}\mathbf{PC}$, (T)	(K)	$\widetilde{X}$
$\mathcal{C}\mathbf{D}$	$\mathcal{C}\mathbf{PC}$, (D)	(K)	$Sf\neg Sf\Box\widetilde{X}$
$\mathcal{C}\mathbf{D}'$	$\mathcal{C}\mathbf{PC}$	(KD)	$\widetilde{X}$
$\mathcal{C}\mathbf{KB}$	?	?	?
$\mathcal{C}\mathbf{K4}$	$\mathcal{C}\mathbf{PC}$	$(K4)$	$\widetilde{X}$
$\mathcal{C}\mathbf{K5}$	?	?	?
$\mathcal{C}\mathbf{KDB}$	?	?	?
$\mathcal{C}\mathbf{KD5}$	?	?	?
$\mathcal{C}\mathbf{K4D}$	$\mathcal{C}\mathbf{PC}$, (D)	$(K4)$	$Sf\neg Sf\Box\widetilde{X}$
$\mathcal{C}\mathbf{K45}$	$\mathcal{C}\mathbf{PC}$	(45)	$\widetilde{X}$
$\mathcal{C}\mathbf{K45D}$	$\mathcal{C}\mathbf{PC}$	$(45D)$	$\widetilde{X}$
$\mathcal{C}\mathbf{S4}$	$\mathcal{C}\mathbf{PC}$, (T)	$(S4)$	$\widetilde{X}$
$\mathcal{C}\mathbf{S5}\pi^-$	$\mathcal{C}\mathbf{PC}$, (T)	$(S5)$	$\widetilde{X}$
$\mathcal{C}^\dagger\mathbf{K45}$	$\mathcal{C}\mathbf{PC}$, (sfc)	(45)	$\widetilde{X}$
$\mathcal{C}^\dagger\mathbf{K45D}$	$\mathcal{C}\mathbf{PC}$, (sfc)	$(45D)$	$\widetilde{X}$
$\mathcal{C}^\dagger\mathbf{K4B}$	$\mathcal{C}\mathbf{PC}$, (sfc), $(T_\Diamond)$, (5)	$(K4)$	$Sf\neg Sf\Box\widetilde{X}$
$\mathcal{C}^\dagger\mathbf{S4}$	$\mathcal{C}\mathbf{PC}$, (T), $(sfcT)$	$(S4)$	$\widetilde{X}$
$\mathcal{C}^\dagger\mathbf{B}$	$\mathcal{C}\mathbf{PC}$, $(T), (B), (sfcT)$	(K)	$Sf\neg Sf\Box\widetilde{X}$
$\mathcal{C}^\dagger\mathbf{S5}$	$\mathcal{C}\mathbf{PC}$, $(T), (5), (sfcT)$	$(S4)$	$Sf\neg Sf\Box\widetilde{X}$
$\mathcal{C}^\dagger\mathbf{S5}'$	$\mathcal{C}\mathbf{PC}$, $(T), (sfcT)$	$(S5)$	$\widetilde{X}$

Fig. 7. Tableau Calculi for Basic Systems

transitivity of R, any and all successors of w' must also make X true, hence w' makes $\Box X$ true. If w' does not have successors then it makes $\Box X$ true vacuously.

Intuitions for $(S4)$ **:** if the numerator represents a world w where $\Box X$ and $\neg\Box P$ are true, then by transitivity of R there must be a world w' representing the denominator, with wRw', such that w' makes $\Box X$ true and makes P false.

Intuitions for (B) **:** if R is symmetric and reflexive and the numerator represents a world w where X and $\neg\Box P$ are true, we know that this world either makes P true or makes P false. If w makes P true then we have the left denominator. If w makes P false, then we have the right denominator which also contains $\Box\neg\Box P$ since $A \to \Box\Diamond A$ is a theorem of $\mathbf{B}$.

Intuitions for (5) **:** Suppose R is euclidean and the numerator represents a world w where X and $\neg\Box P$ are true. Then we immediately have that w also makes $\Box\neg\Box P$ true since $\neg\Box A \rightarrow \Box\neg\Box A$ is just another way of writing the axiom 5 which we know must be valid in all euclidean Kripke frames.

Intuitions for (sfc) **:** if the numerator represents a world w where $\neg(P \wedge Q)$ is true, then we know that w either makes both P and Q false; or makes P false and Q true; or makes P true and Q false. The other cases use similar intuitions.

Intuitions for $(sfcT)$ **:** as for the (sfc) rule except that by reflexivity we cannot have both $\Box P$ and $\neg P$ true at w so one of the cases cannot occur.

Example 4.12.1. The following example shows that copying the principal formula into the denominator is crucial, since the first $\mathcal{C}$**KT**-tableau (using a non-copying application of a rule (T') does not close but the second one (using (T) does.

$$
\begin{array}{c}
\Box p; \neg(p \wedge \neg\Box q); \neg\Box(p \wedge q) \quad (T') \\
\hline
p; \neg(p \wedge \neg\Box q); \neg\Box(p \wedge q) \quad (\vee) \\
\hline
p; \neg p; \neg\Box(p \wedge q)\ (\bot) \mid p; \neg\neg\Box q; \neg\Box(p \wedge q)\ (\neg) \\
\hline
\mid p; \Box q; \neg\Box(p \wedge q)\ (\theta) \\
\hline
\mid \Box q; \neg\Box(p \wedge q)\ (K) \\
\hline
\mid q; \neg(p \wedge q)\ (\vee) \\
\hline
\mid q; \neg p\ (open) \mid q; \neg q\ (\bot)
\end{array}
$$

$$
\begin{array}{c}
\Box p; \neg(p \wedge \neg\Box q); \neg\Box(p \wedge q) \quad (T) \\
\hline
p; \Box p; \neg(p \wedge \neg\Box q); \neg\Box(p \wedge q) \quad (\vee) \\
\hline
\begin{array}{c|c}
p; \Box p; \neg p; \neg\Box(p \wedge q)\ (\bot) & p; \Box p; \neg\neg\Box q; \neg\Box(p \wedge q)\ (\neg) \\
\hline
 & p; \Box p; \Box q; \neg\Box(p \wedge q)\ (\theta) \\
\hline
 & \Box p; \Box q; \neg\Box(p \wedge q)\ (K) \\
\hline
 & p; q; \neg(p \wedge q)\ (\vee) \\
\hline
 & p; q; \neg p\ (\bot) \mid p; q; \neg q\ (\bot)
\end{array}
\end{array}
$$

Example 4.12.2. The following example shows that the order of the modal rule applications is important, since the $\mathcal{C}$**KT**-tableau below does not close precisely because (K) is applied at the start:

$$
\cfrac{\Box p; \neg(p \wedge \neg\Box q); \neg\Box(p \wedge q)}{\cfrac{\Box p; \neg\Box(p \wedge q)}{\cfrac{p; \neg(p \wedge q)}{p; \neg p\ (\bot) \mid p; \neg q \quad (\text{open})}\,(\vee)}\,(K)}\,(\theta)
$$

4.12.2 Soundness

Theorem 4.12.3 (Soundness). *Each calculus $\mathcal{C}$**L** and $\mathcal{C}^\dagger$**L** listed in Figure 7 on page 28 (without question marks!) is sound with respect to **L**-frames.*

Proof Outline : For each rule in $\mathcal{C}$**L** we have to show that if the numerator of the rule is **L**-satisfiable then so is at least one of the denominators. The $\mathcal{C}$**PC** rules are obviously sound since each world behaves classically. The rules (sfc) and $(sfcT)$ are also sound for Kripke frames because any particular world in any model either satisfies P or satisfies $\neg P$ for any formula P. For each modal rule we prove that it is sound with respect to some known property of R as enforced by the **L**-frames restrictions. The proofs are fairly straightforward and intuitive so we give a sketch only.

We often use annotated names like w_1 and w' to denote possible worlds. Unless stated explicitly, there is no reason why w_1 and w' cannot name the same world.

Proof for (K)**:** We show that (K) is sound with respect to all Kripke frames. Suppose $\mathcal{M} = \langle W, R, V\rangle$ is any Kripke model, $w_0 \in W$ and that w_0 satisfies the numerator of (K). That is, suppose $w_0 \models \Box X; \neg\Box P$. We have to show that there exists some world that satisfies the denominator of (K). By definition of the satisfaction relation, $w_0 \models \neg\Box P$ implies that there exists a $w_1 \in W$ with $w_0 R w_1$ and $w_1 \models \neg P$. Since $w_0 \models \Box X$ and $w_0 R w_1$, the definition of $\models$ implies that $w_1 \models X$, hence $w_1 \models (X; \neg P)$, which is what we had to show.

Proof for (T)**:** We show that (T) is sound with respect to all reflexive Kripke frames. Suppose $\mathcal{M} = \langle W, R, V \rangle$ is any Kripke model where R is reflexive, $w_0 \in W$ and $w_0 \models \Box X; \Box P$. Then the reflexivity of R and the definition of $\models$ implies that $w_0 \models \Box X; \Box P; P$.

Proof for (D) **:** We show that (D) is sound with respect to all serial Kripke frames. So suppose $\mathcal{M} = \langle W, R, V \rangle$ is any Kripke model where R is serial. That is, $\forall w \in W, \exists w' \in W : wRw'$. Suppose $w_0 \in W$ and $w_0 \models X; \Box P$. By seriality there exists some $w_1 \in W$ with $w_0 R w_1$. And since $w_0 \models \Box P$ we must have $w_1 \models P$. But then there is a world (namely w_1) accessible from w_0 that satisfies P, and hence $w_0 \models \Diamond P$. By definition, $\Diamond P = \neg\Box\neg P$, hence $w_0 \models \neg\Box\neg P$, thus satisfying the denominator of (D).

Proof for (KD) **:** We show that (KD) is sound with respect to all serial Kripke frames. So suppose $\mathcal{M} = \langle W, R, V \rangle$ is any Kripke model where R is serial. Suppose $w_0 \in W$ and $w_0 \models \Box X$. By seriality there exists some $w_1 \in W$ with $w_0 R w_1$, and since $w_0 \models \Box X$ we must have $w_1 \models X$ thus satisfying the denominator of (KD) when the $\neg\Box P$ part is missing from the numerator. On the other hand, if $w_0 \models \Box X; \neg\Box P$ for some P then, by definition, there is a world w_2 accessible from w_0 with $w_2 \models X; \neg P$.

Proof for $(K4)$**:** We show that $(K4)$ is sound with respect to all transitive Kripke frames. So suppose $\mathcal{M} = \langle W, R, V \rangle$ is any Kripke model where R is transitive. Suppose $w_0 \in W$ and $w_0 \models \Box X; \neg\Box P$. Thus there exists $w_1 \in W$ with $w_0 R w_1$ and $w_1 \models X; \neg P$. Since R is transitive, all successors of w_1 are reachable from w_0, hence $w_0 \models \Box X$ implies that every successor of w_1, if there are any, must also satisfy X. By the definition of $\models$ this gives $w_1 \models X; \Box X; \neg P$. If w_1 has no successors then it vacuously satisfies $\Box A$ for any formula A, hence it vacuously satisfies $\Box X$, and we are done.

Proof for $(S4)$**:** The proof for $(K4)$ also shows that $(S4)$ is sound with respect to all transitive Kripke models.

Proof for (45)**:** Let $\mathcal{M} = \langle W, R, V \rangle$ be any Kripke model where R is transitive and euclidean. Suppose that $w_0 \in W$ and $w_0 \models \Box X; \neg\Box Y; \neg\Box P$. We have to show that there exists a $w' \in W$ such that $w' \models X; \Box X; \neg\Box Y; \neg\Box P; \neg P$.

We need only prove that there exists a $w' \in W$ such that $w' \models \neg\Box Y; \neg\Box P; \neg P$ since the $X; \Box X$ part will follow from the transitivity of R. Since $w_0 \models \neg\Box P$ we know that there exists some w' with $w_0 R w'$ and $w' \models \neg P$. By the definition of euclideaness $w_0 R w'$ and $w_0 R w'$ (sic) implies $w' R w'$. Hence w' is reflexive and we have $w' \models \neg\Box P$. Now, if Y is empty then we are done; otherwise if $Y = \{Q_1, Q_2, \cdots, Q_n\}$, $n \geq 1$, there will be worlds w_1, w_2, $\ldots$,w_n (not necessarily distinct) where $w_0 R w_i$ for each

$1 \leq i \leq n$ and such that $w_i \models \neg Q_i$. Since R is euclidean, $w_0 R w'$ and $w_0 R w_i$ implies that $w' R w_i$ for each $1 \leq i \leq n$. But then $w' \models \neg\Box Y$ and we are done.

Proof for $(45D)$**:** Let $\mathcal{M} = \langle W, R, V\rangle$ be any Kripke model where R is serial, transitive and euclidean, and suppose that $w_0 \in W$ and $w_0 \models \Box X; \neg\Box Y; \neg\Box P$. We have to show that there exists a $w' \in W$ such that $w' \models X; \Box X; \neg\Box Y; \neg\Box P; \neg P$ allowing for the case where the $\neg\Box Y; \neg\Box P$ part is missing. Since R is transitive and euclidean the proof for $\mathcal{C}$**K45** applies when the $\neg\Box Y; \neg\Box P$ part is present. If there are no formulae of the form $\neg\Box P$ in w_0 then seriality guarantees that there is some world w' with wRw', and then transitivity of R ensures that $w' \models X; \Box X$.

Proof for (B)**:** We show that (B) is sound with respect to all symmetric Kripke frames. Suppose $\mathcal{M} = \langle W, R, V\rangle$ is any Kripke model where R is symmetric, $w_0 \in W$ and $w_0 \models X; \neg\Box P$. We show that $w_0 \models P$ or $w_0 \models \neg P; \Box\neg\Box P$. If $w_0 \models P$ then $w_0 \models X; \neg\Box P; P$ and we are done. Otherwise $w_0 \models \neg P$. In this latter case, suppose $w_0 \not\models \Box\neg\Box P$. Then $w_0 \models \neg\Box\neg\Box P$, that is $w_0 \models \Diamond\Box P$, so there exists some $w_1 \in W$ with $w_0 R w_1$ and $w_1 \models \Box P$. Since R is symmetric, $w_0 R w_1$ implies $w_1 R w_0$ which together with $w_1 \models \Box P$ gives $w_0 \models P$. But this contradicts the supposition that $w_0 \models \neg P$. Hence $w_0 \models X; \neg\Box P; P$ or $w_0 \models X; \neg\Box P; \neg P; \Box\neg\Box P$ and we are done.

Proof for $(T_\Diamond)$**:** We show that $(T_\Diamond)$ is sound with respect to all Kripke frames that are symmetric and transitive. Suppose $\mathcal{M} = \langle W, R, V\rangle$ is any Kripke model where R is symmetric and transitive, $w_0 \in W$ and $w_0 \models X; \Box P; \neg\Box Q$. Then there exists some $w_1 \in W$ with $w_0 R w_1$ and $w_1 \models \neg Q$. By symmetry, $w_0 R w_1$ implies $w_1 R w_0$. By transitivity, $w_0 R w_1$ and $w_1 R w_0$ implies $w_0 R w_0$. Therefore $w_0 \models P$ and we are done.

Proof for (5)**:** We show that (5) is sound with respect to all euclidean Kripke frames. Suppose $\mathcal{M} = \langle W, R, V\rangle$ is any Kripke model where R is euclidean, and suppose $w_0 \in W$ with $w_0 \models X; \neg\Box P$. We have to show that $w_0 \models \Box\neg\Box P$. Assume for a contradiction that $w_0 \not\models \Box\neg\Box P$; that is, $w_0 \models \neg\Box\neg\Box P$, which is the same as $w_0 \models \Diamond\Box P$. Thus there exists some $w_1 \in W$ with $w_0 R w_1$ and $w_1 \models \Box P$. Since $w_0 \models \neg\Box P$ there is also some w_2 with $w_0 R w_2$ and $w_2 \models \neg P$. Since R is euclidean, $w_0 R w_1$ and $w_0 R w_2$ implies $w_1 R w_2$. And since $w_1 \models \Box P$ we must have $w_2 \models P$. Contradiction; hence $w_0 \models \Box\neg\Box P$ as desired.

Proof for $(S5)$**:** We show that $(S5)$ is sound with respect to all Kripke frames that are transitive and euclidean. Suppose $\mathcal{M} = \langle W, R, V\rangle$ is any Kripke model where R is transitive and euclidean. Suppose $w_0 \in W$ and $w_0 \models \Box X; \neg\Box Y; \neg\Box P$. Thus there exists some world $w' \in W$ with $w_0 R w'$ and $w' \models \neg P$. Suppose $Y = \{Q_1, Q_2, \cdots, Q_n\}$, $n \geq 1$. Thus there exist (not necessarily distinct) worlds $w_1, w_2, \cdots, w_n$ such that $w_0 R w_i$ and $w_i \models \neg Q_i$, for $1 \leq i \leq n$. Since R is euclidean, $w' R w'$ and $w' R w_i$ for each i. The first gives $w' \models \neg\Box P$, and the second gives $w' \models \neg\Box Q_i$, $1 \leq i \leq n$. Hence

$w' \models \neg P; \neg\Box P; \neg\Box Y$. If Y is empty then we just get $w' \models \neg P; \neg\Box P$. Now choose any arbitrary world w'' such that $w'Rw''$ (there is at least one since w' is a reflexive world). By transitivity of R, w_0Rw'', hence $w'' \models X$. Since w'' was an arbitrary successor for w' this holds for all successors of w'. Hence $w' \models \Box X$ as well giving $w' \models \Box X; \neg P; \neg\Box P; \neg\Box Y$.

■

4.12.3 Completeness

As we saw in Subsection 4.9 (page 23) , proving completeness boils down to proving the following: if X is a finite set of formulae and no $\mathcal{C}\mathbf{L}$-tableau for X is closed then there is an $\mathbf{L}$-model for X on an $\mathbf{L}$-frame $\langle W, R\rangle$.

We call a formula $\neg\Box P$ an **eventuality** since it entails that eventually $\neg P$ must hold. A world w is said to **fulfill** an eventuality $\neg\Box P$ when $w \models \neg P$. A sequence of worlds $w_1Rw_2R\cdots Rw_m$ is said to fulfill an eventuality $\neg\Box P$ when $w_i \models \neg P$ for some w_i in the sequence.

As expected we shall associate sets of formulae with possible worlds and use an explicit immediate successor relation $\prec$ from which we will obtain R. We abuse notation slightly by using w, w' and w_1 to sometimes denote worlds in a model, and sometimes to denote sets of formulae (in a model under construction). Thus, a *set* w is said to fulfill an eventuality $\neg\Box P$ when $\neg P \in w$. A sequence $w_1 \prec w_2 \prec \cdots \prec w_m$ of sets is said to fulfill an eventuality $\neg\Box P$ when $\neg P \in w_i$ for some w_i in the sequence.

Recall that a set X is **$\mathcal{C}$L-saturated** iff it is $\mathcal{C}\mathbf{L}$-consistent and closed with respect to the static rules of $\mathcal{C}\mathbf{L}$ (excluding (θ)). We now have to check that the $X^*_{\mathcal{C}\mathbf{L}}$ defined in Figure 7 on page 28 allow (the Saturation) Lemma 4.9.1 (page 24) to go through.

Lemma 4.12.4. *If there is a closed $\mathcal{C}\mathbf{L}$-tableau for X then there is a closed $\mathcal{C}\mathbf{L}$-tableau for X with all nodes in the finite set $X^*_{\mathcal{C}\mathbf{L}}$.*

Proof: Obvious from the fact that all rules for $\mathcal{C}\mathbf{L}$ operate with subsets of $X^*_{\mathcal{C}\mathbf{L}}$ only. ■

Lemma 4.12.5. *For each $\mathcal{C}\mathbf{L}$-consistent X there is an effective procedure to construct some finite $\mathcal{C}\mathbf{L}$-saturated X^s with $X \subseteq X^s \subseteq X^*_{\mathcal{C}\mathbf{L}}$.*

Proof: Same as on page 24. ■

A set X is **subformula-complete** if $P \in Sf(X)$ implies either $P \in X$ or $\neg P \in X$. Some of the completeness proofs make extensive use of the following lemma.

Lemma 4.12.6 (sfc). *If X is closed with respect to (the static rules)* { $(\bot)$, $(\neg)$, $(\wedge)$, $(\vee)$, (sfc) }, *or* { $(\bot)$, $(\neg)$, $(\wedge)$, $(\vee)$, (sfc), $(T_\Diamond)$ } *or* { $(\bot)$, $(\neg)$, $(\wedge)$, $(\vee)$, (T), $(sfcT)$ } *then X is subformula-complete.*

Proof: The first case is obvious. The $(sfcT)$ rule is just a special case of (sfc) and always appears with (T). Thus, the lemma also holds if we have both $(sfcT)$ and (T) instead of (sfc). ■

Theorem 4.12.7 (Completeness). *If X is a finite set of formulae and X is $\mathcal{C}\mathbf{L}$-consistent then there is an $\mathbf{L}$-model for X on a finite $\mathbf{L}$-frame.*

Proof Outline: For each $\mathcal{C}\mathbf{L}$ we give a way to construct a finite model graph $\langle W_0, R\rangle$ for X. Recall that a model graph for some finite fixed set of formulae X is a finite $\mathbf{L}$-frame $\langle W_0, R\rangle$ such that all $w \in W_0$ are $\mathcal{C}\mathbf{L}$-saturated sets with $w \subseteq X^*_{\mathcal{C}\mathbf{L}}$ and

(i) $X \subseteq w_0$ for some $w_0 \in W_0$;
(ii) if $\neg\Box P \in w$ then there exists some $w' \in W_0$ with wRw' and $\neg P \in w'$;
(iii) if wRw' and $\Box P \in w$ then $P \in w'$.

The first step is to create a $\mathcal{C}\mathbf{L}$-saturated w_0 from X with $X \subseteq w_0 \subseteq X^*_{\mathcal{C}\mathbf{L}}$. This is possible via Lemma 4.9.1 (page 24). So w_0, and in general w (possibly annotated) stands for a finite $\mathcal{C}\mathbf{L}$-saturated set of formulae (that corresponds to a world of W_0). Since w_0 is $\mathcal{C}\mathbf{L}$-consistent, we know that *no* $\mathcal{C}\mathbf{L}$-tableau for w_0 closes. We use this fact to construct a graph of $\mathcal{C}\mathbf{L}$-saturated worlds, always bearing in mind that the resulting model graph must be based on an $\mathbf{L}$-frame. The construction is a meta-level one since we are free to inspect all $\mathcal{C}\mathbf{L}$-tableaux for w_0, choosing nodes at will, since all such $\mathcal{C}\mathbf{L}$-tableaux are open. We use a successor relation $\prec$ while building this graph and then form R from $\prec$. Also, if w is a set of formulae in this construction then $w^\Box = \{P : \Box P \in w\}$.

By Lemma 4.9.2 (page 24), $w_0 \models X$ under the truth valuation $\vartheta : p \mapsto \{w \in W_0 \ : \ p \in w\}$, giving an $\mathbf{L}$-model for X at w_0 as desired.

Proof for $\mathcal{C}\mathbf{K}$: If no $\neg\Box P$ occurs in w_0 then $\langle W, R\rangle = \langle\{w_0\}, \emptyset\rangle$ is the desired model graph since this is a $\mathbf{K}$-frame and (i)-(iii) are satisfied.

Otherwise, let $Q_1, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$. Since w_0 is $\mathcal{C}\mathbf{K}$-consistent, no application of (θ) can lead to a closed $\mathcal{C}\mathbf{K}$-tableau; in particular, the set $\{\Box A : \Box A \in w_0\} \cup \{\neg\Box Q_i\}$ must be $\mathcal{C}\mathbf{K}$-consistent for each $1 \leq i \leq m$. Each of these sets matches the numerator of (K) so (K) is applicable to each of them. But we know that an application of (K) to any of these sets could not have led to a closed $\mathcal{C}\mathbf{K}$-tableau either, so each of their respective denominators $(w_0^\Box; \neg Q_i)$ for $i = 1, \cdots, m$ must be $\mathcal{C}\mathbf{K}$-consistent (by (θ) and (K)). Note that these nodes come from different $\mathcal{C}\mathbf{K}$-tableaux.

Create a $\mathcal{C}\mathbf{K}$-saturated $v_i \subseteq X^*_{\mathcal{C}\mathbf{K}}$ from each $(w_0^\Box; \neg Q_i)$ for $i = 1, \cdots, m$, by using the static rules, and (the Saturation) Lemma 4.9.1. Put $w_0 \prec v_i$ for $i = 1, \cdots, m$, giving the nodes of level 1. Continue to create the nodes of further levels using (θ) and (K) as above.

Note that the denominator of the (K) rule has a maximum modal degree which is strictly less than that of its numerator, and that the $\mathcal{C}\mathbf{K}$-saturation process does not increase the maximum modal degree. Hence a path $w_0 \prec w_1 \prec w_2 \cdots$ must terminate (without cycles) because each successor created by (K) has a maximum modal degree strictly lower than that of the parent node.

Let R be $\prec$ and let W_0 consist of all the nodes created in this process, then $\langle W_0, R\rangle$ is a finite, irreflexive and intransitive tree and a model graph for X. Hence by Lemma 4.9.2, there is a **K**-model for X with root w_0.

Proof for $\mathcal{C}$T: If no $\neg\Box P$ occurs in w_0 then $\langle W, R\rangle = \langle \{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph since (i)-(iii) are satisfied. Otherwise, let Q_1, Q_2, $\cdots$, Q_m be all the formulae such that $\neg\Box Q_i \in w_0$ and $\neg Q_i \notin w_0$. Proceed as for $\mathcal{C}$**K**, noting that $\mathcal{C}$**T**-saturation now involves (T) as well, but ignoring the successor for $\neg\Box Q \in w$ if $\neg Q \in w$. Let R be the reflexive closure of $\prec$; that is, put wRw for all worlds in the tree and also put wRw' if $w \prec w'$. Termination is as for $\mathcal{C}$**K**.

Proof for $\mathcal{C}$D: If no $\neg\Box P$ occurs in w_0 then $\langle \{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph since (i)-(iii) are satisfied. Otherwise, proceed as for $\mathcal{C}$**K**, except that $\mathcal{C}$**D**-saturation now involves (D) as well, and let W_{end} be the nodes of (the resulting tree) W_0 that have no successors. For each $w, w' \in W_0$, put wRw' if $w \prec w'$ and put wRw if $w \in W_{end}$. We have to show that (i)-(iii) are satisfied by this R. The only interesting case is to show that $\Box P \in w$ implies $P \in w$ for $w \in W_{end}$. This is true since $w \in W_{end}$ implies that w contains no $\Box P$, as otherwise, w would contain $\neg\Box\neg P$ by (D) and hence would have a successor node by (K), contradicting that $w \in W_{end}$. Termination is as for $\mathcal{C}$**K**.

Proof for $\mathcal{C}$D$'$: If no $\neg\Box Q$ occurs in w_0 and no $\Box P$ occurs in w_0 then $\langle \{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph. Otherwise, let $Z = \{Q_1, \cdots, Q_m\}$ be all the formulae such that $\neg\Box Q_i \in w_0$, $1 \le i \le m$, and let $Y = \{P_1, P_2, \cdots, P_n\}$ be all the formulae such that $\Box P_j \in w_0$, $1 \le j \le n$. We know $m + n \ge 1$. Since w_0 is $\mathcal{C}$**D**$'$-consistent each $\neg Q_i; Y$ is $\mathcal{C}$**D**$'$-consistent, for $i = 1, 2, \cdots, m$ by (θ) and (KD). Also, Y itself is $\mathcal{C}$**D**$'$-consistent by (θ) and (KD). If Z is non-empty then create a Q_i-successor v_i using (KD) containing $(\neg Q_i; Y)$ for each Q_i. But if Z is empty then create a single P-successor y using (KD) containing Y. Put $w_0 \prec v_i$ for each $i = 1 \cdots m$, or $w_0 \prec y$, as the case may be, giving the node(s) of level one. Continuing in this way obtain the node(s) of level two etc. Again, a sequence $w_0 \prec w_1 \prec w_2 \cdots$ must terminate since (KD) reduces the maximum modal degree and $\mathcal{C}$**D**$'$-saturation does not increase it. As in the first proof for $\mathcal{C}$**D** put wRw if $w \in W_{end}$ and put wRw' if $w \prec w'$. Property (iii) holds for $w \in W_{end}$ as end nodes do not contain any $\Box P$, as otherwise, w would have a successor by (KD), contradicting that $w \in W_{end}$.

Proof for $\mathcal{C}$K4: If no $\neg\Box P$ occurs in w_0 then $\langle \{w_0\}, \emptyset\rangle$ is the desired model graph since it is an **K4**-frame and (i)-(iii) are satisfied. Otherwise, let $Q_1, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$.

We can form the sets $\{\Box A : \Box A \in w_0\} \cup \neg\Box Q_i$ for $1 \le i \le m$, by (θ), each of which is a numerator for $(K4)$. Hence by $(K4)$ each denominator $X_i = \{A : \Box A \in w_0\} \cup \{\Box A : \Box A \in w_0\} \cup \neg Q_i$ for $1 \le i \le m$, is also $\mathcal{C}$**K4**-consistent.

Clearly for each X_i we can find some $\mathcal{C}$**K4**-saturated $v_i \supseteq X_i$, with

$v_i \subseteq X^*_{\mathcal{C}\mathbf{K4}}$. Put $w_0 \prec v_i$, $i = 1, \cdots, m$ and call v_i the Q_i-successor of w_0. These are the immediate successors of w_0. Now repeat the construction with each v_i thus obtaining the nodes of level 2 and so on.

In general, the above construction of $\langle W_0, \prec\rangle$ runs ad infinitum. However, since $w \in W_0$ implies $w \subseteq X^*_{\mathcal{C}\mathbf{K4}}$, (a finite set), a sequence $w_0 \prec w_1 \prec \cdots$ in $\langle W_0, \prec\rangle$ either terminates, or a node repeats. If in the latter case $n > m$ are minimal with $w_n = w_m$ we stop the construction and identify w_n and w_m in $\langle W_0, \prec\rangle$ thus obtaining a circle instead of an infinite path. One readily confirms that $\langle W_0, R\rangle$ is a model graph for X where R is the transitive closure of $\prec$. It is obvious that clusters in $\langle W_0, R\rangle$ form a tree.

Proof for $\mathcal{C}$K4D: If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph. Otherwise, proceed as for $\mathcal{C}$**K4**, except that $\mathcal{C}$**K4D**-saturation also involves (D). A sequence either terminates or cycles since $X^*_{\mathcal{C}\mathbf{K4D}}$ is finite. Put $w \prec w$ for all $w \in W_{end}$ and let R be the transitive closure of $\prec$. Property (iii) is satisfied by $w \in W_{end}$ just as in the proof for $\mathcal{C}$**D**.

Proof for $\mathcal{C}$K45: Suppose X is $\mathcal{C}$**K45**-consistent and create a $\mathcal{C}$**K45**-saturated superset $w_0 \subseteq X^*_{\mathcal{C}\mathbf{K45}}$ of X as usual. If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \emptyset\rangle$ is the desired model graph since (i)-(iii) are satisfied.

Otherwise let $Q_i, Q_2, \cdots, Q_k$ be all the formulae such that $\neg\Box Q_i \in w_0$ and create a Q_i-successor for each Q_i using (θ) and the (45) rule. Continue construction of one such sequence $S = w_0 \prec w_1 \prec \cdots$ always choosing a successor that is new to the current sequence. Note that a successor may be new either because it fulfills an eventuality that is not fulfilled by the current sequence, or because it contains formulae that do not appear in previous nodes that fulfill the same eventuality. Since $X^*_{\mathcal{C}\mathbf{K45}}$ is finite, we must sooner or later come to a node w_m such that the sequence $S = w_0 \prec w_1 \prec \cdots \prec w_m$ already contains *all* the successors of w_m. That is, it is not possible to choose a new successor.

Now, the $(K45)$ rule guarantees that if $\neg\Box P \in w_0$ then $\neg\Box P \in w_i$, $i > 0$, so one of the successors of w_m must fulfill $\neg\Box P$, and furthermore, this successor must already appear in the sequence. However, there is no guarantee that this successor is w_1. So, choose the successor w_x of w_m that fulfills some eventuality in w_m, but that appears earliest in S and put $w_m \prec w_x$ giving $S = w_0 \prec w_1 \prec \cdots \prec w_x \prec \cdots \prec w_m \prec w_x$. There are two cases to consider depending on whether $x = 0$ or $x \neq 0$.

Case 1: If $x = 0$, put R as the reflexive, transitive and symmetric closure of $\prec$ over $W_0 = \{w_0, w_1, \cdots, w_m\}$. This gives a frame $\langle W_0, R\rangle$ which is a nondegenerate cluster.

Case 2: If $x \neq 0$, put $W_0 = \{w_0, w_x, w_{x+1}, \cdots, w_m\}$, discarding w_1, w_2, $\cdots$, w_{x-1}, and let R' be the reflexive, transitive and *symmetric* closure of $\prec$ over $W_0 \setminus \{w_0\}$. That is, $R' = \{(w_i, w_j) | w_i \in W_0, w_j \in W_0, i \geq x, j \geq x\}$. Now put $R'' = R' \cup \{(w_0, w_x)\}$ and let R be the transitive closure of R''. The frame $\langle W_0, R\rangle$ now consists of a degenerate cluster w_0 followed by a nondegenerate cluster $w_x R w_{x+1} R \cdots R w_m R w_x$ where R is transitive and euclidean.

Property (i) is satisfied by $\langle W_0, R\rangle$ by construction. We show that (ii) and (iii) are satisfied as follows.

Proof of (ii): The (45) rule also carries *all* eventualities from the numerator to the denominator, including the one it fulfills. Therefore, for all $w_i \in W_0$ we have: $\neg\Box P \in w_i$ implies $\neg\Box P \in w_m$. But we stopped the construction at w_m because no new Q_i-successors for w_m could be found. Hence there is a Q_i-successor for each eventuality of w_m. Since we have a cycle, and eventualities cannot disappear, these are all the eventualities that appear in the cycle. Furthermore, we chose w_x to be the successor of w_m that was earliest in the sequence S. Hence all of the eventualities of w_m are fulfilled by the sequence $w_x R \cdots R w_m$. All the eventualities of w_0 are also in w_m, hence (ii) holds.

Proof of (iii): The (45) rule carries all formulae of the form $\Box P$ from its numerator to its denominator. Hence $\Box P \in w$ and $w \prec v$ implies that $P \in v$ and $\Box P \in v$. But we know that $w_x \prec \cdots \prec w_m \prec w_x$ forms a cycle, hence (iii) holds as well.

Proof for $\mathcal{C}$K45D: Based on the previous proof. If the $(45D)$ rule is ever used with no eventualities present then this can only happen when w_0 contains no eventualities. For if w_0 contained an eventuality then so would all successors.

So if w_0 contains no eventualities and no formulae of the form $\Box P$ then $\langle\{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph. This gives a frame which is a simple (nondegenerate) cluster.

Otherwise, let $Q_1, \cdots, Q_k$ be all the formulae such that $\neg\Box Q_i \in w_0$ and let $P_1, \cdots, P_m$ be all the formulae such that $\Box P_j \in w_0$. Create a successor w_1 for w_0 using $(45D)$ for some Q_i or P_j and continue creating successors using $(45D)$, always choosing a successor new to the sequence until no new successors are possible. Choose w_x as the successor nearest to w_0 giving a cycle $w_0 \prec \cdots \prec w_x \prec \cdots \prec w_m \prec w_x$ and discard $w_1, w_2, \cdots w_{x-1}$ as in the previous proof.

Form R as in the proof for $\mathcal{C}$**K45** where $x = 0$ gives a frame which is a simple cluster and $x \neq 0$ gives a frame which is a degenerate cluster followed by a nondegenerate cluster.

Properties (i)-(iii) can be proved in a similar manner.

Note that the requirement to continually choose a new successor is tantamount to following an infinite path in Shvarts' formulation [Shv89]. That is, the inevitable cycle that we encounter constitutes an infinite branch if it is unfolded out.

Proof for $\mathcal{C}$S4: If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph. Otherwise proceed as for $\mathcal{C}$**K4** except create a successor for eventuality $\neg\Box P \in w$ only if $\neg P \notin w$, and use $(S4)$ to create successors instead of $(K4)$. Then, a successor for w will be based on

$\{\Box A : \Box A \in w\} \cup \neg P$. Let R be the reflexive and transitive closure of $\prec$ (instead of the transitive closure of $\prec$). We can add reflexivity because of closure with respect to (T).

Proof for $\mathcal{C}$S5π^-: see page 44.

Proof for $\mathcal{C}^\dagger$K45: Suppose X is $\mathcal{C}^\dagger$**K45**-consistent and create a $\mathcal{C}^\dagger$**K45**-saturated superset w_0 with $X \subseteq w_0 \subseteq X^*_{\mathcal{C}^\dagger\mathbf{K45}}$ as usual. If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \emptyset\rangle$ is the desired model graph since (i)-(iii) are satisfied.

Otherwise, let $Q_i, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$ and create a Q_i-successor v_i for each Q_i using the (45) rule. This gives all the nodes of level 1, so put $w_0 \prec v_i$, for each $i = 1 \cdots m$, and stop!

Consider any two nodes v_i and v_j with $i \neq j$. Using the facts that each node is subformula-complete and there are no building up rules, we show that

(a) $\Box P \in v_i$ implies $\Box P \in w_0$ implies $P \in v_j$, $P \in v_i$ and $\Box P \in v_j$;

(b) $\neg\Box P \in v_i$ implies $\neg\Box P \in w_0$ implies there exists a v_k such that $\neg P \in v_k$.

Proof of (a): Suppose $\Box P \in v_i$. Then $\Box P \in Sf(w_0)$ since there are no building up rules, and so $\Box P \in w_0$ or $\neg\Box P \in w_0$ since w_0 is subformula-complete. If $\neg\Box P \in w_0$ then $\neg\Box P \in v_i$ by (45), contradicting the $\mathcal{C}^\dagger$**K45**-consistency of v_i. Hence $\Box P \in w_0$. Note that this holds only because the (45) rule carries $\neg\Box P$ into its denominator along with $\neg\Box Y$.

Proof of (b): As for (a) except uniformly replace $\neg\Box P$ by $\Box P$ and vice-versa. The crux of the proof is that (45) preserves all formulae of the form $\Box P$ and $\neg\Box P$.

Hence we can put $v_i R v_j R v_i$ for all v_i and v_j giving a reflexive, transitive and symmetric nondegenerate cluster. If we also put $w_0 R v_i$ for all $i = 1 \cdots m$, and take the transitive closure, then we obtain a degenerate cluster followed by a nondegenerate cluster. If some $v_k = w_0$ then we obtain a lone nondegenerate cluster. In each case the frame is a **K45**-frame.

In either case, (i)-(iii) are satisfied giving a model graph and hence a **K45**-model for X.

Proof for $\mathcal{C}^\dagger$K45D: Similar to the proofs for $\mathcal{C}^\dagger$**K45** and $\mathcal{C}$**KD**.

Proof for $\mathcal{C}^\dagger$KB4: Suppose no $\mathcal{C}^\dagger$**KB4**-tableau for X is closed. Construct a $\mathcal{C}^\dagger$**KB4**-saturated w_0 from X as usual. If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \emptyset\rangle$ is the desired model graph as (i)-(iii) are satisfied. Otherwise, create a successor v_i for each eventuality in w_0 using (θ) and $(K4)$ giving the nodes of level one, put $w_0 \prec v_i$ and stop. Since w_0 contains at least one eventuality, w_0 must be closed with respect to $(T_\Diamond)$, hence $\Box Q \in w_0$ implies $Q \in w_0$. We show that

(a) $\neg\Box P \in v_i$ implies $\neg\Box P \in w_0$; and

(b) $\Box P \in v_i$ implies $\Box P \in w_0$

from which properties (i)-(iii) follow.

(a) Suppose $\neg\Box P \in v_i$ and $\neg\Box P \notin w_0$. The only super-formulae are of the form $\Box A$ hence $\neg\Box P \in Sf(w_0)$ or $\neg\Box P \in \neg Sf(w_0)$ whence $\Box P \in Sf(w_0)$. Since w_0 is subformula-complete we must have $\Box P \in w_0$ and hence $\Box P \in v_i$ by $(K4)$; contradiction.

(b) Suppose $\Box P \in v_i$.

  (i) If $\Box P \in Sf(w_0)$ then $\Box P \in w_0$ or $\neg\Box P \in w_0$. The latter implies $\Box\neg\Box P \in w_0$ by (5) which implies $\neg\Box P \in v_i$; contradiction. Hence if $\Box P \in v_i$ and $\Box P \in Sf(w_0)$ then $\Box P \in w_0$ whence $P \in v_i$ by $(K4)$ and $P \in w_0$ by $(T_\diamond)$.

  (ii) If $\Box P \notin Sf(w_0)$ then $\Box P = \Box\neg\Box Q$ for some eventuality $\neg\Box Q$ of v_i. Hence $\neg\Box Q \in v_i$. By (a) we then have $\neg\Box Q \in w_0$, which by (5) gives $\Box\neg\Box Q \in w_0$. But $\Box\neg\Box Q$ is $\Box P$, hence $\Box P \in Sf(w_0)$; contradiction. Hence case $\Box P \notin Sf(w_0)$ is impossible.

Now let R be the reflexive, transitive and symmetric closure of $\prec$. Note that reflexivity for w_0 comes from saturation with respect to $(T_\diamond)$ and reflexivity for v_i comes from property (b) via $(K4)$. Thus when w_0 contains at least one eventuality, we get an **S5**-frame (showing that **K4B** is "almost" **S5**).

Proof for $\mathcal{C}^\dagger$S4: If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\},\{(w_0,w_0)\}\rangle$ is the desired model graph. Otherwise, let $Q_1, Q_2, \cdots, Q_k$ be all the formulae such that $\neg\Box Q_i \in w_0$ and $\neg Q_i \notin w_0$. Create a Q_i-successor v_i of level 1 for each Q_i using the (θ) and $(S4)$ rules, and continue in this way to obtain the nodes of level 2 and so on with the following termination condition:

(*) if $w_0 \prec w_1 \prec \cdots \prec w_{i-1} \prec w_i$ is a path in this construction and $i \geq 1$ is the least index such that $\Box A \in w_i$ implies $\Box A \in w_{i-1}$, then put $w_i \prec w_{i-1}$ giving a cycle on this path and stop!

First of all, this termination condition is satisfactory since $(S4)$ ensures that $\Box A \in w_j$ implies $\Box A \in w_{j+1}$ so that $\Box$-formulae accumulate and we eventually run out of new $\Box$-formulae since $X^*_{\mathcal{C}^\dagger\mathbf{S4}}$ is finite.

Second, note that $\mathcal{C}^\dagger$**S4** contains $(sfcT)$ and hence each w_i is subformula-complete. Since there are no building up rules, the only new formulae that may appear by saturating with the $(sfcT)$ rules are the negations of subformulae from the predecessor. Therefore, each $w_{n+1} \subseteq Sf(\widetilde{w_n})$ where $\widetilde{w} = Sf(w) \cup \neg Sf(w)$.

Let R be the reflexive and transitive closure of $\prec$. It is obvious that clusters of R form a tree. To prove that $\langle W_0, R\rangle$ is a model graph for X we have to prove (i)-(iii).

(i) Clearly (i) holds by construction;

(ii) Suppose $\neg\Box P \in w_j$ where w_j is some arbitrary world of some arbitrary path of our construction. If the termination condition was not applied to w_j, then either $\neg P \in w_j$ or w_j has a P-successor fulfilling

$\neg\Box P$ by $(S4)$ and so (ii) is satisfied. That is (ii) holds for any world to which the termination condition was not applied.
If the termination condition was applied to w_j, then it could not have been applied to w_{j-1}. Hence (ii) holds for w_{j-1}. So all we have to show is that $\neg\Box P \in w_{j-1}$ because, in this case, (ii) would then hold for w_j from the fact that $w_j R w_{j-1}$ and the transitivity of R.
Suppose to the contrary that $\neg\Box P \notin w_{j-1}$. Since $\neg\Box P \in w_j$ by supposition, we must have $\Box P \in Sf(w_{j-1})$ by the second point we noted above. Then $\Box P \in w_{j-1}$ by (the subformula-completeness) Lemma 4.12.6, and $\Box P \in w_j$ by $(S4)$ contradicting the $\mathcal{C}^\dagger\mathbf{S4}$-consistency of w_j since $\neg\Box P \in w_j$. Hence (ii) also holds.

(iii) Suppose $\Box P \in w_j$. If (*) was not applied to w_j then (iii) holds as for $\mathcal{C}\mathbf{S4}$ by (T) since $(S4)$ preserves $\Box$-formulae. If (*) was applied to w_j then (iii) would follow from $\Box P \in w_{j-1}$ by $(S4)$ and (T). But this is exactly what (*) guarantees. Hence (iii) holds as well.

Proof for $\mathcal{C}^\dagger\mathbf{B}$: If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph as (i)-(iii) are satisfied. Otherwise, let $Q_1, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$ and $\neg Q_i \notin w_0$. Since w_0 is $\mathcal{C}^\dagger\mathbf{B}$-saturated, w_0 is subformula-complete, hence $Q_i \in w_0$ for each Q_i. Create a Q_i-successor for each Q_i using (θ) and (K) giving the nodes of level one. Repeat this procedure to give the nodes of level two and so on. For any node w in this construction let $s(w)$ be the number of formulae with $P \in w$ and $\neg\Box P \in w$. Let $t(w) = s(w) + mdeg(w)$. To quote Rautenberg *"It is easily seen that $w \prec v \rightarrow t(v) < t(w)$, so that W_0 is finite."*, but as shown in [Gor92] Rautenberg's definition of *mdeg* is not sufficient. We accept Rautenberg's claim for the moment and return to this issue after completing the model construction.

Let R be the reflexive and symmetric closure of $\prec$ so that $\langle W_0, R\rangle$ is a $\mathbf{B}$-frame. We have to show that (i)-(iii) hold. The only difficulty is to show symmetry: that is, $\Box P \in w_{i+1}$ and $w_i \prec w_{i+1}$ implies $P \in w_i$. So suppose that $w_i \prec w_{i+1}$ and $\Box P \in w_{i+1}$. We have to show that $P \in w_i$. There are two cases: $\Box P \in Sf(w_i)$ or $\Box P \notin Sf(w_i)$.

Case 1: If $\Box P \in Sf(w_i)$, then $\Box P \in w_i$ or $\neg\Box P \in w_i$ since w_i is subformula-complete. If $\Box P \in w_i$ then $P \in w_i$ by (T) and we are done. Otherwise, if $\neg\Box P \in w_i$ and $P \notin w_i$ then $\neg P \in w_i$ and $\Box\neg\Box P \in w_i$ by (B) and so $\neg\Box P \in w_{i+1}$ contradicting the consistency of w_{i+1} since $\Box P \in w_{i+1}$ by supposition. Hence $\neg\Box P \in w_i$ also implies that $P \in w_i$.

Case 2: If $\Box P \notin Sf(w_i)$ then $\Box P = \Box\neg\Box Q$ for some $\neg\Box Q \in w_{i+1}$ and $\neg Q \in w_{i+1}$. Hence $\neg\Box Q \in Sf(w_i)$ or $\neg\Box Q \in \neg Sf(w_i)$ whence $\Box Q \in Sf(w_i)$. By subformula-completeness we then have $\Box Q \in w_i$ or $\neg\Box Q \in w_i$. If $\Box Q \in w_i$, then $Q \in w_{i+1}$ contradicting the $\mathcal{C}^\dagger\mathbf{B}$-consistency of w_{i+1} since $\neg Q \in w_{i+1}$. Hence $\neg\Box Q \in w_i$. But then $P \in w_i$ since P is $\neg\Box Q$ and we are done.

Now, we still have to show that this construction terminates. The crux of the matter is to use a definition of a metric *mdg* say, which is like our *mdeg* but where $mdg(A \wedge B) = mdg(A) + mdg(B)$ rather than $max\{mdg(A), mdg(B)\}$ [Mas95a]. Similarly, for a set X, $mdg(X) = \Sigma_{A \in X} mdg(A)$ rather than $max\{mdg(A) \mid A \in X\}$. Then, a rather tedious counting exercise, which we omit for brevity, suffices to show that if $w \prec v$ then $t(v) < t(w)$, which is enough to show termination. We have retained our version of *mdeg* because it is useful for other puposes.

Proof for $\mathcal{C}^\dagger$S5: If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph as (i)-(iii) are satisfied. Otherwise, let $Q_1, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$. Since w_0 is $\mathcal{C}^\dagger$**S5**-saturated, $\Box\neg\Box Q_i \in w_0$ for each Q_i by (5). Create a Q_i-successor for each Q_i using (θ) and $(S4)$ giving the nodes v_i of level one, put $w_0 \prec v_i$, for each $i = 1, 2, \cdots, m$ and stop! Let R be the reflexive, transitive and symmetric closure of $\prec$. By construction, $\langle W_0, R\rangle$ is an **S5**-frame. We have to show that (i)-(iii) hold.

For any k, with $1 \le k \le m$, and $w_0 \prec v_k$, we show that:

(a) $\neg\Box P \in v_k$ implies $\neg\Box P \in w_0$; and

(b) $\Box P \in v_k$ implies $\Box P \in w_0$

from which (i)-(iii) follow.

(a) Suppose $w_0 \prec v_k$, $\neg\Box P \in v_k$ and $\neg\Box P \notin w_0$. Since $\neg\Box P \in Sf(w_0)$, and w_0 is subformula-complete, we have $\Box P \in w_0$. But then, by $(S4)$, $\Box P \in v_k$, contradicting the $\mathcal{C}^\dagger$**S5**-consistency of v_k. Hence $\neg\Box P \in w_0$.

(b) Suppose $w_0 \prec v_k$ and $\Box P \in v_k$, then $\Box P \in Sf(w_0)$ or $\Box P \notin Sf(w_0)$.

(b1) If $\Box P \in Sf(w_0)$ and $\Box P \notin w_0$, then $\neg\Box P \in w_0$ since w_0 is subformula-complete. Then $\Box\neg\Box P \in w_0$ by (5) and $\neg\Box P \in v_k$ by $(S4)$, contradicting the $\mathcal{C}^\dagger$**S5**-consistency of v_k. Hence, if $\Box P \in v_k$ and $\Box P \in Sf(w_0)$ then $\Box P \in w_0$.

(b2) If $\Box P \notin Sf(w_0)$ then $\Box P = \Box\neg\Box Q$ for some $\neg\Box Q \in v_k$ since this is the only way that formulae from outside $Sf(w_0)$ can appear in v_k. By (a), $\neg\Box Q \in v_k$ implies $\neg\Box Q \in w_0$ which by (5) implies $\Box\neg\Box Q \in w_0$. Since $\Box\neg\Box Q$ *is* $\Box P$, we have $\Box P \in w_0$. But this is absurd since it implies that $\Box P \in Sf(w_0)$ and our supposition was that $\Box P \notin Sf(w_0)$. Hence the subcase (b2) cannot occur.

Proof for $\mathcal{C}^\dagger$S5$'$: For completeness suppose X is $\mathcal{C}$**S5**$'$-consistent and create a $\mathcal{C}^\dagger$**S5**$'$-saturated superset w_0 with $X \subseteq w_0 \subseteq X^*_{\mathcal{C}\mathbf{S5}'}$ as usual.

If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph. Otherwise, let $Q_1, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$ and $\neg Q_i \notin w_0$. Create a Q_i-successor v_i of level 1 for each Q_i using the $(S5)$ rule and stop!

Let $W_0 = \{w_0, v_1, v_2, \cdots, v_m\}$. Consider any two nodes v_i and v_j of level 1 so that $w_0 \prec v_i$ and $w_0 \prec v_j$ with $i \neq j$. We claim that:

(a) $\Box P \in v_i$ implies $\Box P \in w_0$ implies $\Box P \in v_j$; and

(b) $\neg\Box P \in v_i$ implies $\neg\Box P \in w_0$ implies there exists a $w \in W_0$ with $\neg P \in w$.

Proof of (a): Suppose $\Box P \in v_i$, then $P \in v_i$ by (T). Also, $\Box P \in Sf(w_0)$ as there are no building up rules, hence $\Box P \in w_0$ or $\neg\Box P \in w_0$ by $(sfcT)$. If $\neg\Box P \in w_0$ then either $\neg P \in v_i$ or $\neg\Box P \in v_i$ by $(S5)$. The first contradicts the $\mathcal{C}^\dagger\mathbf{S5}'$-consistency of v_i since $P \in v_i$ and so does the second since $\Box P \in v_i$. Hence $\Box P \in w_0$. And then $\Box P \in v_j$ by $(S5)$ and $P \in v_j$ by (T).

Proof of (b): Suppose $\neg\Box P \in v_i$. Then as there are no building up rules, $\neg\Box P \in Sf(w_0)$. Hence $\Box P \in w_0$ or $\neg\Box P \in w_0$ since w_0 is subformula-complete. If $\Box P \in w_0$ then $\Box P \in v_i$ by $(S5)$, contradicting the $\mathcal{C}^\dagger\mathbf{S5}'$-consistency of v_i since $\neg\Box P \in v_i$ by supposition. Hence $\neg\Box P \in w_0$. And then either $\neg P \in w_0$, or there is some v_k such that $\neg P \in v_k$ by $(S5)$. That is, the w we seek is either w_0 itself, or one of the nodes of level 1.

Putting R equal to the reflexive, symmetric and transitive closure of $\prec$ gives an **S5**-model graph since (i)-(iii) follow from (a) and (b). ∎

4.12.4 Bibliographic Remarks and Discussion

The cut-free calculi $\mathcal{C}\mathbf{K}$, $\mathcal{C}\mathbf{T}$, $\mathcal{C}\mathbf{D}$, $\mathcal{C}\mathbf{D}'$, $\mathcal{C}\mathbf{K4}$, $\mathcal{C}\mathbf{K4D}$ and $\mathcal{C}\mathbf{S4}$ can all be traced back to Fitting [Fit73] via Fitting [Fit83] although our presentation is based on the work of Hintikka [Hin55] and Rautenberg [Rau83]. The system $\mathcal{C}\mathbf{K4D}$ is an obvious extension of Rautenberg's system $\mathcal{C}\mathbf{D}$, and $\mathcal{C}\mathbf{D}'$ is lifted straight from Fitting [Fit83]. The advantage of $\mathcal{C}\mathbf{D}'$ is that it has the subformula property whereas $\mathcal{C}\mathbf{D}$ does not. Clearly, the $(K4)$ rule can be extended to handle seriality as done in the (KD) rule to give a $(K4D)$ rule, but we omit details. The tableau systems $\mathcal{C}\mathbf{K45}$ and $\mathcal{C}\mathbf{K45D}$ are based on the work of Shvarts [Shv89] (also known as Schwarz), while $\mathcal{C}\mathbf{K4B}$ and the $(T_\Diamond)$ rule come from the work of Amerbauer [Ame93].

Some of the desired properties of R can be obtained in two different ways. For example, Rautenberg encodes the seriality of **D**-frames by the *static* (D) rule which adds an eventuality $\Diamond P$ for every formula of the form $\Box P$. The transitional (K) rule then fulfills that eventuality. On the other hand Shvarts [Shv89] and Fitting [Fit83] use the *transitional* rule (KD). Similarly, the $(S5)$ *transitional* rule due to Fitting builds in the effect of Rautenberg's *static* rule (5) by carrying $\neg\Box P$ and $\neg\Box Y$ from the numerator into the denominator.

Rautenberg [Rau83] does not explicitly distinguish transitional and static modal rules. Hence his rules for (T), (D), (B), (sfc) and $(sfcT)$ do not carry all the numerator formulae into their denominators. For example, Rautenberg's (T) rule is shown below left whereas ours is shown below right:

$$\frac{X; \Box P}{X; P} \qquad (T)\,\frac{X; \Box P}{X; \Box P; P}$$

Thus contraction is implicit in his systems and as we saw in Example 4.12.1 (page 29), contraction is necessary for some modal systems.

The $\mathcal{C}^{\dagger}$**S4** system is based on ideas of Hanson [Han66] where he gives Kripke-like tableau systems for **S4** and **S5** using a form of $(sfcT)$ as early as 1966. The tableau system $\mathcal{C}^{\dagger}$**S4** is not exactly Hanson's system but the ideas are his. The advantage of adding $(sfcT)$ is that the termination condition in the completeness proof is much easier to check than the one for $\mathcal{C}$**S4** where we have to look at all predecessors in order to detect a cycle. However, the overheads associated with any sort of cut rule are significant, and a more detailed analysis shows that $\mathcal{C}^{\dagger}$**S4** performs much useless work. Hanson also suggests a tableau system for **S5** along these lines, but in it he uses a rule which explicitly adds a formula to the parent node to obtain symmetry. This is forbidden for our tableau systems since we cannot return to previous nodes.

The tableau systems of Heuerding *et al* [HSZ] are further refinements of our tableau systems which allow for a more efficient check for cyclic branches. However, they are nonstandard in that the denominators and numerators carry extra sets to store the necessary information.

Notice that the effects of $(sfcT)$ on w_0 when R is to be transitive and there are no building up rules like (5) is to flush out all the eventualities that could possibly appear in any successor. That is, if $\neg\Box P$ is going to appear in a successor, it must be in $Sf(w_0)$. But then it must be in w_0 since otherwise by $(sfcT)$, we would have $\Box P \in w_0$ contradicting the appearance of $\neg\Box P$ in any consistent successor. Hence the number of eventualities never increases as all the eventualities that will ever appear are already in w_0. Indeed this fact may actually make things worse since we will have to fulfill $\neg\Box P$ at the first level of the model construction as well as at deeper levels where $\neg\Box P$ reappears. The refinements of Heuerding et al [HSZ] maye be useful in such cases since one of their ideas addresses exactly this point.

The idea behind (sfc) and $(sfcT)$ is to put extra information into a node before leaving it for good. That is, once we leave a node in our tableau procedure, we can never return to it. Also, the transitional rules usually lose information in the transition from the numerator to the denominator. The (sfc) and $(sfcT)$ rules are used to make up for this "destructive" aspect of our transitional rules.

The completeness proofs in this section go through unchanged [Mas95a] if we replace the (sfc) and $(sfcT)$ rules by the "modal cut" rule (mc) shown below:

$$(mc)\ \frac{X}{\Box P; X \mid \neg\Box P; X} \text{ where } \Box P \in Sf(X)$$

Also, many of the rule combinations can be further refined. For example, the (B) rule subsumes the modal aspects of the $(sfcT)$ rule so that only the non-modal part is necessary in $\mathcal{C}^{\dagger}\mathbf{B}$; see also [Ame93] for further refinements.

The tableau systems $\mathcal{C}^{\dagger}\mathbf{B}$ and $\mathcal{C}^{\dagger}\mathbf{S5}$ are due to Rautenberg while $\mathcal{C}^{\dagger}\mathbf{S5}'$, $\mathcal{C}^{\dagger}\mathbf{K45}$ and $\mathcal{C}^{\dagger}\mathbf{K45D}$ are an amalgamation of ideas of Fitting, Hanson and Rautenberg. Note that in the latter, we add (sfc), not $(sfcT)$ since **K45**-frames and **K45**-frames are not reflexive. The advantage over the cut-free counterparts $\mathcal{C}\mathbf{K45}$ and $\mathcal{C}\mathbf{K45D}$ is that the completeness proofs, and hence the satisfiability tests based upon them, are much simpler. Note that $\mathcal{C}^{\dagger}\mathbf{S5}$ does not have the subformula property, but $\mathcal{C}^{\dagger}\mathbf{S5}'$ does.

Fitting [Fit83, page 201] gives tableau calculi for the symmetric logics **KB**, **KDB**, **KTB**, and **S5** using a **semi-analytic cut** rule (sac), which he attributes to Osamu Sonabe. The (sac) rule is allowed to cut on subformulae of formulae that are in the numerator, and also on superformulae obtained by repeatedly prefixing modalities $\Box$, $\neg\Box$, $\Diamond$ and $\neg\Diamond$, to these subformulae. Since the superformulae are not bounded, as they are in Rautenberg's systems, the semi-analytic cut rule cannot give a decision procedure.

Fitting's semi-analytic system for **S5** is essentially $\mathcal{C}\mathbf{T} + (S5) + (sac)$. Fitting [Fit83, page 226] replaces the semi-analytic cut rule with a (static) building up rule of the form

$$(\pi)\ \frac{X; P}{X; \Diamond P; P}$$

and proves that his system $\mathcal{C}\mathbf{S5}\pi = \mathcal{C}\mathbf{T} + (S5) + (\pi)$ is sound and (weakly) complete with respect to **S5**-frames. But notice that the (π) rule is not "once off" since it can lead to an infinite chain $A \in w, \Diamond A \in w, \Diamond\Diamond A \in w, \cdots$ so this system cannot give a decision procedure for **S5** either. That is, we have merely traded one non-analytic rule for another.

Fitting then proves the curious fact that a single formula A is an **S5**-theorem if and only if a $\mathcal{C}\mathbf{S5}\pi$-tableau for $\{\neg A\}$ closes, and furthermore, that the (π) rule needs to be used only *once* at the beginning of the $\mathcal{C}\mathbf{S5}\pi$-tableau to lift $\neg A$ to $\neg\Box A$ [Fit83, page 229]. That is, the system $\mathcal{C}\mathbf{S5}\pi^-$ *without* the (π) rule is (weakly) complete for **S5** in the sense that A is an **S5**-theorem if and only if a $\mathcal{C}\mathbf{S5}\pi^-$-tableau for $\{\neg\Box A\}$ closes. Fitting gives a completeness proof in terms of maximal consistent sets, but a constructive completeness for this system is also easy as given below.

Completeness Proof for $\mathcal{C}\mathbf{S5}\pi^-$: Suppose no $\mathcal{C}\mathbf{S5}\pi^-$-tableau for the singleton set $\{\neg\Box A\}$ closes. Construct some $\mathcal{C}\mathbf{S5}\pi^-$-saturated set w_0

from $\neg\Box A$ by applying all the non-structural static rules; obtaining $w_0 = \{\neg\Box A\}$! Now construct a tree of $\prec$-successors as in the $\mathcal{C}\mathbf{S4}$ completeness proof except that we use the transitional rule $(S5)$ instead of $(S4)$ to create $\prec$-successors. Let R be the reflexive and transitive closure of $\prec$ to obtain a finite tree of finite clusters as in the $\mathcal{C}\mathbf{S4}$ case. Consider some final cluster C of this tree. Since C is final, any eventuality in any of its sets must be fulfilled by some set of C itself, as otherwise, C could not be final. But note that the $(S5)$ rule carries *all* its eventualities from its numerator into its denominator. Thus, in this case, $\neg\Box A$ is in every member of C, and hence some set $w_1 \in C$ has $\{\neg\Box A, \neg A\} \subseteq w_1$. But a *final* cluster is also symmetric, hence C is an $\mathbf{S5}$-frame and hence an $\mathbf{S5}$-model for $\{\neg A, \neg\Box A\}$ at w_1 under the usual valuation $\vartheta(p) = \{w : p \in w\}$. This completes the unusual proof for $\mathcal{C}\mathbf{S5}\pi^-$ that: if there is no closed $\mathcal{C}\mathbf{S5}\pi^-$-tableau for $\{\neg\Box A\}$ then $\neg A$ is $\mathbf{S5}$-satisfiable. That is, if $\models_{\mathbf{S5}} A$ then $\vdash_{\mathcal{C}\mathbf{S5}\pi^-} \Box A$.

For the logics with a symmetric R we seem to need analytic cut, either as (sfc) or as $(sfcT)$. The subformula property can be regained for some logics by changing the transitional rules to carry more information from the numerator to the denominator. But note that a building up rule seems essential for $\mathcal{C}\mathbf{B}$, so not all the systems are amenable to this trick.

4.13 Modal Logics of Knowledge and Belief

In this section we give a brief overview of tableau systems for the modal logics **S4R**, **S4F** and **S4.2**. These logics, together with the logics **K45** and **K45D**, have proved useful as nonmonotonic modal logics where the formula $\Box A$ is read as "A is believed" or as "A is known" [Moo85], [Sch92, Tru, Tru91, ST92, MST91]. In these logics, the reflexivity axiom, $\Box A \to A$, is deliberately omitted on the grounds that believing A should not imply that A is true. The logic **K45D** is another candidate for such logics of belief because its extra axiom, $\Box A \to \Diamond A$, which can be written as $\Box A \to \neg\Box\neg A$, encodes the intuition that "if A is believed then $\neg A$ is not believed".

Figure 8 shows the tableau rules we require. The tableau calculi we consider are shown below:

$\mathcal{C}\mathbf{L}$	Static Rules	Transitional Rules	$X^*_{\mathcal{C}\mathbf{L}}$
$\mathcal{C}\mathbf{S4R}$	$\mathcal{C}\mathbf{PC}$, (T), (R)	$(S4)$	$Sf \neg Sf \Box \widetilde{X}$
$\mathcal{C}^\dagger\mathbf{S4.2}$	$\mathcal{C}\mathbf{PC}$, $(sfcT)$, (T), $(S4.2)$	$(S4)$	$Sf \neg Sf \Box X^*_{\mathcal{C}\mathbf{S4R}}$
$\mathcal{C}^\dagger\mathbf{S4F}$	$\mathcal{C}\mathbf{PC}$, $(sfcT)$, (T), $(S4.2)$	$(S4F), (S4)$	$Sf \neg Sf \Box X^*_{\mathcal{C}\mathbf{S4R}}$

The $(S4F)$ rule is odd in that its left denominator is static whilst its right denominator is transitional. The $(S4.2)$ rule is the only potentially dangerous rule since its denominator contains a formula to which the rule can be applied in an endless fashion. To forbid this the new formula is

$$(R)\ \frac{X; \neg\Box P}{X; \neg\Box P; \neg P \quad | \quad X; \neg\Box P; \Box\neg\Box P; P}$$

$$(S4F)\ \frac{U; \Box X; \neg\Box P; \neg\Box Y}{U; \Box X; \neg\Box P; \neg\Box Y; \Box\neg\Box P \quad | \quad \Box X; \neg\Box P; \neg\Box Y; \neg P}$$

$$(S4.2)\ \frac{X; \neg\Box P}{X; \neg\Box P; \Box\neg\Box P \quad | \quad X; \neg\Box P; \Box(\neg\Box\neg\Box P)^*}\ \neg\Box P \text{ not starred}$$

Fig. 8. Tableau rules for **S4R**, **S4F** and **S4.2**.

marked with a star and the $(S4.2)$ rule is restricted to apply only to non-starred formulae. All other rules must treat starred formula as if they were non-starred.

The soundness and completeness of these calculi is proved in detail by Goré [Gor91]. Goré actually proves soundness and completeness with respect to a class of finite frames, each of which is an **L**-frame as defined here. Consequently, these logics are also characterised by the classes of finite-**L**-frames shown in Figure 13. Note that the values of $X^*_{\mathcal{C}\mathbf{L}}$ are different from those in [Gor91] but it is easy to see that the new ones are the correct ones due to the effect of $(sfcT)$.

Tableau systems for the logics **K4.2** and **K4.2G** can be found in Amerbauer's dissertation [Ame93].

$$(G)\ \frac{\Box X; \neg\Box P}{X; \Box X; \neg P; \Box P} \qquad\qquad (Grz)\ \frac{\Box X; \neg\Box P}{X; \Box X; \neg P; \Box(P \to \Box P)}$$

Fig. 9. Tableau Rules for logics of provability

Example 4.13.1. The formula $\Diamond\Box p \to \Box\Diamond p$ is an instance of the axiom 2, and hence is a theorem of **S4.2**. The following closed $\mathcal{C}^{\dagger}$**S4.2**-tableau for its negation $(\Diamond\Box p \land \neg\Box\Diamond p)$ which in primitive notation is $(\neg\Box\neg\Box p) \land (\neg\Box\neg\Box\neg p)$ illustrates the use of starred formulae.

$$(\neg\Box\neg\Box p) \land (\neg\Box\neg\Box\neg p) \quad (\land)$$

$$\neg\Box\neg\Box p, \neg\Box\neg\Box\neg p \quad (S4.2)$$

$$\begin{array}{ll|rr}
(S4) & \neg\Box\neg\Box p, \neg\Box\neg\Box\neg p, \Box\neg\Box\neg\Box p & \neg\Box\neg\Box p, \neg\Box\neg\Box\neg p, \Box(\neg\Box\neg\Box\neg\Box p)* & (S4) \\
(\neg\neg) & \neg\neg\Box\neg p, \Box\neg\Box\neg\Box p & \neg\neg\Box p, \Box(\neg\Box\neg\Box\neg\Box p)* & (\neg\neg) \\
(T) & \Box\neg p, \Box\neg\Box\neg\Box p & \Box p, \Box(\neg\Box\neg\Box\neg\Box p)* & (T) \\
(S4) & \Box\neg p, \neg\Box\neg\Box p, \Box\neg\Box\neg\Box p & \Box p, (\neg\Box\neg\Box\neg\Box p)*, \Box(\neg\Box\neg\Box\neg\Box p)* & (S4) \\
(\neg\neg) & \Box\neg p, \neg\neg\Box p, \Box\neg\Box\neg\Box p & \Box p, \neg\neg\Box\neg\Box p, \Box(\neg\Box\neg\Box\neg\Box p)* & (\neg\neg) \\
(T) & \Box\neg p, \Box p, \Box\neg\Box\neg\Box p & \Box p, \Box\neg\Box p, \Box(\neg\Box\neg\Box\neg\Box p)* & (T) \\
(T) & \neg p, \Box\neg p, \Box p, \Box\neg\Box\neg\Box p & \Box p, \neg\Box p, \Box\neg\Box p, \Box(\neg\Box\neg\Box\neg\Box p)* & (S4) \\
(\bot) & \neg p, \Box\neg p, p, \Box p, \Box\neg\Box\neg\Box p & \Box p, \neg p, \Box\neg\Box p, \Box(\neg\Box\neg\Box\neg\Box p)* & (T) \\
 & & p, \Box p, \neg p, \Box\neg\Box p, \Box(\neg\Box\neg\Box\neg\Box p)* & (\bot)
\end{array}$$

4.14 Modal Logics With Provability Interpretations

In this section we give tableau calculi for the modal logics that have important readings as logics of "provability" where $\Box A$ is read as "it is provable in Peano Arithmetic that A holds"; see Fitting [Fit83, page 241] and Boolos [Boo79]. These systems are obtained either by adding the axiom G:$\Box(\Box A \to A) \to \Box A$, named after Gödel-Löb and sometimes called GL, or adding the axiom Grz:$\Box(\Box(A \to \Box A) \to A) \to A$, named after Grzegorczyk, or adding the axioms 4 and G_o:$\Box(\Box(A \to \Box A) \to A) \to \Box A$, to **K**.

It is known that both G and Grz imply the transitivity axiom 4 when they are respectively added to **K** [vB78]. But the logic $\mathbf{K4G_o}$ whose frames share some of the properties of **G**-frames and **Grz**-frames, explicitly contains 4 as an axiom. It is also known that Grz implies reflexivity.

Once again, all the tableau calculi contain the rules of $\mathcal{C}$**PC** and one or more logical rules from Figure 9 on page 47 as shown below:

$\mathcal{C}\mathbf{L}$	Static Rules	Transitional Rules	$X^*_{\mathcal{C}\mathbf{L}}$
$\mathcal{C}\mathbf{G}$	$\mathcal{C}\mathbf{PC}$	(G)	$\widetilde{X}$
$\mathcal{C}\mathbf{K4G_o}$	$\mathcal{C}\mathbf{PC}$	(Grz)	$Sf\Box(\widetilde{X} \to \Box\widetilde{X})$
$\mathcal{C}\mathbf{Grz}$	$\mathcal{C}\mathbf{PC}$, (T)	(Grz)	$Sf\Box(\widetilde{X} \to \Box\widetilde{X})$

The semantic and axiomatic intuitions behind these rules are more enlightening than any technical proof (of soundness) so we present these as well.

Intuitions for (G) **:** We know that axiomatically formulated logic **G** is characterised by **G**-frames. Therefore, axiom G must be valid on any **G**-frame; hence true in any world of any **G**-model. The axiom G is

$$\Box(\Box A \to A) \to \Box A.$$

Its contrapositive is

$$\neg\Box A \to \neg(\Box(\Box A \to A))$$

which is the same as

$$\neg\Box A \to \Diamond(\Box A \land \neg A).$$

Thus, if the numerator represents a world w where $\neg\Box P$ is true, then there exists another world w' where $\Box P$ is true and P is false, and w' is reachable from w. The denominator represents this world.

Intuitions for (Grz) **:** The axiom Grz is

$$\Box(\Box(A \to \Box A) \to A) \to A.$$

It is known that 4 and T are theorems of **Grz** [HC84, page 111], hence **S4** $\subseteq$ **Grz**. Segerberg [Seg71, page 107], and more recently Goré et al [GHH95], show that **Grz** = **S4Grz** = $\mathbf{S4G_o}$ where G_o is

$$\Box(\Box(A \to \Box A) \to A) \to \Box A$$

which gives the following (contraposed formulae) as theorems of **Grz**:

$$\begin{array}{rcl} \neg\Box A & \to & \neg\Box(\Box(A \to \Box A) \to A) \\ \neg\Box A & \to & \Diamond(\Box(A \to \Box A) \land \neg A). \end{array}$$

Thus, if $\neg\Box P$ is true at the numerator, then there exists some world where $\Box(P \to \Box P) \land \neg P$ eventually becomes true. The denominator of (Grz) represents this world.

Theorem 4.14.1 (Soundness). *The calculi $\mathcal{C}\mathbf{G}$, $\mathcal{C}\mathbf{Grz}$ and $\mathcal{C}\mathbf{K4G_o}$ are sound with respect to* **G***-frames,* **Grz***-frames and* $\mathbf{K4G_o}$*-frames respectively.*

Proof Outline : For each rule in $\mathcal{C}\mathbf{L}$ we have to show that if the numerator of the rule is **L**-satisfiable then so is at least one of the denominators.

Proof of $\mathcal{C}\mathbf{G}$: Suppose $\mathcal{M} = \langle W, R, V\rangle$ is a **G**-model, $w_0 \in W$ and $w_0 \models \Box X; \neg\Box P$. Thus there exists some $w_1 \in W$ with $w_0 R w_1$ and $w_1 \models X; \Box X; \neg P$ by the transitivity of R. Since R is irreflexive, $w_0 \neq w_1$. Suppose $w_1 \not\models \Box P$. Then $w_1 \models \neg\Box P$ and there exists some $w_2 \in W$ with $w_1 R w_2$ and $w_2 \models X; \Box X; \neg P$ by transitivity of R. Since R is irreflexive, $w_1 \neq w_2$. Since R is transitive, $w_2 = w_0$ would give $w_1 R w_0 R w_1$ implying $w_1 R w_1$ and contradicting the irreflexivity of R, hence $w_0 \neq w_2$. Suppose $w_2 \not\models \Box P$ then ... Continuing in this way, it is possible to obtain an infinite path of distinct worlds in $\mathcal{M}$ contradicting the **G**-frame condition on $\mathcal{M}$. Thus there must exist some $w_i \in W$ with $w_0 R w_i$ and $w_i \models X; \Box X; \neg P; \Box P$ and we are done.

Proof of (T) for $\mathcal{C}\mathbf{Grz}$: The (T) rule is sound for **Grz**-frames since every **Grz**-frame is reflexive.

Proof of (Grz) for $\mathcal{C}\mathbf{K4G_o}$: Suppose $\mathcal{M} = \langle W, R, V\rangle$ is a $\mathbf{K4G_o}$-model, then R is transitive, there are no proper clusters, and there are no proper ∞-R-chains. Suppose $w_0 \in W$ is such that $w_0 \models \Box X; \neg\Box P$. We have to show that there exists some $w_n \in W$ with $w_0 R w_n$ and $w_n \models X; \Box X; \neg P; \Box(P \to \Box P)$. Since R is transitive, $w_0 \models \Box X$ means that $\forall w \in W, w_0 R w$ implies $w \models X; \Box X$. Thus our task is reduced to showing that there exists some $w_n \in W$ such that $w_0 R w_n$ and $w_n \models \neg P; \Box(P \to \Box P)$. Suppose for a contradiction that no such world exists in W. That is,

(a) $\forall w \in W, w_0 R w$ implies $w \not\models \neg P; \Box(P \to \Box P)$.

Since $w_0 \models \neg\Box P$, there exists some $w_1 \in W$ with $w_0 R w_1$ and $w_1 \models \neg P$. By (a), $w_1 \not\models \Box(P \to \Box P)$ and hence $w_1 \models \neg\Box(P \to \Box P)$. Thus there exists some $w_2 \in W$ with $w_1 R w_2$ and $w_2 \models \neg(P \to \Box P)$, that is, $w_2 \models P \wedge \neg\Box P$. Since $w_1 \models \neg P$, $w_1 \neq w_2$ and since $\mathbf{K4G_o}$-models cannot contain proper clusters, $w_0 \neq w_2$. Since $w_2 \models \neg\Box P$ there exists some $w_3 \in W$ with $w_3 \models \neg P$. Since $w_2 \models P$, $w_3 \neq w_2$. And $w_3 \neq w_0$ and $w_3 \neq w_1$ as either would give a proper cluster. By (a), $w_3 \not\models \Box(P \to \Box P)$ and hence $w_3 \models \neg\Box(P \to \Box P)$. Continuing in this way, we either obtain an infinite path of distinct points, giving a proper ∞-R-chain, or we obtain a cycle, giving a proper cluster. Both are forbidden in $\mathbf{K4G_o}$-frames. Hence (a) cannot hold and $\exists w \in W, w_0 R w$ and $w \models \neg P; \Box(P \to \Box P)$. That is, the desired w_n exists.

Proof of (Grz) for $\mathcal{C}\mathbf{Grz}$: Every **Grz**-frame is a $\mathbf{K4G_o}$-frame, hence the proof above suffices. ∎

As we saw in Subsection 4.9, proving completeness boils down to proving the following: if X is a finite set of formulae and no $\mathcal{C}\mathbf{L}$-tableau for X is closed then there is an $\mathbf{L}$-model for X on an $\mathbf{L}$-frame $\langle W, R\rangle$.

Lemma 4.14.2. *If there is a closed* $\mathcal{C}\mathbf{L}$*-tableau for* X *then there is a closed* $\mathcal{C}\mathbf{L}$*-tableau for* X *with all nodes in the finite set* $X^*_{\mathcal{C}\mathbf{L}}$*.*

Proof: Obvious from the fact that all rules for $\mathcal{C}\mathbf{L}$ operate with subsets of $X^*_{\mathcal{C}\mathbf{L}}$ only. ∎

Lemma 4.14.3. *For each* $\mathcal{C}\mathbf{L}$*-consistent* X *there is an effective procedure to construct some finite* $\mathcal{C}\mathbf{L}$*-saturated* X^s *with* $X \subseteq X^s \subseteq X^*_{\mathcal{C}\mathbf{L}}$*.*

Theorem 4.14.4 (Completeness). *If* X *is a finite set of formulae and* X *is* $\mathcal{C}\mathbf{L}$*-consistent then there is an* $\mathbf{L}$*-model for* X *on a finite* $\mathbf{L}$*-frame.*

As usual we construct some $\mathcal{C}\mathbf{L}$-saturated w_0 from X with $X \subseteq w_0 \subseteq X^*_{\mathcal{C}\mathbf{L}}$.

Proof for $\mathcal{C}\mathbf{G}$: If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \emptyset\rangle$ is the desired model graph as (i)-(iii) are satisfied. Otherwise, let $Q_1, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$. Create a $\mathcal{C}\mathbf{G}$-saturated Q_i-successor for each Q_i using (θ) and (G) giving the nodes v_i of level one. Repeating this construction on the nodes of level one gives the nodes of level two, and so on for other levels. Consider any sequence $w_i \prec w_{i+1} \prec w_{i+2} \cdots$. Since w_i has a successor, there is some $\neg\Box Q \in w_i$ and $\Box Q \in w_{i+j}$ for all $j \geq 1$ by (G). Thus $w_i \neq w_{i+j}$ for any $j \geq 1$ and each such sequence must terminate since $X^*_{\mathcal{C}\mathbf{G}}$ is finite. Let R be the transitive closure of $\prec$; that is put wRw' if $w \prec w'$ and put wRv if $w \prec w' \prec v$. The resulting tree is a model graph $\langle W_0, R\rangle$ for X which is also a $\mathbf{G}$-frame.

Proof for $\mathcal{C}\mathbf{Grz}$: If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \{(w_0, w_0)\}\rangle$ is the desired model graph as (i)-(iii) are satisfied. Otherwise, let $Q_1, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$ and $\neg Q_i \notin w_0$. Create a $\mathcal{C}\mathbf{Grz}$-saturated Q_i-successor for each Q_i using (θ) and (Grz) giving the nodes v_i of level one, and so on for other levels. Consider any sequence $w_i \prec w_{i+1} \prec w_{i+2} \cdots$. Since w_i has a successor, there is some Q such that $\neg\Box Q \in w_i$, $\neg Q \notin w_i$, and by (Grz), $\Box(Q \to \Box Q) \in w_{i+j}$ for all $j \geq 1$. Suppose $w_{i+j} = w_i$, then $\Box(Q \to \Box Q) \in w_i$ and hence $Q \to \Box Q \in w_i$ by (T). Since $Q \to \Box Q$ is just abbreviation for $\neg(Q \wedge \neg\Box Q)$, we know that $\neg Q \in w_i$ or $\neg\neg\Box Q \in w_i$. We created a successor w_{i+1} for w_i precisely because $\neg Q \notin w_i$ and so the first case is impossible. And if $\neg\neg\Box Q \in w_i$ then $\Box Q \in w_i$ by $(\neg)$, contradicting the $\mathbf{Grz}$-consistency of w_i since $\neg\Box Q \in w_i$ by supposition. Thus each such sequence must terminate (without cycles). Let R be the reflexive and transitive closure of $\prec$ to obtain a model graph $\langle W_0, R\rangle$ for X which is also a $\mathbf{Grz}$-frame.

Proof for $\mathcal{C}\mathbf{K4G_o}$: If no $\neg\Box P$ occurs in w_0 then $\langle\{w_0\}, \{\emptyset\}\rangle$ is the desired model graph as (i)-(iii) are satisfied. Otherwise, let $Q_1, Q_2, \cdots, Q_m$ be all the formulae such that $\neg\Box Q_i \in w_0$. A $\mathcal{C}\mathbf{K4G_o}$-saturated set v is reflexive iff $\Box A \in v$ implies $A \in v$. If v is non-reflexive then there exists some $\Box B \in v$ but $B \notin v$.

If w_0 is reflexive then create a $\mathcal{C}\mathbf{K4G_o}$-saturated Q_i-successor for each $\neg\Box Q_i$ with $\neg Q_i \notin w_0$, otherwise if w_0 is non-reflexive then create a $\mathcal{C}\mathbf{K4G_o}$-saturated Q_i-successor for each $\neg\Box Q_i$, $1 \le i \le m$. This gives the nodes of level one. Continue creating successors in this fashion for these nodes using (θ) and (Grz).

Consider any sequence $w_i \prec w_{i+1} \prec w_{i+2} \cdots$. Since w_i has a successor, there is some $\neg\Box Q \in w_i$ that gives rise to w_{i+1}. Also, $\Box(Q \to \Box Q) \in w_{i+j}$ for all $j \ge 1$.

If w_i is reflexive then $\neg Q \notin w_i$, and yet $\neg Q \in w_{i+1}$ by (Grz); hence $w_i \ne w_{i+1}$. Suppose $w_{i+j} = w_i$, $j \ge 2$. That $j \ge 2$ is crucial! Then $\Box(Q \to \Box Q) \in w_i$ and $Q \to \Box Q \in w_i$ by (Grz). Since $Q \to \Box Q$ is just abbreviation for $\neg(Q \wedge \neg\Box Q)$, we know that $\neg Q \in w_i$ or $\neg\neg\Box Q \in w_i$. Since w_i is reflexive, we created a successor w_{i+1} for w_i precisely because $\neg Q \notin w_i$ and so the first case is impossible. And if $\neg\neg\Box Q \in w_i$ then $\Box Q \in w_i$ by $(\neg)$, contradicting the $\mathbf{K4G_o}$-consistency of w_i since $\neg\Box Q \in w_i$ by supposition.

If w_i is non-reflexive then there is some $\Box B \in w_i$, with $B \notin w_i$, and yet both $\Box B$ and B are in w_{i+j} by (Grz), for all $j \ge 1$; hence $w_i \ne w_{i+j}$, $j \ge 1$.

Thus each such sequence must terminate (without cycles). Let R be the transitive closure of $\prec$ and also put wRw if w is reflexive to obtain a model graph $\langle W_0, R\rangle$ for X which is also a $\mathbf{K4G_o}$-frame.

As Amerbauer [Ame93] points out, this means that $\mathbf{K4G_o}$ is characterised by finite transitive trees of non-proper clusters refuting the conjecture of Goré [Gor92] that $\mathbf{K4G_o}$ is characterised by finite transitive trees of degenerate non-final clusters and simple final clusters.

4.14.1 Bibliographic Remarks and Related Systems

The tableau system $\mathcal{C}\mathbf{G}$ is from Fitting [Fit83] who attributes it to [Boo79], while $\mathcal{C}\mathbf{Grz}$ is from Rautenberg [Rau83]. Rautenberg gives a hint on how to extend these to handle $\mathcal{C}\mathbf{K4G_o}$ but Goré [Gor92] is unable to give an adequate system for $\mathcal{C}\mathbf{K4G_o}$, leaving it as further work. The given $\mathcal{C}\mathbf{K4G_o}$ is due to Martin Amerbauer [Ame93] who following suggestions of Rautenberg and Goré also gives systems for $\mathbf{KG.2}$ and $\mathbf{KGL}$ (which Amerbauer calls $\mathbf{K4.3G}$).

Provability logics have also been studied using Gentzen systems and cut-elimination proofs have been given by Avron [Avr84], Bellin [Bel85], Borga [Bor83], Borga and Gentilini [BG86], Sambin and Valentini [SV80, VS83, SV82], and Valentini [Val83, Val86].

4.15 Monomodal Temporal Logics

In this section, which is based heavily on [Gor94], we give tableau systems for normal modal logics with natural temporal interpretations where $\Box A$ is read as "A is true always in the future" and $\Diamond A$ is read as "A is true some time in the future". All logics are "monomodal" in that the reverse analogues of these operators, namely "always in the past" and "sometime in the past", are not available. That is, the reachability relation R is taken to model the flow of time in a forward direction, and each possible world represents a point in this flow with some point deemed to be "now". We are allowed to look forwards but not backwards. In all cases time is taken to be transitive and the variations between the logics comes about depending on whether we view time as linear or branching; as dense or discrete; and as reflexive or non-reflexive (which is not the same as irreflexive). We explain these notions below.

4.15.1 Reflexive Monomodal Temporal Logics

The logics **S4.3**, **S4.3.1** and **S4Dbr** are all normal extensions of **S4** and are axiomatised by taking the appropriate formulae from Figure 1 as axiom schemas. Their respective axiomatisations are: **S4** is $KT4$; **S4.3** is $KT43$; **S4.3.1** is $KT43Dum$; and **S4Dbr** is $KT4Dbr$.

The Diodorean modal logics **S4.3** and **S4.3.1** have received much attention in the literature because of their interpretation as logics of dense and discrete *linear* time [Bul65]. That is, it can be shown that $\langle \mathcal{I}, \leq \rangle \models A$ iff $\vdash_{\mathbf{S4.3}} A$ where $\mathcal{I}$ is either the set of real numbers or the set of rational numbers and $\leq$ is the usual (reflexive and transitive) ordering on numbers [Gol87, page57]. Consequently, between any two points there is always a third and **S4.3** is said to model **linear dense** time. It can be shown that $\langle \omega, \leq \rangle \models A$ iff $\vdash_{\mathbf{S4.3.1}} A$ where ω is the set of natural numbers [Gol87]. Hence, between any two points there is always a finite number (possibly none) of other points and **S4.3.1** is said to model **linear discrete** time. The formal correspondence between $\langle \mathcal{I}, \leq \rangle$ and **S4.3**-frames, and between $\langle \omega, \leq \rangle$ and **S4.3.1**-frames can be obtained by using a technique known as bulldozing and defining an appropriate mapping called a p-morphism [Gol87] [HC84].

The logics **S4** and **S4Dbr** can be given interpretations as logics of dense and discrete *branching* time. That is, it can be shown that **S4** is also characterised by the class of all reflexive transitive (and possibly infinite) trees [HC84, page 120]. That is, by bulldozing each proper cluster of an **S4**-frame we can obtain an infinite dense sequence so that **S4** is the logic that models branching dense time. The axiomatic system **S4Dbr** is proposed by Zeman [Zem73, page 249] as the temporal logic for branching discrete time, but Zeman and Goré [Gor94] call this logic **S4.14**.

Therefore, the logics **S4**, **S4.3**, **S4.3.1** and **S4Dbr** cover the four possible combinations of discreteness and density paired with linearity and branching.

$$(S4Dbr) \quad \frac{\Box X; \neg\Box P}{\Box X; \Box\neg\Box P \quad | \quad \Box X; \neg P; \Box(P \rightarrow \Box P)}$$

$$(S4.3) \quad \frac{\Box X; \neg\Box\{P_1, \cdots, P_k\}}{\Box X; \neg\Box\overline{Y_1}; \neg P_1 \quad | \quad \cdots \quad | \quad \Box X; \neg\Box\overline{Y_k}; \neg P_k}$$

where $Y = \{P_1, \cdots, P_k\}$ and $\overline{Y_i} = Y \setminus \{P_i\}$

$$(S4.3.1) \quad \frac{U; \Box X; \neg\Box\{Q_1, \cdots, Q_k\}}{S_1 \quad | \quad S_2 \quad | \quad \cdots \quad | \quad S_k \quad | \, S_{k+1} \, | \quad S_{k+2} \quad | \quad \cdots \quad | \quad S_{2k}}$$

where

$Y = \{Q_1, \cdots, Q_k\}$;

$\overline{Y_j} = Y \setminus \{Q_j\}$;

$S_j = U; \Box X; \neg\Box\overline{Y_j}; \Box\neg\Box Q_j$

$S_{k+j} = \Box X; \neg Q_j; \Box(Q_j \rightarrow \Box Q_j); \neg\Box\overline{Y_j}$

for $1 \leq j \leq k$

Fig. 10. Tableau rules $(S4Dbr)$, $(S4.3)$ and $(S4.3.1)$.

Figure 10 on page 53 shows the rules we need to add to $\mathcal{C}$**S4** in order to obtain tableau systems for **S4.3**, **S4.3.1** and **S4Dbr**. The tableau calculi $\mathcal{C}$**S4.3**, $\mathcal{C}$**S4.3.1** and $\mathcal{C}$**S4Dbr** are respectively the calculi for the logics **S4.3**, **S4.3.1** and **S4Dbr** as shown below:

$\mathcal{C}$**L**	Static Rules	Transitional Rules	$X^*_{\mathcal{C}\mathbf{L}}$
$\mathcal{C}$**S4.3**	$\mathcal{C}$**PC**, (T)	$(S4.3)$	$\widetilde{X}$
$\mathcal{C}$**S4.3.1**	$\mathcal{C}$**PC**, (T)	$(S4), (S4.3.1)$	$Sf(\Box(\widetilde{X} \to \Box\widetilde{X}); \Box\widetilde{X})$
$\mathcal{C}$**S4Dbr**	$\mathcal{C}$**PC**, (T)	$(S4), (S4Dbr)$	$Sf(\Box(\widetilde{X} \to \Box\widetilde{X}); \Box\widetilde{X})$

Note that $\mathcal{C}$**S4.3** does not contain the rule $(S4)$ and that $\mathcal{C}$**S4.3.1** does not contain the rule $(S4.3)$ but does contain the rule $(S4)$. Also note that the $(S4.3.1)$ rule contains some static denominators and some transitional denominators.

Lemma 4.15.1. *If there is a closed* $\mathcal{C}$**L** *tableau for the finite set* X *then there is a closed* $\mathcal{C}$**L** *tableau for* X *with all nodes in the finite set* $X^*_{\mathcal{C}\mathbf{L}}$.

Proof: Obvious from the fact that all rules for $\mathcal{C}$**L** operate with subsets of $X^*_{\mathcal{C}\mathbf{L}}$ only. ∎

Lemma 4.15.2. *For each* $\mathcal{C}$**L***-consistent* X *there is an effective procedure to construct some finite* $\mathcal{C}$**L***-saturated* X^s *with* $X \subseteq X^s \subseteq X^*_{\mathcal{C}\mathbf{L}}$.

Proof: As on page 24. ∎

Theorem 4.15.3. *The* $\mathcal{C}$**L** *rules are sound with respect to* **L***-frames.*

Proof: We omit details since the proofs can be found in [Gor94], although note that there, the definition of **L**-frames is slightly different.

The intuition behind the $(S4.3)$ rule is based on a consequence of the characteristic **S4.3** axiom 3. Adding 3 to **S4** gives a weakly-connected R for **S4.3** so that eventualities can be weakly-ordered. If there are k eventualities, one of them must be fulfilled first. The $(S4.3)$ rule can be seen as a disjunctive choice between which one of the k eventualities is fulfilled first and an appropriate "jump" to the corresponding world.

The intuition behind the $(S4.3.1)$ rule is that each eventuality is either "eternal", because it is fulfilled an infinite number of times in the sequence of worlds that constitute an **S4.3.1**-model, or "non-eternal". If the eventuality $\neg\Box P$ is "eternal" then it can be stashed away (statically) as $\Box\neg\Box P$ and ignored until "later". Otherwise it must be dealt with immediately by fulfilling it via a transition. But there may be many such eventualities and since R is weakly-connected, they must be ordered.

Theorem 4.15.4. *If* X *is a finite set of formulae and* X *is* $\mathcal{C}$**L***-consistent then there is an* **L***-model for* X *on a finite* **L***-frame* $\langle W, R\rangle$.

Again we omit details since they can be found in [Gor94] but note that there we used **S4.14** for **S4Dbr**. However, the proof for $\mathcal{C}$**S4.3** is reproduced below to give an idea of how to handle linearity.

Proof sketch for $\mathcal{C}$S4.3: The completeness proof of $\mathcal{C}$**S4.3** is similar to the completeness proof for $\mathcal{C}$**S4**. The differences are that only *one* sequence is constructed, and that in doing so, the $(S4.3)$ rule is used instead of the $(S4)$ rule. Note that the $(S4.3)$ rule guarantees only that *at least one* eventuality gives a $\mathcal{C}$**S4.3**-consistent successor whereas $(S4)$ guarantees that *every* eventuality gives a $\mathcal{C}$**S4**-consistent successor. And this crucial difference is why thinning seems essential. The basic idea is to follow one sequence, always attempting to choose a successor new to the sequence. Sooner or later, no such successor will be possible giving a sequence $S = w_0 \prec w_1 \prec w_2 \prec \cdots \prec w_m \prec w_{m+1} \prec \cdots \prec w_{n-1} \prec w_m$ containing a cycle $C = w_m \prec w_{m+1} \prec \cdots \prec w_{n-1} \prec w_m$ which we write pictorially as

$$S = w_0 \prec w_1 \prec w_2 \prec \cdots \prec \overline{w_m \prec w_{m+1} \cdots \prec w_{n-1}}.$$

The cycle C fulfills at least one of the eventualities in w_{n-1}, namely the $\neg\Box Q$ that gave the duplicated Q-successor w_m of w_{n-1}. But C may not fulfill *all* the eventualities in w_{n-1}.

Let $Y = \{P \mid \neg\Box P \in w_{n-1} \text{ and } \neg P \notin w_j,\ m \leq j \leq n-1\}$, so that $\neg\Box Y$ is the set of eventualities in w_{n-1} that remain unfulfilled by C. Let $w' = \{P \mid \Box P \in w_{n-1}\}$. Since $(\Box w'; \neg\Box Y) \subseteq w_{n-1}$ is $\mathcal{C}$**S4.3**-consistent by (θ), so is at least *one* of

$$X_j = \Box w' \cup \{\neg P_j\} \cup \neg\Box\overline{Y_j}, \text{ for } j = 1, \cdots, k$$

by $(S4.3)$. As before, choose the $\mathcal{C}$**S4.3**-consistent X_i that gives a **S4.3**-saturated P_i-successor for w_{n-1} which is new to S to sprout a continuation of the sequence, thus escaping out of the cycle. If no such new successor is possible then choose the successor $w_{m'}$ that appears earliest in S. This successor *must* precede w_m, as otherwise, C would already fulfill the eventuality that gives this successor. That is, we can extend C by putting $w_{n-1} \prec w_{m'}$. Recomputing Y using m' instead of m must decrease the size of Y since w_{n-1} has remained fixed. Repeating this procedure will eventually lead either to an empty Y or to a new successor. In the latter case we carry on the construction of S. In the former case we form a final cycle that fulfills all the eventualities of w_{n-1} and stop.

Sooner or later we must run out of new successors since $X^*_{\mathcal{C}\mathbf{S4.3}}$ is finite and so only the former case is available to us. Let R be the reflexive and transitive closure of $\prec$ so that the overlapping clusters of $\prec$ become maximal disjoint clusters of R. It should be clear that $\langle W, R\rangle$ is a linear order of maximal, disjoint clusters that satisfies properties (i)-(iii), and hence that $\langle W, R\rangle$ is a model-graph for X.

Note that thinning seems essential. That is, in computing Y, we *have* to exclude the eventualities that are already fulfilled by the current cycle

C in order to escape out of the cycle that they cause. We return to this point later. ∎

4.15.2 Non-reflexive Monomodal Temporal Logics

The logics **S4.3** and **S4.3.1** respectively have counterparts called **K4DLX** and **K4DLZ** [Gol87] that omit reflexivity where the new axiom schema are D, L, X, Z, and Zbr; see Figure 1 on page 5.

It is known that $\langle \mathcal{I}, < \rangle \models A$ iff $\vdash_{\mathbf{K4DLX}} A$ and $\langle \omega, < \rangle \models A$ iff $\vdash_{\mathbf{K4DLZ}} A$ where $\mathcal{I}$ is either the set of real numbers or the set of rational numbers and ω is the set of natural numbers [Gol87]. Hence these logics model transitive non-reflexive linear dense, and transitive non-reflexive linear discrete time respectively. I am not aware of a proof of completeness for the non-reflexive counterpart of **S4Dbr** but it seems reasonable to conjecture that **K4DZbr** is this counterpart.

The simplest way to handle the seriality axiom D is to use the static (D) rule of Rautenberg even though it breaks the subformula property. But (D) and $(K4Zbr)$ can conspire to give an infinite sequence of building up operations,[1] so we use the transitional $(KD4)$ and $(KD4L)$ rules instead; see Figure 11.

Another minor complication is the need for an explicit tableau rule to capture density (no consecutive degenerate clusters, see [Gol87]) for **K4DLX** but this is handled by the transitional rule $(K4DX)$, which is sound for **K4DLX**-frames.

The non-reflexive analogue of the $(S4.3)$ rule becomes very clumsy since it is based on the **K4LX**-theorem:

$$\Diamond P \wedge \Diamond Q \rightarrow \Diamond(P \wedge \Diamond Q) \vee \Diamond(Q \wedge \Diamond P) \vee \Diamond(P \vee Q)$$

and it is easier to use the rule $(K4L)$ which makes explicit use of subsets. The $(K4L)$ rule is similar to a rule given by Valentini [Val86]. By using rules from Figure 11 it is possible to obtain cut-free tableau calculi possessing the analytic superformula property for these logics as:

$\mathcal{C}\mathbf{L}$	Static Rules	Transitional Rules	$X^*_{\mathcal{C}\mathbf{L}}$
$\mathcal{C}$**K4DLX**	$\mathcal{C}$**PC**	$(K4DX)$, $(K4L)$	$\widetilde{X}$
$\mathcal{C}$**K4DLZ**	$\mathcal{C}$**PC**	$(K4D)$, $(K4LZ)$	$Sf\neg Sf\Box\widetilde{X}$
$\mathcal{C}$**K4DZbr**	$\mathcal{C}$**PC**	$(K4D)$, $(K4Zbr)$	$Sf\neg Sf\Box\widetilde{X}$

First of all note that $(K4DX)$ is a transitional rule, not a static rule.

Now, it may appear as if the explicit subset notation would allow us to dispense with (θ) but this is not so. For (θ) allows us to *ignore* certain eventualities, whereas $(K4L)$ and $(K4LZ)$ only allow us to *delay* them.

[1] I missed this aspect in [Gor94]

$$(K4D)\quad \frac{\Box X; \neg\Box P}{X; \Box X; \neg P} \text{ where } \{\neg\Box P, \neg P\} \text{ may be empty}$$

$$(K4DX)\quad \frac{\Box X; \neg\Box Y}{X; \Box X; \neg\Box Y} \text{ where } \neg\Box Y \text{ may be empty}$$

$$(K4Zbr)\quad \frac{\Box X; \neg\Box P}{X; \Box X; \Box\neg\Box P \quad | \quad X; \Box X; \neg P; \Box P}$$

$$(K4L)\quad \frac{\Box X; \neg\Box\{P_1, \cdots, P_k\}}{S_1 \mid S_2 \mid \cdots \mid S_m}$$

where $m = 2^k - 1, \quad 1 \leq i \leq m$;
$Y^1, \cdots, Y^m$ is an enumeration of the non-empty subsets of Y;
$\overline{Y^i} = Y \setminus Y^i$
$S_i = (X; \Box X; \neg\Box\overline{Y^i}; \neg Y^i)$

$$(K4LZ)\quad \frac{U; \Box X; \neg\Box\{Q_1, \cdots, Q_k\}}{S_1 \mid S_2 \mid \cdots \mid S_k \mid S_{k+1} \mid S_{k+2} \mid \cdots \mid S_{k+m}}$$

where :

$Y = \{Q_1, \cdots, Q_k\}; m = 2^k - 1;$

$Y^1, \cdots, Y^m$ is an enumeration of the non-empty subsets of Y;

$\overline{Y_j} = Y \setminus \{Q_j\}$ for $1 \leq j \leq k$;

$\overline{Y^i} = Y \setminus Y^i$ for $1 \leq i \leq m$;

$S_j = U; \Box X; \neg\Box\overline{Y_j}; \Box\neg\Box Q_j$ for $1 \leq j \leq k$;

$S_{k+i} = X; \Box X; \neg Y^i; \Box Y^i; \neg\Box\overline{Y^i}$ for $1 \leq i \leq m$

Fig. 11. Tableau rules for non-reflexive Diodorean logics

Thus using the reflexive analogues of these rules for **S4.3** and **S4.3.1** does not help to eliminate (θ).

The Saturation Lemma (Lemma 4.9.1 on page 24) will go through as for the other logics since the tableau systems have the analytic superformula property.

Theorem 4.15.5. *The $\mathcal{C}$**L** rules are sound with respect to **L**-frames.*

Proof: We omit details since the proofs are similar to the ones for the reflexive temporal logics and are not difficult.

Theorem 4.15.6. *If X is a finite set of formulae and X is $\mathcal{C}$**L**-consistent then there is an **L**-model for X on a finite **L**-frame $\langle W, R\rangle$.*

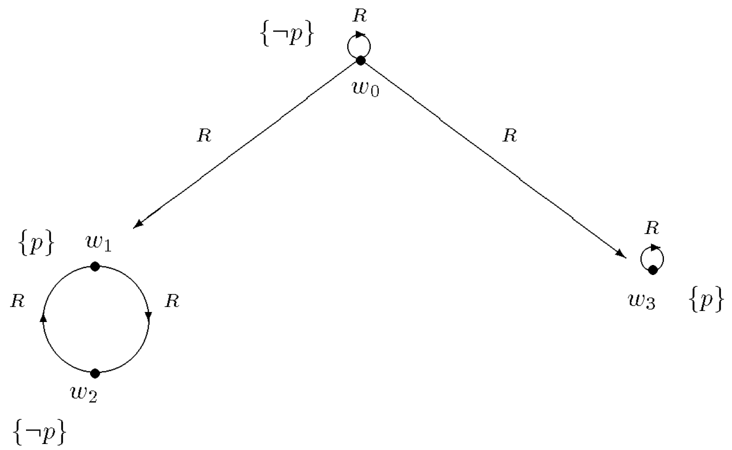
Again we omit details since they are similar to the proofs given in [Gor94] but note that *there* we used the name Z_{14} for the axiom we here dub Zbr. However, the proof for $\mathcal{C}$**K4DLX** is reproduced below to give an idea of how to handle the density requirement.

Proof sketch for $\mathcal{C}$K4DLX: The construction of the model graph is similar to the construction for $\mathcal{C}$**S4.3** except that we now know that every eventuality gives rise to two $\mathcal{C}$**K4DLX**-consistent successors; one from $(K4DX)$ and at least one from $(K4L)$ (and (θ)). We again construct just one sequence but with the following twist.

A $\mathcal{C}$**K4DLX**-saturated set v is reflexive iff $\Box A \in v$ implies $A \in v$. If v is non-reflexive then there exists some $\Box Q \in v$ but $Q \notin v$. If v is non-reflexive then create a successor v_1 for v using $(K4DX)$. If v_1 is non-reflexive then create a successor v_2 for v_1 using $(K4DX)$. Repeating this procedure must eventually give a $(K4DX)$-successor v_n that *is* reflexive. Note that $v \subseteq v_1 \subseteq v_2 \subseteq \cdots \subseteq v_n$ hence the sole purpose of v_n is to carry v *and* be reflexive; thus it need not fulfill any eventualities. Now discard $v_1, v_2, \cdots, v_{n-1}$ and put $v \prec v_n$.

So in the general $\mathcal{C}$**K4DLX** construction, if we are constructing a successor for w and w is reflexive then create a possibly non-reflexive $(K4L)$-successor, else create a reflexive $(K4DX)$-successor (like v_n) as shown above. In either case the sequence produced using $\prec$ satisfies the following criterion: there are no consecutive non-reflexive sets in the sequence.

Once again, this procedure may produce a cycle, and we may need thinning to escape from the cycle if it does not fulfill all its own eventualities as in the case for $\mathcal{C}$**S4.3**. Nevertheless, eventually we will produce a sequence, possibly containing cycles, that fulfills all its eventualities, and furthermore that has no consecutive non-reflexive worlds in the sequence. Let R be the transitive closure of $\prec$ but also put wRw if w is reflexive. The resulting model is a finite reflexive and transitive linear sequence of R-clusters with no consecutive degenerate R-clusters. The density condition is met because if we have w_1Rw_2 then one of them must be reflexive, as otherwise they would form two consecutive degenerate R-clusters. Hence between any w_1



Dum can be written as: $\Box(\neg p \to \Diamond(p \wedge \Diamond\neg p)) \wedge \Diamond\Box p \to p$;

$w_0 \models \Diamond\Box p$ because $w_3 \models \Box p$;

$w_0 \models \neg p \to \Diamond(p \wedge \Diamond\neg p)$ because of w_1 and w_2;

$w_0 \models \Box(\neg p \to \Diamond(p \wedge \Diamond\neg p))$

but $w_0 \not\models p$.

Fig. 12. A finite reflexive-and-transitive model in which all but the final clusters are simple in which *Dum* is false at w_0.

and w_2 we can always put a third world w which is a copy of the one that is reflexive.

The observation that we can detect reflexive worlds is due to Martin Amerbauer [Ame93].

4.15.3 A Note on S4Dbr

In a chapter on modal logic by Segerberg and Bull [BS84, page 51], it is claimed that the logic **S4Dum** "is characterised by the finite reflexive-and-transitive frames in which all but the final clusters are simple". We show that this second claim is not correct by giving a finite reflexive-and-transitive model in which all but the final clusters are simple, but in which *Dum* is false. The model is pictured in Figure 12.

The explanation rests on the fact that $\Box(\Box(P \to \Box P) \to P)$ can be written as $\Box(\neg P \to \Diamond(P \wedge \Diamond\neg P))$. Thus *Dum* can be written as: $\Box(\neg P \to \Diamond(\neg P \wedge \Diamond P)) \wedge \Diamond\Box P \to P$.

This is just as well because we have just shown that **S4Dbr** characterises this class and *Dum* and *Dbr* are different. But note that the extra

$\Box$ modality in Dbr is exactly what is needed since, in the counter-example of Figure 12, $w_0 \not\models \Box\Diamond\Box p$. That is, the counter-example does not falsify Dbr because the extra modality handles the branching inherent in **S4Dbr**-models which is absent in **S4.3.1**-models.

4.15.4 Related Work and Extensions

Zeman [Zem73] appears to have been the first to give a tableau system for **S4.3** but he is unable to extract the corresponding cut-free sequent system [Zem73, page 232]. Shimura [Shi91] has given a syntactic proof of cut-elimination for the corresponding sequent system for **S4.3**, whereas we give a semantic proof. Apparently, Serebriannikov has also obtained this system for **S4.3** but I have been unable to trace this paper. Rautenberg [Rau83] refers to "a simple tableau" system for **S4.3** but does not give details since his main interest is in proving interpolation, and **S4.3** lacks interpolation. In subsequent personal communications I have been unable to ascertain the **S4.3** system to which Rautenberg refers [Rau90]. Bull [Bul85] states that *"Zeman's* **Modal Logic** *(XLII 581), gives tableau systems for* **S4.3** *and* **D** *in its Chapter 15, ... ".* The **D** mentioned by Bull is **S4.3.1** but Zeman [Zem73, page 245] merely shows that his tableau procedure for **S4.3** goes into unavoidable cycles when attempting to prove Dum. Zeman does not investigate remedies and consequently does *not* give a tableau system for **S4.3.1**. In fact, Bull [Bul65] mentions that Kripke used semantic tableau for **S4.3.1**, in 1963, but he gives no reference and subsequent texts that use semantic tableau do not mention this work [Zem73]. Presumably Kripke would have used tableaux where an explicit auxiliary relation is used to mimic the desired properties (like linearity) of R as is done in the semantic diagrams of Hughes and Cresswell [HC68, page 290]. Note that no such explicit representation of R is required in our systems where the desired properties of R are obtained by appropriate tableau rules. I know of no other (cut-free) sequent or tableau systems for the logics **S4.3.1** and **S4Dbr** or their non-reflexive counterparts **K4DLZ** and **K4DLZbr**.

Finally, these techniques extend easily to give a cut-free tableau system for **S4.3Grz** = **KGrz.3** [vB78] which is axiomatised as $KGrz.3$ where Grz is the Grzegorczyk axiom schema Grz: $\Box(\Box(A \to \Box A) \to A) \to A$. This logic is characterised by finite linear sequences of simple clusters but note that Shimura [Shi91] has already given a sequent system for this logic, and it is easy to turn his system into a tableau system.

The non-reflexive counterpart of **S4.3Grz** is **KLG** (sometimes called **G.3** or $\mathbf{GL_{lin}}$ or **K4.3W**) where L is as above and G is the Gödel-Löb axiom $\Box(\Box A \to A) \to \Box A$. Rautenberg [Rau83] shows that **KG** is characterised by the class of finite transitive trees of irreflexive worlds. Thus **KLG** is characterised by finite linear sequences of irreflexive worlds, but note that Valentini [Val86] has already given a cut-free sequent system for this logic.

4.16 Eliminating Thinning

The structural rule (θ) corresponds to the sequent rule of weakening which explicitly enforces monotonicity; see page 18. From a theorem proving perspective, (θ) introduces a form of nondeterminism into each $\mathcal{C}\mathbf{L}$ since we have to guess which formula are really necessary for a proof. It is therefore desirable to eliminate (θ). There are two places where we resort to applications of (θ) in our completeness proofs. We consider each in turn.

The main applications of (θ) in our completeness proofs are the ones used to eliminate the formulae that do not match elements of the numerator, prior to an application of a transitional rule; see page 19. These applications of (θ) can be eliminated by building thinning in a deterministic way into the transitional rules. For example, we can change the $(S5)$ rule shown below left to the $(S5\theta)$ rule shown below right:

$$(S5)\ \frac{\Box X; \neg\Box Y; \neg\Box P}{\Box X; \neg\Box Y; \neg\Box P; \neg P} \qquad\qquad (S5\theta)\ \frac{X; \neg\Box P}{X'; \neg P}$$

where $X' = \{\Box A : \Box A \in X\} \cup \{\neg\Box B : \neg\Box B \in X\} \cup \{\neg\Box P\}$; see Fitting [Fit83]. The new transitional rule $(S5\theta)$ does the work of (θ) and $(S5)$. The crucial point is that we can specify X' exactly because we know exactly which formulae to throw away: namely, the ones that do not match the numerator of $(S5)$.

In some completeness proofs we also avoid creating a successor for $\neg\Box Q \in w$ if $\neg Q \in w$, thus pre-empting the reflexivity of R. This is *not* an application of (θ) when the transitional rule in question is non-branching like $(S4)$, because a consistent successor also exists for these eventualities, it is just that we are not interested in these successors.

However, (θ) appears essential for some of the branching transitional rules like $(S4.3)$, $(K4L)$ and $(S4.3.1)$ etc. even though we can also build thinning into these rules as well. For in the counter-model construction for $\mathcal{C}\mathbf{S4.3}$, we may reach a stage where all $\mathcal{C}\mathbf{S4.3}$-consistent successors already appear in S but no such cycle fulfills all the eventualities of the last node. At this stage it is essential to invoke applications of (θ) on subsets of the eventualities. That is, we must be able to *ignore* some of the eventualities in w_{n-1} using (θ) and this means that (θ) is now an essential rule of $\mathcal{C}\mathbf{S4.3}$.

The crucial difference between the branching transitional rules like $(S4.3)$ and the non-branching transitional rules like $(S4)$ is that the former guarantee only that *at least one* denominator is consistent, whereas the non-branching rules guarantee that every denominator is consistent (since they only have one denominator). But note that not all branching transitional rules are bad, for the $(S4Dbr)$ rule also branches, but the completeness proof (see [Gor94]) goes through without recourse to (θ) because we can make a second pass of the initial model graph to obtain the desired frame.

It may be possible to eliminate thinning by using cleverer completeness proofs. For example, an alternate proof for $\mathcal{C}$**S4.3** may be possible by considering all ($S4.3$)-successors for every node, giving a tree of nondegenerate clusters, and then showing that any two worlds in this tree can be ordered as is done by Hughes and Cresswell [HC84, page 30-31]. Note however that this seems to require a cut rule since Hughes and Cresswell use maximal consistent sets rather than saturated sets as we do.

Clearly the intuitions inherent in our semantic methods are no longer enough to prove that weakening is eliminable. We have obtained a syntactic proof of elimination of weakening in the sequent system containing the sequent analogues of the modified tableau rule ($S4.3\theta$), but this is beyond the scope of this chapter.

4.17 Eliminating Contraction

As we have seen, contraction is built into our tableau rules by the ability to carry a copy of the principal formula into the denominator. But we believe it can be limited to the explicit contractions we have shown in our modal rules. Unfortunately, our set-based rules and completeness proofs are not sophisticated enough to *prove* this since (the saturation) Lemma 4.9.1 on page 24 requires that we copy the principal formula into the denominator. It is possible to rework all of our work using multisets instead of sets, but the proofs become very messy. For a more detailed study of contraction in modal tableau systems see the work of Hudelmaier [Hud94] and Miglioli *et al* [MMO95].

4.18 Finite L-frames

In all our completeness proofs we construct *finite* model graphs, hence our logics are also characterised by the *finite* frames shown in Figure 13. The frames in Figure 13 are all based on trees of clusters or trees of worlds where we assume that clusters immediately imply transitivity. Consequently, each logic has the finite model property, and is decidable. These finer-grained results are not always obtainable when using other tableau methods.

4.19 Admissibility of Cut and Gentzen Systems

The cut rule is sound with respect to all our **L**-frames and each $\mathcal{C}$**L** is sound and complete with respect to the appropriate **L**-frames. Thus, putting (ρ) equal to (*cut*) in Lemma 4.6.1 (page 20) gives:

Theorem 4.19.1. *The rule (cut) is admissible in each* $\mathcal{C}$**L**.

Tableau systems are (upside down) cousins of proof systems called Gentzen systems or sequent systems; see Fitting [Fit83]. For example, the Gentzen system $\mathcal{G}K$ shown in Figure 14 is a proof system for modal

L	finite-**L**-frames
K	finite intransitive tree of irreflexive worlds
T	finite intransitive tree of reflexive worlds
D	finite intransitive tree of worlds with reflexive final worlds
K4	finite tree of finite clusters
KDB	a single reflexive world; or a finite intransitive and symmetric tree of at least two worlds
K4D	finite tree of finite clusters with finite nondegenerate final clusters
K45	a single finite cluster; or a degenerate cluster followed by a finite nondegenerate cluster
K45D	a single finite nondegenerate cluster; or a degenerate cluster followed by a finite nondegenerate cluster
S4	finite tree of finite nondegenerate clusters
KB4	single finite cluster
S5	single finite nondegenerate cluster
B	finite symmetric tree of reflexive worlds
S4R **S4.3Zem**	a single finite nondegenerate cluster; or a simple cluster followed by a finite nondegenerate cluster
S4F	a sequence of at most two finite nondegenerate clusters
S4.2	a finite tree of finite nondegenerate clusters with one last cluster
S4.3	finite sequence of finite nondegenerate clusters
S4.3.1	finite sequence of finite nondegenerate clusters with no proper non-final clusters
S4Dbr	finite tree of finite nondegenerate clusters with no proper non-final clusters
K4L	finite sequence of finite clusters
K4DL	finite sequence of finite clusters with a nondegenerate final cluster
K4DLX	finite sequence of finite clusters with a nondegenerate final cluster, and no consecutive degenerate clusters
K4DLZ	finite sequence of degenerate clusters with a final simple cluster
K4DLZbr	finite tree of degenerate clusters with final simple clusters
G	finite transitive tree of irreflexive worlds
Grz **S4Grz** **S4MDum**	finite transitive tree of reflexive worlds
$\mathbf{K4G_o}$	finite transitive tree of worlds
GL	finite transitive sequence of irreflexive worlds

Fig. 13. Definition of finite-**L**-frames.

$$X, P \longrightarrow P, Y \quad (\mathrm{Ax})$$

$$\frac{X, P, Q \longrightarrow Y}{X, P \wedge Q \longrightarrow Y}\ (\wedge \rightarrow) \qquad \frac{X \longrightarrow P, Y \quad X \longrightarrow Q, Y}{X \longrightarrow P \wedge Q, Y}\ (\rightarrow \wedge)$$

$$\frac{X \longrightarrow P, Y}{X, \neg P \longrightarrow Y}\ (\neg \rightarrow) \qquad \frac{X, P \longrightarrow Y}{X \longrightarrow \neg P, Y}\ (\rightarrow \neg)$$

$$\frac{X \longrightarrow Y}{X, U \longrightarrow V, Y}\ (\theta) \qquad \frac{X \longrightarrow P}{\Box X \longrightarrow \Box P}\ (\rightarrow \Box P : K)$$

Fig. 14. Sequent rules for $\mathcal{G}K$

logic **K**. That is, a formula A is valid in all **K**-frames (and hence a theorem of **K**) iff the sequent $\longrightarrow A$ is provable in $\mathcal{G}K$. Each of our tableau rules has a sequent analogue so it is possible to convert each tableau system $\mathcal{C}\mathbf{L}$ into a sequent system $\mathcal{G}L$. Then, $\mathcal{G}L$ is cut-free as long as $\mathcal{C}\mathbf{L}$ does not use (sfc) or $(sfcT)$. By induction it is straightforward to show that the sequent $X \longrightarrow Y$ is provable in $\mathcal{G}L$ iff there is a closed $\mathcal{C}\mathbf{L}$-tableau for $X; \neg Y$.

Our sequent systems do not possess all the elegant properties usually demanded of (Gentzen) sequent systems. For example, not only do some of our systems break the subformula property, but most do not possess separate rules for introducing modalities into the right and left sides of sequents.

Elegant modal sequent systems respecting these ideals of Gentzen have proved elusive although the very recent work of Avron [Avr94], Cerrato [Cer93], Masini [Mas92, Mas91] and Wansing [Wan94] are attempts to redress this dearth. However, some of these methods have their own disadvantages. The systems of Cerrato enjoy the subformula property and separate introduction rules but do not enjoy cut-elimination in general (although the systems for **K** do so). The systems of Masini enjoy cut-elimination and give direct proofs of decidability but (currently) apply only to the logics **K** and **KD**. The systems of Wansing enjoy cut-elimination and clear introduction rules but do not immediately give decision procedures, and cannot handle logics like **S4.3.1** and **S4Dbr** [Kra96]. The hypersequents of Pottinger [Pot83] and Avron [Avr94] seem to retain most of the desired properties since they give cut-free systems with the subformula property for most of the basic modal logics including **S5**. It would be interesting to see if they can be extended to handle the Diodorean or provability logics.

5 Tableau Systems For Multimodal Temporal Logics

In this section we briefly survey tableau systems for multimodal temporal logics with future and past time connectives which have proved useful in Computer Science. The brevity is justified since the survey by Emerson [Eme90] covers tableau methods for these logics. Here we just try to show how these logics and their tableau methods relate to the methods we have seen so far.

In Computer Science the term "temporal logic" is used to describe logics where the frames are discrete in the sense of **S4.3.1**-frames and **S4Dbr**-frames. The term "linear temporal logic" is used when the frames are linear (discrete) sequences and the term "branching temporal logic" is used when the frames are (discrete and) branching. If we wish to refer to the past then we can use a multimodal tense logic where $\blacksquare A$ is read as "A is true at all point in the past" and $\blacklozenge A$ is read as "A is true at some point in the past" [Bur84]. However, certain *binary* modal connectives have proved more useful.

The impetus for studying linear binary modal operators started with the seminal results of Kamp [Kam68]. Kamp showed that linear tense logic equipped with monomodal tense connectives like $\blacksquare$, $\blacklozenge$, $\Diamond$ and $\Box$ are "expressively incomplete" because there are simple properties of linear orders that cannot be expressed using only these connectives together with the usual boolean connectives. One example is the property "A is true now and remains true until B becomes true". Kamp then showed that certain *binary* modal connectives are "expressively complete" in that they capture *any* property expressible in the first-order theory of linear orders; that is, expressible using time point variables like t_1, t_2, the quantifiers $\forall$, $\exists$, the boolean connectives and the predicate $\leq$ familiar from number theory. Wolper then showed that even these connectives could not express all desirable properties of sequences [Wol83]; for example, properties that correspond to regular expressions from automata theory like "A is true in every second state". Wolper introduced extra connectives corresponding to regular expressions but these are beyond the scope of this article; see [Wol83].

5.1 Linear Temporal Logics

5.1.1 Syntax of Linear Temporal Logics

We add the unary modal connectives $\bullet$, $\circ$, $\blacklozenge$ and $\blacksquare$, and the binary modal connectives $\mathcal{U}$, $\mathcal{W}$, $\mathcal{S}$ and $\mathcal{Z}$. Any primitive proposition p is a formula, and if A and B are formulae, then so are: $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(\Box A)$, $(\Diamond A)$, $(\blacksquare A)$, $(\blacklozenge A)$, $(\circ A)$, $(\bullet A)$, $(A\,\mathcal{U} B)$, $(A\,\mathcal{W} B)$, $(A\,\mathcal{S} B)$ and $(A\,\mathcal{Z} B)$.

Intuitively, $\circ A$ means "A is true in the next state", $\bullet A$ means "A is true in the previous state", $A\,\mathcal{U} B$ means "A is true until B becomes true", and $A\,\mathcal{S}B$ means "A has been true since B became true". The others are explained shortly.

5.1.2 Semantics of Linear Temporal Logics

For brevity we concentrate on the linear temporal logic with future connectives only and dub it **PLTL** for propositional linear temporal logic, and follow Goldblatt [Gol87].

A **state sequence** is a pair $\langle S, \sigma\rangle$ where σ is a function from the natural numbers ω onto S enumerating the members of S as an infinite sequence $\sigma_0, \sigma_1, \cdots, \sigma_n \cdots$ (with repetitions when S is finite). A **model** $\mathcal{M} = \langle S, \sigma, V\rangle$ is a state sequence together with a valuation V that maps every primitive proposition onto a subset of S as usual. A model **satisfies** a formula at state σ_i according to:

$$
\begin{array}{lll}
(\mathcal{M}, \sigma_i) \models p & \text{iff} & \sigma_i \in V(p); \\[1ex]
(\mathcal{M}, \sigma_i) \models \neg A & \text{iff} & (\mathcal{M}, \sigma_i) \not\models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models A \wedge B & \text{iff} & (\mathcal{M}, \sigma_i) \models A \text{ and } (\mathcal{M}, \sigma_i) \models B; \\[1ex]
(\mathcal{M}, \sigma_i) \models A \vee B & \text{iff} & (\mathcal{M}, \sigma_i) \models A \text{ or } (\mathcal{M}, \sigma_i) \models B; \\[1ex]
(\mathcal{M}, \sigma_i) \models \bigcirc A & \text{iff} & (\mathcal{M}, \sigma_{i+1}) \models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models \Box A & \text{iff} & \forall j, j \geq i, (\mathcal{M}, \sigma_j) \models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models \Diamond A & \text{iff} & \exists j, j \geq i, (\mathcal{M}, \sigma_j) \models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models A\,\mathcal{U}B & \text{iff} & \exists k, k \geq i, (\mathcal{M}, \sigma_k) \models B \text{ and} \\
 & & \forall j, i \leq j < k, (\mathcal{M}, \sigma_j) \models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models A\,\mathcal{W}B & \text{iff} & (\mathcal{M}, \sigma_i) \models A\,\mathcal{U}B \text{ or } (\mathcal{M}, \sigma_i) \models \Box A.
\end{array}
$$

Intuitively imagine the states $\sigma_0, \sigma_1, \cdots$ to form an infinite sequence where $\sigma_i R \sigma_j$ iff $j = i + 1$ and R is functional. Now if we let $\leq$ be the reflexive and transitive closure of R, then $\Box$ is interpreted using $\leq$ while $\bigcirc$ is interpreted using R. For example, the formula $\bigcirc A$ is true at some state σ_i if A is true at *the* successor state σ_{i+1}. Note that the clause for $A\,\mathcal{U}B$ demands that there is some future state σ_k where B becomes true but does not specify a value for A at this state. A weaker version of $\mathcal{U}$ called $\mathcal{W}$ (for weak until) drops the first demand by allowing for the possibility that there is no future state where B is true as long as $\Box A$ is true at σ_i.

Note that we could also obtain $\Box$ and $\Diamond$ by defining $\Box A$ as $A\,\mathcal{W}\bot$ and $\Diamond A$ as $\top\,\mathcal{U}A$, and still maintain that $\Box A$ is $\neg\Diamond\neg A$.

If we wish to allow reasoning about the past we can also allow backward looking operators. The function σ must now map the set of integers onto S. Some care is needed to ensure the correct behaviour of the definitions below if time does not extend ad infinitum in the past [Fis91]:

$$
\begin{array}{lll}
(\mathcal{M}, \sigma_i) \models \bullet A & \text{iff} & (\mathcal{M}, \sigma_{i-1}) \models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models \blacksquare A & \text{iff} & \forall j, j \leq i, (\mathcal{M}, \sigma_j) \models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models \blacklozenge A & \text{iff} & \exists j, j \leq i, (\mathcal{M}, \sigma_j) \models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models A\,\mathcal{S} B & \text{iff} & \exists k, k \leq i, (\mathcal{M}, \sigma_k) \models B \text{ and} \\
 & & \forall j, k \leq j < i, (\mathcal{M}, \sigma_j) \models A; \\[1ex]
(\mathcal{M}, \sigma_i) \models A\,\mathcal{Z} B & \text{iff} & (\mathcal{M}, \sigma_i) \models A\,\mathcal{S} B \text{ or } (\mathcal{M}, \sigma_i) \models \blacksquare A.
\end{array}
$$

5.1.3 Axiomatisations

A Hilbert system for **PLTL** taken from Goldblatt [Gol87] is given below:

$$
\begin{array}{l}
K : \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B) \\
K_o : \bigcirc(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B) \\
Fun : \bigcirc\neg A \leftrightarrow \neg\bigcirc A \\
Mix : \Box A \rightarrow (A \wedge \bigcirc\Box A) \\
Ind : \Box(A \rightarrow \bigcirc A) \rightarrow (A \rightarrow \Box A) \\
U1 : A\,\mathcal{U} B \rightarrow \Diamond B \\
U2 : A\,\mathcal{U} B \leftrightarrow B \vee (A \wedge \bigcirc(A\,\mathcal{U} B))
\end{array}
$$

We also need the inference rules of universal substitution **US**, modus ponens **MP** and an extended rule of necessitation **RN** viz: if $A \in \mathbf{L}$ then both $\Box A \in \mathbf{L}$ and $\bigcirc A \in \mathbf{L}$; see page 4.

The recursive nature of the Mix and $U2$ axioms gives rise to a fix-point characterisation of these operators which is the key to the tableau procedures for these logics; see [Wol83], [BB87]. Notice also that the axiom Ind encodes an induction principle: if it is always the case that A being true now implies A is true in the next state, then A being true now implies A is true always in the future. It is this property that makes Gentzen systems for these logics difficult to obtain; see Section 5.1.6.

5.1.4 Finite Model Property, Decidability and Complexity

Wolper [Wol83] shows that although our models are infinite state sequences, linear temporal logic is also characterised by a class of finite frames. In fact, it is characterised by our finite-**S4.3.1**-frames; see [Gol87]. A tableau procedure is given by Wolper where he also shows that the problem of deciding satisfiability in **PLTL** is PSPACE-complete. Further complexity results for linear and branching time logics can be found in [ES84, SC85]. Decidability and incompleteness results for first order linear temporal logics have been studied by Merz [Mer92].

5.1.5 Tableau Systems

Tableau systems for the fragment of linear temporal logic containing only future connectives have been studied by Wolper. He gives a tableau-based decision procedure for this logic, and extensions involving regular operators; see [Wol83, Wol85].

The linear temporal logic including both future and past modalities has been extensively studied by Gough [Gou84]. Gough uses (the appropriately defined analogues of downward saturated) Hintikka sets to build a model graph for a given formula of this logic. A second phase then prunes nodes from this model graph to check that all eventualities can be fulfilled on a linear sequence. If this is not possible then the graph is pruned by removing the nodes that contain unfulfillable eventualities. If the initial node is removed by this pruning procedure then the initial formula is unsatisfiable on a linear model hence its negation is a theorem of this logic. The procedure has been automated and the resulting prover called dp is available by anonymous ftp from Graham Gough (gdg@cs.man.ac.uk) at the University of Manchester, England.

A system for temporal logic has also been implemented in the MGTP theorem prover by Koshimura and Hasegawa [KH94].

The (informal) gist of any tableau procedure for linear temporal logics involving next-time modalities is to use the fix-point nature of the modalities to create a cyclic graph of (state) nodes. This graph is then pruned by deleting nodes that contain unfulfillable eventualities. For example, the following logical equivalences hold in linear temporal logic:

$$\begin{array}{rl} (A\,\mathcal{U}B) \equiv (B \vee \bigcirc(A\,\mathcal{U}B)) & \Diamond B \equiv (\top\,\mathcal{U}B) \\ (A\,\mathcal{W}B) \equiv (A\,\mathcal{U}B) \vee \Box A & \Box A \equiv (A\,\mathcal{W}\bot) \end{array}$$

Suppose we are given an initial node n node containing a set of formulae X. For every formula in n that is an instance of the left hand of the above equivalences, we can add the appropriate instance of the right hand side formula and mark the left hand instance as "processed". We can use the usual boolean rules for $\neg\neg$ and $\wedge$ to saturate this node by adding the appropriate subformulae to node n, again marking all parent formulae as "processed". For $\vee$ we put one disjunct in n and create a copy of the old n containing the other disjunct giving a branch in the tableau. Repeating this process on the new formulae means that n contains "processed" formulae and unprocessed formulae. But all unprocessed formulae begin with $\bigcirc$ since these are the only formulae not touched by the above procedure. That is, all unprocessed formulae are in outermost-$\bigcirc$-form. For each node x we then create a successor node y and fill it with $\{A : \bigcirc A \in x\}$. Repeating this procedure on such successors produces a graph because the number of different formulae that can be generated from this process is finite, hence some nodes reappear. Note that we now allow arbitrary cycles whereas in the completeness proofs of Section 4 we confined cycles to nodes on the same branch. Some of these nodes contain eventualities like $\Diamond B$ or $A\,\mathcal{U}B$ since each of these demands the existence of some node that fulfills B. Now we make a second pass and delete nodes that contain both P and $\neg P$ for some formula; delete any node s whose eventualities cannot *all* be jointly fulfilled by some linear path through the graph beginning at s; and delete any nodes without successors. If the initial node ever gets deleted by this procedure then it can be shown that the initial set of formulae cannot be satisfied on a linear discrete model [Wol83, Wol85]. Otherwise there will be a linear sequence of nodes that satisfies all the formulae in the initial node, thus demonstrating a linear discrete model for X.

5.1.6 Gentzen Systems

Gentzen systems for temporal logics have been given by various authors but almost all require either a cut rule or an infinitary rule for completeness [Kaw87, Kaw88]. The exceptions appear to be the work of Gudzhinskas [EG82] and Pliuskevicius [Pli91] but these articles are extremely difficult to read.

5.2 Branching Temporal Logics

Just as **S4Dbr** and **S4.3.1** are branching and linear respectively, there are branching analogues of the linear temporal logics we have seen using $\bigcirc$, $\mathcal{U}$ and even $\mathcal{S}$. We briefly cover the syntax and semantics of one of the most powerful of these branching time logics called $\mathbf{CTL}^*$, and point to the abundant literature for tableau methods for these logics.

5.2.1 Syntax of Branching Temporal Logics

We again concentrate on the future fragment only and follow Emerson and Srinivasan [ES88] using new modal connectives E and X in addition to $\mathcal{U}$. The syntax of branching time logics is given in terms of "state" formulae and "path" formulae where "state" formulae are true or false at some state (world) and where "path" formulae are true or false of (rather than on) a linear sequence of states (worlds). More formally:

any atomic formula p is a *state* formula;

if P and Q are *state* formulae then so are $P \wedge Q$ and $\neg P$;

if P is a *path* formula then EP is a *state* formula;

any *state* formula P is also a *path* formula ;

if P and Q are *path* formulae then so are $P \wedge Q$ and $\neg P$;
if P and Q are *path* formulae then so are XP and $(P\,\mathcal{U}Q)$;

The other boolean connectives are introduced in the usual way while AP abbreviates $\neg E \neg P$, and FP abbreviates $\top \mathcal{U} P$, and GP abbreviates $\neg F \neg P$. Note the absence of $\bigcirc$, $\bullet$, $\Box$, $\Diamond$, $\blacksquare$, and $\blacklozenge$.

5.2.2 Semantics of Branching Temporal Logics

The semantics of **CTL*** are again in terms of a Kripke structure $\mathcal{M} = \langle S, R, L\rangle$ where S is a non-empty set of states or worlds; R is a binary relation on S such that each state has at least one successor; and L is a function which assigns to each state a set of atomic propositions (those that are intended to be true at that state). Note that L is a slight variation on our usual V since the latter assigns atomic propositions to sets of worlds, but the two are equivalent in our classical two-valued setting.

A **fullpath** $x = s_0, s_1, s_2, \ldots$ in $\mathcal{M}$ is an infinite sequence of states such that $s_i R s_{i+1}$ for each i, $i \geq 0$. By $(\mathcal{M}, s) \models P$ and $(\mathcal{M}, x) \models P$ we mean that the state formula P is true at state s in model $\mathcal{M}$, and the path formula P is true of the path x in model $\mathcal{M}$, respectively. If $\mathcal{M}$ is understood then we just write $s \models P$ or $x \models P$. The formal definition of $\models$ is as below where s is an arbitrary state of some $\mathcal{M}$, where $x = s_0, s_1, s_2, \ldots$ is a fullpath in $\mathcal{M}$, and where x^i denotes the suffix fullpath $s_i, s_{i+1}, s_{i+2} \ldots$ of x:

$s \models p$ iff $p \in L(s)$;
$s \models P \wedge Q$ iff $s \models P$ and $s \models Q$;
$s \models \neg P$ iff $s \not\models P$;
$s \models EP$ iff for some fullpath y starting at s, $y \models P$;
$x \models P$ iff $s_0 \models P$ for any state formula P;
$x \models P \wedge Q$ iff $x \models P$ and $x \models Q$;
$x \models \neg P$ iff $x \not\models P$;
$x \models XP$ iff $x^1 \models P$;
$x \models P\,\mathcal{U}Q$ iff $\exists i \geq 0$, $x^i \models Q$ and $\forall j, 0 \leq j < i$, $x^i \models P$.

These definitions are enough to give a semantics for the modalities obtained via definitions: AP is true at state s if P is true of all paths beginning at s; FP is true of a path x if P is true of some suffix fullpath x^i $(i \geq 0)$ of x; and GP is true of a path x if P is true of all suffix fullpaths x^i, $(i \geq 0)$.

The notions of satisfiability and validity are the same as before for state formulae. A path formula P is **satisfiable** if there is some model $\mathcal{M}$ containing some path x such that $x \models P$, and is **valid** if for every model $\mathcal{M}$ and every fullpath path x in $\mathcal{M}$ we have $x \models P$.

As Emerson and Srinivasan note, a menagerie of branching time temporal logics can be obtained by restricting or extending these definitions

[ES88].

Note that path formulae cannot be evaluated at states since there are no clauses in the definition of $\models$ for evaluating XP or $P\mathcal{U}Q$ at a state. But a state formula P can be evaluated on a fullpath simply by checking if the first state of the fullpath satisfies P. Hence, if P is a state formula, then formula XP cannot be evaluated at some state s, it must be evaluated with respect to a path x. Once a path is chosen however, it is just the same as the linear time formula $\bigcirc P$ since XP is true on path x if the second state of the path satisfies (state formula) P. Similarly, if P is a state formula, then FP and GP are just $\Diamond P$ and $\Box P$, except that they are evaluated over a *linear* sequence. But note that **CTL*** is strictly more expressive than **PLTL**.

5.2.3 Tableau Systems

The logic **CTL*** is known to have the finite model property, in fact, it is characterised by finite-**S4Dbr**-frames, but once again, note the presence of the extra modalities. Emerson and Srinivasan [ES88] compare the expressiveness of various such branching time logics. Tableau methods for branching time logics can be found in Emerson [Eme85]. Once again, these tableau methods are based on the appropriate analogues of Hintikka-structures (see [EH85]) and use the following logical equivalences to expand formulae that match the left hand sides into an "outermost-EX-normal" or "outermost-AX-normal" form [ES88]:

$$
\begin{array}{ll}
E(P \vee Q) \equiv EP \vee EQ & A(P \wedge Q) \equiv AP \wedge AQ \\
EGP \equiv P \wedge EXEGP & AGP \equiv P \wedge AXAGP \\
EFP \equiv P \vee EXEFP & AFP \equiv P \vee AXAFP \\
E(P\mathcal{U}Q) \equiv Q \vee (P \wedge EXE(P\mathcal{U}Q)) & A(P\mathcal{U}Q) \equiv Q \vee (P \wedge AXA(P\mathcal{U}Q))
\end{array}
$$

There are no clauses to "expand" formulae beginning with EX or AX in some given set w. Each formula of the form EXP_i gives us reason to create a successor state w_i containing P_i just as $\neg\Box P \in w$ gave rise to a successor v containing $\neg P$ in the modal tableau completeness proofs of Section 4. Now any formula of the form $AXQ \in w$ allows us to put Q into each next state since XQ must be true of all paths that begin at w.

Again the procedure gives a cyclic graph since only a finite number of different sets can be built in this manner and some set reappears. Again, we form arbitrary graphs, not cyclic trees. And once again, a second phase prunes nodes that are inconsistent, that contain unfulfillable eventualities, or have no successor.

Note that all these tableau methods break the subformula property in a weak way since they introduce superformulae of the form $\bigcirc P$ or AXP or EXP where P is built from subformulae of the initial set. But we never apply an expansion rule to these superformulae, thus, EX and AXP

act like "wrappers" to keep this building up procedure in check, just as $\bigcirc$ acted as a wrapper for the **PLTL** procedure. These "wrappers" are removed by creating successor state(s) and filling these with the "unwrapped" (sub)formulae.

5.2.4 Gentzen Systems

Once we start to use graphs rather than trees, the connection with Gentzen systems becomes very tenuous. Gentzen systems for some branching time logics (without A and E) have been studied by Paech [Pae88]. Unfortunately, these systems require a (partly hidden) cut rule which means that they are not the proof-theoretic analogues of the tableau procedures mentioned above.

5.3 Bibliographic Remarks and Related Systems

The complexity of the decision problem for branching time logics has been studied by Emerson and Sistla [ES83]. Temporal logics are known to be related to Büchi automata [VW86] and so their decision problems can be studied from an automata-theoretic perspective as well [MSS88].

All these branching time logics exclude past-time operators, but they can be added. The work of Gabbay [Gab87] and Gabbay *et al* [GHR94] is particularly interesting because many temporal logics have the "separation property": that is, any complicated formula A has a *logically equivalent* form A' where A' is a conjunction $B \wedge C \wedge D$ such that B involves only past-time modalities, C involves no modalities, and D involves only future-time modalities. Thus the decision problems for these logics can often be handled by separate routines for just past-time modalities, just future-time modalities and just pure propositional reasoning.

6 Modal Tableau Systems With Explicit Accessibility

We now turn to tableau systems where the reachability relation R is represented explicitly. There are essentially two ways to represent R. One is to maintain a network of named nodes, where each node contains a set of formulae, and also maintain a separate relation $R(x, y)$ to represent that the node named y is reachable from the node named x. The names x and y are merely indices to allow cross-reference between these two "data-structures". The second is to incorporate *complex or structured* world names into the syntax, attaching the label l_1 to every formula that belongs to the world named l_1 and attaching l_2 to every formula that belong to the world named l_2. No separate reachability relation is kept since the reachability relation is built into the *structure* of the labels.

6.1 History of Explicit Tableau Systems

The most celebrated work is of course that of Kripke [Kri59] where possible worlds related by an accessibility relation are first proposed as a semantics for modal logics. Bull and Segerberg [BS84] give an account of the genesis of the possible worlds approach and suggest that credit is also due to Hintikka and Kanger. Zeman [Zem73] even credits C. S. Pierce with the idea of "a book of possible worlds" as far back as 1911!

Kripke follows Beth [Bet55] and divides each tableau into a left hand side and a right hand side where the left side is for formulae that must be assigned "true" and the right side is for formula that must be assigned "false"; see [Fit93] for examples using this style of tableau. Thus it is clear that this is a refutation procedure and we are attempting to obtain a falsifying model of possible worlds for the given formula. To handle the added complexities of modal formulae like $\Box A$ and $\neg\Box A$, Kripke uses auxiliary tableaux, where a new tableau is used for each possible world and these auxiliary tableaux are interrelated by an auxiliary reachability relation R. Auxiliary tableaux may have tableaux auxiliary to them and so on, obtaining a complex web of tableaux.

Kripke uses two basic rules to handle modal formulae: one to handle $\Box A$ on the left of a tableau and one to handle $\Box A$ on the right of a tableau. They are,

- Yl: If $\Box A$ appears on the left of a tableau t, then for every tableau t' such that tRt', put A on the left of t';
- Yr: If $\Box A$ appears on the right of a tableau t, then start out a new tableau t', with A on the right, and such that tRt'.

Different constraints on this auxiliary relation give different tableau systems. That is, the definition of the auxiliary relation R changes with each logic, so that the auxiliary relation directly mimics the required accessibility relation. For example, the auxiliary relation R for **S4** is defined to be reflexive and transitive, so for any tableau t we have tRt *by definition.* These constraints form an extra theory about R that must be taken into account at each rule application.

Note also that the application of the Yl rule can have delayed consequences. For example, if a new auxiliary tableau t'' is created and it happens to be auxiliary to the tableau t in which the Yl rule has already been applied, then we have to keep track of this previous application of Yl and add A to the left of t''. Thus, the meaning of "every tableau t' such that tRt'" includes tableaux that may come into existence via the Yr rule at any later point of the construction. The rules are therefore like constraints that may be activated at a later time.

This is essentially a way to keep track of all worlds in the counter model being sought. When a new world comes into existence, it is immediately linked into this counter-model according to the constraints on R. That is, Kripke's method is a refutation procedure where extra modal information is kept in the auxiliary relation between tableaux. The construction is on a global level in that we can return to previous nodes of the tableau construction at will. In our tableau systems $\mathcal{C}\mathbf{L}$ we cannot return to nodes higher up in the tree.

The semantic diagrams of Hughes and Cresswell [HC68] and the tableau systems of Zeman [Zem73] use essentially the same ideas except that Hughes and Cresswell use annotations of ones and zeros instead of using a left and right side. Slaght [Sla77] goes one step further than usual and adds rules for quantifiers and also incorporates a form of negated normal form by translating $\neg\Box P$ into $\Diamond\neg P$, $\neg\Diamond P$ into $\Box\neg P$, $\neg\exists x(\cdots)$ into $\forall x\neg(\cdots)$ and $\neg\Box x(\cdots)$ into $\exists x\neg(\cdots)$.

These ideas have been implemented by Catach in his TABLEAUX theorem prover [Cat91, Cat88]. Although labels are used in the TABLEAUX prover, they are used only as indices into an *explicit and separate* representation of the reachability relation. Indeed, Catach even laments the lack of modularity in this method [Cat91, page 503].

Kanger's spotted formulae [Kan57], which precede Kripke's work, are the precursors of the second explicit approach which we call the labelled tableau method. In this method, each formula is prefixed with a label to retain its modal context and the reachability relation is encoded in the structure of the labels. Given two labels we can tell whether they are related by the reachability relation simply by inspecting their structure. Fitting's prefixed tableaux are direct applications of Kanger's idea to handle many different modal logics [Fit83, chapter 8]. And as we shall soon see, Massacci [Mas94, Mas95b] has refined these ideas even further to give modular prefixed tableau systems for many modal logics. If we permit labels to contain variables then specialised "string unification" methods can be used to detect closed tableau branches as is done by Wallen [Wal89], and Artosi and Governatori [AG94]. The principle of using labels to "bring some of the semantics into the syntax" is also the basis of Gabbay's Labelled Deductive Systems [Gab9x].

Prefixes are one way to separate the modal component from the classical component. Another is to explicitly translate the modalities into a restricted subset of first-order logic. Specialised routines for first-order deduction, like resolution, can then be applied to this restricted subset. Such "translations methods" have been investigated by Morgan [Mor76], Ohlbach [Ohl90, Ohl93], Auffray and Enjalbert [AE89], Frisch and Scherl [FS91], and Gent [Gen91b, Gen93, Gen91a].

In all these translational methods, the modal logics **K**, **T**, **K4**, **S4** and **S5** are easily handled and Gent has also obtained systems for **B** and **S4.3**. The most striking feature of Gent's work is that he is unable to give a system for **S4.3.1** and this is essentially due to the fact that the reachability relation R for **S4.3.1**-frames is not first order definable. It is known that a formula of second order logic is required to express the reachability relation for **S4.3.1** [vB83]. This deficiency of translational methods is also mentioned by Auffray and Enjalbert [AE89] while the method of Frisch and Scherl [FS91] is limited to serial logics.

The biggest disadvantage of the translational methods is that first-order logic is known to be only semi-decidable, thus the translated system may not be decidable even though the original modal logic is decidable. Clearly it must be possible to identify decidable classes of first-order logic into which these translations will fall, but I am not aware of any such detailed investigations.

In all fairness, it must be mentioned that the translational methods seem to be much better for automated deduction in *first order modal logics* where various domain restrictions can complicate matters for the first order versions of our implicit tableau systems $\mathcal{C}$**L**; see [Ohl90]. At the first order level, all modal logics are only semi-decidable since they all include classical first order logic. Then, decidability is no longer an important issue.

There is a subtle but deep significance to the use of labels which explains their increased power over implicit tableau methods. Our implicit tableaux were *local* in that, at all times, we worked with a set of formulae (denoting one particular world), with no explicit reference to the particular properties of the reachability relation since these properties were built into the rules. Labelled tableaux are *global* in that the labels allow us to "see" the reachability relation and hence allows us to keep a picture of the whole model under construction.

6.2 Labelled Tableau Systems Without Unification

As stated previously the idea of labelled tableau systems goes back to at least Kripke and Kanger. The most attractive feature of labels for modal tableau systems is the ability to handle the symmetric logics like **S5** which require some form of analytic cut rule in the implicit systems we have studied so far, and also logics like **KB** for which I know no implicit tableau system formulation. We now review in some detail recent work of Massacci [Mas94] which gives simple labelled tableau systems for all the 15 distinct basic normal modal logics obtainable from **K** by the addition of any combination of the axioms T, D, 4, 5, and B in a *modular way*. The prefixed tableaux of Fitting can be obtained as derived rules in this method. Hence our labelled tableau systems are a mixture of the methods of Fitting and Massacci.

The irony is that this method is essentially Kripke's "reformulated method" based on his observation [Kri63, page 80] that:

> "These considerations suggest that the rules, which we have stated in terms of R, could instead be stated in terms of the basic tree relation S defined in the preceding paragraph (letting

R drop out of the picture altogether)."

Using trees it is possible to isolate the individual atomic aspects of reflexivity, transitivity, symmetry etc. To model combinations of these properties both Kripke and Fitting merge the respective atomic aspects into new rules. Fitting goes one step further by building in the *closure* of these properties as side-conditions, thereby requiring explicit reference to the underlying reachability relation. Massacci, on the other hand, merely adds the individual atomic rules as they are, and thereby obtains modularity. The closure is obtained by repeated applications of the atomic rules.

As an aside, note that [Mas94] contains some minor errors; for example, the system given there for **K5** is incomplete. Massacci has reworked, corrected, and extended his work into a journal version [Mas95b], but most of this section was written independently of [Mas95b]. Thus there is a lot of overlap between this section and [Mas95b], but there are also some subtle differences. In particular, we do not use an empty label at any stage, whereas Massacci sometimes uses an empty label to capture the **L**-accessibility conditions between labels.

We now switch to the tableau formulation of Fitting and Smullyan [Fit83] rather than sticking to the formulation of Rautenberg because the labels allow us to distinguish formulae that belong to one world (label) from those that belong to another (label), so there is no need to delete formulae when "traversing" from one world to another. Consequently, we can work with a *single* set of *labelled* formulae.

A **label** is a nonempty sequence of positive integers separated by dots. We use lowercase Greek letters like σ, τ for labels and often omit the dots using σn instead of $\sigma.n$ if no confusion can arise. We use Γ to denote a set of labels. The **length** of a label σ is the number of integers it contains (or the number of dots plus one), and is denoted by $|\sigma|$. For example, 1, 1.21, and 1.2.1 are three labels respectively of lengths 1, 2 and 3. A label τ is a **simple extension** of a label σ if $\tau = \sigma.n$ for some $n \geq 1$. A label τ is an **extension** of a label σ if $\tau = \sigma.n_1.n_2.\cdots.n_k$ for some $k \geq 1$ with each $n_i \geq 1$.

A set of labels Γ is **strongly generated** (with root ρ) if:

1. there is some (root) label $\rho \in \Gamma$ such that every other label in Γ is an extension of ρ; and
2. $\sigma.n \in \Gamma$ implies $\sigma \in \Gamma$.

In what follows, we always assume that $\rho = 1$ as it simplifies some technical details.

As we shall soon see, the labels capture a basic reachability relation between the worlds they name where the world named by $\sigma.n$ is accessible from the world named by σ. A set of strongly generated labels can be viewed as a tree with root ρ where $\sigma.n$ is an immediate child of σ (whence the name "strongly generated").

A **labelled formula** is a structure of the form $\sigma :: A$ where σ is a label and A is a formula. A **labelled tableau rule** has a numerator and one or more denominators as before, except that each numerator is comprised of a *single* labelled formula, and each denominator is comprised of at most two labelled formulae. There may be side conditions on the labels that appear in the rule. A **labelled tableau calculus** is simply a collection of labelled tableau rules.

A **labelled tableau** for a finite set of formulae $X = \{A_1, A_2, \cdots, A_n\}$ is a tree, where each node contains a single labelled formula, constructed by the systematic construction described in Figure 15. A **tableau branch** is any path from the root downwards in such a tree. A **branch is closed** if it contains some labelled formula $\sigma :: P$ and also contains $\sigma :: \neg P$. Otherwise it is **open**. A **tableau is closed** if every branch is closed, otherwise it is open.

A label σ is **used** on a branch if there is some labelled formula $\sigma :: P$ on that branch. A label σ is **new** to a branch if there is no labelled formula $\sigma :: P$ on that branch.

If $\mathcal{X}$ is a set of labelled formulae then we let $lab(\mathcal{X}) = \{\sigma | \sigma :: P \in \mathcal{X}\}$ be the set of all labels that appear in $\mathcal{X}$. Although a branch $\mathcal{B}$ of a tableau is defined as a set of nodes, each of which contains a formula, we often drop this pedantic distinction and use $\mathcal{B}$ to mean the set of labelled formulae on the branch. Then $lab(\mathcal{B})$ is just the set of labels that are used on branch $\mathcal{B}$.

In Figure 16 we list the rules we need, and in Figure 17 we show how they can be used to give labelled tableau systems for many basic modal logics including some symmetric logics that proved elusive using implicit tableau systems. All are based on those of Massacci [Mas94].

The rules are categorised into three types: the **PC**-rules are just the usual ones needed for classical propositional logic; the ν-rules are all the rules applicable to formulae of the form $\sigma :: \Box P$ (such formulae are called ν-formulae in many tableau formulations); and the single π-rule is the only rule applicable to formulae of the form $\sigma :: \neg\Box P$ (such formulae are called π-formulae in many tableau formulations).

As expected, there is no modal aspect to the **PC**-rules since the labels in the numerator and denominator(s) are identical. The π-rule is a "successor creator" since it is the only rule allowed to create new labels. Each ν-rule is a licence to add the formula in the denominator to the already existing world named by the label of the denominator. It is the power to look *backwards* along the reachability relation (in rules like (lB) and $(l4^r)$ that allows us to handle the symmetric and euclidean logics with such ease.

Notice that none of the rules explicitly mention the reachability relation between labels in their side-conditions. Furthermore, in all rules, the world named by the label in the denominator is at most *one step away* from the world named by the label in the numerator. For example, the (lT) rule

Stage 1: Put the labelled formulae $1 :: A_i$, $1 \leq i \leq n$, in a vertical linear sequence of nodes, one beneath the other, in some order and mark them all as awake.

While the tableau is open and some formula is awake do:

Begin Stage n+1: Choose an awake labelled formula $\sigma :: A$ as close to the root as possible. If there are several awake formulae at the same level then choose the one on the leftmost branch. If $\sigma :: A$ is atomic then mark this formula as finished and stop stage $n+1$. Otherwise update the tableau as follows where "updating a branch with a labelled formula" means adding the formula to the end of the branch and marking it as awake if it does not already appear on the branch (with any mark), but doing nothing if the formula already appears on the branch (with any mark). For every *open* branch $\mathcal{B}$ which passes through $\sigma :: A$, do:

- ($\wedge$) if $\sigma :: A$ is of the form $\sigma :: P \wedge Q$ then update $\mathcal{B}$ with $\sigma :: P$ and then update the new $\mathcal{B}$ with $\sigma :: Q$;
- ($\vee$) if $\sigma :: A$ is of the form $\sigma :: \neg(P \wedge Q)$ then split the end of $\mathcal{B}$ and update the left fork with $\sigma :: \neg P$ and update the right fork with $\sigma :: \neg Q$. If any of these updates fails to add the corresponding formula then delete that fork, possibly leaving $\mathcal{B}$ unaltered or with no fork;
- ($\neg$) if $\sigma :: A$ is of the form $\sigma :: \neg\neg P$ then update $\mathcal{B}$ with $\sigma :: P$;
- (ν) if $\sigma :: A$ is of the form $\sigma :: \Box P$ then, for every ν-rule rule in the calculus which is applicable to $\sigma :: \Box P$, update $\mathcal{B}$ with the corresponding denominator;
- (π) if $\sigma :: A$ is of the form $\sigma :: \neg\Box P$ then let k be the smallest integer such that the label σk is new on branch $\mathcal{B}$, update $\mathcal{B}$ with $\sigma k :: \neg P$, and mark all formula on $\mathcal{B}$ of the form $\sigma :: \Box Q$ as awake;

End Stage n+1: Once this has been done for every open branch that passes through $\sigma :: A$, if $\sigma :: A$ is of the form $\sigma :: \Box P$ then mark it as asleep, otherwise mark $\sigma :: A$ as finished, and terminate Stage n+1.

Fig. 15. Systematic tableau construction for $X = \{A_1, A_2, \cdots, A_n\}$.

$$(l\neg)\ \frac{\sigma :: \neg\neg P}{\sigma :: P} \qquad (l\wedge)\ \frac{\sigma :: P \wedge Q}{\begin{array}{c}\sigma :: P\\ \sigma :: Q\end{array}} \qquad (l\vee)\ \frac{\sigma :: \neg(P \wedge Q)}{\sigma :: \neg P \mid \sigma :: \neg Q}$$

$$(l\pi)\ \frac{\sigma :: \neg\Box P}{\sigma.n :: \neg P} \text{ where } \sigma.n \text{ is new to the current branch}$$

$$(lK)\ \frac{\sigma :: \Box P}{\sigma.n :: P} \qquad (lD)\ \frac{\sigma :: \Box P}{\sigma :: \neg\Box\neg P} \qquad (lT)\ \frac{\sigma :: \Box P}{\sigma :: P}$$

$$(lB)\ \frac{\sigma.n :: \Box P}{\sigma :: P} \qquad (l4)\ \frac{\sigma :: \Box P}{\sigma.n :: \Box P} \qquad (l5)\ \frac{1.n :: \Box P}{1 :: \Box\Box P}$$

$$(l4^r)\ \frac{\sigma.n :: \Box P}{\sigma :: \Box P} \qquad (l4^d)\ \frac{\sigma.n :: \Box P}{\sigma.n.m :: \Box P}$$

Note: except for σn in the rule $(l\pi)$, each label in the numerator and denominator must already exist on the branch.

Fig. 16. Single Step Rules for the Basic Modal Logics

adds the formula P to the same world, whereas the (lK) and (lB) rules add P to a successor and predecessor respectively.

At first sight, the "single step" nature of the ν-rules seems a drawback since we know that a ν-formula can affect *all* successors, regardless of how many primitive steps it takes to reach them. One is immediately tempted to add side conditions that explicitly mention the reachability relation to capture this notion as is done by Fitting [Fit83]. But it is precisely this "single step" nature that allows the rules to ignore the reachability relation and which gives us the modularity apparent in the calculi of Figure 17.

A particular rule may not capture a property of accessibility completely, but some combination of the rules will do so. For example, for transitivity we require $\sigma :: \Box P$ to be able to give $\sigma.\theta :: P$, for any $|\theta| \geq 1$, assuming that both these labels (worlds) σ and $\sigma.\theta$ exist. As Massacci [Mas94] points out, instead of building this transitive closure property into a side condition for $(l4)$, it is obtained by the combination of $(l4)$ and (lK), one step at a time, as shown below extreme left where we assume that $\theta = n.m$. That is, we cannot derive Fitting's actual rule for transitivity since that rule captures the *closure* of the transitivity property by referring to **L**-accessibility in the side condition. But we can derive *every instance* of transitivity, thereby

$\mathcal{LCL}$	**PC**-Rules	ν-Rules	π-Rule
$\mathcal{LC}PC$	$(l\neg)$, $(l\wedge)$, $(l\vee)$	—	—
$\mathcal{LC}K$	$\mathcal{LC}PC$	(lK)	$(l\pi)$
$\mathcal{LC}T$	$\mathcal{LC}PC$	(lK), (lT)	$(l\pi)$
$\mathcal{LC}D$	$\mathcal{LC}PC$	(lK), (lD)	$(l\pi)$
$\mathcal{LC}KB$	$\mathcal{LC}PC$	(lK), (lB)	$(l\pi)$
$\mathcal{LC}K4$	$\mathcal{LC}PC$	(lK), $(l4)$	$(l\pi)$
$\mathcal{LC}K5$	$\mathcal{LC}PC$	(lK), $(l4^d)$, $(l4^r)$, $(l5)$	$(l\pi)$
$\mathcal{LC}KDB$	$\mathcal{LC}PC$	(lK), (lB), (lD)	$(l\pi)$
$\mathcal{LC}KD5$	$\mathcal{LC}PC$	(lK), (lD), $(l4^d)$, $(l4^r)$, $(l5)$	$(l\pi)$
$\mathcal{LC}K4D$	$\mathcal{LC}PC$	(lK), (lD), $(l4)$	$(l\pi)$
$\mathcal{LC}K45$	$\mathcal{LC}PC$	(lK), $(l4)$, $(l4^r)$	$(l\pi)$
$\mathcal{LC}K45D$	$\mathcal{LC}PC$	(lK), $(l4)$, $(l4^r)$, (lD),	$(l\pi)$
$\mathcal{LC}K4B$	$\mathcal{LC}PC$	(lK), (lB), $(l4)$, $(l4^r)$	$(l\pi)$
$\mathcal{LC}B$	$\mathcal{LC}PC$	(lK), (lT),(lB)	$(l\pi)$
$\mathcal{LC}S4$	$\mathcal{LC}PC$	(lK), (lT), $(l4)$	$(l\pi)$
$\mathcal{LC}S5$	$\mathcal{LC}PC$	(lK), (lT), $(l4)$, $(l4^r)$	$(l\pi)$

Fig. 17. Labelled Tableau Systems for the Basic Logics

computing the closure by repeated applications of the single step rules. We can also derive other useful rules. For example, the rule of "delayed reflexivity" (lT^d) below center says something like "all worlds $(\sigma.n)$ that have a predecessor (σ) are reflexive". It can be derived in $\mathcal{LC}K5$ and $\mathcal{LC}K4B$ as shown below extreme right:

$$\dfrac{\dfrac{\sigma :: \Box P}{\sigma.n :: \Box P}(l4)}{\sigma.n.m :: P}(lK) \qquad (lT^d)\ \dfrac{\sigma.n :: \Box P}{\sigma.n :: P} \qquad \dfrac{\dfrac{\sigma.n :: \Box P}{\sigma :: \Box P}(l4^r)}{\sigma.n :: P}(lK)$$

derivation of transitivity derivation of (lT^d)

As an aside, note that in a symmetric frame, like those for **K4B**, any world that has a predecessor also has a successor, hence (lT^d) captures the essence of the $(T_\Diamond)$ rule of $\mathcal{C}^\dagger$**K4B** on page 27.

The systematic construction is based on the one given by Fitting [Fit83, page 402] for his prefixed tableau, and the one given by Massacci [Mas94], except that we have amalgamated two of Fitting's procedures in one here. Fitting first works with *occurrences* of labelled formulae in order to mark them as finished, adding fresh unfinished *occurrences* to handle necessary repetitions. Later he refines the procedure to stop explicit repetitions since this is just a form of contraction where such formulae may have to be used more than once for completeness.

We work with labelled formulae *per se*, avoiding repetitions right from the beginning, and mark most formulae as finished once we have dealt with them. But we do not mark ν-formulae as finished since they may need to be used again and again. Because we always start a stage at the highest awake formulae, these formulae get considered over and over again as desired.

Notice that the systematic procedure constructs only *one* tableau and that it traverses this tableau in a *breadth-first* manner (except that some formulae may change from asleep to awake and temporarily interrupt this traversal). Massacci [Mas94] gives an alternative systematic procedure where the formulae on a branch are processed using a different strategy; all formulae of the form $\sigma :: \neg\Box P$ on a branch are processed before all formulae of the form $\tau :: \Box Q$ for example. Space forbids us from comparing these strategies in more detail.

Example 6.2.1. Below we show a closed systematic $\mathcal{LCK}$-tableau for $X = \{\Box(p \to q), \neg(\Box p \to \Box q)\}$. We assume that $A \to B$ is written as $\neg(A \wedge \neg B)$ and that $\neg(A \to B)$ is rewritten and simplified to $A \wedge \neg B$. We use "a", "s" and "f" for **a**wake, a**s**leep and **f**inished respectively. The notation s/a indicates that the formula was asleep but was woken up during the stage.

Systematic $\mathcal{LCK}$-tableau for $X = \{\Box(p \to q), \neg(\Box p \to \Box q)\}$

Extant Tableau at Stage End	**Marks at Stage End** 1	2	3	4	5	6	7	8	9
$1 :: \Box(p \to q)$	a	s	s	s	s/a	s	s	s	s
$1 :: \neg(\Box p \to \Box q)$	a	a	f	f	f	f	f	f	f
$1 :: \Box p$			a	s	s/a	a	s	s	s
$1 :: \neg\Box q$			a	a	f	f	f	f	f
$1.1 :: \neg q$					a	a	a	f	f
$1.1 :: p \to q$						a	a	a	f
$1.1 :: p$							a	a	a
$1.1 :: \neg p$									a
$1.1 :: \neg\neg q$									a
(closed) (closed)									

Example 6.2.2. The formula $(\Box\Diamond p) \wedge (\Diamond p)$ can be written in negated normal form as $(\Box\neg\Box\neg p) \wedge (\neg\Box\neg p)$. As the reader can verify, the systematic **S4**-tableau for $X = \{(\Box\neg\Box\neg p) \wedge (\neg\Box\neg p)\}$ neither terminates nor closes.

6.3 Soundness of Single Step Tableau Rules

The soundness of the tableau rules is proved using a method from Fitting [Fit83], but modified to cater for the strongly generated property. We first extend the primitive notion of reachability between labels σ and $\sigma.n$ into a general notion of **L**-accessibility *between labels* σ and τ, and show that it captures the conditions on **L**-frames.

Recall that label ρ is the root of a strongly generated set of labels if every other label in the set is an extension of ρ.

A set $\mathcal{X}$ of labelled formulae is **strongly generated** if $lab(\mathcal{X})$ is strongly generated. For any two labels σ and τ from some strongly generated set Γ of labels with root $\rho = 1$ we define an **L**-accessibility relation $\rhd$ according to Figure 18. These conditions are calculated by taking the appropriate closure of the underlying basic reachability relation between σ and $\sigma.n$. For example, the condition on **K45**-frames is calculated by computing the transitive and euclidean closure of the basic reachability relation. It is here that our assumption that the root $\rho = 1$ simplifies the conditions for **L**-accessibility, but there is still a slight complication for the serial logics.

For any nonserial logic $\mathbf{L_1}$ we say that σ is an $\mathbf{L_1}$**-deadend** if there is no τ that is $\mathbf{L_1}$-accessible from σ. Now we can express the seriality condition for the serial counterpart $\mathbf{L} = \mathbf{L_1D}$ by demanding that all $\mathbf{L_1}$-deadends be reflexive. In particular, we say that $\sigma \in \Gamma$ is a **K**-deadend if no label in Γ is a simple extension of σ. In Figure 18 we have computed the forms of the $\mathbf{L_1}$-deadends and added an extra condition to make them reflexive for each logic $\mathbf{L_1D}$. The notation $|\Gamma|$ means the number of labels in Γ.

We leave it to the reader to generalise these conditions to account for the case where ρ is an arbitrary label. Note that the conditions on **L**-accessibility in Figure 18 and the conditions on accessibility in the finite-**L**-frames of Figure 13 on page 63 are closely related. We return to this point later.

But first we relate **L**-accessibility with the **L**-frames of Figure 4 on page 9.

Theorem 6.3.1. *If* Γ *is a strongly generated set of labels with root* $\rho = 1$ *then* $\mathcal{F} = \langle \Gamma, \rhd \rangle$ *is an* **L**-*frame.*

Proof: It is obvious that **KT**-accessibility, **K4**-accessibility and **KB**-accessibility forces $\mathcal{F}$ to be respectively reflexive, transitive and symmetric. We consider only the case for **K45** in detail.

We have to show that **K45**-accessibility forces $\mathcal{F}$ to be euclidean and transitive. **K45**-accessibility $\rhd$ is euclidean if $\sigma_0 \rhd \sigma_1$ and $\sigma_0 \rhd \sigma_2$ implies $\sigma_1 \rhd \sigma_2$, where **K45**-accessibility $\rhd$ is defined as:

$$\sigma \rhd \tau \text{ iff } (\tau = \sigma.\theta \text{ and } |\theta| \geq 1) \text{ or } (|\sigma| \geq 2 \text{ and } |\tau| \geq 2)$$

By substitution we get:

Definition of $\sigma \rhd \tau$ where σ and τ are nonempty and drawn from a strongly generated set of labels Γ with root $\rho = 1$	
Logics	τ is **L**-accessible from σ iff
K	$\tau = \sigma.n$ for some $n \geq 1$
KT	$\tau = \sigma.n$ or $\tau = \sigma$
KB	$\tau = \sigma.n$ or $\sigma = \tau.m$
K4	$\tau = \sigma.\theta$ and $\lvert\theta\rvert \geq 1$
K5	$\tau = \sigma.n$ or $(\lvert\sigma\rvert \geq 2$ and $\lvert\tau\rvert \geq 2)$
K45	$(\tau = \sigma.\theta$ and $\lvert\theta\rvert \geq 1)$ or $(\lvert\sigma\rvert \geq 2$ and $\lvert\tau\rvert \geq 2)$
KD	**K**-condition or (σ is a **K**-deadend and $\sigma = \tau$)
KDB	**KB**-condition or $(\lvert\Gamma\rvert = 1$ and $\sigma = \tau = 1)$
KD4	**K4**-condition or (σ is a **K**-deadend and $\sigma = \tau$)
KD5	**K5**-condition or $(\lvert\Gamma\rvert = 1$ and $\sigma = \tau = 1)$
KD45	**K45**-condition or $(\lvert\Gamma\rvert = 1$ and $\sigma = \tau = 1)$
KB4	$((\tau = \sigma.\theta$ or $\sigma = \tau.\theta)$ and $\lvert\theta\rvert \geq 1)$ or $(\tau = \sigma$ and $\lvert\Gamma\rvert \geq 2)$
B	$\tau = \sigma$ or $\tau = \sigma.n$ or $\sigma = \tau.m$
S4	$(\tau = \sigma.\theta$ and $\lvert\theta\rvert \geq 1)$ or $(\tau = \sigma)$
S5	$\lvert\sigma\rvert \geq 1$ and $\lvert\tau\rvert \geq 1$

Fig. 18. Definition of **L**-accessibility $\rhd$.

Hypotheses	Expanded Hypothesis
$\sigma_0 \rhd \sigma_1$ and $\sigma_0 \rhd \sigma_2$	$(\sigma_1 = \sigma_0.\theta_1$ and $\lvert\theta_1\rvert \geq 1)$ or $(\lvert\sigma_0\rvert \geq 2$ and $\lvert\sigma_1\rvert \geq 2)$ and $(\sigma_2 = \sigma_0.\theta_2$ and $\lvert\theta_2\rvert \geq 1)$ or $(\lvert\sigma_0\rvert \geq 2$ and $\lvert\sigma_2\rvert \geq 2)$
Goal	Expanded Goal
$\sigma_1 \rhd \sigma_2$	$(\sigma_2 = \sigma_1.\theta_3$ and $\lvert\theta_3\rvert \geq 1)$ or $(\lvert\sigma_1\rvert \geq 2$ and $\lvert\sigma_2\rvert \geq 2)$

Now, we know that σ_0 is nonempty, hence $|\sigma_0| \geq 1$. But this together with $(\sigma_1 = \sigma_0.\theta_1$ and $|\theta_1| \geq 1)$ in the left disjunct of the first hypothesis immediately gives $|\sigma_1| \geq 2$. Thus both disjuncts of the first line of the hypothesis imply $|\sigma_1| \geq 2$.

Similarly, $|\sigma_0| \geq 1$ together with $(\sigma_2 = \sigma_0.\theta_2$ and $|\theta_2| \geq 1)$ in the left disjunct of the second hypothesis gives $|\sigma_2| \geq 2$. Thus both disjuncts of the second hypothesis imply $|\sigma_2| \geq 2)$.

And the conjunction of these two gives the second disjunct of the goal showing that **K45**-accessibility relation $\rhd$ is indeed euclidean.

To show that **K45**-accessibility is also transitive, we must show that

$\sigma_0 \rhd \sigma_1$ and $\sigma_1 \rhd \sigma_2$ implies $\sigma_0 \rhd \sigma_2$. The same expansions can be used but the roles of hypotheses and goal are slightly altered. The argument is almost identical, except for one subcase which relies on the fact that $|\sigma_0| = 1$ implies $\sigma_0 = 1$. ∎

Let $\mathcal{X}$ be a strongly generated set of labelled formulae, let $lab(\mathcal{X})$ be the set of labels that appear in $\mathcal{X}$ and let $\mathcal{M} = \langle W, R, V \rangle$ be some **L**-model where **L** is any one of the 15 distinct basic normal modal logics obtainable by adding any combination of the axioms T, D, B, 4 and 5 to logic **K**. Call a world in $\mathcal{M}$ **idealisable** iff it has an R-successor in $\mathcal{M}$.

An **L-interpretation of** (a strongly generated set of labelled formulae) $\mathcal{X}$ **in** $\mathcal{M}$ is a mapping $I : lab(\mathcal{X}) \mapsto W$ that satisfies: if $\sigma \rhd \tau$ and $I(\tau)$ is idealisable then $I(\sigma)RI(\tau)$, where $\rhd$ is the appropriate **L**-accessibility relation from Figure 18 [Fit83].

A strongly generated set $\mathcal{X}$ of labelled formulae is **L-satisfiable under the L-interpretation** I if $I(\sigma) \models A$ for each $\sigma :: A$ in $\mathcal{X}$; and is **L-satisfiable** if it is **L**-satisfiable under some **L**-interpretation. A branch of a labelled tableau is **L**-satisfiable if the set of labelled formulae on it is **L**-satisfiable, and a tableau is **L**-satisfiable if some branch of the tableau is **L**-satisfiable.

Proposition 6.3.2. *The set of labelled formulae $lab(\mathcal{B})$ from any branch $\mathcal{B}$ of a labelled tableau is a strongly generated set.*

Proof: By the fact that the initial label is always $\rho = 1$, and the fact that the only new labels that may be created are labels of the form $\sigma.n$, $n \geq 1$, which are all simple extensions of some $\sigma \in lab(\mathcal{B})$. ∎

We now prove soundness of some of the rules leaving the others to the reader. Since the systematic procedure updates all branches that pass through the chosen formula, the soundness theorem states the following: if a tableau $\mathcal{T}$ is **L**-satisfiable and we apply rule $(l\rho)$ to get tableau $\mathcal{T}'$, then $\mathcal{T}'$ is also **L**-satisfiable. Since every rule has at most two denominators, a rule can cause a given branch to split into at most branches. Consequently we have to prove that if a branch $\mathcal{B}$ is **L**-satisfiable, and applying rule $(l\rho)$ causes it to be updated into branches $\mathcal{C}$ and $\mathcal{D}$, then at least one of the new branches is also **L**-satisfiable.

Soundness of (lK) **for L-frames:** Suppose $\mathcal{B}$ is an **L**-satisfiable branch and that we apply the (lK) rule to some awake $\sigma :: \neg\Box P$ on $\mathcal{B}$ to obtain branch $\mathcal{C}$ containing $\sigma n :: \neg P$ where σn is a simple extension of σ that is new to $\mathcal{B}$. We have to show that $\mathcal{C}$ is **L**-satisfiable.

Since $\mathcal{B}$ is **L**-satisfiable, there is some **L**-model $\mathcal{M} = \langle W, R, V \rangle$ and some **L**-interpretation I in $\mathcal{M}$ such that $I(\sigma) \in W$ and $I(\sigma) \models \neg\Box P$. Hence $I(\sigma)$ is idealisable as there is some $w \in W$ with $I(\sigma)Rw$ and $w \models \neg P$. Since σn is new, it does not appear in $\mathcal{B}$ and hence has no image under I. Extend I by putting $I(\sigma n) = w$. We then have $\sigma \rhd \sigma n$, $I(\sigma)RI(\sigma n)$, and $I(\sigma n) \models \neg P$ meaning that $\mathcal{C}$ is indeed **L**-satisfiable under the extended I in $\mathcal{M}$. ∎

Soundness of $(l4^d)$ for K5-frames: Suppose $\mathcal{B}$ is a **K5**-satisfiable branch and that we apply the $(l4^d)$ rule to some $\sigma n :: \Box P$ to get a branch $\mathcal{C}$ containing $\sigma nm :: \Box P$. We have to show that $\mathcal{C}$ is also **K5**-satisfiable.

Since $\mathcal{B}$ is **K5**-satisfiable and the labels σn and σnm must already exist on $\mathcal{B}$, there is some **K5**-model $\mathcal{M} = \langle W, R, V \rangle$ and some **K5**-interpretation I in $\mathcal{M}$ such that $I(\sigma n) \in W$, $I(\sigma nm) \in W$ and $I(\sigma n) \models \Box P$. The label σn can exist on $\mathcal{B}$ only if σ also exists on $\mathcal{B}$ since $\mathcal{B}$ is strongly generated. Hence there is some $I(\sigma) \in W$. The configuration $\sigma \rhd \sigma n \rhd \sigma nm$ immediately implies $I(\sigma)RI(\sigma n)RI(\sigma nm)$ by the definition of I. Because R is euclidean we know that $I(\sigma n)RI(\sigma n)$; that is $I(\sigma n)$ is reflexive. Then $I(\sigma n)RI(\sigma nm)$ and $I(\sigma n)RI(\sigma n)$ gives $I(\sigma nm)RI(\sigma n)$. Hence $I(\sigma nm) \models \Diamond\Box P$. Euclidean frames must validate axiom 5 ($\Diamond\Box A \rightarrow \Box A$) hence $I(\sigma nm) \models \Box P$. We have not altered I in any way, so by definition, $\mathcal{C}$ is **K5**-satisfiable under I in $\mathcal{M}$. ∎

Soundness of $(l5)$ for K5-frames: Suppose $\mathcal{B}$ is a **K5**-satisfiable branch and that we apply the $(l5)$ rule to some $1.n :: \Box P$ to get a branch $\mathcal{C}$ containing $1 :: \Box\Box P$. We have to show that $\mathcal{C}$ is also **K5**-satisfiable.

As before there is some **K5**-model $\mathcal{M} = \langle W, R, V \rangle$ and some **K5**-interpretation I in $\mathcal{M}$ such that $I(1.n) \in W$ and $I(1.n) \models \Box P$. Since 1 is used on $\mathcal{B}$ and $1 \rhd 1.n$, there must be some $I(1) \in W$ with $I(1)RI(1.n)$.

Now suppose for a contradiction that $I(1) \models \neg\Box\Box P$; then there is some $w \in W$ such that $I(1)Rw$ and $w \models \neg\Box P$, which in turn implies that there is some $w' \in W$ such that wRw' and $w' \models \neg P$. Since R is euclidean, $I(1)RI(1.n)$ and $I(1)Rw$ gives $wRI(1.n)$, and then wRw' gives $I(1.n)Rw'$. But then $I(1.n) \models \Box P$ implies $w' \models P$; contradiction. Hence $I(1) \models \Box\Box P$ and $\mathcal{C}$ is **K5**-satisfiable under I in $\mathcal{M}$. ∎

Theorem 6.3.3. *If the systematic tableau for X closes then X is* **L**-*unsatisfiable.*

Proof: For a contradiction, suppose the tableau for X is closed and that X is **L**-satisfiable. The latter means that there is some **L**-model $\mathcal{M} = \langle W, R, V \rangle$ and some world $w \in W$ such that $w \models X$. Our tableau begins with nodes $1 :: A_i$, for each $A_i \in X$ so define an **L**-interpretation I in $\mathcal{M}$ such that $I(1) = w$. Then the initial tableau comprising of the linear sequence of these nodes $1 :: A_i$ is **L**-satisfiable (under I in $\mathcal{M}$). Since each of our tableau rules is sound, any tableau obtained from this initial tableau by these rules is also **L**-satisfiable. Hence our tableau is **L**-satisfiable.

Suppose $\mathcal{B}$ is some branch of this closed tableau. Then $\mathcal{B}$ itself is closed and hence contains some labelled formula $\sigma :: P$ and also contains $\sigma :: \neg P$. Now any **L**-interpretation I' for $\mathcal{B}$ in any **L**-model $\mathcal{M}'$ would entail that $I'(\sigma) \models P$ and also that $I'(\sigma) \models \neg P$, which is clearly impossible. Hence $\mathcal{B}$ is not **L**-satisfiable. Since $\mathcal{B}$ was an arbitrary branch this must be true for all branches of this closed tableau. Then, by definition, our tableau is not **L**-satisfiable. Contradiction, hence if the tableau for X closes then X is **L**-unsatisfiable. ∎

Corollary 6.3.4 (soundness). *If the systematic tableau for $\{\neg A\}$ is closed then A is **L**-valid.*

6.4 Fairness, Infinite Tableaux, Chains and Periodicity

The systematic tableau construction may go on ad infinitum in some cases. We now prove some useful properties of our systematic labelled tableau procedure giving some insight into its behaviour.

We have already noted that the systematic procedure is essentially a breadth-first traversal of the tableau under construction except that certain formulae may awaken to interrupt this traversal. In what follows we refer to the *uninterrupted* sequence of node traversal as the **visit sequence**. That is, the visit sequence is the sequence in which the systematic procedure would visit the nodes if no ν-formula is reawakened. It has little to do with the sequence of nodes on a *particular* branch.

The systematic tableau is a finitely generated tree in that each node has at most two immediate children (since branches are caused only by the $(\vee)$ rule). By Königs lemma, an infinite but finitely generated tree must contain an infinite branch (see Fitting [Fit83, pages 404-407]). Hence there are four ways in which the systematic procedure can go on ad infinitum:

1. by constructing an *infinite branch* containing a sequence of distinct labelled formulae $\sigma :: P_1, \sigma :: P_2, \sigma :: P_3, \cdots, \sigma :: P_n, \cdots$ all with the same label σ ;
2. by constructing an *infinite branch* containing a sequence of labelled formulae $\sigma.1 :: P_1, \sigma.2 :: P_2, \sigma.3 :: P_3, \cdots, \sigma.n :: P_n, \cdots$ all simple extensions of some common σ;
3. by constructing an *infinite branch* containing a sequence of labelled formulae $\sigma_1 :: P_1, \sigma_2 :: P_2, \sigma_3 :: P_3, \cdots, \sigma_n :: P_n, \cdots$ all with different labels ; and
4. by traversing a set of formulae that repeatedly switch from asleep to awake and vice-versa on the *visit sequence*.

We show that items (1), (2) and (4) cannot occur.

Lemma 6.4.1. *In any branch of a systematic tableau for the finite set of formulae X, the maximum number of formulae with some given label σ is finite.*

Proof: By induction on the length of σ. If $|\sigma| = 1$ then $\sigma = 1$ and the only possible formulae with this label are either subformulae of X, negations of a subformula of X, or are obtained from some subformulae of X by the building up rules $(l5)$ and (lD). But no infinite sequence of building up rules is possible. If $|\sigma| \geq 1$ then σ must have been created by

$(l\pi)$ which adds only the negation of a *subformula* of its numerator. For details see Fitting [Fit83, page 411]. ∎

Item 1 above is then impossible since any branch has but a finite number of formulae with label σ and we do not permit the branch to contain repetitions. We leave it to the reader to compute actual bounds noting the presence of the "building up rules" (lD) and $(l5)$; see Massacci [Mas94]

Lemma 6.4.2. *In any branch of a systematic tableau for the finite set of formulae X, the number N_k of different labels of length k is finite.*

Proof: Proof by induction on k and the fact that the systematic tableau construction avoids repetitions. See Fitting [Fit83, pages 410-412] and Massacci [Mas94] for more exact bounds but once again beware that these need to be adjusted for the "building up rules". ∎

Thus no branch can contain an infinite number of labels all of the same length k for any k, and item 2 above is also impossible.

We now turn to item 4 in some detail since these details cannot be found elsewhere. First note that although a branch does not contain repetitions, the visit sequence may do so.

Lemma 6.4.3. *A particular labelled formula occurrence $\sigma :: \Box Q$ on the visit sequence can be awakened only a finite number of times.*

Proof: The only way to awaken a ν-formula occurrence $\sigma :: \Box Q$ is to visit some π-formula occurrence $\sigma :: \neg\Box P$ that appears on the *same branch* as $\sigma :: \Box Q$. Since the systematic tableau is finitely branching, the number of such branches is finite. A branch can contain $\sigma :: \neg\Box P$ at most once, hence the number of *occurrences* of $\sigma :: \neg\Box P$ on the visit sequence is (also) finite. Since σ must be of finite length, Lemma 6.4.1 guarantees that there are only a finite number of formulae with label σ on any branch of the tableau. Hence there are a finite number of π-formulae occurrences that can awaken $\sigma :: \Box Q$.

If none of these π-formulae occurrences is visited then $\sigma :: \Box Q$ is never awakened. On the other hand, whenever one of these π-formulae occurrences is visited, it is marked as finished, and π-formulae are never reawakened, hence $\sigma :: \Box Q$ can be awakened only a finite number of times. Since this formula occurrence was an arbitrary ν-formula occurrence we know that *every* ν-formula occurrence can be awakened only a finite number of times. ∎

Lemma 6.4.4 (fairness). *If a labelled formula occurrence $\sigma :: A$ on the visit sequence is awake at the end of Stage n, the systematic procedure is guaranteed to visit it at some later stage.*

Proof: By induction on the number of π-formulae occurrences that precede $\sigma :: A$ in the visit sequence. Clearly, if $\sigma :: A$ is the root then it is immediately visited at Stage $n+1$. Similarly, if there are no π-formulae occurrences between the root and $\sigma :: A$ on the visit sequence then every subsequent stage will visit the next intervening formulae occurrence in the visit sequence and mark it as asleep or finished. The absence of intervening π-formulae occurrences means that no formulae occurrences can awaken until after $\sigma :: A$ is visited. Hence there must come a stage that visits $\sigma :: A$.

Suppose the lemma holds for any labelled formula occurrence with j π-formulae occurrences preceding it in the visit sequence.

Consider some $\sigma :: A$ occurrence that is awake at the end of stage n but that has $j + 1$ π-formulae occurrences preceding it in the visit sequence. Let $\tau :: \neg\Box B$ be the last π-formula occurrence in the visit sequence that precedes $\sigma :: A$.

If $\tau :: \neg\Box B$ is not awake at the end of stage n then it must be finished, meaning that all π-formula occurrences preceding $\sigma :: A$ in the visit sequence must be finished. Each subsequent stage must visit one of the awake ν-formulae occurrences preceding $\sigma :: A$ and mark each one as asleep. No ν-formulae occurrences can awaken during this process since there are no awake π-formula occurrences preceding $\sigma :: A$. Hence there must come a stage that visits $\sigma :: A$.

If $\tau :: \neg\Box B$ is awake at the end of stage n then it satisfies the induction hypothesis, so it will eventually be visited at some later stage, and marked as finished, meaning that no π-formula occurrences preceding $\sigma :: A$ in the visit sequence are awake. Some ν-formulae occurrences preceding $\sigma :: A$ may be awakened by the visit to $\tau :: \neg\Box B$ but each of these will be visited in turn and put to sleep in the stages that follow. Again, no formulae occurrences will be awakened in this process. Hence there must come a stage when we visit the formula occurrence immediately after $\tau :: \neg\Box B$ in the visit sequence. If this is $\sigma :: A$ then we are done. Otherwise this stage and subsequent stages must bring us closer and closer to $\sigma :: A$ since none of these intervening formulae occurrences is a π-formula. ∎

Lemma 6.4.5. *No labelled formula occurrence on the visit sequence can remain awake for ever.*

Proof: Suppose the occurrence $\sigma :: A$ is awake at stage n. Lemma 6.4.4 guarantees that $\sigma :: A$ will be visited at some later stage m with $m > n$. If $\sigma :: A$ is not a ν-formula then it will be marked as finished and will remain so hereafter. Else $\sigma :: A$ is a ν-formula and it will be marked as asleep at the end of stage m. If $\sigma :: A$ ever awakens at some later stage k then Lemma 6.4.4 again guarantees that it will be visited and put back to sleep. But this can happen only a finite number of times since Lemma 6.4.3 guarantees that $\sigma :: A$ can awaken only a finite number of times. Hence there must come a stage when $\sigma :: A$ is put to sleep, never to awaken again. ∎

Thus the systematic procedure is "fair" in that item 4 is also impossible. The only way the systematic procedure can go ad infinitum is for some branch to have at least one infinite sequence of longer and longer labels of the form $\sigma, \sigma.n_1, \sigma.n_1.n_2 \cdots$ where each label is a simple extension of its predecessor. In fact, since every label starts with a 1 we can be more precise as below (again following Fitting [Fit83]).

A **chain** is a sequence of labels $1, \sigma_1, \sigma_2 \cdots$ where each label in the sequence is a simple extension of its predecessor [Fit83]. A chain of labels $1, \sigma_1, \sigma_2 \cdots$ from branch $\mathcal{B}$ is **periodic** if there exist distinct labels σ_i and σ_j in the chain $(i < j)$ such that $\sigma_i :: A$ is on $\mathcal{B}$ iff $\sigma_j :: A$ is on $\mathcal{B}$; that is if $\{A | \sigma_i :: A \text{ on } \mathcal{B}\} = \{B | \sigma_j :: B \text{ on } \mathcal{B}\}$. A branch is **periodic** if every infinite chain (of labels) on $\mathcal{B}$ is periodic.

Lemma 6.4.6. *If any branch of a systematic tableau for the finite set of formulae X is infinite, then it must be periodic [Fit83].*

Proof: Basically, given a finite X, there is a limit to the number of different (unlabelled) formulae we can play with, even with the building up rules. Thus any infinite chain of prefixed formulae from any one branch must repeat formulae at some stage. Since this is true for every chain on an infinite branch, the branch must become periodic. ∎

We thus have a handle on the systematic construction since an infinite branch is not as bad as it first seemed. If we could keep track of cycles then we could obtain a decision procedure. We briefly return to this point later.

6.5 Completeness

Again we follow Fitting [Fit83, pages 408-410] but make adjustments for the strongly generated property. A strongly generated set $\mathcal{X}$ of labelled formulae is **L-downward-saturated** if it satisfies the following conditions, where $\rhd$ is the appropriate **L**-accessibility relation between labels from Figure 18 (page 83):

0) there is no formula A such that both $\sigma :: A$ and $\sigma :: \neg A$ are in $\mathcal{X}$;

1) if $\sigma :: \neg\neg A \in \mathcal{X}$ then $\sigma :: A \in \mathcal{X}$

2) if $\sigma :: A \wedge B \in \mathcal{X}$ then $\sigma :: A \in \mathcal{X}$ and $\sigma :: B \in \mathcal{X}$;

3) if $\sigma :: \neg(A \wedge B) \in \mathcal{X}$ then $\sigma :: \neg A \in \mathcal{X}$ or $\sigma :: \neg B \in \mathcal{X}$;

4) if $\sigma :: \Box A \in \mathcal{X}$ then $\tau :: A \in \mathcal{X}$ for every τ such that $\sigma \rhd \tau$;

5) if $\sigma :: \neg\Box A \in \mathcal{X}$ then $\tau :: \neg A \in \mathcal{X}$ for some τ such that $\sigma \rhd \tau$.

Lemma 6.5.1. *If $\mathcal{X}$ is a strongly generated set of labelled formulae that is* **L***-downward-saturated and $lab(\mathcal{X})$ has root $\rho = 1$, then $\mathcal{X}$ is* **L***-satisfiable in a model whose possible worlds are the labels that appear in $\mathcal{X}$.*

Proof: Suppose $\mathcal{X}$ is **L**-downward-saturated and let $lab(\mathcal{X})$ be the set of labels that appear in $\mathcal{X}$. Since $\mathcal{X}$ is strongly generated, so is $lab(\mathcal{X})$. Now define a model $\langle W, R, V\rangle$ as follows:

1. let $W = lab(\mathcal{X})$;

2. let $\sigma R \tau$ iff $\sigma \rhd \tau$ (that is, iff τ is **L**-accessible from σ);
3. for each primitive proposition p let $V(p) = \{\sigma | \sigma :: p \in \mathcal{X}\}$.

It is then easy to show by induction on the degree of a formula A and the **L**-downward-saturated property that: if $\sigma :: A \in \mathcal{X}$ then $\sigma \models A$ in the model $\langle W, R, V \rangle$. The identity mapping $I(\sigma) = \sigma$ is then an **L**-interpretation for $\mathcal{X}$ in the model $\langle W, R, V \rangle$.

Once again, the condition that $\rho = 1$ is forced upon us by our reliance on this condition in the definitions of **L**-accessibility. ∎

We have already noted that the systematic procedure is essentially a breadth-first traversal of the tableau under construction. We have also identified the mode in which this procedure can go ad infinitum. Keeping these in mind, we say that an open tableau is **completed** if it is infinite or if no formulae in it is awake. But before we can prove the completeness theorem we need to show that our systematic procedure "does everything that is necessary" in the following sense.

Lemma 6.5.2. *If $\mathcal{B}$ is an open branch of a completed systematic tableau then $\mathcal{B}$ is closed with respect to every tableau rule in the calculus in that: every rule that could have been applied to a formula in $\mathcal{B}$ must have been applied at some stage.*

Proof: By fairness, every formula is visited at least once. Thus the **PC**-rules and the π-rules must have been applied whenever it was possible. For the ν-rules, suppose $\sigma :: \Box P$ is some ν-formula on $\mathcal{B}$ and suppose some instance of a ν-rule $(l\rho)$ is applicable to it because some label τ of the required form is used on $\mathcal{B}$.

Now, when $\sigma :: \Box P$ was first visited, if τ was used on the extant part of $\mathcal{B}$ then we are done for the given instance of rule $(l\rho)$ must have been applied then.

Else, τ must be $\sigma.n$ and must have been created at some later stage by some awake π-formula $\sigma :: \neg\Box Q$ on $\mathcal{B}$. The creation of $\sigma.n$ must have awakened $\sigma :: \Box P$. Since $\mathcal{B}$ is completed, and our systematic procedure is fair, the procedure must have visited $\sigma :: \Box P$ at some later stage still. The given instance of rule $(l\rho)$ must have been applied at that later stage since τ was used on the extant part of $\mathcal{B}$. ∎

Lemma 6.5.3. *If $\mathcal{B}$ is an open branch of a completed systematic tableau then $\mathcal{B}$ is* **L***-downward-saturated.*

Proof: By Lemma 6.5.2, $\mathcal{B}$ is closed with respect to every rule of the calculus (in the appropriate sense). We now have to go through the necessary clauses (see page 89) to show that $\mathcal{B}$ is **L**-downward-saturated.

Clause **0)** is satisfied since $\mathcal{B}$ is open. Clauses **1)**, **2)** and **3)** are satisfied since $\mathcal{B}$ must be closed with respect to the classical propositional rules. Clause **5)** must be satisfied because $\mathcal{B}$ is closed with respect to $(l\pi)$. For clause **4)** assume that $\sigma :: \Box A \in \mathcal{B}$ and that $\sigma \rhd \tau$ for some τ in $lab(\mathcal{B})$. We have to show that $\tau :: A \in \mathcal{B}$ for each definition of **L**-accessibility $\triangleright$ from Figure 18.

We give the proof for **K5** only. By the definition of **K5**-accessibility, $\sigma \triangleright \tau$ means that

$$\tau = \sigma.n \text{ or } (|\sigma| \geq 2 \text{ and } |\tau| \geq 2)$$

Case 1: If $\tau = \sigma.n$ then $\sigma :: \Box A \in \mathcal{B}$ implies $\sigma n :: A \in \mathcal{B}$ by the fact that $\mathcal{B}$ is closed with respect to the rule (lK).
Case 2: Otherwise, if $(|\sigma| \geq 2$ and $|\tau| \geq 2)$ then $\sigma = 1.n_1.n_2 \cdots n_k$ for some $k \geq 1$ and $\tau = 1.m_1.m_2 \cdots m_l$ for some $l \geq 1$. Then starting from $(\sigma :: \Box A) = (1.n_1.n_2 \cdots n_k :: \Box A)$ we can obtain $1.n_1 :: \Box A \in \mathcal{B}$ and $1 :: \Box A \in \mathcal{B}$ by closure of $\mathcal{B}$ with respect to $(l4^r)$. From the first we can obtain $1 :: \Box\Box A \in \mathcal{B}$ by closure of $\mathcal{B}$ with respect to $(l5)$, and from this we obtain $1.m_1 :: \Box A \in \mathcal{B}$ by (lK). Now, if $l = 1$ then $\tau = 1.m_1$ and $1 :: \Box A \in \mathcal{B}$ immediately implies $\tau :: A \in \mathcal{B}$ by (lK). Otherwise, if $l \geq 2$ then $1.m_1 :: \Box A \in \mathcal{B}$ and closure with respect to $(l4^d)$ guarantee that $1.m_1.m_2 \cdots m_{l-1} :: \Box A \in \mathcal{B}$ from which we get $(1.m_1.m_2 \cdots m_l :: A) = (\tau :: A) \in \mathcal{B}$ by (lK) as desired. ■

Theorem 6.5.4. *If the systematic tableau for X does not close then X is* **L***-satisfiable.*

Proof: Suppose the systematic tableau for X does not close. Then the tableau must be completed, and must contain some open branch $\mathcal{B}$ by definition. Lemma 6.5.3 guarantees that $\mathcal{B}$ is an **L**-downward-saturated set. Since $lab(\mathcal{B})$ must have root $\rho = 1$, Lemma 6.5.1 then guarantees that $\mathcal{B}$ is **L**-satisfiable (under the identity **L**-interpretation $I(\sigma) = \sigma$) in an **L**-model $\mathcal{M} = \langle lab(\mathcal{B}), \triangleright, V\rangle$. Furthermore, if $\sigma :: A \in \mathcal{B}$ then $\sigma \models A$ in $\mathcal{M}$. The tableau started with a linear sequence of nodes $1 :: A_i$ for every $A_i \in X$, hence $1 :: A_i \in \mathcal{B}$ for every $A_i \in X$. But then $1 \models X$ in $\mathcal{M}$. ■

Corollary 6.5.5 (completeness). *If A is* **L***-valid then the systematic tableau for $\{\neg A\}$ must close.*

These methods extend easily to cater for "strong completeness" where we are allowed both "global" and "local" assumptions; see Fitting [Fit83] and Massacci [Mas94].

6.6 Cycles, Termination and Decidability

In the previous sections we have seen how an infinite tableau must give rise to a counter-model. But it is also possible to modify the systematic procedure to identify potential periodic chains and keep tabs on them during the systematic procedure. That is, once a chain of labels becomes periodic because σ_i and σ_j label identical sets of formulae, all formulae with the longer label are put to sleep. They are awakened only when periodicity for this chain is broken by the appearance of some new formula with a label

$$(l\pi G)\quad \frac{\sigma :: \neg\Box P}{\begin{array}{l}\sigma.n :: \neg P\\ \sigma.n :: \Box P\end{array}} \text{ where } \sigma.n \text{ is new to the current branch}$$

$$(l\pi Grz)\quad \frac{\sigma :: \neg\Box P}{\begin{array}{l}\sigma.n :: \neg P\\ \sigma.n :: \Box(P \rightarrow \Box P)\end{array}} \text{ where } \sigma.n \text{ is new to the current branch}$$

$\mathcal{LCL}$	**PC**-Rules	ν-Rules	π-Rule	**L**-accessibility $\rhd$
$\mathcal{LC}G$	$\mathcal{LC}PC$	(lK), $(l4)$	$(l\pi G)$	**K**
$\mathcal{LC}K4G_o$	$\mathcal{LC}PC$	(lK), $(l4)$	$(l\pi Grz)$	**K4**
$\mathcal{LC}Grz$	$\mathcal{LC}PC$	(lK), $(l4)$, (lT)	$(l\pi Grz)$	**S4**

Fig. 19. Labelled Tableau Systems for Provability Logics

σ_i or σ_j; see Massacci [Mas94]. Lemma 6.4.6 guarantees that every infinite branch will eventually become periodic, hence the modified systematic procedure will terminate for finite X. If the tableau has not closed then we are still guaranteed the same model as if we had allowed it to run ad infinitum. Thus these labelled tableau can be used as decision procedures for the 15 basic logics. By keeping tabs on cycles we can also prove the finite model property for these logics since the resulting **L**-frames are exactly the finite-**L**-frames of Figure 13 (page 63).

The details are considerably more intricate than the preceding paragraph suggests since we have to preserve "fairness" and completeness. But there simply is no space. Massacci [Mas94, Mas95b] gives alternative proofs of decidability for his systematic procedure based on an interpretation of the tableau rules as term rewriting rules. But a check for periodicity cannot be avoided for the transitive logics.

6.7 Extensions and Further Work

The most obvious extensions of this approach are to multi-modal logics where different sorts of labels are used to model the different reachability relations.

An alternative extension is to change the π-rule, thereby obtaining systems for the provability logics, as shown in Figure 19. Note that first-order definability is not a hurdle for these labelled tableau systems since the class of **G**-frames and **Grz**-frames are *not* first order definable. It may also be possible to extend these systems to handle some of the Diodorean modal logics.

We noted on page 82 that the **L**-accessibility relation $\rhd$ and the finite-**L**-frames of Figure 13 (page 63) are closely related. We also mentioned on page 11 that there is a duality between the explicit tableau methods and the implicit tableau methods. We now briefly explain these comments by way of an alternative labelled tableau system $\mathcal{LC}^*K45$ for logic **K45**.

Consider the system $\mathcal{LC}^*K45 = \mathcal{LC}PC \cup \{\ (l\pi_1)\ ,\ (l4^r\pi)\ ,\ (l4^r_1)\ ,\ (lK)\ \}$ where the new rules are as given below:

$$(l4^r\pi)\quad \frac{1.n :: \neg\Box P}{1 :: \neg\Box P} \qquad (l4^r_1)\quad \frac{1.n :: \Box P}{1 :: \Box P} \qquad (lK)\quad \frac{\sigma :: \Box P}{\sigma.n :: P}$$

$$(l\pi_1)\quad \frac{1 :: \neg\Box P}{1.n :: \neg P} \text{ where } 1.n \text{ is new to the current branch}$$

Note: except for $1.n$ in the rule $(l\pi_1)$, each label in the numerator and denominator must already exist on the branch.

The system $\mathcal{LC}^*K45$ does not fit into the mould of our other labelled systems since: it has *two* π-rules, neither of which is the usual (π) rule; the $(l4^r\pi)$ rule does *not* create a successor but merely moves a π-formula from world $1.n$ to the root world 1; and the $(l\pi_1)$ rule is a special case of the usual (π) rule, and creates a successor for a π formula only if its label is the root label 1. We therefore need to modify the systematic procedure slightly so that one of the mutually exclusive rules $(l4^r\pi)$ or $(l\pi_1)$ is applied to the chosen (awake) π-formula as is appropriate. Then a π-formula with a label $\sigma \neq 1$ cannot cause the creation of a successor and a systematic $\mathcal{LC}^*K45$-tableau for a finite X will contain labels of length at most 2. Furthermore, even though the logic is transitive, we do not need any check for periodicity since every systematic tableau is guaranteed to terminate for finite X.

Theorem 6.7.1. *The rules of $\mathcal{LC}^*K45$ are sound for* **K45***-frames.*

Proof: We have to show that if the numerator is **K45**-satisfiable then so is each denominator. So as in Section 6.3 (page 82), suppose there is some **K45**-model $\mathcal{M}$ and an **L**-interpretation I under which each numerator is **K45**-satisfiable in $\mathcal{M}$.

Proof for $(l4^r\pi)$ **:** If $I(1.n) \models \neg\Box P$ then $1 \rhd 1.n$ gives $I(1)RI(1.n)$ which gives $I(1) \models \Diamond\neg\Box P$ which is $I(1) \models \Diamond\Diamond\neg P$. Then, by the variant $\Diamond\Diamond A \rightarrow \Diamond A$ of the transitivity axiom 4 we have $I(1) \models \Diamond\neg P$, that is, $I(1) \models \neg\Box P$ as required.

Proof for $(l4^r_1)$ **:** If $I(1.n) \models \Box P$ then $1 \rhd 1.n$ gives $I(1)RI(1.n)$, giving $I(1) \models \Diamond\Box P$, which by the euclidean axiom $\Diamond\Box A \rightarrow \Box A$ gives $I(1) \models \Box P$, as required.

Proof for $(l\pi_1)$ **:** The rule $(l\pi_1)$ is just an instance of $(l\pi)$ and we know the latter is sound for all Kripke frames. ∎

Theorem 6.7.2. *The calculus* $\mathcal{LC}^*K45$ *is complete with respect to* **K45**-*frames.*

Proof: We have to show that if the systematic tableau for X is open then some open branch $\mathcal{B}$ gives an **K45**-downward-saturated set of labelled formulae (see page 89).

Very well, suppose the systematic tableau for X is open. Choose an open branch $\mathcal{B}$. The branch must be closed with respect to all the rules of $\mathcal{LC}^*K45$ in the appropriate sense (page 90) since this is a consequence of the systematic procedure itself rather than the form of the rules. The clauses **0)** to **3)** of the definition of **K45**-downward-saturated go through as before. For clause **4)** note that $1 \rhd 1.n$ and $1.n \rhd 1.m$ for all n and m, where n and m are integers, captures **K45**-accessibility over $lab(\mathcal{B})$ completely since $\mathcal{B}$ contains labels of length at most 2. The derivation below left shows that clause **4)** must be satisfied while the derivation below right shows that clause **5)** must also be satisfied

$$\dfrac{\dfrac{1.n :: \Box P}{1 :: \Box P}\,(l4^r_1)}{1.m :: P}\,(lK) \qquad \dfrac{\dfrac{1.n :: \neg\Box P}{1 :: \neg\Box P}\,(l4^r\pi)}{1.m :: \neg P}\,(l\pi_1)$$

Thus X is **K45**-satisfiable under the identity **L**-interpretation $I(\sigma) = \sigma$ in the **K45**-model $\langle lab(\mathcal{B}), \rhd, V\rangle$ as defined in Lemma 6.5.1 on page 89. ∎

The new rules of $\mathcal{LC}^*K45$ are essentially the operations that we required in the completeness proofs for $\mathcal{C}^\dagger$**K45** on page 38. Thus $\mathcal{LC}^*K45$ *implements* the completeness proof for $\mathcal{C}^\dagger$**K45**, but $\mathcal{LC}^*K45$ is cut-free! Furthermore, the **K45**-model created by the completeness proof for $\mathcal{LC}^*K45$ (above) is also a finite-**K45**-frame as defined on page 63. The extra power of rules that look backward against R, like $(l4^r_1)$ and $(l4^r\pi)$, have allowed us to eliminate even analytic cut.

For most cases, $\mathcal{LC}^*K45$ will be more efficient than $\mathcal{LC}K45$ due to the restriction that labels be at most length 2. Given a finite X, the number of prefixes of length 2 on any branch of a systematic tableau for X can be bounded by extending Lemma 6.4.2; see Massacci [Mas94] or Fitting [Fit83]. Hence, as pointed out to me by Massacci, we may even be able determine the complexity of the decision and satisfiability problems for **K45** using this system, although such results are already known for most of the basic logics; see [Lad77, HM85].

The system **KE** of Mondadori [DM94] has already been described in another chapter in this handbook. Clearly, it should be possible to extend all our modal tableau systems by modifying our tableau rules to incorporate the rule (PB). The only work along these lines that I know of is the work of Artosi, Governatori and coworkers [AG94] who use both (PB) and labelled tableaux, but where the labels are allowed to contain *variables* as well as constants. A branch is now closed if it contains some $\sigma :: A$ and some $\tau :: \neg A$ as long as the labels σ and τ are unifiable as strings with different string unification algorithms for different modal logics. The rule (PB) is also driven by string unification of labels. That is, the restrictions on the reachability relation are not built into a notion like **L**-accessibility, but into the unification algorithms. The main advantage is that we can now "detect" closure subject to a constraint that two given labels unify.

Ohlbach [Ohl93] has also studied such systems but in a different guise, for Ohlbach literally translates modal logics into classical first-order logic.

Any method that uses labels is really translating the modal logic into classical first-order logic since all these methods use a label of "universal force" for $\Box$-formulae and use a label of "existential force" for $\Diamond$-formulae. The recent work of Russo [Rus95] makes these intuitions explicit.

Acknowledgements:

I would like to thank Melvin Fitting, Jean Goubault, Alain Heuerding and Minh Ha Quang for their comments on an earlier draft. Particular thanks to Fabio Massacci for many useful comments and fruitful discussions.

References

[AE89] Y. Auffray and P. Enjalbert. Modal theorem proving: An equational viewpoint. In *11th International Joint Conference on Artificial Intelligence*, pages 441–445, 1989.

[AG94] A. Artosi and G. Governatori. Labelled model modal logic. In *Proceedings of the CADE-12 Workshop on Automated Model Building*, pages 11–17, 1994.

[Ame93] Martin Amerbauer. *Schnittfreie Tableau- und Sequenzenkalküle für Normale Modale Ausagenlogiken.* PhD thesis, Naturwissenschaftliche Fakultät der Universität Salzburg, 1993.

[Avr84] Arnon Avron. On modal systems having arithmetical interpretations. *Journal of Symbolic Logic*, 49:935–942, 1984.

[Avr94] Arnon Avron. The method of hypersequents in proof theory of propositional non-classical logics. Technical Report 294-94, Institute of Computer Science, Tel Aviv University, Israel, 1994.

[BB87] Behnam Banieqbal and Howard Barringer. Temporal logic with fixed points. In *Proc. Workshop on Temporal Logic in Specification, LNCS 398*, 1987.

[Bel85] G. Bellin. A system of natural deduction for GL. *Theoria*, 51:89–114, 1985.

[Bet53] E. W. Beth. On Padoa's method in the theory of definition. *Indag. Math.*, 15:330–339, 1953.

[Bet55] E. W. Beth. Semantic entailment and formal derivability. *Mededelingen der Koninklijke Nederlandse Akademie van Wetenschappen, Afd. Letterkunde*, 18:309–342, 1955.

[BG86] M. Borga and P. Gentilini. On the proof theory of the modal logic Grz. *ZML (now called Mathematical Logic Quarterly)*, 32:145–148, 1986.

[BHE95] P. Baumgartner, R. Hähnle, and J. Possega (Ed.). Proceedings of the fourth workshop on theorem proving with analytic tableaux and related methods. LNAI 918, 1995.

[Boo79] G. Boolos. *The Unprovability of Consistency.* Cambridge University Press, 1979.

[Bor83] M. Borga. On some proof theoretical properties of the modal logic GL. *Studia Logica*, 42:453–459, 1983.

[Bor93] Tijn Borghuis. Interpreting modal natural deduction in type theory. In Maarten de Rijke, editor, *Diamonds and Defaults*, pages 67–102. Kluwer Academic Publishers, 1993.

[BP95] Bernhard Beckert and Joachim Posegga. leanT^AP: Lean tableau-based deduction. *Journal of Automated Reasoning*, 15(3):339–358, 1995.

[BS84] R. A. Bull and K. Segerberg. Basic modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Volume II: Extensions of Classical Logic*, pages 1–88. D. Reidel, 1984.

[Bul65] R. A. Bull. An algebraic study of Diodorean modal systems. *Journal of Symbolic Logic*, 30(1):58–64, 1965.

[Bul85] R. A. Bull. Review of 'Melvin Fitting, Proof Methods for Modal and Intuitionistic Logics, Synthese Library, Vol. 169, Reidel, 1983'. *Journal of Symbolic Logic*, 50:855–856, 1985.

[Bur84] J. Burgess. Basic tense logic. In D. Gabbay and F. Guenthner, editors, *Extensions of Classical Logic*, volume II of *Handbook of Philosophical Logic*, pages 1–88. Reidel, Dordrecht, 1984.

[Cat88] L Catach. TABLEAUX: A general theorem prover for modal logics. In *Proc. International Computer Science Conference: Artificial Intelligence: Theory and Applications*, pages 249–256, 1988.

[Cat91] Laurant Catach. TABLEAUX: A general theorem prover for modal logics. *Journal of Automated Reasoning*, 7:489–510, 1991.

[Cer93] Claudio Cerrato. Modal sequents for normal modal logics. *Mathematical Logic Quarterly (previously ZML)*, 39:231–240, 1993.

[Che80] B. F. Chellas. *Modal Logic: An Introduction.* Cambridge University Press, 1980.

[CL73] G. L. Chang and R. G. T. Lee. *Symbolic logic and mechanical theorem proving.* Academic Press, New York, 1973.

[Cur52] H. B. Curry. The elimination theorem when modality is present. *Journal of Symbolic Logic*, 17:249–265, 1952.

[DM94] M. D'Agostino and M. Mondadori. The taming of the cut. classical refutations with analytic cut. *Journal of Logic and Computation*, 4:285–319, 1994.

[EG82] Éigirdas Gudzhinskas. Syntactical proof of the elimination theorem for von Wright's temporal logic. *Mat. Logika Primenen*, 2:113–130, 1982.

[EH85] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30:1–24, 1985.

[Eme85] E. Allen Emerson. Automata, tableaux, and temporal logics. In *Proc. Logics of Programs 1985, LNCS 193*, pages 79–87, 1985.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume Volume B: Formal Models and Semantics, chapter 16. MIT Press, 1990.

[ES83] E. A. Emerson and A. Prasad Sistla. Deciding branching time logic: A triple exponential decision procedure for CTL^*. *Proc. Logics of Programs, LNCS 164*, pages 176–192, 1983.

[ES84] E. A. Emerson and A. Prasad Sistla. Deciding branching time logic. In *Proc. 16th ACM Symposium Theory of Computing*, pages 14–24, 1984.

[ES88] E. Allen Emerson and Jai Srinivasan. Branching time temporal logic. In *Proc. School/Workshop on Linear Time, Branching Time and Partial Order In Logics and Models of Concurrency, Nederlands, 1988, LNCS 354*, pages 123–172, 1988.

[Fis91] Michael Fisher. A resolution method for temporal logic. In *Proc. 12th Internatinal Joint Conference on Artificial Intelligence 1991*, pages 99–104. Morgan-Kaufmann, 1991.

[Fit73] Melvin Fitting. Model existence theorems for modal and intuitionistic logics. *Journal of Symbolic Logic*, 38:613–627, 1973.

[Fit83] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 169 of *Synthese Library*. D. Reidel, Dordrecht, Holland, 1983.

[Fit93] Melvin Fitting. Basic modal logic. In D. Gabbay et al, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming: Logical Foundations*, volume 1, pages 365–448. Oxford University Press, 1993.

[FS91] Alan M. Frisch and Richard B. Scherl. A general framework for modal deduction. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proc. 2nd Conference on Principles of Knowledge Representation and Reasoning*. Morgan-Kaufmann, 1991.

[Gab87] Dov Gabbay. The declarative past and imperative future: Executable temporal logic for interactive systems. In *Proc. Workshop on Temporal Logic in Specification, LNCS 398*, pages 409–448. Springer-Verlag, 1987.

[Gab9x] Dov Gabbay. *Labelled Deductive Systems*. Oxford University Press (to appear), 199x.

[Gen35] G. Gentzen. Unterschungen über das logische schliessen. *Mathematische Zeitschrift*, 39:176–210 and 405–431, 1935. English translation: Investigations into logical deduction, in The Collected Papers of Gerhard Gentzen, edited by M. E. Szabo, pp 68-131, North-Holland, 1969.

[Gen91a] Ian Gent. *Analytic Proof Systems for Classical and Modal Logics of Restricted Quantification*. PhD thesis, Dept. of Computer Science, University of Warwick, Coventry, England, 1991.

[Gen91b] Ian Gent. Theory tableaux. Technical Report 91-62, Mathematical Sciences Institute, Cornell University, 1991.

[Gen93] Ian Gent. Theory matrices (for modal logics) using alphabetical monotonicity. *Studia Logica*, 52(2):233–257, 1993.

[GHH95] R. Goré, W. Heinle, and A. Heuerding. Relations between propositional normal modal logics: an overview. Technical Report TR-16-95, Automated Reasoning Project, Australian National University, 1995. Available via `http://arp.anu.edu.au/` on WWW.

[GHR94] D M Gabbay, I. M. Hodkinson, and M A Reynolds. *Temporal logic - mathematical foundations and computational aspects, Volume 1.* Oxford University Press, 1994.

[Gol87] R. I. Goldblatt. *Logics of Time and Computation.* CSLI Lecture Notes Number 7, Center for the Study of Language and Information, Stanford, 1987.

[Gor91] Rajeev Goré. Semi-analytic tableaux for modal logics with applications to nonmonotonicity. *Logique et Analyse*, 133-134:73–104, 1991. (printed in 1994).

[Gor92] Rajeev Goré. Cut-free sequent and tableau systems for propositional normal modal logics. Technical Report 257, University of Cambridge, England, June, 1992.

[Gor94] Rajeev Goré. Cut-free sequent and tableau systems for propositional Diodorean modal logics. *Studia Logica*, 53:433–457, 1994.

[Gou84] G. Gough. Decision procedures for temporal logics. Master's thesis, Dept. of Computer Science, University of Manchester, England, 1984.

[Han66] William Hanson. Termination conditions for modal decision procedures (abstract only). *Jornal of Symbolic Logic*, 31:687–688, 1966.

[HC68] G. E. Hughes and M. J. Cresswell. *Introduction to Modal Logic.* Methuen, London, 1968.

[HC84] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic.* Methuen, London, 1984.

[Hin55] K. J. J. Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:3–55, 1955.

[HM85] J. Y. Halpern and Y. Moses. A guide to the modal logics of knowledge and belief: Preliminary draft. In *Proc. International Joint Conference on Artificial Intelligence*, pages 480–490, 1985.

[HSZ] Alain Heuerding, Michael Seyfried, and Heinrich Zimmermann. Efficient loop-check for backward proof search in some non-classical logics. Submitted to Tableaux 96.

[Hud94] J. Hudelmaier. On a contraction free sequent calculus for the modal logic S4. In *Proceedings of the 3rd Workshop on Theorem Proving with Analytic Tableaux and Related Methods.* Technical Report TR-94/5, Department of Computing, Imperial College, London, 1994.

[Kam68] Johan Anthony Willem Kamp. *Tense Logic and the Theory of Linear Order.* PhD thesis, Dept. of Philosophy, University of California, USA, 1968.

[Kan57] S. Kanger. *Provability in Logic.* Stockholm Studies in Philosophy, University of Stockholm, Almqvist and Wiksell, Sweden, 1957.

[Kaw87] Hiroya Kawai. Sequential calculus for a first order infinitary temporal logic. *ZML (now Mathematical Logic Quarterly)*, 33:423–432, 1987.

[Kaw88] Hiroya Kawai. Completeness theorems for temporal logics T_Ω and $\Box T_\Omega$. *ZML (now Mathematical Logic Quarterly)*, 34:393–398, 1988.

[KH94] M. Koshimura and R. Hasegawa. Modal propositional tableaux in a model generation theorem prover. In *Proceedings of the 3rd Workshop on Theorem Proving with Analytic Tableaux and Related Methods.* Technical Report TR-94/5, Department of Computing, Imperial College, London, 1994.

[Kra96] M. Kracht. Power and weakness of the modal display calculus. In H. Wansing, editor, *Proof Theory of Modal Logics.* Kluwer, 1996. to appear.

[Kri59] Saul Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24(1):1–14, March 1959.

[Kri63] Saul Kripke. Semantical analysis of modal logic I: Normal modal propositional calculi. *Zeitschrift für Mathematik Logik und Grundlagen der Mathematische*, 9:67–96, 1963.

[Lad77] Richard Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing*, 6(3):467–480, 1977.

[Mas91] Andrea Masini. 2-sequent calculus: Classical modal logic. Technical Report TR-13/91, Universita Degli Studi Di Pisa, 1991.

[Mas92] Andrea Masini. 2-sequent calculus: A proof theory of modalities. *Annals of Pure and Applied Logic*, 58:229–246, 1992.

[Mas93] Andrea Masini. 2-sequent calculus: Intuitionism and natural deduction. *Journal of Logic and Computation*, 3(5):533–562, 1993.

[Mas94] Fabio Massacci. Strongly analytic tableaux for normal modal logics. In Alan Bundy, editor, *Proc. CADE-12, LNAI 814*, pages

723–737. Springer, 1994.

[Mas95a] Fabio Massacci. Personal communication, December 1995.

[Mas95b] Fabio Massacci. Simple steps tableaux for modal logics. Submitted for publication, 1995.

[Mer92] Stephan Merz. Decidability and incompleteness results for first-order temporal logics of linear time. *Journal of Applied Non-Classical Logic*, 2(2), 1992.

[MMO95] P. Miglioli, U. Moscato, and M. Ornaghi. Refutation systems for propositional modal logics. In P. Baumgartener, R. Hähnle, and J. Posegga, editors, *Proceedings of the 4th Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, volume LNAI 918, pages 95–105. Springer-Verlag, 1995.

[Moo85] R. C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25:272–279, 1985.

[Mor76] C. G. Morgan. Methods for automated theorem proving in non-classical logics. *IEEE Transactions on Computers*, C-25(8):852–862, 1976.

[MSS88] David E Muller, Ahmed Saoudi, and Paul E Schupp. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *Proc. Logics in Computer Science*, pages 422–427, 1988.

[MST91] W. Marek, G. Schwarz, and M. Truszczynski. Modal nonmonotonic logics: ranges, characterisation, computation. Technical Report 187-91, Dept. of Computer Science, University of Kentucky, USA, 1991.

[NFKT87] H Nakamura, M Fujita, S Kono, and H Tanaka. Temporal logic based fast verification system using cover expressions. In C H Séquin, editor, *VLSI '87, Proceedings of the IFIP TC 10/WG 10.5 International Conference on VLSI, Vancouver*, pages 101–111, 1987.

[Ohl90] H. J. Ohlbach. Semantics based translation methods for modal logics. Technical Report SEKI Report SR-90-11, Universität Kaiserslautern, Postfach, 3049, D-6750, Kaiserslautern, Germany, 1990.

[Ohl93] H. J. Ohlbach. Translation methods for non-classical logics: An overview. *Bulletin of the Interest Group in Pure and Applied Logics*, 1(1):69–89, 1993.

[OM57a] M. Ohnishi and K. Matsumoto. Corrections to our paper 'Gentzen method in modal calculi I'. *Osaka Mathematical Journal*, 10:147, 1957.

[OM57b] M. Ohnishi and K. Matsumoto. Gentzen method in modal calculi I. *Osaka Mathematical Journal*, 9:113–130, 1957.

[OM59] M. Ohnishi and K. Matsumoto. Gentzen method in modal calculi II. *Osaka Mathematical Journal*, 11:115–120, 1959.

[OS88] F Oppacher and E Suen. HARP: A tableau-based theorem prover. *Journal of Automated Reasoning*, 4:69–100, 1988.

[Pae88] B Paech. Gentzen-systems for propositional temporal logics. In *Proc. 2nd Workshop on Computer Science Logics, LNCS 385*, pages 240–253, 1988.

[Pli91] Regimantas Pliuskevicius. Investigations of finitary calculus for a discrete linear time logic by means of finitary calculus. In *LNCS vol 502*, pages 504–528. Springer-Verlag, 1991.

[Pot83] Garrel Pottinger. Uniform, cut-free formulations of T, S4 and S5. *Abstract in JSL*, 48:900–901, 1983.

[Rau79] W. Rautenberg. *Klassische und Nichtklassische Aussagenlogik*. Vieweg, Wiesbaden, 1979.

[Rau83] W. Rautenberg. Modal tableau calculi and interpolation. *Journal of Philosophical Logic*, 12:403–423, 1983.

[Rau85] W. Rautenberg. Corrections for modal tableau calculi and interpolation by W. Rautenberg, JPL 12 (1983). *Journal of Philosophical Logic*, 14:229, 1985.

[Rau90] W. Rauteberg. Personal communication, December 5th, 1990.

[Rus95] Alessandra Russo. Modal labelled deductive systems. Technical Report TR-95/7, Dept. of Computing, Imperial College, London, 1995.

[SC85] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the Association for Computing Machinery*, 32(3):733–749, 1985.

[Sch92] Grigory Schwarz. Minimal model semantics for nonmonotonic modal logics. In *Proc. Logics in Computer Science*, 1992.

[Seg71] Krister Segerberg. An essay in classical modal logic (3 vols.). Technical Report Filosofiska Studier, nr 13, Uppsala Universitet, Uppsala, 1971.

[Shi91] Tatsuya Shimura. Cut-free systems for the modal logic S4.3 and S4.3GRZ. *Reports on Mathematical Logic*, 25:57–73, 1991.

[Shv89] Grigori F. Shvarts. Gentzen style systems for K45 and K45D. In A. R. Meyer and M. A. Taitslin, editors, *Logic at Botik '89, Symposium on Logical Foundations of Computer Science, LNCS 363*, pages 245–256. Springer-Verlag, 1989.

[Sla77] Ralph L. Slaght. Modal tree constructions. *Notre Dame Journal of Formal Logic*, 18(4):517–526, 1977.

[ST92] G. E. Schwarz and M. Truszczynski. Modal logic S4F and the minimal knowledge paradigm. In *Proc. of Theoretical Aspects of Reasoning About Knowledge*, 1992.

[Sti92] C. Stirling. Modal and temporal logics for processes. Technical report, Dept of Computer Science, Edinburgh University, 1992. ECS-LFCS-92-221.

[SV80] G. Sambin and S. Valentini. A modal sequent calculus for a fragment of arithmetic. *Studia Logica*, 34:245–256, 1980.

[SV82] G. Sambin and S. Valentini. The modal logic of provability: the sequential approach. *Journal of Philosophical Logic*, 11:311–342, 1982.

[Tru] M. Truszczynski. Embedding default logic into modal nonmonotonic logics. In W. Marek, A. Nerode, and V. S. Subramanian, editors, *Proc. of the First International Workshop on Logic Programming and Non-monotonic Reasoning.*

[Tru91] M. Truszczynski. Modal interpretations of default logic. In *International Joint Conference on Artificial Intelligence*, pages 393–398. Morgan Kaufmann, 1991.

[Val83] S. Valentini. The modal logic of provability: Cut-elimination. *Journal of Philosophical Logic*, 12:471–476, 1983.

[Val86] S. Valentini. A syntactic proof of cut elimination for GL_{lin}. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 32:137–144, 1986.

[vB78] J. F. A. K. van Benthem and W. Blok. Transitivity follows from Dummett's axiom. *Theoria*, 44:117–118, 1978.

[vB83] J. F. A. K. van Benthem. *The Logic of Time: a model-theoretic investigation into the varieties of temporal ontology and temporal discourse.* Synthese library; vol. 156, Dordrecht: Reidel, 1983.

[vB84] Johan van Benthem. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 167–247. D. Reidel, 1984.

[VS83] S. Valentini and U. Solitro. The modal logic of consistency assertions of Peano arithmetic. *ZML (now Mathematical Logic Quarterly)*, 29:25–32, 1983.

[VW86] Moshe Vardi and Pierre Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32, 1986.

[Wal89] L. A. Wallen. *Automated Deduction in Nonclassical Logics: Efficient Matrix Proof Methods for Modal and Intuitionistic Logics.* MIT Press, 1989.

[Wan94] Heinrich Wansing. Sequent calculi for normal modal propositional logics. *Journal of Logic and Computation*, 4, 1994.

[Wol83] P. Wolper. Temporal logic can be more expressive. *Information and Control*, 56:72–99, 1983.

[Wol85] P. Wolper. The tableau method for temporal logic: an overview. *Logique et Analyse*, 110-111:119–136, 1985.

[Zem73] J. J. Zeman. *Modal Logic: The Lewis-Modal Systems*. Oxford University Press, 1973.