On Axiomatizations for Propositional Logics of Programs

P.M.W. Knijnenburg

RUU-CS-88-34
November 1988

Rijksuniversiteit Utrecht

Vakgroep informatica

Padualaan 14 3584 CH Utrecht
Corr. adres: Postbus 80.089, 3508 TB Utrecht
Telefoon 030-531454
The Netherlands

On Axiomatizations for Propositional Logics of Programs

P.M.W. Knijnenburg

Technical Report RUU-CS-88-34
November 1988

Department of Computer Science
University of Utrecht
P.O.Box 80.089, 3508 TB Utrecht
The Netherlands

Contents

Chapter 1

Introduction

Logics of Programs are formal systems for reasoning about the behavior of computer programs. To this end, computer programs are viewed as a means to enable certain logical formulae. The formulae may be propositional or first order, giving rise to propositional and first order program logics, respectively. In this paper, we focus attention on a propositional program logic, namely Propositional Dynamic Logic or PDL in short.

1.1 A historical note

Elements of the logic of programs can be traced back to the nineteen forty's where they appear in work by A.M. Turing and J. McCarthy. The subject, as we view it nowadays, originated with papers of Engeler [5] and of Floyd [7]. The ideas of Floyd were developed further by many authors and the logic of partial correctness, also called *Floyd-Hoare logic*, has been studied intensively.

In 1969 Salwicki [31] formulated the algorithmic logic **AL**, following the work of Engeler. **AL** was developed further by a group in Warsaw. Later, Mirkowska [19] gave a propositional version of **AL**. In 1976, Pratt [25] introduced Modal Logic to computer science, which proved to be very fruitful. Fisher and Ladner [6] gave the definition of Propositional Dynamic Logic, following Pratt, and proved decidability of the logic by means of a filtration technique, borrowed from Modal Logic. Segerberg [33] gave a complete axiomatization for **PDL** and several completeness proofs have now appeared in the literature, notably a proof by Berman [3], using in fact a standard completeness technique from Modal Logic (*c.f.* [8]).

1.2 Modal logic

The origins of Modal Logic seem to date back to Aristotle; it was the subject of intensive research in the Middle Ages. In the first half of the twentieth century, Modal Logic appeared in its commonly known form. Modal Logic can be viewed as an extension of classical propositional logic, by introducing the operator $\Box$. This operator has several

readings (which in turn define different logics such as Temporal Logic or Deontic Logic) but is always of a dynamic nature. Given a formula ϕ, the readings of $\Box\phi$ include:

- ϕ is always true;
- It is necessarily true that ϕ;
- ϕ ought to be true;
- It is known that ϕ;
- After the program terminates, ϕ holds.

We define the operator $\Diamond$ to be $\neg\Box\neg$. Readings of $\Diamond$ follow from the interpretations of $\Box$. The precise nature of $\Box$ is given by axioms like $\Box\phi \rightarrow \Box\Box\phi$. Different sets of such axioms define different Modal Logics. In 1959, Kripke introduced the notion of the (later called) *Kripke model* as any structure underlying Modal Logic. Basicly, a Kripke model is a triple $\mathcal{M} = (S, R, V)$ where

- S is a set of states;
- $R \subseteq S \times S$ is a binary relation on S;
- V is a valuation for the predicate symbols.

We can now define the relation $\models$, where $\mathcal{M}, s \models \phi$ means that "ϕ holds in $\mathcal{M}$ at state s" by induction on the complexity of ϕ:

- $\mathcal{M}, s \models p$ iff $s \in V(p)$ for p a primitive predicate symbol;
- $\mathcal{M}, s \models \phi \vee \psi$ iff $\mathcal{M}, s \models \phi$ or $\mathcal{M}, s \models \psi$;
- $\mathcal{M}, s \models \neg\phi$ iff $\mathcal{M}, s \not\models \phi$.

A formula $\Box\phi$ is interpreted in a Kripke model as

$$\mathcal{M}, s \models \Box\phi \text{ iff for each } t \in S, \text{ if } (s,t) \in R, \text{ then } \mathcal{M}, t \models \phi.$$

Different sets of axioms for $\Box$ were proved to coincide with different first-order definable properties of R (*e.g.*, $\Box\phi \rightarrow \Box\Box\phi$ coincides with the property of R being transitive). It can be shown, however, that there are first-order definable properties of R that are not axiomatizable in Modal Logic, irreflexivity of R is a noteworthy example. On the other hand, there exist schemata for $\Box$ that do not define any first-order property of R; the schema

$$\Box\Diamond\phi \rightarrow \Diamond\Box\phi$$

is one example.

An immediate extension is obtained by allowing a set of relations $\{R_i \mid i \in I\}$ to be incorporated in the logic, with each relation R_i having its own necessity operator $\Box_i$. This is called *multimodal logic*.

1.3 Dynamic logic

Pratt [25] recognized the possibility of modeling program logics by means of Multimodal Logic. If we view a program to be defined by its *input/output*, or *before/after*, behavior then Modal Logic provides a natural framework in which we can develop such a program logic. Each program α has associated its "own" modal operator $\Box_\alpha$, or $[\alpha]$. For a propositional program logic we can take a set of primitive programs and rules that determine how more complex programs can be built. With each rule we can define how the modal operator for the more complex program relates to the modal operators of the building blocks. For instance, program composition is defined by the rule: at a state s, $[\alpha;\beta]\phi$ holds if and only if $[\alpha][\beta]\phi$ holds. The modal operators for the primitive programs are parameters in this approach.

Propositional Dynamic Logic is defined to be the Multimodal Logic in which the programs are regular expressions over the set of primitive programs. Thus the program connectives are ";", "$\cup$" and "$\star$" which are usually interpreted as composition, choice and iteration, respectively. Note, however, that we can take any program construction as long as we can express their modality. An important restriction, however, is the requirement of the algorithmic solvability of the validity problem of the resulting logic. PDL with regular programs is known to be decidable, but PDL with linear context-free programs is not decidable. In fact, the latter problem is known to be Π^1_1-complete, that is, highly undecidable. Furthermore, it can be proved that PDL with all r.e. programs equals the infinitary logic of equality $L_{\omega_1\omega}$.

Several variants have been proposed of the original definition of PDL. These variants include

- only allowing deterministic primitive programs (DPDL);
- a primitive assertion *repeat* for programs, which holds of a program if that program can be executed ad infinitum (RPDL);
- a primitive assertion *loop* for programs, which holds of a program if that program may never terminate (LPDL);
- a converse operator for programs which yields a program that executes the original program "backwards" (CPDL).

See [10] for precise definitions and results and for references to the original literature.

1.4 Outline

This paper is organized as follows. In chapter 2, the basic definitions, syntax and semantics for PDL are given. We give the Segerberg axiomatization which we prove complete in chapter 3. We also give an infinitary axiom system which we prove complete using a technique proposed by Berman [3]. We then state a slightly more general technique for proving completeness of axiomatizations for (variants) of PDL based on this infinitary axiom system and give applications. In chapter 3, we prove the existence

of a Universal Model using model theoretic arguments and prove completeness of the axiomatization using this Universal Model. We also use the Universal Model to give a different proof of the Small Model theorem of Fisher and Ladner [6]. In chapter 4 we discuss the above mentioned assertion *repeat* and show that this assertion is definable in the infinitary logic. In chapter 5, we describe a fragment of Propositional Dynamic Logic of Context-Free Programs. We give an axiomatization and prove it complete using the technique of chapter 2. In chapter 6, some related topics are discussed. We review Propositional Algorithmic Logic as formulated by Mirkowska [19] and compare the proof of completeness from that paper with ours. Finally, we discuss two different approaches to the problem of introducing time in the logic, namely, the linear time and the branching time approach.

Acknowledgements

The author wishes to thank Jan van Leeuwen for introduction to the subject and Pim Kars for references to Modal and Temporal Logic.

Chapter 2

Propositional Dynamic Logic

In this chapter we give the definitions of the syntax and semantics of a formal system for reasoning about programs. To this end, we define a class of programs which can enable propositions by means of a *possibility operator* $\Diamond$. Thus, when α is a program and ϕ is a proposition, $\alpha\Diamond\phi$ (which we will abbreviate as $\langle\alpha\rangle\phi$) states "program α can terminate with ϕ holding upon termination". The resulting logic is interpreted over Kripke structures and we will give an axiomatization for the logic that is complete, *i.e.* validity and derivability coincide.

2.1 Syntax and semantics

The syntax of Propositional Dynamic Logic PDL has as its basis two disjoint countable sets of primitive symbols, namely the set

$$\Phi_0 = \{p_0, p_1, \ldots\}$$

of primitive *predicate symbols*, and the set

$$\Pi_0 = \{a_0, a_1, \ldots\}$$

of primitive *program symbols*. From these base sets we recursively define the sets of PDL propositions Φ and programs Π:

1. $\Phi_0 \subseteq \Phi$;
2. if $\phi, \psi \in \Phi$ then $\phi \vee \psi, \neg\phi \in \Phi$;
3. if $\alpha \in \Pi$ and $\phi \in \Phi$ then $\langle\alpha\rangle\phi \in \Phi$;
4. $\Pi_0 \subseteq \Pi$;
5. if $\alpha, \beta \in \Pi$ then $\alpha \cup \beta, \alpha;\beta, \alpha^\star \in \Pi$;
6. if $\phi \in \Phi$ then $\phi? \in \Pi$.

In addition we abbreviate $\neg(\neg\phi \vee \neg\psi)$ to $\phi \wedge \psi$; $\neg\phi \vee \psi$ to $\phi \rightarrow \psi$; $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ to $\phi \leftrightarrow \psi$. We further abbreviate $\neg\langle\alpha\rangle\neg\phi$ to $[\alpha]\phi$.

First we give an informal semantics for the above constructions: the meaning of the propositional connectives is exactly like in ordinary, classical propositional logic CPC. Therefore, PDL can be seen as an extension of CPC, *i.e.* all tautologies of CPC are valid PDL formulae. Primitive programs are exactly what their name suggests: uninterpreted programs or *input/output relations*, which is essentially the way we view programs in general. That is, programs are black boxes and their input/output behavior completely characterizes their relevant aspects; we identify two programs if and only if they constitute the same input/output relation. The meaning of the operator ; is program concatenation; thus, $\alpha;\beta$ means "first execute program α and then execute β". $\cup$ means nondeterministic choice; $\alpha \cup \beta$ means "choose nondeterministically program α or β and execute it". The $\star$-operator is a nondeterministic looping operator and $\alpha^\star$ means "execute α a nondeterministically chosen number of times". In the sequel we often abbreviate $\alpha;\alpha;\cdots;\alpha$ (n times) to α^n. Thus $\alpha^\star$ can be viewed as "choose n nondeterministically and execute α^n". The operator ? is a testing operator and $\phi?$ means "test ϕ and proceed if true".

The operator $\Diamond$ is the usual modal operator and the meaning of $\langle\alpha\rangle\phi$ is "program α can be executed with ϕ holding upon termination". Its dual, $[\alpha]\phi$, therefore means "whenever program α terminates, ϕ holds". Note that these operators give rise to two important aspects of programs, namely, when $\langle\alpha\rangle$*true* is valid, then α can terminate, and when $[\alpha]$*false* is valid, then α never terminates. We are also able to express *partial correctness* of programs, $\phi \rightarrow [\alpha]\psi$.

Formally, PDL formulae are interpreted over Kripke structures.

Definition 2.1 *A* Kripke structure *is a triple* $\mathcal{A} = (W^{\mathcal{A}}, \pi^{\mathcal{A}}, \rho^{\mathcal{A}})$ *where*

- $W^{\mathcal{A}}$ *is a set of states;*
- $\pi^{\mathcal{A}} : \Phi_0 \mapsto 2^{W^{\mathcal{A}}}$ *is an interpretation function for the primitive predicate symbols;*
- $\rho^{\mathcal{A}} : \Pi_0 \mapsto 2^{W^{\mathcal{A}} \times W^{\mathcal{A}}}$ *is an interpretation function for the primitive program symbols.*

Usually we write a Kripke structure as $\mathcal{A} = (W, \pi, \rho)$ when no confusion can arise. We further use the terms "Kripke structure", "Kripke model", "structure" and "model" interchangeably. The interpretation functions extend to the whole sets Φ and Π:

- $\rho(\alpha \cup \beta) = \rho(\alpha) \cup \rho(\beta)$;
- $\rho(\alpha;\beta) = \rho(\alpha) \circ \rho(\beta)$, where $\circ$ is relation composition;
- $\rho(\alpha^\star) = \bigcup_{i<\omega} \rho(\alpha^i)$, the reflexive transitive closure of $\rho(\alpha)$;
- $\rho(\phi?) = \{(s,s) \in W \times W \mid s \in \pi(\phi)\}$;
- $\pi(\phi \vee \psi) = \pi(\phi) \cup \pi(\psi)$;
- $\pi(\neg\phi) = W - \pi(\phi)$;

- $\pi(\langle\alpha\rangle\phi) = \{s \in W \mid \exists t \in W.((s,t) \in \rho(\alpha) \wedge t \in \pi(\phi))\}$;

We say that a proposition ϕ is *satisfiable* in a structure $\mathcal{A}$ if and only if there exists a state s in $\mathcal{A}$ such that $s \in \pi(\phi)$ and we write $\mathcal{A}, s \models \phi$. We omit $\mathcal{A}$ when it is clear from the context. We say that ϕ is *$\mathcal{A}$-valid* and write $\mathcal{A} \models \phi$ if $\mathcal{A}, s \models \phi$ for each $s \in W$. We say that ϕ is *valid* and write $\models \phi$ if ϕ is $\mathcal{A}$-valid for every structure $\mathcal{A}$. Clearly, ϕ is valid if and only if $\neg\phi$ is not satisfiable.

In the sequel of this paper we use $\phi, \psi, \ldots$ to denote propositions and $\alpha, \beta, \ldots$ to denote programs.

2.2 Axiomatization

In this section we present an axiomatization for PDL as proposed by Segerberg [33]. He claimed this axiomatization to be complete and several completeness theorems are established in the literature. In the next chapter we give another proof of completeness of the axiom system by a technique which resembles the proof method proposed by Berman [3], which we review below.

Definition 2.2 *The set of axioms AX for* PDL *contains*

1. *axioms for propositional logic;*
2. $\langle\alpha\rangle\phi \wedge [\alpha]\psi \rightarrow \langle\alpha\rangle(\phi \wedge \psi)$;
3. $\langle\alpha\rangle(\phi \vee \psi) \leftrightarrow \langle\alpha\rangle\phi \vee \langle\alpha\rangle\psi$;
4. $\langle\alpha \cup \beta\rangle\phi \leftrightarrow \langle\alpha\rangle\phi \vee \langle\beta\rangle\phi$;
5. $\langle\alpha;\beta\rangle\phi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\phi$;
6. $\langle\psi?\rangle\phi \leftrightarrow \psi \wedge \phi$;
7. $\phi \vee \langle\alpha\rangle\langle\alpha^*\rangle\phi \rightarrow \langle\alpha^*\rangle\phi$;
8. $\langle\alpha^*\rangle\phi \rightarrow \phi \vee \langle\alpha^*\rangle(\neg\phi \wedge \langle\alpha\rangle\phi)$.

In addition we have the following inference rules:

1. *modus ponens: from ϕ, $\phi \rightarrow \psi$, infer ψ;*
2. *modal generalization: from ϕ, infer $[\alpha]\phi$, for any $\alpha \in \Pi$.*

As usual, we define a *derivation* to be a finite sequence of well-formed formulae, each of which is an instance of an axiom or the conclusion of an inference rule whose premisses occur earlier in the derivation. The last formula occurring in the derivation is called the *conclusion of the derivation*. If, for any formula ϕ, there exists a derivation of which ϕ is the conclusion, we say that ϕ is *derivable* and write $\vdash \phi$.

Axioms 1–3 are not particular for PDL but hold in all modal systems. Axiom 2 is easier in its dual form

$$[\alpha](\phi \to \psi) \to ([\alpha]\phi \to [\alpha]\psi).$$

This is the axiom K of Modal Logic and any logic which satisfies K and has a modal generalization rule, is called *normal* (*c.f.* [8]). Axiom 8 is called the *induction axiom*, and is better known in its dual form

$$\phi \wedge [\alpha^*](\phi \to [\alpha]\phi) \to [\alpha^*]\phi.$$

Note the resemblance between this axiom and the induction axiom in arithmetic. The intuition behind axiom 8 is that if a program α^* enables a proposition ϕ, then the proposition is always true or there is a point in the looping of the program where the proposition becomes true for the first time.

Note that we may not assume ϕ and then infer, with modal generalisation, $\vdash \phi \to [\alpha]\phi$ for all propositions ϕ and programs α. This schema is obviously unsound. This derivation is only valid when ϕ is.

Inspection of the system AX immediately gives us the next proposition.

Theorem 2.3 (Soundness Theorem) *If* $\vdash \phi$ *then* $\models \phi$.

A familiar fact of PDL is its lack of compactness. For an easy example, consider the infinite set Γ:

$$\begin{aligned} \Gamma &= \{\neg\phi, \neg\langle\alpha\rangle\phi, \neg\langle\alpha^2\rangle\phi, \ldots\} \cup \{\langle\alpha^*\rangle\phi\} \\ &= \Delta \cup \{\langle\alpha^*\rangle\phi\} \end{aligned}$$

Every finite subset $\Gamma' \subseteq \Gamma$ has a model: suppose $\langle\alpha^*\rangle\phi \in \Gamma'$ and let i be the largest integer such that $\neg\langle\alpha^i\rangle\phi \in \Gamma'$. Then each model $\mathcal{M}$ that satisfies $\neg\langle\alpha^j\rangle\phi$ for $j \leq i$ and $\langle\alpha^{i+1}\rangle\phi$, satisfies Γ'. Yet the whole set Γ cannot have a model, for Δ is precisely the definition of $\neg\langle\alpha^*\rangle\phi$.

2.3 An infinitary axiom system

Intuitively, the nature of the $\star$-operator requires an *infinitary* axiom system. We define the system AX_∞ as such an infinitary system. The induction axiom is replaced by an inference rule with an infinite set of premisses.

Definition 2.4 *The infinitary axiom system* AX_∞ *contains the following axioms.*

1. *All* **PDL** *axioms, except the Induction Axiom;*
2. $[\alpha^*]\phi \to [\alpha^i]\phi$, *for each* $i < \omega$;

In addition, we have the following inference rules:

1. *modus ponens: from* ϕ, $\phi \to \psi$, *infer* ψ;

2. *modal generalization: from* ϕ, *infer* $[\alpha]\phi$, *for any* $\alpha \in \Pi$;

3. ∞*-rule: from* $\{\psi \to [\alpha^i]\phi\}_{i<\omega}$, *infer* $\psi \to [\alpha^*]\phi$.

We treat $[\alpha^*]\phi$ as an abbreviation for $\bigwedge_{i<\omega}[\alpha^i]\phi$. By contraposition, we have, for each $i < \omega$,

$$\langle\alpha^i\rangle\phi \to \langle\alpha^*\rangle\phi.$$

We define a derivation in AX_∞ to be a countable sequence of well-formed formulae, each of which is either an instance of an axiom or the conclusion of an inference rule whose premisses occur earlier in the sequence. The last formula in the sequence is called the conclusion of the derivation and any formula ϕ for which such a derivation exists is called derivable or provable and we write $\vdash_\infty \phi$.

From the Soundness Theorem for AX, we immediately get a Soundness Theorem for AX_∞.

Theorem 2.5 (Soundness Theorem) *If* $\vdash_\infty \phi$, *then* $\models \phi$.

Theorem 2.6 1. *In the infinitary system* AX_∞, *the induction axiom is derivable.*

2. *In the Segerberg system* AX, $\vdash [\alpha^*](\phi \to [\alpha]\phi) \to (\phi \to [\alpha^n]\phi)$ *for each* $n < \omega$.

Proof.

1. Let $\psi = \phi \wedge [\alpha^*](\phi \to [\alpha]\phi)$. Then, by **CPC**, $\vdash_\infty \psi \to \psi$, or

$$\vdash_\infty \psi \to \phi \wedge [\alpha^*](\phi \to [\alpha]\phi).$$

An instance of Axiom 7 in its dual form is

$$([\alpha^*](\phi \to [\alpha])) \to ((\phi \to [\alpha]\phi) \wedge [\alpha][\alpha^*](\phi \to [\alpha]\phi)).$$

Hence

$$\vdash_\infty \psi \to \phi \wedge ((\phi \to [\alpha]\phi) \wedge [\alpha][\alpha^*](\phi \to [\alpha]\phi)).$$

and

$$\vdash_\infty \psi \to [\alpha]\phi \wedge [\alpha][\alpha^*](\phi \to [\alpha]\phi)$$

and by Axiom 3,

$$\vdash_\infty \psi \to [\alpha](\phi \wedge [\alpha^*](\phi \to [\alpha]\phi)).$$

So $\vdash_\infty \psi \to [\alpha]\psi$. With induction, we can show $\vdash_\infty \psi \to [\alpha^n]\psi$ for each $n < \omega$. We now may infer $\vdash_\infty \psi \to [\alpha^*]\psi$ from which, with propositional reasoning, follows

$$\vdash_\infty ([\alpha^*](\phi \to [\alpha]\phi)) \to (\phi \to [\alpha^*]\phi).$$

2. Using axiom 7, we have

$$\vdash [\alpha^*]\psi \to (\psi \wedge [\alpha][\alpha^*]\psi)$$

where $\psi = \phi \rightarrow [\alpha]\phi$. Assume $\vdash [\alpha^*]\psi$. Then $\vdash \phi \rightarrow [\alpha]\phi$ and $\vdash [\alpha][\alpha^*](\phi \rightarrow [\alpha]\phi)$. Another application of axiom 7 yields

$$\vdash [\alpha](\phi \rightarrow [\alpha]\phi) \wedge [\alpha^2][\alpha^*](\phi \rightarrow [\alpha]\phi).$$

We may now infer

$$[\alpha]\phi \rightarrow [\alpha^2]\phi.$$

As we have $\vdash \phi \rightarrow [\alpha]\phi$, we can deduce

$$\vdash \phi \rightarrow [\alpha^2]\phi.$$

And the theorem follows by induction on n. □

One sound inference rule in the system AX is the so-called *reflexive transitive closure rule* (*c.f.* [17]) which reads:

$$\frac{(\phi \vee \langle\alpha\rangle\psi) \rightarrow \psi}{\langle\alpha^*\rangle\phi \rightarrow \psi}$$

When we substitute $\langle\alpha^*\rangle\phi$ for ψ in the premise of this rule, the conclusion is *true*, for the premise is valid by axiom 7. Thus this rule says that $\langle\alpha^*\rangle\phi$ is the *least* (with respect to logical implication) PDL proposition to do so, which is consistent with the infinitary axiomatization for the $\star$-operator.

2.3.1 A completeness technique

In the following chapter we prove the completeness of the Segerberg axiom system using a Universal Model. A model $\mathcal{U}$ is called *universal* (*c.f.* [21]) if, for each model $\mathcal{M}$, there exists a mapping $\theta_{\mathcal{M}} : W^{\mathcal{M}} \mapsto W^{\mathcal{U}}$ such that for each state $s \in W^{\mathcal{M}}$ and each PDL formula ϕ:

$$\mathcal{M}, s \models \phi \text{ iff } \mathcal{U}, \theta_{\mathcal{M}}(s) \models \phi.$$

Berman [3] gave a completeness technique for PDL which we review in this section. We use this technique to prove completeness of AX_∞.

We first give some definitions. Let $\Pr(AX_\infty) = \{\phi \mid \vdash_\infty \phi\}$ be the set of all provable formulas of the axiom system AX_∞.

Definition 2.7 *Let Σ be a set of formulas and ϕ a formula.*

1. $\Sigma \vdash_\infty \phi$ *if and only if there is a (finite or countable) subset $\Sigma' \subseteq \Sigma$ such that $\vdash_\infty \bigwedge \Sigma' \rightarrow \phi$.*
2. *We say that Σ is* inconsistent *iff $\Sigma \vdash_\infty$ false.*
3. *We say that Σ is* consistent *iff Σ is not inconsistent.*
4. *Σ is* maximally consistent *iff Σ is consistent and for each $\phi \in \Phi$, either ϕ or $\neg\phi \in \Sigma$.*

We now define a model $\mathcal{A}$ by:

- $W^{\mathcal{A}} = \{s \subseteq \Phi \mid \Pr(AX_\infty) \subseteq s \text{ and } s \text{ is maximally consistent }\}$;
- $\pi^{\mathcal{A}}(p) = \{s \mid p \in s\}$ for primitive predicate p;
- $\rho^{\mathcal{A}}(a) = \{(s,t) \mid \forall\psi.([a]\psi \in s \Longrightarrow \psi \in t)\}$ for primitive program a.

Lemma 2.8 *For each proposition ϕ,*

$$\mathcal{A}, s \models \phi \text{ iff } \phi \in s.$$

Proof.

We proceed by induction on the complexity of ϕ. For ϕ a primitive predicate, the theorem holds by definition.

($\phi = \psi \vee \chi$). $\mathcal{A}, s \models \psi \vee \chi$ iff $\mathcal{A}, s \models \psi$ or $\mathcal{A}, s \models \chi$ iff, by induction hypothesis, $\psi \in s$ or $\chi \in s$ iff $\psi \vee \chi \in s$, by construction.

($\phi = \neg\psi$). $\mathcal{A}, s \models \neg\psi$ iff $\mathcal{A}, s \not\models \psi$ iff $\psi \notin s$ iff $\neg\psi \in s$.

($\phi = \langle\alpha\rangle\psi$). The only nontrivial case. We prove this case by induction on the structure of α.

First let $\alpha = a$ be a primitive program. $\mathcal{A}, s \models \langle a\rangle\psi$ iff there exists a state t such that $(s,t) \in \rho(a)$ and $\mathcal{A}, t \models \psi$. By induction hypothesis, $\psi \in t$ and by the definition of $\rho(a)$, $\langle a\rangle\psi \in s$. Conversely, suppose $\langle a\rangle\psi \in s$. Consider the set

$$\Gamma = \{\phi \mid [a]\phi \in s\}.$$

Claim. Γ is consistent.
Proof of claim. Suppose Γ is inconsistent. Then there exists $\Gamma' \subseteq \Gamma$ such that

$$\vdash_\infty \bigwedge \Gamma' \rightarrow \mathit{false}$$

or

$$\vdash_\infty \phi_1 \wedge \cdots \wedge \phi_n \wedge \cdots \rightarrow \mathit{false}$$

$$\vdash_\infty [a]\phi_1 \wedge \cdots \wedge [a]\phi_n \wedge \cdots \rightarrow [a]\mathit{false}$$

As $\vdash_\infty \mathit{false} \rightarrow \phi$ for all ϕ, we get, by Modal Generalization and axiom 2,

$$\vdash_\infty [a]\mathit{false} \rightarrow [a]\neg\psi$$

$$\vdash_\infty [a]\phi_1 \wedge \cdots \wedge [a]\phi_n \wedge \cdots \rightarrow [a]\neg\psi$$

Hence $[a]\neg\psi \in s$ or $\neg\langle a\rangle\psi \in s$. Contradiction.

Extend Γ to the set $\Gamma' = \Gamma \cup \{\psi\}$.

Claim. Γ' is consistent.
Proof of claim. Suppose Γ' is inconsistent. The only way inconsistency can occur is by $\{\psi\}$. So suppose there are $\phi_1, \ldots, \phi_m, \ldots \in \Gamma$ such that

$$\vdash_\infty \phi_1 \wedge \cdots \wedge \phi_m \wedge \psi \wedge \cdots \rightarrow \mathit{false}$$

Hence,

$$\vdash_\infty \phi_1 \wedge \cdots \wedge \phi_m \wedge \cdots \rightarrow \neg\psi$$

$$\vdash_\infty [a]\phi_1 \wedge \cdots \wedge [a]\phi_m \wedge \cdots \rightarrow [a]\neg\psi$$

Hence $[a]\neg\psi \in s$. Contradiction.

It is easy to see that the set $\Pr(AX_\infty) \cup \Gamma'$ is consistent. Hence this set can be extended to a maximally consistent set t.

Claim. $(s,t) \in \rho(a)$.
Proof of claim. $t \in W^{\mathcal{A}}$ by definition. And for all propositions ϕ:

$$[a]\phi \in s \Longrightarrow \phi \in \Gamma \subseteq \Gamma' \subseteq t$$

Hence $(s,t) \in \rho(a)$ by the definition of ρ.

By the last claim, since $\psi \in t$, $\mathcal{A}, s \vDash \langle a\rangle\psi$ and the case α is primitive, is proved. The other cases follow easily.

$\mathcal{A}, s \vDash \langle\chi?\rangle\psi$ iff $\mathcal{A}, s \vDash \chi \wedge \psi$ iff, by induction hypothesis, $\chi \wedge \psi \in s$ iff $\langle\chi?\rangle\psi \in s$.

$\mathcal{A}, s \vDash \langle\alpha \cup \beta\rangle\psi$ iff $\mathcal{A}, s \vDash \langle\alpha\rangle\psi \vee \langle\beta\rangle\psi$ iff $\langle\alpha\rangle\psi \vee \langle\beta\rangle\psi \in s$ iff $\langle\alpha \cup \beta\rangle\psi \in s$.

$\mathcal{A}, s \vDash \langle\alpha;\beta\rangle\psi$ iff $\mathcal{A}, s \vDash \langle\alpha\rangle\langle\beta\rangle\psi$ iff $\langle\alpha\rangle\langle\beta\rangle\psi \in s$ iff $\langle\alpha;\beta\rangle\psi \in s$.

Dually we prove $[\alpha^\star]\psi \in s$ iff $\mathcal{A}, s \vDash [\alpha^\star]\psi$. $\mathcal{A}, s \vDash [\alpha^\star]\psi$ iff, by definition of Kripke models, $\mathcal{A}, s \vDash [\alpha^n]\psi$ for each $n < \omega$, iff, by induction hypothesis, $[\alpha^n]\psi \in s$ for each $n < \omega$, iff, by the ∞-rule $[\alpha^\star]\psi \in s$. □

Corollary 2.9 *For each program α and proposition ϕ,*

$$\langle\alpha\rangle\phi \in s \text{ iff there exists a } t \in W \text{ such that } (s,t) \in \rho(\alpha) \text{ and } \phi \in t.$$

With Lemma 2.8 we can easily prove the completeness of the system AX_∞:

Theorem 2.10 (Completeness Theorem) *For each* **PDL** *formula ϕ, $\vdash_\infty \phi$ iff $\vDash \phi$.*

Proof.

One direction is the Soundness Theorem; for the other direction: let ϕ be such that $\nvdash_\infty \phi$. Then $\Pr(AX_\infty) \cup \{\neg\phi\}$ is consistent and can be extended to a maximally consistent set s by Lindenbaum's Theorem. Hence, $s \in W^{\mathcal{A}}$ and $\mathcal{A}, s \vDash \neg\phi$ by Lemma 2.8, which implies that ϕ is not valid or $\nvDash \phi$. □

The following theorem abstracts our technique for proving completeness.

Lemma 2.11 (Completeness Lemma) *Let AX' be any sound axiomatization for (a variant of)* **PDL**, *that is, $AX_\infty \subseteq AX'$. Construct the model $\mathcal{A}$ as indicated using AX'. Then, if*

$$\mathcal{A}, s \vDash \phi \text{ if and only if } \phi \in s$$

then

1. *AX' is complete;*

2. *$\mathcal{A}$ is a Universal Model.*

Proof.

1. Suppose ϕ is such that $\not\vdash \phi$. Then $\Pr(AX') \cup \{\neg\phi\}$ is consistent and can be extended to a maximally consistent set s by Lindenbaum's Theorem. Then $s \in W^{\mathcal{A}}$ and $\mathcal{A}, s \models \neg\phi$. So ϕ is not valid.

2. For each model $\mathcal{M}$ define the mapping $\theta_{\mathcal{M}} : W^{\mathcal{M}} \mapsto W^{\mathcal{A}}$ by

$$\theta_{\mathcal{M}}(s) = \{\phi \mid \mathcal{M}, s \models \phi\}.$$

$\theta_{\mathcal{M}}(s)$ is maximally consistent, for if $\psi \notin \theta_{\mathcal{M}}(s)$ then $\mathcal{M}, s \not\models \psi$; hence $\mathcal{M}, s \models \neg\psi$ and $\neg\psi \in \theta_{\mathcal{M}}(s)$. By Soundness, $\Pr(AX') \subseteq \theta_{\mathcal{M}}(s)$. Hence $\theta_{\mathcal{M}}(s) \in W^{\mathcal{A}}$. Then

$$\mathcal{M}, s \models \phi \text{ iff } \phi \in \theta_{\mathcal{M}}(s) \text{ iff } \mathcal{A}, \theta_{\mathcal{M}}(s) \models \phi$$

which implies that $\mathcal{A}$ is universal. □

2.3.2 Application: PDL with concurrency

In this section we give an application of our completeness technique for a variant of **PDL**. In this application we use the infinitary system AX_ω and the variant is an addition of axioms to this system.

Peleg [22] defined a variant of **PDL** by introducing the *concurrency operator* $\cap : \Pi \times \Pi \mapsto \Pi$ which has the following semantics for any Kripke model $\mathcal{M}$:

$$\mathcal{M}, s \models \langle \alpha \cap \beta \rangle \phi \text{ iff } \mathcal{M}, s \models \langle \alpha \rangle \phi \text{ and } \mathcal{M}, s \models \langle \beta \rangle \phi.$$

An axiomatization for the resulting logic follows easily.

Definition 2.12 *The set AX_C of axioms for* **PDL** *with concurrency contains:*

1. *the system AX_ω;*
2. $\langle \alpha \cap \beta \rangle \phi \leftrightarrow \langle \alpha \rangle \phi \wedge \langle \beta \rangle \phi$.

Using the technique from the previous section, we readily get a completeness result for the logic.

Theorem 2.13 *The system AX_C is complete.*

Proof.

By the Completeness Lemma, we only need to consider the additional case $\phi = \langle \alpha \cap \beta \rangle \psi$. This can be proved by a simple extension of the proof of Lemma 2.8.

Let $\phi = \langle \alpha \cap \beta \rangle \psi$. Then $\mathcal{A}, s \models \phi$ iff, by definition, $\mathcal{A}, s \models \langle \alpha \rangle \psi$ and $\mathcal{A}, s \models \langle \beta \rangle \psi$ iff, by induction hypothesis, $\langle \alpha \rangle \psi \in s$ and $\langle \beta \rangle \psi \in s$ iff, by construction, $\langle \alpha \rangle \psi \wedge \langle \beta \rangle \psi \in s$ iff $\langle \alpha \cap \beta \rangle \psi \in s$. □

In chapter 4 another application of this completeness technique is given for the case of an infinitary axiom system for a fragment of Propositional Dynamic Logic of Context-Free Programs.

2.3.3 Discussion

We have introduced a completeness technique for PDL which rests on an infinitary axiom system. One might ask whether this technique is applicable to the "normal" axiomatization as well. The answer to this question is "No". The difficulty in proving a lemma such as Lemma 2.8 lies in the case $\phi = [\alpha^*]\psi$. Let us see what happens when we try to prove the case. We can prove that $\mathcal{A}, s \models [\alpha^*]\psi$ implies $[\alpha^n]\psi \in s$ for each $n < \omega$, but we may not infer that then $[\alpha^*]\psi \in s$. In fact, we can prove the following theorem.

Theorem 2.14 *Let*

$$\begin{aligned}\Gamma &= \Pr(AX) \cup \{\phi, [a]\phi, [a^2]\phi, \ldots\} \cup \{\neg[a^*]\phi\} \\ &= \Pr(AX) \cup \Delta \cup \{\neg[a^*]\psi\}\end{aligned}$$

Then Γ *is consistent.*

Proof.

Suppose Γ inconsistent. Then for some *finite* subset $\Gamma' = \{\phi_0, \phi_1, \ldots, \phi_n\} \subseteq \Gamma$,

$$\Gamma' \vdash \mathit{false}.$$

Or

$$\vdash \phi_0 \wedge \cdots \wedge \phi_n \rightarrow \mathit{false}.$$

Without loss of generality, we may assume that $\phi_n = \neg[a^*]\psi$ and the other $\phi_j \in \Delta$. By Soundness, then, for all models $\mathcal{M}$ and states $s \in W^{\mathcal{M}}$, $\mathcal{M}, s \models \phi_0 \wedge \cdots \wedge \phi_{n-1} \rightarrow [\alpha^*]\phi$. But counterexamples are easily found. Hence Γ is consistent. □

Essentially, this is the same argument as we used for proving incompactness. There we saw that an infinite, semantically inconsistent set could not be proved to be inconsistent, by proving inconsistency of each of its finite subsets. In fact, each of its finite subsets was consistent. For exactly the same reason, namely syntactic consistency of each of the finite subsets of Γ, we must conclude that Γ itself is syntactically consistent. Yet it surely is *not* semantically consistent. We therefore conclude that *syntactic* and *semantic* consequence are two different notions in the case of the axiom system AX.

The Theorem does not hold, however, in the case of an infinitary axiom system. We may infer $[a^*]\psi$ from Δ and Γ proves inconsistent. In this case, syntactic and semantic consequence do coincide.

It is interesting to compare the discussion which arose between Kozen and Pratt concerning the equational definition of an algebraic structure underlying PDL and the above remarks. Pratt defined a Dynamic Algebra which had the Segerberg axiom system as its equational definition; Kozen proposed a $\star$-continuous Dynamic Algebra which incorporated the infinitary system AX_∞. Kozen proved the following theorem.

Theorem 2.15 *1. Standard Kripke models and $\star$-continuous Dynamic Algebras share the same $L_{\omega_1\omega}$ theory.*

2. *There exists an first-order sentence σ such that for each standard Kripke model $\mathcal{A}$, $\mathcal{A} \models \sigma$, but there exists a Dynamic Algebra D such that $D \models \neg\sigma$.*

Kozen then concluded that looping is "inherently infinitary", which agrees with our findings using model theoretic arguments.

The Segerberg axiom system is, however, complete for PDL [16] (see also Chapter 3 of the present paper). As a result we have, for each $\phi \in \Phi$,

$$\vdash \phi \text{ iff } \models \phi \text{ iff } \vdash_\infty \phi$$

since both logics are interpreted in the same class of models. Hence provability in both axiom systems coincide.

Chapter 3

A Universal Model Theorem for Kripke Structures

In this chapter we prove the existence of a Universal Model for PDL in a very natural and intuitively appealing way following Parikh [21]. Using this model, we prove a Completeness Theorem for the Segerberg axiom system AX and give a different proof for the Small Model Theorem.

3.1 The Universal Model Theorem

In this section we establish a nontrivial property of Kripke structures, namely the existence of an structure $\mathcal{U}$ that is universal in the sense that every other structure can be isomorphically embedded in it. We further exhibit some immediate consequences of this fact.

3.1.1 Some model theory

In this section we establish some facts about models for PDL and the theory of Kripke models.

Definition 3.1 *For each model $\mathcal{M}$, the theory of $\mathcal{M}$ is the set*

$$Th(\mathcal{M}) = \{\phi \mid \exists s \in W^{\mathcal{M}}.\mathcal{M}, s \models \phi\}.$$

This definition gives a notion of equivalence of models: two models $\mathcal{M}_1$ and $\mathcal{M}_2$ are equivalent iff $Th(\mathcal{M}_1) = Th(\mathcal{M}_2)$. Notice, however, that $\mathcal{M}_1$ and $\mathcal{M}_2$ need not be isomorphic, as the following example shows.

Example. Let $W^{\mathcal{M}_1} = W^{\mathcal{M}_2} = \{0, 1, 2, \ldots, \omega\}$. Let $\mathcal{M}_1, n \models p_i$ iff $\mathcal{M}_2, n \models p_i$ iff $i < n$. $\mathcal{M}_1, \omega \models p_j$ and $\mathcal{M}_2, \omega \models p_j$ for all j. Now let $\rho^{\mathcal{M}_1}(a) = \{(0, n) \mid n < \omega\}$ and $\rho^{\mathcal{M}_2}(a) = \{(0, n) \mid n \leq \omega\}$. Then $\mathcal{M}_1$ and $\mathcal{M}_2$ are not isomorphic but $Th(\mathcal{M}_1) = Th(\mathcal{M}_2)$ and both models even have the same formulae holding at the same states of W. □

Definition 3.2 *For each model $\mathcal{M}$ the relation $\equiv$ on the state space $W^{\mathcal{M}}$ is defined by:*

$$s \equiv t \text{ iff } \mathcal{M}, s \models \phi \iff \mathcal{M}, t \models \phi.$$

For each model $\mathcal{M}$ we now define the *collapse of* $\mathcal{M}$ to be the model $\mathcal{M}_c = \mathcal{M}/\equiv$:

$$\begin{aligned} s_c &= \{t \mid s \equiv t\} \\ W^{\mathcal{M}_c} &= \{s_c \mid s \in W^{\mathcal{M}}\} \\ \pi^{\mathcal{M}_c}(p_i) &= \{s_c \mid s \in \pi^{\mathcal{M}}(p_i)\} \\ \rho^{\mathcal{M}_c}(a_j) &= \{(s_c, t_c) \mid (s,t) \in \rho^{\mathcal{M}}(a_j)\} \end{aligned}$$

The following lemma is immediate.

Lemma 3.3 *For each proposition ϕ,*

$$\mathcal{M}, s \models \phi \text{ iff } \mathcal{M}_c, s_c \models \phi.$$

The lemma in effect states that we only need to consider models of cardinality at most $\aleph_1$, that is, the cardinality of the power set of Φ.

Lemma 3.4 *For every model $\mathcal{M}$ and program α,*

1. *if $(s,t) \in \rho(\alpha)$, then $\forall\phi.(\mathcal{M}, t \models \phi \Longrightarrow \mathcal{M}, s \models \langle\alpha\rangle\phi)$;*
2. *if $(s,t) \in \rho(\alpha)$, then $\forall\phi.(\mathcal{M}, s \models [\alpha]\phi \Longrightarrow \mathcal{M}, t \models \phi)$;*
3. *$\forall\phi.(\mathcal{M}, t \models \phi \Longrightarrow \mathcal{M}, s \models \langle\alpha\rangle\phi)$ iff $\forall\phi.(\mathcal{M}, s \models [\alpha]\phi \Longrightarrow \mathcal{M}, t \models \phi)$.*

Proof.

Clauses (1) and (2) follow immediately from the definition of $\models$. For clause (3): $\forall\phi.(\mathcal{M}, s \models [\alpha]\phi \Longrightarrow \mathcal{M}, t \models \phi)$ iff $\forall\phi.(\mathcal{M}, t \not\models \phi \Longrightarrow \mathcal{M}, s \not\models [\alpha]\phi)$ iff $\forall\phi.(\mathcal{M}, t \models \neg\phi \Longrightarrow \mathcal{M}, s \models \neg\langle\alpha\rangle\neg\phi)$ iff $\forall\psi.(\mathcal{M}, t \models \psi \Longrightarrow \mathcal{M}, s \models \langle\alpha\rangle\psi)$. □

For each program α we define the mappings $Dom(\alpha)$ and $Ran(\alpha)$ by:

$$Dom(\alpha) = \{s \in W \mid \exists t \in W.(s,t) \in \rho(\alpha)\}$$

$$Ran(\alpha) = \{t \in W \mid \exists s \in W.(s,t) \in \rho(\alpha)\}$$

In the light of Lemma 3.4 we can define for each model $\mathcal{M}$ another model $\mathcal{M}_{ex}$, called the *extension of* $\mathcal{M}$, by:

$$\begin{aligned} W^{\mathcal{M}_{ex}} &= W^{\mathcal{M}}; \\ \pi^{M_{ex}} &= \pi^{\mathcal{M}}; \\ \rho^{\mathcal{M}_{ex}}(a) &= \{(s,t) \mid \forall\phi(\mathcal{M}, s \models [a]\phi \Longrightarrow \mathcal{M}, t \models \phi)\} \text{ for } a \text{ primitive} \end{aligned}$$

By Lemma 3.4, $\rho^{\mathcal{M}}(a) \subseteq \rho^{\mathcal{M}_{ex}}(a)$ for each primitive program a. Note that $\rho^{\mathcal{M}}(a)$ need not equal $\rho^{\mathcal{M}_{ex}}(a)$. Consider for example the case in which $\mathcal{M}, s \models [a]\phi$ only if ϕ is valid. Then, for every $t \in W^{\mathcal{M}}$, $(s,t) \in \rho^{\mathcal{M}_{ex}}(a)$. Obviously, $\rho^{\mathcal{M}_{ex}}(a)$ can be substantially larger than $\rho^{\mathcal{M}}(a)$. We extend $\rho^{\mathcal{M}_{ex}}$ to the whole set Π in the usual way.

Lemma 3.5 *For each proposition ϕ,*

$$\mathcal{M}_{ex}, s \models \phi \text{ iff } \mathcal{M}, s \models \phi.$$

Proof.

($\Longleftarrow$) Since $\rho^{\mathcal{M}}(a) \subseteq \rho^{\mathcal{M}_{ex}}(a)$ for each primitive program a, it is easy to see that for each $\alpha \in \Pi$, $\rho^{\mathcal{M}}(\alpha) \subseteq \rho^{\mathcal{M}_{ex}}(\alpha)$. The proof proceeds by induction on the complexity of ϕ. The only non-trivial case is $\phi = \langle\alpha\rangle\psi$, which follows from the inclusion given above.

($\Longrightarrow$) Let $\mathcal{M}_{ex}, s \models \phi$. We define the mapping $R : \Pi \mapsto 2^{W^{\mathcal{M}} \times W^{\mathcal{M}}}$ by:

$$\begin{aligned} R(\alpha) &= \{(s,t) \mid \forall\psi.(\mathcal{M}, s \models [\alpha]\psi \Longrightarrow \mathcal{M}, t \models \psi)\} \\ &= \{(s,t) \mid \forall\psi.(\mathcal{M}, t \models \psi \Longrightarrow \mathcal{M}, s \models \langle\alpha\rangle\psi\} \end{aligned}$$

for $\alpha \in \Pi$. Note that, by Lemma 3.4(3), we may use both conditions interchangeably in the definition of R.

Claim 1. $\mathcal{M}, s \models \phi$ iff $\mathcal{M}, s \models_R \phi$, where $\models_R$ is defined as the relation $\models$ except that we use $R(\alpha)$ instead of $\rho(\alpha)$.

Proof of claim. Induction on the structure of ϕ. The only non-trivial case is $\phi = \langle\alpha\rangle\psi$. Let $\mathcal{M}, s \models \langle\alpha\rangle\psi$. Then there exists a state t such that $\mathcal{M}, t \models \psi$ and $(s,t) \in \rho(\alpha)$. But then $(s,t) \in R(\alpha)$ by the construction of R and $\mathcal{M}, s \models_R \langle\alpha\rangle\psi$. Conversely, let $\mathcal{M}, s \models_R \langle\alpha\rangle\psi$; then there is a state t such that $(s,t) \in R(\alpha)$ and $t \models \psi$. Suppose that there exists no state t such that $(s,t) \in \rho(\alpha)$ and $\mathcal{M}, t \models \psi$. Then $\mathcal{M}, s \models [\alpha]\neg\psi$ and, by the definition of R, if $(s,t) \in R(\alpha)$, then $t \models \neg\psi$. Contradiction.

Claim 2. For each $\alpha \in \Pi$, $\rho^{\mathcal{M}_{ex}}(\alpha) \subseteq R(\alpha)$.

Proof of claim. Induction on the complexity of α. For α primitive, the claim holds by definition. Next we consider more complex programs α.

Case 1: $\alpha = \beta \cup \gamma$.

Clearly, $\rho(\beta \cup \gamma) = \rho(\beta) \cup \rho(\gamma) \subseteq R(\beta) \cup R(\gamma)$. The last union equals:

$$\{(s,t) \mid \forall\phi.(t \models \phi \Longrightarrow s \models \langle\beta\rangle\phi) \vee \forall\phi.(t \models \phi \Longrightarrow s \models \langle\gamma\rangle\phi)\}$$

It is easy to see that this set is contained in:

$$\{(s,t) \mid \forall\phi.(t \models \phi \Longrightarrow s \models \langle\beta\rangle\phi \vee s \models \langle\gamma\rangle\phi)\}$$

which is $R(\beta \cup \gamma)$.

Case 2: $\alpha = \beta;\gamma$.

$\rho(\beta;\gamma) = \rho(\beta) \circ \rho(\gamma) \subseteq R(\beta) \circ R(\gamma)$. Now,

$$R(\beta) \circ R(\gamma) = \{(s,t) \mid \exists u.((s,u) \in R(\beta) \wedge (u,t) \in R(\gamma))\}$$

Let $(s,t) \in R(\beta) \circ R(\gamma)$. Then, for each ϕ,

$$t \models \phi \Longrightarrow s \models \langle\beta\rangle\langle\gamma\rangle\phi$$

hence $(s,t) \in R(\beta;\gamma)$ and $R(\beta) \circ R(\gamma) \subseteq R(\beta;\gamma)$.
Case 3: $\alpha = \beta^\star$.
By the former argument we get

$$\rho(\beta^n) \subseteq R(\beta^n)$$

for each $n < \omega$. We further have, for each $n < \omega$,

$$R(\beta^n) \subseteq R(\beta^\star)$$

Suppose $(s,t) \in R(\beta^n)$; then $t \models \psi \Longrightarrow s \models \langle\beta^n\rangle\psi$ for all ψ. Surely $t \models \psi \Longrightarrow s \models \langle\beta^\star\rangle\psi$ for all ψ, by the definition of $\rho(\beta^\star)$. Hence $(s,t) \in R(\beta^\star)$. Hence, by induction on n,

$$\rho(\beta^\star) = \bigcup_{i<\omega} \rho(\beta^i) \subseteq \bigcup_{i<\omega} R(\beta^i) \subseteq R(\beta^\star).$$

Note that this is the place where we use the infinitary properties of $\beta^\star$.
Case 4: $\alpha = \psi?$.
Clearly, $\rho(\psi?) = R(\psi?)$ follows immediately by the definitions of ρ and R.

The proof of the lemma now follows by induction on the structure of ϕ. Again, the only non-trivial case is $\phi = \langle\alpha\rangle\psi$. If $\mathcal{M}_{ex}, s \models \langle\alpha\rangle\psi$ then, by claim 2, $\mathcal{M}, s \models_R \langle\alpha\rangle\psi$ and hence, by claim 1, $\mathcal{M}, s \models \langle\alpha\rangle\psi$. □

Next we define, for each model $\mathcal{M}$, the model $\tilde{\mathcal{M}}$ by replacing every state in $W^{\mathcal{M}}$ by the set of propositions that hold at that state. We denote the state in $W^{\tilde{\mathcal{M}}}$ corresponding to s by $\tilde{s}$. It is easy to see that

$$\mathcal{M}, s \models \phi \Longleftrightarrow \tilde{\mathcal{M}}, \tilde{s} \models \phi \Longleftrightarrow \phi \in \tilde{s}$$

for each proposition $\phi \in \Phi$.

Definition 3.6 *For each model $\mathcal{M}$, the* canonical model for $\mathcal{M}$ *is* $[\mathcal{M}] = (\tilde{\mathcal{M}}_c)_{ex}$.

Theorem 3.7 *For each proposition ϕ and each model $\mathcal{M}$, $\mathcal{M}, s \models \phi$ iff $[\mathcal{M}], [s] \models \phi$.*

Proof.

Immediate from Lemma 3.3 and Lemma 3.5. □

3.1.2 The Universal Model

We can now define a universal Kripke structure $\mathcal{U}$. Consider the class $\mathcal{K}$ of all Kripke structures. For each $\mathcal{M} \in \mathcal{K}$ we define the mapping $\theta_{\mathcal{M}} : W^{\mathcal{M}} \mapsto W^{\mathcal{U}}$ by:

$$\theta_{\mathcal{M}}(s) = \{\phi \mid \mathcal{M}, s \models \phi\}.$$

We let the set of states $W^{\mathcal{U}}$ of the universal model be exactly the set of all subsets of Φ that can be obtained this way (when $\mathcal{M}$ ranges over all Kripke structures). That is, for $\Psi \subseteq \Phi$, $\Psi \in W^{\mathcal{U}}$ iff $\Psi = \theta_{\mathcal{M}}(s)$ for some model $\mathcal{M}$ and state $s \in W^{\mathcal{M}}$. We define $\pi^{\mathcal{U}}$ by:

$$\pi^{\mathcal{U}}(p_i) = \{s \in W^{\mathcal{U}} \mid p_i \in s\}$$

for $0 \leq i < \omega$. The interpretation for the primitive programs is defined as:

$$\rho^{\mathcal{U}}(a_j) = \{(s,t) \in W^{\mathcal{U}} \times W^{\mathcal{U}} \mid \forall \phi.([a_j]\phi \in s \Longrightarrow \phi \in t)\}$$

for $0 \leq j < \omega$. Note that the states of $\mathcal{U}$ consist of all semantically consistent complete sets of formulae.

We can also describe the Universal Model as the model which results from "pasting together" all canonical models $[\mathcal{M}]$ for all Kripke models $\mathcal{M}$. All states in $\mathcal{U}$ are "copies" of states in some canonical model $[\mathcal{M}]$.

Lemma 3.8 *For each canonical model* $[\mathcal{M}]$ *and* $\alpha \in \Pi$, $\rho^{[\mathcal{M}]}(\alpha) \subseteq \rho^{\mathcal{U}}(\alpha)$.

Proof.

It follows immediately from the definitions of $\rho^{[\mathcal{M}]}$ and $\rho^{\mathcal{U}}$ that, for primitive a, $\rho^{[\mathcal{M}]}(a) \subseteq \rho^{\mathcal{U}}(a)$. The lemma follows. □

Lemma 3.9 *Consider the universal model* $\mathcal{U}$.

1. *For each* $\phi \in \Phi$ *and* $\alpha \in \Pi$,

$$\langle\alpha\rangle\phi \in s \Longleftrightarrow \exists t.(s,t) \in \rho(\alpha) \wedge \phi \in t.$$

2. *For each* $\phi \in \Phi$,

$$\mathcal{U}, s \vDash \phi \textit{ if and only if } \phi \in s.$$

Proof.

1. ($\Longrightarrow$) Let $\langle\alpha\rangle\phi \in s$. Then there exists a canonical model $[\mathcal{M}]$ and a state $[s] \in W^{[\mathcal{M}]}$ such that $\langle\alpha\rangle\phi \in [s]$. Then there exists a $[t] \in Ran^{[\mathcal{M}]}(\alpha)$ such that $\phi \in [t]$. Hence, by Lemma 3.8, $t \in Ran^{\mathcal{U}}(\alpha)$ and $\phi \in t$.

   ($\Longleftarrow$) Again define the function $R : \Pi \mapsto 2^{W \times W}$ as in Theorem 3.5 except that we use $\in$ instead of $\vDash$. By the proof of that theorem, $\rho(\alpha) \subseteq R(\alpha)$. Hence, if $(s,t) \in \rho(\alpha)$ and $\phi \in t$, then $(s,t) \in R(\alpha)$ and by the definition of R, $\langle\alpha\rangle\phi \in s$.

2. The proof is by induction on the structure of ϕ. For ϕ primitive, the lemma holds by definition.

   Case 1: $(\phi = \psi \vee \chi)$

   $s \vDash \psi \vee \chi$ iff $s \vDash \psi$ or $s \vDash \chi$ iff, by the induction hypothesis, $\psi \in s$ or $\chi \in s$ iff $\psi \vee \chi \in s$ by the maximality of s.

Case 2: $(\phi = \neg\psi)$

Similar.

Case 3: $(\phi = \langle\alpha\rangle\psi)$

$s \models \langle\alpha\rangle\psi$ iff there is a state $t \in W$ such that $(s,t) \in \rho(\alpha)$ and $t \models \psi$ iff $\psi \in t$ by the induction hypothesis and $\langle\alpha\rangle\phi \in s$ by the first part of the lemma. □

The following theorem is an immediate consequence of the lemma.

Theorem 3.10 *There exists a universal Kripke structure $\mathcal{U} = (W^{\mathcal{U}}, \pi^{\mathcal{U}}, \rho^{\mathcal{U}})$ such that for each Kripke structure $\mathcal{M} = (W^{\mathcal{M}}, \pi^{\mathcal{M}}, \rho^{\mathcal{M}})$ there exists an embedding $\theta_{\mathcal{M}} : W^{\mathcal{M}} \mapsto W^{\mathcal{U}}$ such that $\mathcal{M}, s \models \phi$ iff $\mathcal{U}, \theta_{\mathcal{M}}(s) \models \phi$ for each well-formed formula ϕ.*

Proof.

The model $\mathcal{U}$ constructed above and mappings $\theta_{\mathcal{M}}$ for each $\mathcal{M}$ are the required model and mappings. Let $\mathcal{M}$ be any model and $s \in W^{\mathcal{M}}$. Then $\mathcal{M}, s \models \phi$ iff $\phi \in [s]$ for $[s]$ in the canonical model $[\mathcal{M}]$ and hence $\phi \in s$ for $s \in W^{\mathcal{U}}$ by the construction of $\mathcal{U}$. By Lemma 3.9, $\mathcal{U}, s \models \phi$. Conversely, let $\mathcal{U}, s \models \phi$. Then, again by Lemma 3.9, $\phi \in s$ and hence, $\phi \in [s]$ for some canonical model $[\mathcal{M}]$. Then $[\mathcal{M}], [s] \models \phi$ and each model $\mathcal{M}$ such that $[\mathcal{M}]$ is canonical for $\mathcal{M}$, satisfies ϕ. □

Note that the theorem implies that the mapping θ is an *isomorphic embedding* in the terminology of model theory.

3.1.3 Some consequences

In this section we give two immediate consequences of Theorem 3.10 which will be instrumental for obtaining the results of the next two sections.

Lemma 3.11 *For all propositions ϕ, ϕ is satisfiable if and only if ϕ is $\mathcal{U}$-satisfiable.*

Proof.

ϕ is satisfiable iff there exists a model $\mathcal{M}$ and a state $s \in W^{\mathcal{M}}$ such that $\mathcal{M}, s \models \phi$ iff $\mathcal{U}, \theta_{\mathcal{M}}(s) \models \phi$. □

Lemma 3.12 *For all propositions ϕ, ϕ is valid if and only if ϕ is $\mathcal{U}$-valid.*

Proof.

($\Longrightarrow$) Immediate.

($\Longleftarrow$) Suppose ϕ not valid. Then there exists a model $\mathcal{M}$ and a state $s \in W^{\mathcal{M}}$ such that $\mathcal{M}, s \not\models \phi$. Hence $\mathcal{U}, \theta_{\mathcal{M}}(s) \not\models \phi$ and ϕ is not $\mathcal{U}$-valid. □

3.1.4 Some more model theory

Parikh [21] defined two notions of equivalence of models.

Definition 3.13 *For models $\mathcal{M}$ and $\mathcal{N}$ of* PDL, *let*

1. $\mathcal{M} \equiv \mathcal{N}$ *iff* $Th(\mathcal{M}) = Th(\mathcal{N})$;
2. $\mathcal{M} \equiv_s \mathcal{N}$ *iff* $\forall s \in W^{\mathcal{M}} \exists s' \in W^{\mathcal{N}} \forall \phi \in \Phi . \mathcal{M}, s \models \phi \Longleftrightarrow \mathcal{N}, s' \models \phi$ *and* $\forall s' \in W^{\mathcal{N}} \exists s'' \in W^{\mathcal{M}} \forall \psi \in \Phi . \mathcal{N}, s' \models \psi \Longleftrightarrow \mathcal{M}, s'' \models \psi$.

Theorem 3.14 *For all models $\mathcal{M}$ and $\mathcal{N}$ of* PDL,

$$\mathcal{M} \cong \mathcal{N} \Longrightarrow \mathcal{M} \equiv_s \mathcal{N} \Longrightarrow \mathcal{M} \equiv \mathcal{N}.$$

where $\cong$ denotes "is isomorphic to".

The proof of the theorem is trivial, but note that none of the converse implications hold. However, in all equivalence classes induced by these relations we can find canonical elements.

Definition 3.15 *Let $\mathcal{C}$ be any class of Kripke models and $\mathcal{M} \in \mathcal{C}$. Then*

1. $\mathcal{M}$ *is* canonical for $\mathcal{C}$ *iff*
   (a) $\forall s, t \in W^{\mathcal{M}} . (s \neq t \Longrightarrow \exists \phi . (s \models \phi \wedge t \models \neg \phi))$;
   (b) $\forall s, t \in W^{\mathcal{M}} . ((s,t) \in \rho(a) \Longleftrightarrow \forall \phi . (s \models [a]\phi \Longrightarrow t \models \phi)$.
2. $\mathcal{M}$ *is* canonically closed for $\mathcal{C}$ *iff*
   (a) $\mathcal{M}$ *is canonical for* $\mathcal{C}$;
   (b) *for any* $\mathcal{M}' \in \mathcal{C}$ *and* $s' \in W^{\mathcal{M}'}$, *if for all* ϕ *there exists a* $s \in W^{\mathcal{M}}$ *such that* $\mathcal{M}', s' \models \phi \Longrightarrow \mathcal{M}, s \models \phi$, *then there exists a* $s_0 \in W^{\mathcal{M}}$ *such that for all* ϕ, $\mathcal{M}', s' \models \phi \Longrightarrow \mathcal{M}, s_0 \models \phi$.

Note that the "only if" part of condition (1b) holds in all models because of the semantics of $\Box$, but in canonical models, the $\rho(a)$ is "packed full" so that the other direction holds as well. Condition (2b) states that if a canonically closed model $\mathcal{M}$ has arbitrarily close "approximations" to s', then $\mathcal{M}$ contains a "copy" s_0 of s', that is, a state s_0 such that $\{\phi \mid \mathcal{M}, s_0 \models \phi\} = \{\phi \mid \mathcal{M}', s' \models \phi\}$. In this sense the word "closed" can be given a topological meaning.

Let $\mathcal{K}$ be the class of all Kripke models and $\mathcal{K}^{[\mathcal{M}]}$ the class of all models $\mathcal{M}$ such that $[\mathcal{M}]$ is canonical for $\mathcal{M}$.

Theorem 3.16 1. $[\mathcal{M}]$ *is canonically closed for* $\mathcal{K}^{[\mathcal{M}]}$.

2. $\mathcal{U}$ *is canonically closed for* $\mathcal{K}$;

Proof.

Immediate from section 3.1.1. □

3.2 Completeness of AX

To prove completeness of AX we adapt the Lindenbaum construction [1] to PDL: We impose a Boolean algebra structure on the state space $W^{\mathcal{U}}$ of $\mathcal{U}$. With each proposition ϕ we associate the set of states that satisfy ϕ:

$$|\phi| = \{s \in W \mid s \vDash \phi\}.$$

Let P be the set of all such $|\phi|$. We define a partial ordering $\leqslant$ on P:

$$|\phi| \leqslant |\psi| \text{ iff } \vdash \phi \to \psi.$$

Lemma 3.17 *$\mathcal{B} = \langle P, \leqslant \rangle$ is a complemented distributive lattice, that is, a Boolean algebra.*

Proof.

By propositional reasoning we have

$$\vdash \psi \to \mathit{true}$$

$$\vdash \mathit{false} \to \psi$$

for all propositions ψ. Hence we can take $|\mathit{true}| = 1$ and $|\mathit{false}| = 0$ in $\mathcal{B}$.

Let $|\phi| \in P$. Then its complement, $|\phi|^c$, is defined as:

$$\begin{aligned} |\phi|^c &= \{s \mid s \vDash \phi\}^c \\ &= \{s \mid s \nvDash \phi\} \\ &= \{s \mid s \vDash \neg\phi\} \\ &= |\neg\phi| \end{aligned}$$

and $|\neg\phi| \in P$.

Let $|\phi|, |\psi| \in P$. Then:

$$\begin{aligned} |\phi| \cap |\psi| &= \{s \mid s \vDash \phi\} \cap \{s \mid s \vDash \psi\} \\ &= \{s \mid s \vDash \phi \wedge s \vDash \psi\} \\ &= \{s \mid s \vDash \phi \wedge \psi\} \\ &= |\phi \wedge \psi| \end{aligned}$$

Hence $|\phi| \cap |\psi| \in P$. By propositional reasoning,

$$\vdash (\phi \wedge \psi) \to \phi \text{ and } \vdash (\phi \wedge \psi) \to \psi.$$

Hence $|\phi \wedge \psi|$ is a lower bound for $\{|\phi|, |\psi|\}$. Suppose $|\chi|$ is a lower bound too. Then $\vdash \chi \to \phi$ and $\vdash \chi \to \psi$. Hence $\vdash \chi \to (\phi \wedge \psi)$. This shows that $|\phi \wedge \psi|$ is the greatest lower bound, *i.e.* the infimum of $\{|\phi|, |\psi|\}$. Similarly, $|\phi \vee \psi|$ is the supremum of $\{|\phi|, |\psi|\}$. Thus $\mathcal{B}$ is a lattice.

Let $|\phi|, |\psi|, |\chi| \in P$. Then $|(\phi \wedge \psi) \vee \chi| \in P$ and because

$$\vdash ((\phi \wedge \psi) \vee \chi) \leftrightarrow ((\phi \vee \chi) \wedge (\psi \vee \chi))$$

we get from the Soundness Theorem,

$$|(\phi \wedge \psi) \vee \chi| = |(\phi \vee \chi) \wedge (\psi \vee \chi)|.$$

This shows that $\mathcal{B}$ is a complemented distributive lattice. □

Lemma 3.18 *In the Boolean algebra $\mathcal{B}$,*

1. $|\phi| = 1$ *if and only if* $\vdash \phi$;
2. $|\psi| = 0$ *if and only if* $\vdash \neg\psi$.

Proof.

1. Let $|\phi| = 1$. Then for each $|\psi| \in P$, $|\psi| \leqslant |\phi|$. Hence, for each $|\psi|$, $\vdash \psi \to \phi$. Choose ψ so that $\vdash \psi$, then , by modus ponens, $\vdash \phi$. Conversely, suppose $\vdash \phi$. Then, for each ψ, $\vdash \psi \to \phi$. Hence, for each ψ, $|\psi| \leqslant |\phi|$, so $|\phi| = 1$ in $\mathcal{B}$.
2. Similar. □

Lemma 3.19 *For all proposition ϕ, if $\mathcal{U} \models \phi$ then $\vdash \phi$.*

Proof.

Suppose that ϕ is not provable in the system AX. Then, by lemma 3.18, in the Lindenbaum algebra $\mathcal{B}$, $|\phi| \neq 1$ and so $|\neg\phi| \neq 0$. Hence there exists a state $s \in |\neg\phi|$ such that $\mathcal{U}, s \models \neg\phi$. Hence ϕ is not $\mathcal{U}$-valid. □

Theorem 3.20 (Completeness Theorem) $\models \phi$ *if and only if* $\vdash \phi$.

Proof. One direction is the Soundness Theorem. The other direction follows from Lemmas 3.12 and 3.19. □

Remark. Let $\mathcal{A}$ be the model as defined in the previous chapter from the infinitary axiom system AX_∞. An immediate observation leads to the next lemma.

Lemma 3.21 $W^{\mathcal{A}} = W^{\mathcal{U}}$.

Proof.

By Soundness, each $s \in W^{\mathcal{U}}$ is maximally consistent and $\Pr(AX_\infty) \subseteq s$ so $W^{\mathcal{U}} \subseteq W^{\mathcal{A}}$. Conversely, $W^{\mathcal{A}} \subseteq W^{\mathcal{U}}$ by Completeness. □

By the lemma and the constructions of $\mathcal{U}$ and $\mathcal{A}$ we get:

Theorem 3.22 $\mathcal{U} \cong \mathcal{A}$.

In fact we may say that $\mathcal{U}$ and $\mathcal{A}$ are only two different names for the same model and conclude that $\mathcal{U} = \mathcal{A}$.

3.3 The Small Model theorem

We find another application of Theorem 3.10 in a different proof of the Small Model theorem. This theorem is one of the basic results of the theory of PDL and was first discovered by Fischer and Ladner [6]. It states that every proposition ϕ that is satisfiable, is satisfiable in a model with $2^{|\phi|}$ states. This fact immediately gives rise to a naïve doubly-exponential time decision procedure for the validity problem for PDL: to check whether ϕ is valid, generate all models with $2^{|\neg\phi|}$ states and cycle through them in search for a model that satisfies $\neg\phi$. If such a model doesn't exists, then ϕ is valid. Sherman and Harel [10, 32] proved the existence of a singly-exponential time procedure by constructing a model $\mathcal{A}_\phi$ that satisfies ϕ iff ϕ is satisfiable. Thus we can construct a model in polynomial time and check whether this model satisfies $\neg\phi$ in exponential time.

We first need a notion of the "subformulae" of a PDL formula ϕ. This concept is captured by the Fischer-Ladner closure of ϕ [6].

Definition 3.23 *Let $\phi \in \Phi$ be a* PDL *formula. The* Fischer-Ladner closure of ϕ, *denoted by $FL(\phi)$, is the smallest set S of formulae containing ϕ and satisfying the following closure rules for all $a \in \Pi_0$, $\alpha, \beta \in \Pi$ and $\psi, \chi \in \Phi$:*

$$\begin{aligned}
\neg\psi \in S &\implies \psi \in S \\
\psi \vee \chi \in S &\implies \psi, \chi \in S \\
\langle a \rangle \psi \in S &\implies \psi \in S \\
\langle \alpha\beta \rangle \psi \in S &\implies \langle \alpha \rangle \langle \beta \rangle \psi \in S \\
\langle \alpha \cup \beta \rangle \psi \in S &\implies \langle \alpha \rangle \psi, \langle \beta \rangle \psi \in S \\
\langle \alpha^* \rangle \psi \in S &\implies \psi, \langle \alpha \rangle \langle \alpha^* \rangle \psi \in S \\
\langle \psi? \rangle \chi \in S &\implies \psi, \chi \in S
\end{aligned}$$

The Fischer-Ladner closure of ϕ is the set of all "subformulae" that are relevant for the meaning of ϕ. The set $FL(\phi)$ induces an equivalence relation $\equiv_\phi$ on the state space W of any model $\mathcal{M}$:

$$s \equiv_\phi t \text{ iff } \forall \psi \in FL(\phi).(s \vDash \psi \iff t \vDash \psi)$$

In other words, we "collapse" s and t if they are not distinguishable by any formula of $FL(\phi)$. We now define the quotient model $\mathcal{M}/FL(\phi)$:

$$\begin{aligned}
[s] &= \{t \mid s \equiv_\phi t\} \\
W^{\mathcal{M}/FL(\phi)} &= \{[s] \mid s \in W^{\mathcal{M}}\} \\
\pi^{\mathcal{M}/FL(\phi)}(p_i) &= \{[s] \mid s \in \pi^{\mathcal{M}}(p_i)\} \text{ for all } p_i \in \Phi_0 \\
\rho^{\mathcal{M}/FL(\phi)}(a_j) &= \{([s],[t]) \mid (s,t) \in \rho^{\mathcal{M}}(a_j)\} \text{ for all } a_j \in \Pi_0
\end{aligned}$$

$\pi^{\mathcal{M}/FL(\phi)}$ and $\rho^{\mathcal{M}/FL(\phi)}$ are extended inductively to Π and Φ in the usual way. The following lemma, called the *Filtration Lemma*, is crucial for the theorem:

Lemma 3.24 (Filtration Lemma) *For all $\psi \in FL(\phi)$:*

1. *if* $\psi = \langle\alpha\rangle\chi$ *then* $\forall s,t \in W^{\mathcal{M}}\{(s,t) \in \rho^{\mathcal{M}}(\alpha) \Longrightarrow ([s],[t]) \in \rho^{\mathcal{M}/FL(\phi)}(\alpha)\}$;

2. *for all states* s: $\mathcal{M}, s \models \psi \Longleftrightarrow \mathcal{M}/FL(\phi), [s] \models \psi$.

Proof.

Tedious but straightforward induction on the structure of ψ; see [6] for details. □

We now consider the quotient model $\mathcal{U}/FL(\phi)$.

Lemma 3.25 *For each* $\psi \in FL(\phi)$

$$\psi \text{ is satisfiable iff } \psi \text{ is } \mathcal{U}/FL(\phi)\text{-satisfiable.}$$

Proof. The lemma follows from Lemma 3.11 and the Filtration Lemma. □

Next we give another representation for the states of the quotient model $\mathcal{U}/FL(\phi)$: for each $[s] \in W^{\mathcal{U}/FL(\phi)}$, let $\tilde{s}$ be the set

$$\tilde{s} = \{\psi \mid [s] \models \psi \text{ and } \psi \in FL(\phi)\} \cup \{\neg\psi \mid [s] \models \neg\psi \text{ and } \psi \in FL(\phi)\}$$

That is, $\tilde{s}$ is the set of formulae from $FL(\phi)$ that hold at $[s]$ together with the negations of the formulae from $FL(\phi)$ that don't hold. We define the model $\mathcal{U}_\phi$ by mapping in the filtration model $\mathcal{U}/FL(\phi)$ each state $[s]$ onto $\tilde{s}$. The interpretation functions are adapted in the obvious way. From this construction we immediately get the following lemma.

Lemma 3.26 *For each formula* $\psi \in FL(\phi)$ *and* $[s] \in W^{\mathcal{U}/FL(\phi)}$,

$$\mathcal{U}/FL(\phi), [s] \models \psi \text{ iff } \mathcal{U}_\phi, \tilde{s} \models \psi \text{ iff } \psi \in \tilde{s}.$$

Theorem 3.27 *For each formula* $\psi \in FL(\phi)$,

$$\psi \text{ is satisfiable iff } \psi \in \tilde{s}$$

for some state $\tilde{s} \in \mathcal{U}_\phi$.

Proof.

Immediate from Lemmas 3.25 and 3.26. □

The sets of formulae $\tilde{s}$ are called *atoms of* $FL(\phi)$ and play a crucial role in the definition of the model $\mathcal{A}_\phi$.

Definition 3.28 *Let* Z *be the set of* PDL *formulae in which all formulae of* $FL(\phi)$ *and their negations occur. Then an* atom of $FL(\phi)$ *is defined to be a subset* $A \subseteq Z$ *such that for every* $\alpha, \beta \in \Pi$ *and* $\psi, \chi \in \Phi$:

if $\neg\psi \in Z$, *then* $\psi \in A$ *iff* $\neg\psi \notin A$
if $\psi \vee \chi \in Z$, *then* $\psi \vee \chi \in A$ *iff* $\psi \in A$ *or* $\chi \in A$
if $\langle\alpha\beta\rangle\psi \in Z$, *then* $\langle\alpha\beta\rangle\psi \in A$ *iff* $\langle\alpha\rangle\langle\beta\rangle\psi \in A$
if $\langle\alpha \cup \beta\rangle\psi \in Z$, *then* $\langle\alpha \cup \beta\rangle\psi \in A$ *iff* $\langle\alpha\rangle\psi \in A$ *or* $\langle\beta\rangle\psi \in A$
if $\langle\alpha^*\rangle\psi \in Z$, *then* $\langle\alpha^*\rangle\psi \in A$ *iff* $\psi \in A$ *or* $\langle\alpha\rangle\langle\alpha^*\rangle\psi \in A$
if $\langle\psi?\rangle\chi \in Z$, *then* $\langle\psi?\rangle\chi \in A$ *iff* $\psi \in A$ *and* $\chi \in A$.

Note that for all $\psi \in FL(\phi)$, either ψ or $\neg\psi$ is contained in each atom. Denote the set of all atoms of $FL(\phi)$ by $At(\phi)$. From the definition of atoms it follows that an $A \in At(\phi)$ is free of "obvious" or internal contradictions. In the construction of the model $\mathcal{A}_\phi$ we will eliminate the "nonobvious" or external contradictions also. This model will be constructed in phases. For the definition of the interpretation functions π and ρ we limit ourself, without loss of generality, to the primitive predicate and program symbols occurring in ϕ.

$\mathcal{A}_0 = (W_0, \pi_0, \rho_0)$ is defined by:

- $W_0 = At(\phi)$;
- $\pi_0 : \Phi_0 \mapsto 2^{W_0}$ by $A \in \pi_0(p)$ iff $p \in A$;
- $\rho_0 : \Pi_0 \mapsto 2^{W_0 \times W_0}$ by $(A, B) \in \rho_0(a)$ iff
  1. there is a $\langle a \rangle \psi \in A$ with $\psi \in B$, and
  2. for every $[a]\psi \in A$, $\psi \in B$.

For $i > 0$, $\mathcal{A}_{i+1} = (W_{i+1}, \pi_{i+1}, \rho_{i+1})$ is defined by

- $W_{i+1} = \{A \mid A \in W_i$, and for every $\langle \alpha \rangle \psi \in A$, there is $B \in W_i$ with $(A, B) \in \rho'_i(\alpha)$ and $\psi \in B\}$;
- $\pi_{i+1}(p) = \pi_i(p) \cap W_{i+1}$;
- $\rho_{i+1}(a) = \rho_i(a) \cap (W_{i+1} \times W_{i+1})$.

Here ρ'_i is the ordinary extension of ρ_i to Π, except that for $\psi \in Z$ we define $\rho'_i(\psi?) = \{(A, A) \mid \psi \in A\}$. The unprimed ρ is the usual extension.

It follows from the finiteness of $At(\phi)$ and the fact that $W_{i+1} \subseteq W_i$ that there is a j for which the construction closes up; *i.e.* $\mathcal{A}_i = \mathcal{A}_j$ for each $i > j$. Accordingly, set $\mathcal{A}_\phi = \mathcal{A}_j$.

The following lemma is the main technical lemma we need for our final result.

Lemma 3.29 *For every $A \in W^{\mathcal{A}_\phi}$,*

1. *for each $\langle \alpha \rangle \psi \in FL(\phi)$,*
   $\langle \alpha \rangle \psi \in A$ iff there exists a $B \in W^{\mathcal{A}_\phi}$ with $(A, B) \in \rho(\alpha)$ and $\psi \in B$;
2. *for each $\psi \in FL(\phi)$,*
   $\psi \in A$ iff $\mathcal{A}_\phi, A \models \psi$.

Proof.

The proof proceeds by simultaneous induction on the structure of α in (1) and the structure of ψ in (2). See [32] for details. □

Theorem 3.30 (Small Model Theorem) *For all $\psi \in FL(\phi)$, ψ is satisfiable iff $\psi \in A$ for some $A \in W^{\mathcal{A}_\phi}$.*

Proof.

In the light of Theorem 3.27, we only need to prove that $W^{\mathcal{U}_\phi} = W^{\mathcal{A}_\phi}$, from which the theorem follows.

- $W^{\mathcal{A}_\phi} \subseteq W^{\mathcal{U}_\phi}$: immediate from the construction of $\mathcal{U}_\phi$;
- suppose there exists an atom $A \in W^{\mathcal{U}_\phi}$ and $A \notin W^{\mathcal{A}_\phi}$. As we have started from the set of all atoms in W_0, there exists a phase i in which the first such atom is removed from W_{i+1}. Inspection of the algorithm shows that this can only happen if there exists a formula $\langle\alpha\rangle\psi \in A$ such that there exists no $B \in W_i$ with $(A, B) \in \rho'_i(\alpha)$ and $\psi \in B$. But $A \in W^{\mathcal{U}_\phi}$ and hence there exists a state $B \in W^{\mathcal{U}_\phi}$ with $(A, B) \in \rho(\alpha)$ and $\psi \in B$. Because A is the first state to be removed, $B \in W_i$. Contradiction. □

Remark. The above described filtration technique stems from Modal Logic where a somewhat stronger result has been obtained. First define a notion of subformulae of a Modal Logic proposition ϕ. The set of all subformulae $Sf(\phi)$ of a formula ϕ is defined by:

$$\begin{aligned} Sf(p) &= \{p\} \\ Sf(\phi \vee \chi) &= \{\phi \vee \chi\} \cup Sf(\phi) \cup Sf(\chi) \\ Sf(\neg\phi) &= \{\neg\phi\} \cup Sf(\phi) \\ Sf(\Box\phi) &= \{\Box\phi\} \cup Sf(\phi) \end{aligned}$$

We define a subset Γ of the set of well-formed formulae Φ to be a *filtration set* if Γ is closed under subformulae, that is,

$$\phi \in \Gamma \text{ implies } Sf(\phi) \subseteq \Gamma.$$

Γ induces an equivalence relation $\sim_\Gamma$ on the state space of any model $\mathcal{M} = (S, R, V)$:

$$s \sim_\Gamma t \text{ iff for all } \phi \in \Gamma,\ \mathcal{M}, s \vDash \phi \text{ iff } \mathcal{M}, t \vDash \phi.$$

We denote the equivalence class to which a state $s \in S$ belongs by $|s|$. We can define a model $\mathcal{M}' = (S_\Gamma, R', V_\Gamma)$, called the Γ-*filtration of* $\mathcal{M}$, by:

- $S_\Gamma = \{|s| \mid s \in S\}$
- $|s| \in V_\Gamma(p)$ iff $s \in V(p)$ for primitive p

and R' must satisfy:

1. if $(s, t) \in R$, then $(|s|, |t|) \in R'$; and
2. if $(|s|, |t|) \in R'$, then for all ϕ,

$$\text{if } \Box\phi \in \Gamma \text{ and } \mathcal{M}, s \vDash \Box\phi, \text{ then } \mathcal{M}, t \vDash \phi.$$

Note that we still have some liberty in choosing the relation R'. We can prove the following lemma.

Lemma 3.31 (Filtration Lemma) *If $\phi \in \Gamma$, then for any $s \in S$,*

$$\mathcal{M}, s \models \phi \text{ iff } \mathcal{M}', |s| \models \phi.$$

We now give some examples of filtrations.

1. The *smallest* filtration.

$$(|s|, |t|) \in R^{\sigma} \text{ iff } \exists s' \in |s| \exists t' \in |t|.((s', t') \in R).$$

2. The *largest* filtration.

$$(|s|, |t|) \in R^{\lambda} \text{ iff for all } \phi, \Box\phi \in \Gamma \text{ and } \mathcal{M}, s \models \phi \text{ implies } \mathcal{M}.t \models \phi.$$

Observe that, in the context of PDL, the Filtration Lemma is proved for a *smallest* filtration. Observe further, that Lemma 3.5 proves a special case of a largest filtration, namely, the case that we take Γ to be the whole set of well-formed formulae.

Chapter 4

PDL with Repeat

In this chapter we study the effect of adding a predicate, *repeat*, for programs. This predicate holds of a state s_0 iff there are states s_i for $i < \omega$ such that $(s_i, s_{i+1}) \in \rho(\alpha)$. That is, *repeat*($\alpha$) holds when α^* can diverge when executed from s_0. The main tool we employ is the notion of the *computation tree* of a program α. We use it to show that *repeat*(α) is equivalent to an infinitary formula ψ. Thus the predicate *repeat* is definable in infinitary PDL, that is, the variant of PDL in which we allow for infinite conjunctions and disjunctions of arbitrary formulae. This logic is easily definable analogous to AX_∞: we only have to replace the ∞-rule by a general infinitary rule for formulae. Thus we have the following inference rule:

- ∞-rule': from $\{\phi_n \mid n < \omega\}$, infer $\bigwedge_{n<\omega} \phi_n$.

4.1 Computation trees

Definition 4.1 *Let $\mathcal{M} = (W, \pi, \rho)$ be any Kripke structure. For each program $\alpha \in \Pi$ and state $s \in W$, the* computation tree of α rooted in s *is the tree $T_\alpha^s = (V, E)$ where V is a set of vertices and E is a set of edges, defined by*

- $s \in V$
- *if $(t, t') \in \rho(\alpha)$ and $t \in V$, then $(t, t') \in E$ and $t' \in V$.*

Note that each vertex in T_α^s may have countably many descendants and that T_α^s itself may be infinitely deep. Note also that the same state may occur in several distinct nodes in T_α^s and if a state lies on a cycle, then it occurs infinitely often in T_α^s. Let T_α be any computation tree rooted in s on any Kripke structure $\mathcal{M}$. An *α-path* departing from s is any branch in T_α. When $s, t_1, \ldots, t_n$ are the vertices on that branch, then the path is of length n. When the path is infinitely long, its length is ω. Intuitively, an α-path is any computation sequence of α^*.

Definition 4.2 *For any program $\alpha \in \Pi$, repeat(α) holds at state s iff there exists an α-path of length ω departing from s.*

Note that *repeat*(α) holds if and only if $\alpha^\star$ can diverge. We can define a partial ordering $\sqsubseteq_\alpha$ of computation trees by setting $T^s_\alpha \sqsubseteq_\alpha T^t_\alpha$ iff there exists an embedding $f : T^s_\alpha \mapsto T^t_\alpha$ such that if $(t, t') \in E$, then $(f(t), f(t')) \in E'$, where E (E') is the set of edges of T^s_α (T^t_α). Two trees T^s_α and T^t_α are α-equivalent, $T^s_\alpha \equiv_\alpha T^t_{\alpha'}$ iff $T^s_\alpha \sqsubseteq_\alpha T^t_\alpha$ and $T^t_\alpha \sqsubseteq_\alpha T^s_\alpha$. If, for two α-trees T and T', $T \sqsubseteq_\alpha T'$, then T' is *at least* as deep as T. This notion of equivalence between computation trees is somewhat coarse, but good enough for our purposes as we are only interested in whether the depth of a tree is finite or infinite.

For the rest of this section we are not interested in a particular program α or structure $\mathcal{M}$, so we drop sub- and superscripts. We define a class $\mathcal{D}$ of computation trees over some (appropriate) Kripke structure as:

$$\mathcal{D} = \{D_i \mid i \leq \omega\}$$

by the following recursive definition:

- D_1 is the α-tree rooted in s_1 that contains an α-path departing from s_1 of length n for each $n < \omega$ but no path of length ω;
- D_{i+1} is the α-tree rooted in s_{i+1} that consists of a copy of D_1 and has a copy of D_i appended to the endpoint of each finite path;
- D_ω is the α-tree rooted in s_ω such that $D_i \sqsubseteq_\alpha D_\omega$ for each $i < \omega$.

Note that this definition is unambiguous up to α-equivalence. For convenience, we set $D_i = (V_i, E_i)$.

Theorem 4.3 *For each $i < \omega$,*

1. $D_i \sqsubseteq_\alpha D_{i+1}$;
2. $D_{i+1} \not\sqsubseteq_\alpha D_i$.

Proof.

1. Trivial.
2. We prove $D_2 \not\sqsubseteq_\alpha D_1$; the claim then follows by an easy induction on i. Assume towards a contradiction that $D_2 \sqsubseteq_\alpha D_1$ and let f be the witnessing embedding. Let t be the root of the copy of D_1 that is attached to the endpoint of the path of length 1 in D_2. Then $(s_2, t) \in E_2$ and hence $(f(s_2), f(t)) \in E_1$. Necessarily, $f(s_2) = s_1$, hence $f(t)$ lies on a branch of finite length. But t has departing branches of any length. Contradiction. □

Lemma 4.4 *Let $T = (V_T, E_T)$ be any α-tree that only contains finite paths. Then $T \sqsubseteq_\alpha D_k$ for some $k < \omega$.*

Proof.

We define the following embedding $f : T \mapsto D_i$ for some $i < \omega$ which will be determined in the construction of f. Suppose t is the root of T. Then t has at most

countably many descendants $t_1, t_2, \ldots$ at distances $n_1, n_2, \ldots$ such that t_j has departing branches of finite but unbounded length. The other branches departing from t have length bounded by some integer m. Embed these branches in the branch of length m in the uppermost copy of D_1 in D_i. Embed t_j as the root of the copies of D_{i-1} located at the endpoints of the branches of length n_j and repeat the process starting from t_j. This process must eventually, after a finite number of times, stop as T only contains finite branches. In fact, this number determines the i we were looking for. □

We define a collection of formulae $\Psi = \{\psi_i \mid i < \omega\}$ by the following recursive definition:

- $\psi_1 = \bigwedge_{i<\omega} \langle \alpha^i \rangle \mathit{true}$;
- $\psi_{n+1} = \bigwedge_{i<\omega} \langle \alpha^i \rangle \psi_n$.

Note that ψ_1 holds at a state s iff the program α can be executed from s an arbitrarily number of times; ψ_n holds iff we can n times execute the program an arbitrarily number of times. The following theorem follows immediately from the definitions.

Theorem 4.5 *For each $i < \omega$,*

1. $s_i \models \psi_j$ *for* $j \leq i$;
2. $s_i \not\models \psi_j$ *for* $j > i$.

We let ψ be the formula

$$\psi = \bigwedge_{i<\omega} \psi_i.$$

Corollary 4.6 *For each $i < \omega$,*

1. $s_i \not\models \psi$;
2. $s_\omega \models \psi$.

The following theorem is easily seen to hold.

Theorem 4.7 $s_\omega \models \mathit{repeat}(\alpha)$.

We can now state the main result of this chapter.

Theorem 4.8 *Let $\mathcal{M} = (W, \pi, \rho)$ be any Kripke structure, let $\alpha \in \Pi$ and $s \in W$. Then*

$$s \models \mathit{repeat}(\alpha) \text{ iff } s \models \psi.$$

Proof.

Let T_α^s be the α-tree rooted in s. If $s \models \mathit{repeat}(\alpha)$, then there exists an α-path departing from s of infinite length and $T_\alpha^s \equiv_\alpha D_\omega$. Hence, $s \models \psi$. Conversely, if $s \not\models \mathit{repeat}(\alpha)$, then every path in T_α^s is of finite length and $T_\alpha^s \sqsubseteq_\alpha D_k$ for some $k < \omega$. Hence, $s \not\models \psi$. □

4.2 Bisimulation

Recall the definition of the relation $\equiv$ on states in Definition 3.2:

$$s \equiv t \text{ iff } s \vDash \phi \Longleftrightarrow t \vDash \phi.$$

We now define the relation $\cong_\alpha$ on computation trees by $T \cong_\alpha T'$ iff $T \equiv_\alpha T'$ and if $f : T \mapsto T'$, then $s \equiv f(s)$ for all $s \in T$. For two states s and t, we define the relation $\simeq_\alpha$ by $s \simeq_\alpha t$ iff, when T_α is the computation tree of α rooted in s and T'_α is the computation tree of α rooted in t, then $T_\alpha \cong_\alpha T'_\alpha$. We define the relation $\simeq$ on states by $s \simeq t$ iff $s \simeq_\alpha t$ for all $\alpha \in \Pi$. The relation $\simeq$ is the relation of bisimulation on states. Note that we have not required that the two states come from the same model. Thus we can extend to a notion of bisimulation between models. For two models $\mathcal{M}_1$ and $\mathcal{M}_2$, let $\mathcal{M}_1 \lesssim_\alpha \mathcal{M}_2$ iff for every state $s \in W^{\mathcal{M}_1}$ there exists an state $t \in W^{\mathcal{M}_2}$ such that $s \simeq_\alpha t$. We say that $\mathcal{M}_1 \simeq_\alpha \mathcal{M}_2$ iff $\mathcal{M}_1 \lesssim_\alpha \mathcal{M}_2$ and $\mathcal{M}_2 \lesssim_\alpha \mathcal{M}_1$. We let $\mathcal{M}_1 \simeq \mathcal{M}_2$ iff $\mathcal{M}_1 \simeq_\alpha \mathcal{M}_2$ for each $\alpha \in \Pi$.

It is obvious that, in order to study (a variant of) PDL, we only need to consider the equivalence classes of $\simeq$ in the class of all Kripke structures.

4.3 On decidability of PDL with repeat and other philosophical topics

An important question to ask is whether the logic of RPDL remains decidable. Streett [34] has argued that this is the case. His argument appears to be, however, incorrect. It seems to go awry in an inductive proof of the central lemma 4.14 in his paper, where an illegal application of the induction hypothesis is used. This error calls for some attention. Remember the Fisher-Ladner closure of a formula ϕ, $FL(\phi)$. This notion in generally introduced as "a notion of subformulae" for a formula ϕ. Naïvely we can now define a relation on formulae $\leqslant$ by

$$\psi \leqslant \phi \text{ iff } \psi \in FL(\phi).$$

We might use $\leqslant$ as a proper notion of subformulae. The relation $\leqslant$ is, however, not well-founded. Consider the following example of an infinitely decreasing chain, with a primitive,

$$[a^\star]\phi \geqslant [a][a^\star]\phi \geqslant [a^\star]\phi \geqslant \cdots$$

Hence, we may not perform induction over this relation. This is exactly what has happened in [34]. Inspection of $\leqslant$ shows that this happens only in this case: the $\star$ operator causes the trouble. This implies that, when we want to perform induction on the structure of ϕ, we have to prove the case "$\phi = \langle\alpha^\star\rangle\psi$" manually; we have no induction hypothesis to invoke.

Kozen and Tiuryn [17] recognized the problem and went to try to define a somewhat more appropriate notion of subformulae and came up with the relation $\prec$ which indeed proved to be well-founded. The only trouble with $\prec$ is that it contains not only p, for

p a primitive predicate symbol, but also $\langle a^* \rangle p$ as $\prec$ minimal elements. Hence we still have to prove this case specially in an inductive proof.

Note that the difficulty in defining a well-founded notion of subformulae is caused by the incompactness of PDL. They come together in the impossibility of defining syntactically a Universal Model using the (finitary) Segerberg axiom system. Using an infinitary axiom system, we can easily define the subformulae of a formula $[\alpha^*]\phi$ to be the set $\{[\alpha^n]\phi \mid n < \omega\}$. This relation is well-founded and suited for structural inductive arguments.

Streett [34] proved that PDL with Repeat has more expressing power than normal PDL. From this result we immediately infer the expected result that Infinitary PDL has more expressive power than PDL.

Chapter 5

Propositional Dynamic Logic of Context-Free Programs

In this chapter we develop a theory of Propositional Dynamic Logic of Context-Free Programs, or $\mathbf{PDL}_{CF}$ in short. We first address the validity problem for $\mathbf{PDL}_{CF}$ and then give an axiomatization for (a fragment of) $\mathbf{PDL}_{CF}$ and prove the completeness of this axiomatization.

5.1 The Validity Problem

As in ordinary PDL, the propositions and programs are built from two sets Φ_0 and Π_0 of primitive predicates and programs respectively. In contrast to PDL, we allow programs to be all context-free definable expressions over Π_0. The semantics for the resulting propositions in the logic is similar to the semantics for PDL-expressions. We let this brief description be the informal syntax and semantics for $\mathbf{PDL}_{CF}$, for the time being; we digress further on the subject in the subsequent sections.

We associate with each program α a set $L(\alpha) \subseteq \Pi_0^*$, namely, the set of all execution sequences of α. $L(\alpha)$ and $\rho(\alpha)$ are related in that if $w = w_1 \ldots w_n \in L(\alpha)$ and $(s,t) \in \rho(w_1)$ and $(s',t') \in \rho(w_n)$, then $(s,t') \in \rho(\alpha)$. Conversely, if $(s,t') \in \rho(\alpha)$, then there exists a $w \in L(\alpha)$ as described above. In effect, we view a program α as a context-free grammar (*i.e.* a recursive set of production rules) over terminals Π_0 and some fixed set of nonterminals. Then $L(\alpha)$ consists of all words generated by the grammar α. Note that we have restricted our definition for Π by excluding tests. Since the results in this section are negative, they carry over to the more general case.

We first give a handy definition used throughout this section.

Definition 5.1 *Let C be a class of programs over Π_0. Then $\mathbf{PDL}_C$ is the propositional dynamic logic with programs drawn from C. In particular, we use RG for regular programs; CF for context-free programs and L for a fragment of linear context-free programs.*

Note that "ordinary" PDL is $\mathbf{PDL}_{RG}$. The following theorem is due to R. Ladner. The proof presented here is from [11].

Theorem 5.2 *For any* $\alpha, \beta \in \Pi$, $p \in \Phi_0$,

$$\models (\langle\alpha\rangle p \rightarrow \langle\beta\rangle p) \text{ iff } L(\alpha) \subseteq L(\beta).$$

Proof.

($\Leftarrow$) Immediate from the definition of $\langle\alpha\rangle p$.

($\Rightarrow$) Let $w \in L(\alpha)$ and $w = b_1 \cdots b_n$ where the b_i are (not necessarily distinct) elements from Π_0. Define the Kripke structure $\mathcal{A}_w = (W^{\mathcal{A}_w}, \pi^{\mathcal{A}_w}, \rho^{\mathcal{A}_w})$ such that

- $W = \{u_0, \ldots, u_n\}$;
- $\pi(p) = \{u_n\}$;
- $\rho(b_j) = \{(u_i, u_j) \mid j = i + 1\}$.

Clearly, then, $\mathcal{A}_w, u_0 \models \langle\alpha\rangle p$ and hence, by the assumption, $\mathcal{A}_w, u_0 \models \langle\beta\rangle p$. And this implies that $w \in L(\beta)$. □

Corollary 5.3 *The validity problem for* $\mathbf{PDL}_{CF}$ *is undecidable.*

Proof.

The corollary follows from Theorem 5.2 the fact that the equivalence (and hence the inclusion) problem for context-free grammars is undecidable. □

The following theorem shows that the validity problem is in fact highly undecidable. The theorem is due to Harel, Pnueli and Stavi [11, 12].

Theorem 5.4 *There exist atomic programs* a *and* b *such that:*

1. *the validity problem for* $\mathbf{PDL}_{RG+a^\Delta ba^\Delta}$ *is* Π^1_1*-complete;*
2. *the validity problem for* $\mathbf{PDL}_{RG+\{a^\Delta b^\Delta, b^\Delta a^\Delta\}}$ *is* Π^1_1*-complete.*

where $\alpha^\Delta\beta\gamma^\Delta$ *is defined as the program* $\bigcup_{i<\omega} \alpha^i\beta\gamma^i$.

An important issue in the theory is to determine for what class C of the context-free programs, $\mathbf{PDL}_C$ remains decidable. Theorem 5.4 seems to indicate that this class is little more than the regular programs. The following conjecture is due to Olshansky and Pnueli.

Conjecture. $\mathbf{PDL}_{RG+a^\Delta b^\Delta}$ is decidable.

As Harel noted in [10], undecidability of the two classes of logics could be explained by the observation that in the presence of a program $a^\Delta ba^\Delta$ or of both programs $a^\Delta b^\Delta$ and $b^\Delta a^\Delta$, an a-transition in the pushdown automaton recognizing the language, might require either pushing or popping the stack since a plays a dual role in these languages. On the other hand, an a in $a^\Delta b^\Delta$ need only the stack to be pushed and an b to be popped. A treatment of $\mathbf{PDL}_{CF}$, in which we allow pushdown automata to act as descriptions of the programs used, might shed some more light on the subject.

5.2 Syntax and semantics

In this section we define the syntax and semantics of a fragment of $\mathbf{PDL}_{CF}$, namely $\mathbf{PDL}_L$. In this fragment we restrict the admissible set of programs to include only a fragment L of the linear context-free programs. In this fragment, only productions of the form

$$X \to aXb$$
$$X \to a$$

for nonterminals X and terminals (including ϵ) a and b. In this choice we follow Harel [9], who uses exactly the same set of programs to describe the first-order Dynamic Logic with respect to this set. The definition of the set Π follows [9] closely. The reason behind this choice is that the theory would become too cumbersome and technical details would block the view on underlying principles.

$\mathbf{PDL}_L$ is formally defined as follows. Let Φ_0 and Π_0 be sets of primitive predicates and programs, respectively. That is,

$$\Phi_0 = \{p_0, p_1, \ldots\}$$
$$\Pi_0 = \{a_0, a_1, \ldots\}$$

We further have a set Ξ of *program variables*, denoted by $X_1, X_2, \ldots$

The definition of the set Φ of propositions is exactly like in chapter 2.

The set Π' of *program terms* is defined as:

1. $\Pi_0 \subseteq \Pi'$;
2. $\Xi \subseteq \Pi'$;
3. if $\alpha, \beta \in \Pi'$ then $\alpha \cup \beta$ and $\alpha;\beta \in \Pi'$;
4. if $\tau \in \Pi'$ and $X \in \Xi$ then $\mu X.\tau \in \Pi'$.

The term $\mu X.\tau$ is intended, intuitively, to represent the (recursive) program consisting of an execution of τ and whenever the variable X is encountered, we again execute τ. A program variable X is said to be *bound* if it occurs in the scope of a μ; it is said to be *free* otherwise. A program is said to be *closed* if it contains no free $X \in \Xi$. We now define the set Π to be the largest subset of Π' in which all terms are closed.

Convention. We denote $\mu X.\tau(X)$ by $\tau^*(f)$.

For a term τ we write $\tau(X)$ to indicate that τ has a free occurrence of X. Accordingly then, for such a τ we may write $\tau(\alpha)$ for the program τ with each occurrence of X replaced by α.

Define $\tau^0(\alpha) = \alpha$ and $\tau^{i+1}(\alpha) = \tau(\tau^i(\alpha))$. We then can give the definition of $\rho(\tau^*(f))$ as

$$\rho(\tau^*(f)) = \bigcup_{i<\omega} \rho(\tau^i(\mathit{false}?)).$$

$\mathbf{PDL}_L$ is interpreted over Kripke structures and the definitions for ρ, π and $\models$ are exactly like in chapter 2, except that we replace $\rho(\alpha^*)$ by $\rho(\tau^*(f))$.

5.3 Axiomatization

In this section we propose an axiomatization for $\mathbf{PDL}_L$. Recall that the set of valid $\mathbf{PDL}_L$ propositions forms a Π^1_1-complete set by Theorem 5.4: the program construct $\alpha^\Delta\beta\gamma^\Delta$ is defined as

$$\alpha^\Delta\beta\gamma^\Delta = \mu X.(\alpha X\gamma)\cup\beta.$$

Accordingly, this set can't be finitely axiomatizable. We therefore propose an infinitary axiom system (*c.f.* [13]).

Definition 5.5 *The set of axioms AX_L for $\mathbf{PDL}_L$ contains*

1. *the axioms for propositional logic;*
2. $\langle\alpha\rangle\phi\wedge[\alpha]\psi\rightarrow\langle\alpha\rangle(\phi\vee\psi)$;
3. $\langle\alpha\rangle(\phi\vee\psi)\leftrightarrow\langle\alpha\rangle\phi\vee\langle\alpha\rangle\psi$;
4. $\langle\alpha\cup\beta\rangle\phi\leftrightarrow\langle\alpha\rangle\phi\vee\langle\beta\rangle\phi$;
5. $\langle\alpha;\beta\rangle\phi\leftrightarrow\langle\alpha\rangle\langle\beta\rangle\phi$;
6. $\langle\psi?\rangle\phi\leftrightarrow\psi\wedge\phi$;
7. $[\tau^*(f)]\phi\rightarrow[\tau^i(\mathit{false}?)]\phi$ *for all* $0\leq i<\omega$;

In addition we have the following inference rules:

1. *modus ponens: from* ϕ, $\phi\rightarrow\psi$ *infer* ψ;
2. *modal generalisation: from* ϕ *infer* $[\alpha]\phi$ *for any* $\alpha\in\Pi$;
3. ∞*-rule: from* $\{\psi\rightarrow[\tau^i(\mathit{false}?)]\psi\}_{i<\omega}$ *infer* $\psi\rightarrow[\tau^*(f)]\psi$.

Note that, in fact, we have introduced $[\tau^*(f)]\phi$ as an abbreviation for $\bigwedge_{i<\omega}[\tau^i(\mathit{false}?)]\phi$. We get from Axiom 7, by contraposition, $\langle\tau^i(\mathit{false}?)\rangle\phi\rightarrow\langle\tau^*(f)\rangle\phi$ and we can regard $\langle\tau^*(f)\rangle\phi$ as an abbreviation for $\bigvee_{i<\omega}\langle\tau^i(\mathit{false}?)\rangle\phi$.

We define a derivation to be a countable sequence of well-formed formulae, each of which is either an instance of a axiom or the conclusion of an inference rule whose premisses occur earlier in the sequence. The last formula in the sequence is called the conclusion of the derivation. Any formula ϕ for which such a derivation exists is called derivable or provable; we write $\vdash_L\phi$.

Theorem 5.6 (Soundness Theorem) *If* $\vdash_L\phi$ *then* $\models\phi$.

Proof.

Inspection of the system AX_L shows that all axioms are valid and that rules of inference preserve validity. □

5.4 Completeness

In this section we prove the completeness of the system AX_L adapting the completeness proof of Lemma 2.8. First we define the set $\Pr(AX_L) \subseteq \Phi$ as the set of all provable propositions in the system AX_L. Now let the Kripke structure $\mathcal{A} = (W, \pi, \rho)$ be given by:

- $W = \{s \subseteq \Phi \mid \Pr(AX_L) \subseteq s \text{ and } s \text{ is maximally consistent}\}$.
- $\pi(p) = \{s \mid p \in s\}$ for all primitive p;
- $\rho(a) = \{(s,t) \mid \forall \phi.([a]\phi \in s \Longrightarrow \phi \in t)\}$

We extend π and ρ in the usual way to the sets Φ and Π to get a Kripke structure. Note that we can again define ρ by $\rho(a) = \{(s,t) \mid \forall \phi.(\phi \in t \Longrightarrow \langle a \rangle \phi \in s\}$.

Definition 5.7 *Let Σ be a set of formulae. Σ is* consistent *if not $\Sigma \vdash_L$ false.*

Lemma 5.8 *For all propositions ϕ, $\mathcal{A}, s \models \phi$ iff $\phi \in s$.*

Proof.

We use induction on the structure of ϕ. All cases carry over from Lemma 2.8, but now there is the additional case $\phi = \langle \tau^*(f) \rangle \psi$, which we dually prove below.

$s \models [\tau^*(f)]\psi$ iff $s \models [\tau^i(\mathit{false}?)]\psi$ for all $0 \leq i < \omega$ iff $[\tau^i(\mathit{false}?)]\psi \in s$, by induction hypothesis, hence $[\tau^*(f)]\psi \in s$ by construction. Conversely, let $[\tau^*(f)]\psi \in s$. Then, for all i, $[\tau^i(\mathit{false}?)]\psi \in s$. Hence, by induction hypothesis, $s \models [\tau^i(\mathit{false}?)]\psi$ and $s \models [\tau^*(f)]\psi$ by definition. □

Theorem 5.9 (Completeness Theorem) *For all propositions ϕ,*

$$\models \phi \text{ iff } \vdash_L \phi.$$

Proof.

One direction is the Soundness Theorem. The other direction follows from Lemma 5.8 and the Completeness Lemma. □

Chapter 6

Related Topics

In this chapter we review some related topics and give a brief outline of results obtained there which are relevant in our exposition.

6.1 Propositional Algorithmic Logic

Mirkowska [19] defined a propositional version of Algoritmic Logic [31] called **PAL**. **PAL** is closely related to **PDL** but differs in some minor and some more major respects. We treat syntax and semantics of **PAL** in a manner which resembles **PDL**.

6.1.1 Syntax, semantics, axiomatization

Like **PDL**, **PAL** has two syntactic objects, *programs* and *propositions*. Programs are built from a set of primitive programs Π_0 by the following rules.

- If α and β are programs, then $\alpha;\beta$ and $\alpha \cup \beta$ are programs;
- if α and β are programs and ϕ_0 is a formula of **CPC**, then **while** ϕ_0 **do** α **od** and **if** ϕ_0 **then** α **else** β **fi** are programs.

Note that **PAL** has both deterministic and nondeterministic choice operators. Furthermore it has only deterministic looping.

PAL has two modalities $\Diamond$ and $\Box$, which are not related as $\Diamond = \neg\Box\neg$. The semantics of $\langle\alpha\rangle\phi$ is "program α can terminate with ϕ holding" but the meaning of $[\alpha]\phi$ is "all executions of α are successful and ϕ holds upon termination of each of them". The **PAL** formula $[\alpha]\phi$ relates then to the **LPDL** formula $\neg loop(\alpha) \wedge [\alpha]\phi \wedge \langle\alpha\rangle\phi$. The last term is needed to insure that α can be executed, which otherwise causes $\neg loop(\alpha)$ and $[\alpha]\phi$ to hold vacuously. Note that the decidability of **PAL** follows from the decidability of **LPDL**. **PAL** formulae are constructed just like **PDL** formulae.

Like **PDL**, **PAL** is interpreted over Kripke models. The relation ρ is extended in the usual way for ; and $\cup$ but needs special attention for the two other program connectives.

- $(s,t) \in \rho(\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ else } \beta \textbf{ fi})$ iff $\begin{cases} (s,t) \in \rho(\alpha) \text{ and } \mathcal{M}, s \models \phi_0; \\ (s,t) \in \rho(\beta) \text{ and } \mathcal{M}, s \models \neg\phi_0. \end{cases}$
- $(s,t) \in \rho(\textbf{while } \phi_0 \textbf{ do } \alpha \textbf{ od})$ iff, for some $i < \omega$, $(s,t) \in \rho((\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ fi})^i)$ and $\mathcal{M}, t \models \neg\phi_0$.

Note that ρ is well-defined since we only allow CPC formulae in the construction of **while** and **if** programs.

Again the function π induces the relation $\models$ and the two deterministic program connectives need again special care. One can prove the following lemma [19].

Lemma 6.1 *For each program α and* CPC *formula ϕ_0,*

$$\mathcal{M}, s \models \langle \textbf{while } \phi_0 \textbf{ do } \alpha \textbf{ od} \rangle \psi \textit{ iff } \mathcal{M}, s \models \langle (\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ fi})^i \rangle (\neg\phi_0 \wedge \psi)$$

for some $i < \omega$.

In the above lemma, the program

$$\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ fi}$$

is an abbreviation for

$$\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ else } \textit{skip} \textbf{ fi}.$$

Thus, if **if** ϕ_0 **then** α **fi** reaches a state s such that $\mathcal{M}, s \models \neg\phi_0$ after i executions, it also reaches that state after $j > i$ executions. It just "stays there" for some time. Such an equivalence does not hold in general for [**while** ϕ_0 **do** α **od**]ψ, however. The if-direction is valid, because there must be an upperbound to the number of steps a **while** program takes in order to terminate successfully. The only if-direction, although, is not valid. However, if we consider only special structures, then an analogous lemma can be obtained. These structures are required to have the so-called *finite degree of nondeterminism* property, that is, for each $s \in W$ and for any $\alpha \in \Pi$, the set

$$R(\alpha, s) = \{t \in W \mid (s,t) \in \rho(\alpha)\}$$

must be finite.

Lemma 6.2 *For each program α and* CPC *formula ϕ_0, if $\mathcal{M}$ has the finite degree of nondeterminism property, then*

$$\mathcal{M}, s \models [\textbf{while } \phi_0 \textbf{ do } \alpha \textbf{ od}]\psi \textit{ iff } \mathcal{M}, s \models [(\textbf{if } \phi \textbf{ then } \alpha \textbf{ fi})^i](\neg\phi_0 \wedge \psi)$$

for some $i < \omega$.

Mirkowska then defined an infinitary axiom system for **PAL**. We adopt the convention that any occurence of $\circ$ in a axiom or rule must be understood as $\Diamond$ on all places or as $\Box$ on all places (with a slight abuse of notations). *pref* denotes an arbitrary prefix to the expression, that is, *pref* $\in (\{[a] \mid a \text{ primitive}\} \cup \{\langle a \rangle \mid a \text{ primitive}\})^*$. We define notions of derivability etc. as usual.

Definition 6.3 *The axiom system AX_{PAL} for* **PAL** *contains:*

1. $[\alpha]\phi \rightarrow \langle\alpha\rangle\phi$
2. $\langle a\rangle true \rightarrow [a]true$ *for primitive* a
3. $\circ(\alpha;\beta)\phi \leftrightarrow \circ\alpha(\circ\beta\phi)$
4. $\circ(\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ else } \beta \textbf{ fi})\phi \leftrightarrow (\phi_0 \wedge \circ\alpha\phi) \vee (\neg\phi_0 \wedge \circ\beta\phi)$
5. $\circ(\textbf{while } \phi_0 \textbf{ do } \alpha \textbf{ od})\phi \leftrightarrow (\neg\phi_0 \wedge \phi) \vee (\phi_0 \wedge \circ\alpha(\circ(\textbf{while } \phi_0 \textbf{ do } \alpha \textbf{ od})\phi)))$
6. $\langle\alpha \cup \beta\rangle\phi \leftrightarrow \langle\alpha\rangle\phi \vee \langle\beta\rangle\phi$
7. $[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$
8. $\langle\alpha\rangle(\phi \vee \psi) \leftrightarrow \langle\alpha\rangle\phi \vee \langle\alpha\rangle\psi$
9. $[\alpha](\phi \wedge \psi) \leftrightarrow [\alpha]\phi \wedge [\alpha]\psi$
10. $[\alpha]\neg\phi \rightarrow \neg\langle\alpha\rangle\phi$
11. $[\alpha]true \rightarrow (\neg\langle\alpha\rangle\phi \rightarrow [\alpha]\neg\phi)$

We further have the following inference rules:

1. *modus ponens: from* ϕ *and* $\phi \rightarrow \psi$, *infer* ψ;
2. *modal generalisation: from* $\phi \rightarrow \psi$, *infer* $\circ\alpha\phi \rightarrow \circ\alpha\psi$;
3. ∞*-rules: from* $\{pref \circ (\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ fi})^i(\phi \wedge \neg\phi_0) \rightarrow \psi\}_{i<\omega}$, *infer*

$$pref \circ (\textbf{while } \phi_0 \textbf{ do } \alpha \textbf{ od})\phi \rightarrow \psi.$$

In the above definition, α and β denote arbitrary programs, ϕ_0 denotes any **CPC** formula and ϕ and ψ any **PAL** formulae.

Inspection of the axiom system immediately gives the following theorem [19].

Theorem 6.4 *The system AX_{PAL} is sound.*

6.1.2 Completeness

In this section we review the completeness technique used in [19] to prove axiom systems for theories based on **PAL** complete. That is, axiom systems with some additional clauses. For example, we could add the axiom

$$\langle\alpha\rangle\phi \rightarrow [\alpha]\phi$$

to ensure that all programs considered are functional. The additional axioms must ensure, of course, that the resulting theory must be interpreted over f.d.n. models. It is instructive to compare this technique with the one we employ in chapter 3.

We first define Φ to be the set of all well-formed formulae of (a theory based on) **PAL**. We define the equivalence relation $\equiv$ on Φ by:

$$\phi \equiv \psi \text{ iff } \vdash \phi \to \psi \text{ and } \vdash \psi \to \phi.$$

We let $[\phi]$ denote the equivalence class of ϕ under $\equiv$. We now define the *Lindenbaum algebra L.*

Theorem 6.5 *The algebra* $L = \langle \Phi / \equiv, \wedge, \vee, \neg \rangle$ *where:*

- $[\phi] \vee [\psi] = [\phi \vee \psi]$;
- $\neg[\phi] = [\neg\phi]$;
- $[\phi] \wedge [\psi] = [\phi \wedge \psi]$

is a Boolean algebra. In this Boolean algebra:

1. $[\phi] \leqslant [\psi]$ *iff* $\vdash \phi \to \psi$;
2. $[\phi] = 1$ *iff* $\vdash \phi$;
3. $[\phi] = 0$ *iff* $\nvdash \phi$.

Theorem 6.6 *For arbitrary formulae* ϕ, **CPC** *formulae* ϕ_0 *and any program* α, *the following equalities hold.*

1. $[pref\,[\textbf{while } \phi_0 \textbf{ do } \alpha \textbf{ od}]\phi] = \sup_{i<\omega}[pref\,[(\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ fi})^i](\neg\phi_0 \wedge \phi)]$;
2. $[pref\,\langle\textbf{while } \phi_0 \textbf{ do } \alpha \textbf{ od}\rangle\phi] = \sup_{i<\omega}[pref\,\langle(\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ fi})^i\rangle(\neg\phi_0 \wedge \phi)]$.

where pref is an arbitrary prefix.

By this theorem the Lindenbaum algebra can be considered as as a Boolean algebra with an at most enumerable set of infinite operations. Hence the algebra is a *Q-algebra* [30]. By a *Q-filter* in the algebra L with a set of infinite operations Q, we shall understand a maximal filter that preserves all Q-operations. That is, a maximal filter $\mathcal{F}$ such that $\sup_{i<\omega}[pref \circ (\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ fi})^i(\neg\phi_0 \wedge \phi)] \in \mathcal{F}$ implies that there exists an $i_0 < \omega$ such that $[pref \circ (\textbf{if } \phi_0 \textbf{ then } \alpha \textbf{ fi})^{i_0}(\neg\phi_0 \wedge \phi)] \in \mathcal{F}$.

The following theorems from [30] are important.

Theorem 6.7 *For every non-zero element a in a Boolean algebra B with an at most enumerable set of infinite operations Q, there exists a Q-filter* $\mathcal{F}$ *such that* $a \in \mathcal{F}$.

Theorem 6.8 *If the theory T is consistent, then the Lindenbaum algebra L of that theory is a non-degenerate algebra and, by Theorem 6.7, the family of all Q-filters in L is a non-empty set.*

The proof of completeness of axiomatizations for various theories based on **PAL** which are interpreted in f.d.n. models proceeds by showing that we can construct a "canonical structure" for that theory. The state space of that model consists of the set of all Q-filters in the Lindenbaum algebra of that theory. For primitive programs a, $(\mathcal{F}, \mathcal{F}') \in \rho(a)$ iff $\langle a\rangle true \in \mathcal{F}$ and, for every formula ϕ, $[a]\phi \in \mathcal{F}$ implies $\phi \in \mathcal{F}'$. For primitive propositions p , $\mathcal{F} \in \pi(p)$ iff $p \in \mathcal{F}$. We can now show that for this model $\mathcal{M}$,

$$\mathcal{M}, \mathcal{F} \models \phi \text{ iff } [\phi] \in \mathcal{F}$$

for all $\phi \in \Phi$. From this, completeness easily follows.

Theorem 6.9 *The system AX_{PAL} is complete.*

Proof.

Suppose $\not\vdash \phi$, then $[\neg\phi] \neq 0$ in the Lindenbaum algebra of that theory and hence, by Theorem 6.7, $[\neg\phi] \in \mathcal{F}$ for some Q-filter $\mathcal{F}$. From this it follows, $\mathcal{M}, \mathcal{F} \models \neg\phi$ rendering ϕ not valid. □

6.2 Temporal Logic

One major drawback in the theory developed so far is the impossibility to describe the run-time behavior of programs. In fact we view programs as being

1. *computational*, or designed to compute some explicit output;
2. executed *instantly* (as we only keep track of their input/output behavior).

Thus we are unable to model programs that are not supposed to compute something and/or to terminate ever. Networks, operating systems or database systems are prime examples of these programs. We can say that the latter provide an *environment* which *reacts* to inputs, rather than which computes an output from an input. Hence Pnueli [23] proposed the term *reactive systems* for this kind of computer program. He defined a logic, called *Temporal Logic of Programs*, which we briefly discuss below, to describe the behavior of reactive systems.

6.2.1 Background

Before we go into details, let us first develop some intuition behind the logic. One approach can be to associate with each program the set of all execution paths, that is, functions $\sigma : \omega \mapsto W$ which enumerate consecutive states in one computation of a program. This is essentially the way programs are viewed in Temporal Logic. We then introduce modal operators to reason about properties of paths. Things we are interested in to describe include:

- *throughout* the computation of a program a condition ϕ holds, or ϕ holds in each state of the computation path;
- *during* the computation a condition ϕ is enabled, or ϕ holds in some state of the path;
- during a computation, some condition ϕ holds *until* a condition ψ is enabled.

See [20, 29] for a discussion of these and related items.

One aspect of time is implicit in this approach, namely, that time is a linearly ordered set. For this reason, the logic is called *linear time*. A consequence is that we only are able to describe properties of one particular path, or properties of all paths. We cannot state something like "there exists a path which validates such-and-such". In order to be able to do so, we must associate with each program its *computation tree* (*c.f.* [9]). In this approach, in each moment of time there exist (possibly) many alternative "futures", or next states. This logic is therefore called *branching time* as we view time as a branching tree. Now we are able to state properties like the one mentioned above.

Lamport [18] has defined a linear and a branching time logic, and has compared their expressive power. This requires some ingenuity since he proved that basicly the logics are incomparable, that is, there exist sentences which are valid in one logic but not in the other. With an appropriate comparison condition, he proved that linear time logic is more expressive than branching time logic and concluded that we only need the former.

Emerson *et al.* [4] later refuted his argument by showing that Lamport had used two, too restrictive kinds of logic and that his result on more expressiveness of linear time logic does not hold in more general logics. They conclude that, although linear time logic may be more suitable to verify preexisting programs, branching time logic has a right on its own, especially in program specification. We would like to add the argument for branching time logic, that it suits more naturally the concept of *program* as it arises in the context of PDL than linear time logic does.

6.2.2 Linear Time

Pnueli [23] defined a Temporal Logic with linear time. We give basic definitions, following the exposition in [8]. Like PDL, TL has a set of primitive predicates Φ_0. The set Φ of all formulae is defined as:

- $\Phi_0 \subseteq \Phi$;
- if $\phi, \psi \in \Phi$, then $\phi \vee \psi, \neg\phi \in \Phi$;
- if $\phi, \psi \in \Phi$, then $\Box\phi, \bigcirc\phi, \phi U \psi \in \Phi$.

The meaning of the modal connectives is:

- $\Box$ means "henceforth" (*i.e.* from now on, including the present);
- $\bigcirc$ means "next" (*i.e.* at the next state);
- U means "until";
- $\Diamond$, as usual, is shorthand for $\neg\Box\neg$.

The logic is interpreted over a *state sequence model*. A state sequence is a pair (S, σ) where S is a set of states and $\sigma : \omega \mapsto S$ is a *surjective* function, enumerating S as a

sequence

$$\sigma_0, \sigma_1, \ldots, \sigma_n, \ldots.$$

Note that repetitions are allowed and necessary when S is finite.

A model is a triple $\mathcal{M} = (S, \sigma, \pi)$ where π is an interpretation function for primitive predicates, as usual. The relation

$$\mathcal{M}, j \models \phi$$

meaning "ϕ is true at the j-th state σ_j in $\mathcal{M}$", is defined by

$$\begin{array}{lll}
\mathcal{M}, j \models p & \text{iff} & \sigma_j \in \pi(p) \\
\mathcal{M}, j \models \phi \vee \psi & \text{iff} & \mathcal{M}, j \models \phi \text{ or } \mathcal{M}, j \models \psi \\
\mathcal{M}, j \models \neg\phi & \text{iff} & \mathcal{M}, j \not\models \phi \\
\mathcal{M}, j \models \bigcirc\phi & \text{iff} & \mathcal{M}, j+1 \models \phi \\
\mathcal{M}, j \models \Box\phi & \text{iff} & \text{for all } k \geq j, \mathcal{M}, k \models \phi \\
\mathcal{M}, j \models \phi U \psi & \text{iff} & \text{for some } k \geq j, \mathcal{M}, k \models \psi \text{ and} \\
 & & \text{for every } i \text{ such that } j \leq i < k, \\
 & & \mathcal{M}, i \models \phi
\end{array}$$

Intuitively, this semantics amounts to a multimodal logic with two modalities and interpreting $\Box$ by the relation $\leq$, and $\bigcirc$ by the relation $Succ$. Apart from that, we have the connective U. The connection between the two modalities is that $\Box$ is the reflexive, transitive closure of $\bigcirc$. This observation is the key to the completeness theorem to follow.

We define the following set of axioms AX_{TL} for TL:

1. $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$
2. $\bigcirc(\phi \rightarrow \psi) \rightarrow (\bigcirc\phi \rightarrow \bigcirc\psi)$
3. $\bigcirc\neg\phi \leftrightarrow \neg\bigcirc\phi$
4. $\Box\phi \rightarrow \phi \wedge \bigcirc\Box\phi$
5. $\Box(\phi \rightarrow \bigcirc\phi) \rightarrow (\phi \rightarrow \Box\phi)$
6. $\phi U \psi \rightarrow \Diamond\phi$
7. $\phi U \psi \leftrightarrow \psi \vee (\phi \wedge \bigcirc(\phi U \psi))$

We further have the inference rules *modus ponens* and two *necessitation* rules, namely, from ϕ infer $\Box\phi$, or $\bigcirc\phi$. The first two axioms, together with the necessitation rules state that the logic is normal in both modalities. Axiom 3 expresses the interpretation of $\bigcirc$ by a total function. Axioms 4 and 5 correspond to the interpretation of $\Box$ by the reflexive transitive closure of $\bigcirc$. Note the similarity between these axioms and axioms 7 and 8 from PDL. Axiom 4 implies immediately the reflexivity schema $\Box\phi \rightarrow \phi$ indicating that $\Box$ defines a reflexive relation. Axiom 5 expresses the *induction principle* that any set that contains σ_j and is closed under the taking of successor states, must contain all states from σ_j on.

These remarks lead immediately to the following theorem.

Theorem 6.10 *The system AX_{TL} is sound.*

The proof of the completeness of the axiom system involves an interesting use of the Filtration technique described in chapter 3. We give the main points of the completeness proof below.

First, construct a model $\mathcal{M}' = (S', R'_{\Box}, R'_{\bigcirc}, \pi')$ as follows: S' consists of all maximally consistent sets s of formulae such that $\Pr(AX_{TL}) \subseteq s$. The relations $R'_{\Box}$ and $R'_{\bigcirc}$ are defined as

$$(s,t) \in R'_{\Box} \text{ iff } \{\phi \mid \Box\phi \in s\} \subseteq t$$

$$(s,t) \in R'_{\bigcirc} \text{ iff } \{\phi \mid \bigcirc\phi \in s\} \subseteq t$$

π' is defined as usual: $\pi'(p) = \{s \mid p \in s\}$. Fix a formula ϕ such that $\not\vdash \phi$. Then there exists a state $s_\phi \in S'$ such that $\phi \notin s_\phi$. Let

$$S = \{u \mid (s_\phi, u) \in (R'_{\Box})^*\}.$$

We will work with this state space, that is, with the model $\mathcal{M}'' = (S, R'_{\Box}, R'_{\bigcirc}, \pi')$. As $R'_{\Box}$ is provably not the reflexive transitive closure of $R'_{\bigcirc}$, we will have to collapse the associated model by filtration to achieve that property.

As a filtration set Γ, we define

$$\Gamma = Sf(\phi) \cup \{\bigcirc\Box\psi \mid \Box\psi \in Sf(\phi)\} \cup \{\bigcirc(\psi U \chi), \Box\neg\chi, \bigcirc\Box\neg\chi, \neg\chi \mid \psi U \chi \in Sf(\phi)\}$$

The definition of Γ-filtration is adapted as follows.

$$s \equiv_\Gamma t \text{ iff } s \cap \Gamma = t \cap \Gamma.$$

We let $[s]$ be the equivalence class of s under $\equiv_\Gamma$ and define $S_\Gamma = \{[s] \mid s \in S\}$.

A relation $R_{\bigcirc}$ on S_Γ is defined to be a Γ-filtration of $R'_{\bigcirc}$ if and only if

1. $(s,t) \in R'_{\bigcirc}$ implies $([s],[t]) \in R_{\bigcirc}$, and
2. $([s],[t]) \in R_{\bigcirc}$ implies $\{\psi \mid \bigcirc\psi \in s \cap \Gamma\} \subseteq t$.

Likewise for $R_{\Box}$.

Lemma 6.11 *If a relation $R_{\bigcirc}$ on S_Γ is a Γ-filtration of $R'_{\bigcirc}$, then $R^*_{\bigcirc}$ is a Γ-filtration of $R'_{\Box}$.*

We let π be the valuation induced by π' in the obvious way. So now we have constructed a model $\mathcal{M} = (S_\Gamma, R_{\bigcirc}, R^*_{\bigcirc}, \pi)$ in which one relation is the reflexive transitive closure of the other. In order to get our computation path model we have to enumerate the state space S_Γ. We therefore introduce the notion of a *cluster*. Let R be any binary relation on S_Γ. R induces an equivalence relation $\sim_R$ by

$$s \sim_R t \text{ iff } [s = t] \text{ or } [(s,t) \in R \wedge (t,s) \in R].$$

Definition 6.12 *A* cluster *of a binary relation R, is an equivalence class of $\sim_R$.*

We denote such an equivalence class of a state s by C_s. We can order clusters by putting $C_s \leq C_t$ iff $(s,t) \in R$. Likewise, we have an ordering $<$ on clusters. Note that clusters can have one or more states, in general countably many.

We can picture a chain of R-clusters, called an *R-chain*, as follows.

$$\cdots \to \bullet \to \bullet \to \bigcirc \to \bullet \to \bigcirc \to \cdots$$

In this picture, the $\to$ denotes the relation $<$. When an R-chain is finite, we have a *first* and a *last* element in the chain. This will necessarily be so when the state space is finite.

We now define the relation R^c on S_Γ by

$$(x,y) \in R^c \text{ iff } \forall s \in x \exists t \in y((s,t) \in R'_\Box).$$

Then R^c is reflexive, transitive and connected and hence the R^c-chain contains the whole set S_Γ. We further have $R^c \subseteq R^*_\bigcirc$. Thus the R^c-cluster of each point is contained in the $R^*_\bigcirc$-cluster of that point, and so each $R^*_\bigcirc$-cluster decomposes into a sequence of R^c-clusters.

We now "unwind" the $R^*_\bigcirc$-chain, by unwinding all the R^c-clusters in each element of the chain. If C is the first R^c-cluster, then C can be unwound into a finite $R_\bigcirc$-list, starting from any predescribed point $x \in C$, as follows. If $y \in C$, then $(x,y) \in R^c$, so $(x,y) \in R^*_\bigcirc$ and hence there is an $R_\bigcirc$-list $x = x_0, \ldots, x_n = y$. We can extend this list to contain all of C. We then "move to the next" cluster in the chain, and so on, until we reach the last $R^*_\bigcirc$-cluster in the chain (which must exist since S_Γ is finite). We then move circulary through this last cluster, repeating this list ad infinitum. The sequence of points we get from this procedure is our computation chain σ. We can now prove the following lemma.

Lemma 6.13 *Let $\mathcal{M} = (S_\Gamma, \sigma, \pi)$. If $\phi \in \Gamma$, then for any $j \in \omega$ and $s \in \sigma_j$,*

$$\phi \in s \text{ iff } \mathcal{M}, j \models \phi.$$

Now we have our machinery to prove the completeness theorem.

Theorem 6.14 *AX_{TL} is complete.*

Proof.

Remember $\not\vdash \phi$ and $\phi \notin s_\phi$ for some state s_ϕ in the state space of all maximally consistent sets of formulae. Taking a j such that $[s_\phi] = \sigma_j$, Lemma 6.13 gives $\mathcal{M}, j \not\models \phi$ and hence ϕ is not valid. □

Observe that we can easily define a multi-temporal logic in which we allow for (countably) many computation paths, each having its own $\Box$, $\bigcirc$ and U operators. The axiomatization, labeled by the path name, remains complete: we can adopt the completeness proof to generate a falsifying model for any non-theorem of the system.

6.2.3 Branching time

We first observe that linear time Temporal Logic is perfectly suited to reason about existing (specifications of) programs: we can translate the program to a description

of a computation path model and properties of the program we are interested in, are expressible in the language. Among those properties are concepts like fairness, termination etc. See [24] for a survey of current trends in linear time Temporal Logic and what we can do with it.

We also observe that this Temporal Logic is less suited for specifying programs: properties like "from state s there exists a terminating execution of the program" are not expressible. This is, of course, caused by the fact that we have only a single execution path at our disposal. A state s can partake in any number of paths. Once we have selected one particular path, we have lost track of the others.

In branching time Temporal Logic, on the other hand, the execution of a program is modeled as a computation tree. At any state s we have the whole subtree rooted in s at hand. Thus the set $\{\phi \mid s \vDash \phi\}$ contains the information of every execution started in s. Hence we are able to discuss the existence of computation paths and the possibility that a property becomes true at some path. Branching time Temporal Logic, therefore, seems to be well equiped for specification purposes.

Another observation is that a computation tree, which can (and generally will) contain many instances of the same vertex. We can, then, model the computation of a program by a directed computation *graph*. This approach may have some advantages over computation trees in the case when we want to model the computation of a set of interacting programs: they give rise to different sets of edges on the same set of vertices.

As an example of a branching time Temporal Logic, we will briefly discuss the logic **UB** as formulated in [2]. In this logic, a set of both linear time (for properties holding along one particular path) and branching time (for properties holding on some or all paths) modalities are defined. Like the other logics we have discussed, the properties are formulated as propositional formulae.

In the logic, we must quantify over branches as well as over states in one branch. The set of modalities chosen in [2] is:

$\forall G\phi$ holds at s iff ϕ is true at all nodes of the subtree rooted at s (including s).

$\forall F\phi$ holds at s iff on every path departing from s there is some state at which ϕ is true.

$\forall X\phi$ holds at s iff ϕ is true at every immediate successor of s.

$\exists G\phi$ holds at s iff there is a path departing from s such that ϕ is true at all states on this path.

$\exists F\phi$ holds at s iff ϕ is true at some node in the subtree rooted at s, *i.e.* there is a path departing from s such that ϕ is true at some state on this path.

$\exists X\phi$ holds at s iff ϕ is true at some of the immediate successors of s.

Semantics for this logic follow easily. We use the convention to denote states by s, t and branches by b. We will in the semantics quantify over states in the model and branches in the computation tree. The relation R we use is the relation which defines the tree.

1. $s \models p$ iff $s \in \pi(p)$ for atomic p
2. $s \models \phi \vee \psi$ iff $s \models \phi$ or $s \models \psi$
3. $s \models \neg\phi$ iff $s \not\models \phi$
4. $s \models \forall G\phi$ iff $\forall b \forall t(t \in b \Longrightarrow t \models \phi)$
5. $s \models \exists G\phi$ iff $\exists b \forall t(t \in b \Longrightarrow t \models \phi)$
6. $s \models \forall X\phi$ iff $\forall t((s,t) \in R \Longrightarrow t \models \phi)$
7. $s \models \exists F\phi$ iff $\exists b \exists t(t \in b \wedge t \models \phi)$
8. $s \models \forall F\phi$ iff $\forall b \exists t(t \in b \wedge t \models \phi)$
9. $s \models \exists X\phi$ iff $\exists t((s,t) \in R \wedge t \models \phi)$

The following axiomatization is provably complete for **UB** [2].

1. $\forall G(\phi \rightarrow \psi) \rightarrow (\forall G\phi \rightarrow \forall G\psi)$
2. $\forall X(\phi \rightarrow \psi) \rightarrow (\forall X\phi \rightarrow \forall X\psi)$
3. $\forall G\phi \rightarrow \forall X\phi \wedge \forall X\forall G\phi$
4. $\forall G(\phi \rightarrow \forall X\phi) \rightarrow (\phi \rightarrow \forall G\phi)$
5. $\forall G(\phi \rightarrow \psi) \rightarrow (\exists G\phi \rightarrow \exists G\psi)$
6. $\exists G\phi \rightarrow \phi \wedge \exists X\exists G\phi$
7. $\forall G\phi \rightarrow \exists G\phi$
8. $\forall G(\phi \rightarrow \exists X\phi) \rightarrow (\phi \rightarrow \exists G\phi)$

In addition we have the following rules of inference.

1. If ϕ is a substitution instance of a tautology, then $\vdash \phi$.
2. modus ponens, from ϕ, $\phi \rightarrow \psi$ infer ψ.
3. necessitation, if $\vdash \phi$ then $\vdash \forall G\phi$.

Note that, by axioms 1 and 2, the logic is normal in the modalities $\forall G$ and $\forall X$. The other modalities are defined as abbreviations, and so are the other logical connectives. Observe the similarity between this axiomatization and the system AX_{TL}, defined in the previous section.

Bibliography

[1] Bell, J.L. and A.B. Slomson, *Models and Ultraproducts*, North Holland, Amsterdam, 1979.

[2] Ben-Ari, M., A. Pnueli and Z. Manna, "The Temporal Logic of Branching Time", *Acta Inf.* 20 (1983), 207–226.

[3] Berman, F., "A Completeness Technique for D-Axiomatizable Semantics", *Proc. 11th ACM Symp. Theory of Comput.*, 1979, 160–166.

[4] Emerson, E.A. and J.Y. Halpern, "'Sometime' and 'Not Never' Revisited: On Branching versus Linear Time Temporal Logic", *J. ACM* **33** 1 (1986), 151–178.

[5] Engeler, E., "Algorithmic Properties of Structures", *Math. Syst. Theory* 1 (1967), 183–195.

[6] Fischer, M.J. and R.E. Ladner, "Propositional Dynamic Logic of Regular Programs",*J. Comput. Syst. Sci.* 18:2 (1979), 194–211.

[7] Floyd, R.W., "Assigning Meanings to Programs", *Proc. AMS Symp. Appl. Math.* 19, Amer. Math. Soc., Providence, 1967, 19–31.

[8] Goldblatt, R., *Logics of Time and Computation*, Lecture Notes 7, CSLI, Stanford, 1987.

[9] Harel, D., *First Order Dynamic Logic*, LNCS 68, Springer-Verlag, Berlin etc., 1979.

[10] Harel, D., "Dynamic Logic", in: Gabbay and Guenthner (eds.), *Handbook of Philosophical Logic II: Extensions of Classical Logic*, D. Reidel, Boston, 1984, 497–604.

[11] Harel, D., A. Pnueli and J. Stavi, "Propositional Dynamic Logic of Context-Free Programs", *Proc. 22nd IEEE Symp. Found. Comput. Sci.*, 1981, 310–321.

[12] Harel, D., A. Pnueli and J. Stavi, "Further Results on Propositional Dynamic Logic of Non–regular Programs", in: D. Kozen (ed.) *Proc. Workshop on Logics of Programs*, LNCS 131, Springer-Verlag, Berlin etc., 1981, 124–136.

[13] Keisler, J., *Model Theory for Infinitary Logic*, North Holland, Amsterdam, 1971.

[14] Kozen, D., "A Representation Theorem for Models of $\star$–Free PDL", *Proc. 7th Int. Colloq. Automata Lang. Prog.*, LNCS 85, Springer-Verlag, Berlin etc., 1982, 348–359.

[15] Kozen, D., "On Induction vs. $\star$–Continuity", in: D. Kozen (ed.), *Proc. Workshop on Logics of Programs*, LNCS 131, Springer-Verlag, Berlin etc., 1981, 167–176.