# Satisfiability and Completeness of Converse-PDL Replayed

Martin Lange

Institut für Informatik, University of Munich
mlange@informatik.uni-muenchen.de

**Abstract.** This paper reinvestigates the satisfiability problem and the issue of completeness for Propositional Dynamic Logic with Converse. By giving a game-theoretic characterisation of its satisfiability problem using focus games, an axiom system that is extracted from these games can easily be proved to be complete.

## 1 Introduction

Complete axiomatisations are essential for automated reasoning with logics. Propositional Dynamic Logic, PDL, was first introduced in [2] for program verification purposes. In [5] completeness of an axiom system for PDL proposed in [10] was proved. A different proof was given in [8].

The key to proving completeness is to establish that a finite consistent set of formulas is satisfiable. The default way to do this is of course to construct a model for this set. Other methods appeal to canonical structures of maximal consistent sets or to filtrations. Although automata-theory has been very successful for deciding satisfiability of various logics including PDL (cf. [13, 12]) it is in general not known how to use automata-theoretic algorithms in order to establish completeness.

Together with PDL, Fischer and Ladner introduced Converse-PDL, CPDL for short, [2]. It extends PDL by allowing formulas to speak about the backwards execution of a program. Computationally, CPDL is not harder than PDL: model checking can be done in linear time for both logics, satisfiability is EXPTIME-complete and both have the finite model property. However, conceptually the satisfiability problem for CPDL seems to be slightly harder than the one for PDL because of the way how formulas speaking about the forwards and backwards execution of the same program influence each other.

In recent years propositional dynamic logics have become interesting again because of their close connection to description logics, cf. [14, 3]. In this paper we characterise the satisfiability problem for CPDL in terms of simple two-player games. The naive tableau method that eliminates conjuncts and branches at disjuncts does not work because it does not capture the regeneration of least fixed point constructs correctly. To overcome this we employ an additional structure on sets called focus. This approach was first used in [6] to solve the model checking problem for the temporal logics LTL and CTL$^*$ in a game-based way.

In [7] it was shown how this technique is also helpful for solving the satisfiability problem of the temporal logics LTL and CTL, and at the same time led to simple completeness proofs. It is, as this paper shows, also applicable to CPDL. The axiom system can easily be extracted from the satisfiability games. Thus, it is divided into those axioms justifying the game rules and those capturing winning strategies for one of the players.

## 2 Syntax and Semantics

Let $\mathcal{A} = \{a, b, \ldots\}$ be a set of atomic programs and $\mathcal{P}$ be a set of propositional constants including *true* and *false*. We assume $\mathcal{P}$ to be closed under complementary propositions, i.e. $\mathcal{P} = \{\mathtt{tt}, \mathtt{ff}, q_1, \overline{q_1}, \ldots\}$ where $\overline{\overline{q}} = q$ and $\overline{\mathtt{tt}} = \mathtt{ff}$. A labelled *transition system* $\mathcal{T}$ is a tuple $(S, \{\xrightarrow{a} | a \in \mathit{Prog}\}, L)$ with state set $S$. $L : S \to 2^{\mathcal{P}}$ labels the states, such that for all $s \in S$: $\mathtt{tt} \in L(s)$, $\mathtt{ff} \notin L(s)$ and $q \in L(s)$ iff $\overline{q} \notin L(s)$. We will write $s \xrightarrow{a} t$ if $s, t \in S$, and $(s, t) \in \xrightarrow{a}$.

Formulas $\varphi$ and programs $\alpha$ of CPDL are defined in the following way.

$$
\begin{array}{lcl}
\varphi & ::= & q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle\alpha\rangle\varphi \mid [\alpha]\varphi \\
\alpha & ::= & a \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \overline{\alpha} \mid \varphi?
\end{array}
$$

where $q$ ranges over $\mathcal{P}$, and $a$ over $\mathcal{A}$. Greek letters from the end of the alphabet will denote formulas while those from the beginning will stand for programs.

Although formulas are presented in positive form to be suitable for games, negation is needed as a syntactical operation to handle formulas of the form $[\psi?]\varphi$. It is introduced and eliminated using deMorgan's laws, and the equivalences $\neg q \equiv \overline{q}$, $\neg\langle\alpha\rangle\varphi \equiv [\alpha]\neg\varphi$, $\neg\neg\varphi \equiv \varphi$. With $\overline{\varphi}$ we denote the unique formula that results from $\neg\varphi$ when negation is eliminated using these rules.

The set $\mathit{Sub}(\varphi)$ of subformulas of a given $\varphi$ is defined in the usual way for atomic propositions and boolean connectives. For formulas with modalities the subformula set depends on the program inside, e.g.

$$
\begin{array}{ll}
\mathit{Sub}(\langle a\rangle\varphi) & = \{\langle a\rangle\varphi\} \cup \mathit{Sub}(\varphi) \\
\mathit{Sub}(\langle\alpha;\beta\rangle\varphi) & = \{\langle\alpha;\beta\rangle\varphi\} \cup \mathit{Sub}(\langle\alpha\rangle\langle\beta\rangle\varphi) \\
\mathit{Sub}(\langle\alpha\cup\beta\rangle\varphi) & = \{\langle\alpha\cup\beta\rangle\varphi\} \cup \mathit{Sub}(\langle\alpha\rangle\varphi) \cup \mathit{Sub}(\langle\beta\rangle\psi) \\
\mathit{Sub}(\langle\alpha^*\rangle\varphi) & = \{\varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle\varphi, \langle\alpha\rangle\langle\alpha^*\rangle\varphi, \langle\alpha^*\rangle\varphi\} \cup \mathit{Sub}(\varphi) \\
\mathit{Sub}([\alpha^*]\varphi) & = \{\varphi \wedge [\alpha][\alpha^*]\varphi, [\alpha][\alpha^*]\varphi, [\alpha^*]\varphi\} \cup \mathit{Sub}(\varphi) \\
\mathit{Sub}(\langle\psi?\rangle\varphi) & = \{\langle\psi?\rangle\varphi\} \cup \mathit{Sub}(\psi) \cup \mathit{Sub}(\varphi) \\
\mathit{Sub}([\psi?]\varphi) & = \{[\psi?]\varphi\} \cup \mathit{Sub}(\overline{\psi}) \cup \mathit{Sub}(\varphi) \\
\mathit{Sub}(\langle\overline{\alpha\cup\beta}\rangle)\varphi & = \mathit{Sub}(\langle\overline{\alpha}\cup\overline{\beta}\rangle\varphi) \\
\mathit{Sub}(\langle\overline{\alpha;\beta}\rangle)\varphi & = \mathit{Sub}(\langle\overline{\beta};\overline{\alpha}\rangle\varphi) \\
\mathit{Sub}(\langle\overline{\alpha^*}\rangle)\varphi & = \mathit{Sub}(\langle\overline{\alpha}^*\rangle\varphi) \\
\mathit{Sub}(\langle\overline{\psi?}\rangle)\varphi & = \mathit{Sub}(\langle\psi?\rangle\varphi)
\end{array}
$$

The remaining $[\alpha]\varphi$ cases are similar to the corresponding $\langle\alpha\rangle\varphi$ cases. The notion of a *subprogram* is defined in the same way. For a set $\Phi$ of CPDL formulas we set $\mathit{Sub}(\Phi) := \bigcup\{\ \mathit{Sub}(\varphi) \mid \varphi \in \Phi\ \}$. Note that $|\mathit{Sub}(\Phi)| = O(|\Phi|)$.

Sets $\Phi$ of formulas will be interpreted conjunctively, i.e. $\varphi \vee \Phi$ for example is to be read as $\varphi \vee (\bigwedge_{\psi \in \Phi} \psi)$. We will use the following abbreviation: $\Phi^{[\alpha]} := \{\varphi \mid [\alpha]\varphi \in \Phi\}$.

CPDL formulas are interpreted over transition systems. The semantics of a CPDL formula is explained mutually recursively with an extension of the accessibility relation $\xrightarrow{a}$ to full programs $\alpha$.

$$\begin{array}{lll} s \xrightarrow{\alpha;\beta} t & \text{iff} & \exists u \in S \text{ s.t. } s \xrightarrow{\alpha} u \text{ and } u \xrightarrow{\beta} t \\ s \xrightarrow{\alpha\cup\beta} t & \text{iff} & s \xrightarrow{\alpha} t \text{ or } s \xrightarrow{\beta} t \\ s \xrightarrow{\alpha^*} t & \text{iff} & \exists n \in \mathbb{N}, s \xrightarrow{\alpha^n} t \text{ where} \\ & & \forall s,t \in S : s \xrightarrow{\alpha^0} s, \text{ and } s \xrightarrow{\alpha^{n+1}} t \text{ iff } s \xrightarrow{\alpha;\alpha^n} t \\ s \xrightarrow{\overline{\alpha}} t & \text{iff} & t \xrightarrow{\alpha} s \\ s \xrightarrow{\varphi?} s & \text{iff} & s \models \varphi \end{array}$$

We define equivalences of programs $\alpha \equiv \beta$ as $s \xrightarrow{\alpha} t$ iff $s \xrightarrow{\beta} t$ for all $s, t$ of all transition systems. Complementation of programs $\overline{\alpha}$ can be assumed to be applied to atomic progams solely because of $\overline{\alpha;\beta} \equiv \overline{\beta};\overline{\alpha}$, $\overline{\alpha \cup \beta} \equiv \overline{\alpha} \cup \overline{\beta}$, $\overline{\alpha^*} \equiv \overline{\alpha}^*$, and $\overline{\varphi?} \equiv \varphi?$. We set $\mathcal{A}^+ := \mathcal{A} \cup \{\overline{a} \mid a \in \mathcal{A}\}$, and $\overline{\overline{a}} = a$ for every $a \in \mathcal{A}^+$.

Again, assuming a transition system $\mathcal{T}$ to be fixed we define the semantics of a formula $\varphi$ just as $s \models \varphi$ instead of $\mathcal{T}, s \models \varphi$.

$$\begin{array}{lll} s \models q & \text{iff} & q \in L(s) \\ s \models \varphi \vee \psi & \text{iff} & s \models \varphi \text{ or } s \models \psi \\ s \models \varphi \wedge \psi & \text{iff} & s \models \varphi \text{ and } s \models \psi \\ s \models \langle\alpha\rangle\varphi & \text{iff} & \exists t \in S \text{ s.t. } s \xrightarrow{\alpha} t \text{ and } t \models \varphi \\ s \models [\alpha]\varphi & \text{iff} & \forall t \in S : s \xrightarrow{\alpha} t \text{ implies } t \models \varphi \end{array}$$

A formula $\varphi$ is called *satisfiable* if there is a transition system $\mathcal{T} = (S, \{\xrightarrow{a} | a \in Prog\}, L)$ and a state $s \in S$, s.t. $\mathcal{T}, s \models \varphi$. A set $\Phi$ is satisfiable if $\bigwedge \Phi$ is so. A formula $\varphi$ is called *valid*, written $\models \varphi$, if it is true in every state of every transition system. Note that $\not\models \varphi$ iff $\neg\varphi$ is satisfiable.

## 3 Satisfiability Games

A satisfiability game $\Gamma(\Phi_0)$ on a set $\Phi_0$ of CPDL formulas is played by two players, called $\forall$ and $\exists$. It is player $\exists$'s task to show that $\Phi_0$ is satisfiable, whereas player $\forall$ attempts to show the opposite. A *play* is a sequence $C_0, C_1, \ldots, C_n$ of *configurations* where $C_i \in Sub(\Phi_0) \times 2^{Sub(\Phi_0)}$ for all $i = 0, \ldots, n$. Configurations are more than non-empty sets of formulas. In every configuration, one particular formula is highlighted. This formula is said to be *in focus*, indicated by big square brackets.

Every play of $\Gamma(\Phi_0)$ starts with $C_0 = \left[\bigwedge \Phi_0\right]$. Transitions from $C_i$ to $C_{i+1}$ are instances of *game rules* which may require one of the players to make a choice on a formula in $C_i$. Game rules are written

$$\frac{\left[\varphi\right], \Phi}{\left[\varphi'\right], \Phi'}\; p\ c$$

where $C_i$ is the upper configuration, $C_{i+1}$ the lower one. The player $p$ is either $\forall$ or $\exists$, or empty if the rule does not require a player to take a choice. The choice $c$ describes what $p$ has to select. We will write

$$\frac{\ulcorner\varphi\urcorner, \Phi}{\ulcorner\varphi'\urcorner, \Phi'}\; p\ c$$

in order to abbreviate two rules of the form

$$\frac{\left[\varphi\right], \Phi}{\left[\varphi'\right], \Phi'}\; p\ c \qquad \text{and} \qquad \frac{\left[\psi\right], \varphi, \Phi}{\left[\psi\right], \varphi', \Phi'}\; p\ c$$

It might be that in the latter case the role of the choosing player becomes redundant. A class of games has the *subformula property* if the formulas in the premise (lower) of any rule are subformulas of the one in the conclusion (upper).

The CPDL game rules are presented in Figure 1. A disjunction is satisfiable iff one of its disjuncts is satisfiable. Therefore, player $\exists$ chooses one with rule ($\vee$). Conjunctions are preserved, but player $\forall$ can decide with conjunct to keep in focus with rule ($\wedge$). Rules ($\langle ? \rangle$), ($[?]$), ($\langle ; \rangle$), ($[;]$), ($\langle \cup \rangle$) and ($[\cup]$) apply equivalences for programs to reduce the size of the program in the outermost modality. Rules ($\langle * \rangle$) and ($[*]$) unfold the fixed point constructs of CPDL.

At any moment, player $\forall$ can play rule ($\mathtt{FC}$) to move the focus to another formula in the actual configuration. This is particularly necessary if the formula in focus is atomic or the unfolding of a $\langle \alpha^* \rangle \varphi$ which player $\exists$ has just fulfilled. In the first case, the play could not proceed without a focus change. In the second case the new formula in focus might not enable player $\forall$ to win the play anymore. This is because the focus is used by player $\forall$ to track a least fixed point construct, i.e. a formula of the form $\langle \alpha^* \rangle \varphi$, via its unfoldings and to show that it never gets fulfilled.

In the remaining rules ($\langle a \rangle$) and ($[a]$), $a$ can be an arbitrary atomic program or its complement: $a \in \mathcal{A}^+$. Moreover, these rules are only applicable if the set of sideformulas $\Phi$ satisfies the following condition: if $\varphi \in \Phi$ then $\varphi \in \mathcal{P}$, or $\varphi = \langle b \rangle \psi$ or $\varphi = [b]\psi$ for a $b \in \mathcal{A}^+$. Note that applying all the other rules will necessary result in such a configuration unless one of the following winning conditions applies beforehand.

Let $C_0, \ldots, C_n$ be a play of the game $\Gamma(\Phi_0)$. Player $\forall$ wins this play if

1. $C_n = \left[q\right], \overline{q}, \Phi$ or $C_n = \left[\mathtt{ff}\right], \Phi$, or
2. there is an $i < n$ s.t. $C_i = \left[\langle \alpha^* \rangle \varphi\right], \Phi$ and $C_n = \left[\langle \alpha^* \rangle \varphi\right], \Phi'$, and $Sub(\Phi_0) \cap \Phi = Sub(\Phi_0) \cap \Phi'$, and between $C_i$ and $C_n$ player $\forall$ has not used rule ($\mathtt{FC}$).

$$(\vee)\ \frac{\ulcorner \varphi_0 \vee \varphi_1 \urcorner, \Phi}{\ulcorner \varphi_i \urcorner, \Phi}\ \exists i \qquad (\wedge)\ \frac{\ulcorner \varphi_0 \wedge \varphi_1 \urcorner, \Phi}{\ulcorner \varphi_i \urcorner, \varphi_{1-i}, \Phi}\ \forall i \qquad (\langle ? \rangle)\ \frac{\ulcorner \langle \psi ? \rangle \varphi \urcorner, \Phi}{\psi, \ulcorner \varphi \urcorner, \Phi}$$

$$([?])\ \frac{\ulcorner [\psi ?] \varphi \urcorner, \Phi}{\ulcorner \overline{\psi} \vee \varphi \urcorner, \Phi} \qquad (\langle ; \rangle)\ \frac{\ulcorner \langle \alpha ; \beta \rangle \varphi \urcorner, \Phi}{\ulcorner \langle \alpha \rangle \langle \beta \rangle \varphi \urcorner, \Phi} \qquad ([;])\ \frac{\ulcorner [\alpha ; \beta] \varphi \urcorner, \Phi}{\ulcorner [\alpha][\beta] \varphi \urcorner, \Phi}$$

$$(\langle \cup \rangle)\ \frac{\ulcorner \langle \alpha_0 \cup \alpha_1 \rangle \varphi \urcorner, \Phi}{\ulcorner \langle \alpha_i \rangle \varphi \urcorner, \Phi}\ \exists i \qquad ([\cup])\ \frac{\ulcorner [\alpha_0 \cup \alpha_1] \varphi \urcorner, \Phi}{\ulcorner [\alpha_i] \varphi \urcorner, [\alpha_{1-i}] \varphi, \Phi}\ \forall i$$

$$(\langle * \rangle)\ \frac{\ulcorner \langle \alpha^* \rangle \varphi \urcorner, \Phi}{\ulcorner \varphi \vee \langle \alpha \rangle \langle \alpha^* \rangle \varphi \urcorner, \Phi} \qquad ([*])\ \frac{\ulcorner [\alpha^*] \varphi \urcorner, \Phi}{\ulcorner \varphi \wedge [\alpha][\alpha^*] \varphi \urcorner, \Phi} \qquad (\mathtt{FC})\ \frac{\left[\varphi\right], \psi, \Phi}{\varphi, \left[\psi\right], \Phi}\ \forall$$

$$(\langle a \rangle)\ \frac{\left[\langle a \rangle \varphi\right], \Phi}{\left[\varphi\right], \Phi^{[a]}, \langle \overline{a} \rangle (\Phi - \Phi^{[a]})} \qquad ([a])\ \frac{\left[[a] \varphi\right], \Phi}{\left[\varphi\right], \psi, \Phi^{[a]}, \langle \overline{a} \rangle (\Phi - \{\psi\} - \Phi^{[a]})}\ \forall \langle a \rangle \psi \in \Phi$$

**Fig. 1.** The satisfiability game rules for CPDL.

Player $\exists$ wins this play if

3. $C_n = \left[q_1\right], \ldots, q_k$ and $\{q_1, \ldots, q_k\}$ is satisfiable, or
4. there is an $i < n$ s.t. $C_i = \left[[\alpha^*]\varphi\right], \Phi$ and $C_n = \left[[\alpha^*]\varphi\right], \Phi'$, and $\mathit{Sub}(\Phi_0) \cap \Phi = \mathit{Sub}(\Phi_0) \cap \Phi'$, and between $C_i$ and $C_n$ player $\forall$ has not used rule (FC).
5. there is an $i < n$ s.t. $C_i = \left[\varphi\right], \Phi$ and $C_n = \left[\varphi\right], \Phi'$, and $\mathit{Sub}(\Phi_0) \cap \Phi = \mathit{Sub}(\Phi_0) \cap \Phi'$, and between $C_i$ and $C_n$ player $\forall$ has used rule (FC).

A player $p$ has a *winning strategy* for, or simply *wins* the game $\Gamma(\Phi_0)$ if they can enforce a play that is winning for themselves. The *game tree* for player $p$ is a representation of player $p$'s winning strategy and can be obtained from the tree of all plays of the underlying game in the following way. At nodes which require $p$ to make a choice include exactly one successor configuration from which on player $p$ can still win the remaining game. At other nodes retain all successors. Thus, every full path in player $p$'s game tree is a winning play for $p$.

*Example 1.* Take the satisfiable CPDL-formula $\varphi = \overline{q} \wedge \langle \overline{a}^* \rangle q$. A simple model for $\varphi$ consists of two states $s$ and $t$ with $s \xrightarrow{a} t$, $L(s) = \{q\}$ and $L(t) = \{\overline{q}\}$. Then $t \models \varphi$. A play won by player $\exists$ of the game $\Gamma(\varphi)$ is shown in Figure 2.

When the $\langle \overline{a}^* \rangle q$ becomes unfolded she does not choose $q$ in the first place since this would result in a win for player $\forall$. However, after the first application of rule $(\langle \overline{a} \rangle)$ she can fulfil this subformula. From then on, player $\forall$ has to change focus away from atomic propositions so that the play can continue until eventually winning condition 5 applies.

$$
\begin{array}{c}
\boxed{\overline{q} \wedge \langle \overline{a}^* \rangle q} \\
\hline
\overline{q}, \boxed{q \vee \langle \overline{a} \rangle \langle \overline{a}^* \rangle q} \\
\hline
\overline{q}, \boxed{\langle \overline{a} \rangle \langle \overline{a}^* \rangle q} \\
\hline
\boxed{\langle \overline{a}^* \rangle q}, \langle a \rangle \overline{q} \\
\hline
\boxed{q \vee \langle \overline{a} \rangle \langle \overline{a}^* \rangle q}, \langle a \rangle \overline{q} \\
\hline
\boxed{q}, \langle a \rangle \overline{q} \\
\hline
q, \boxed{\langle a \rangle \overline{q}} \\
\hline
\boxed{\overline{q}}, \langle a \rangle q \\
\hline
\overline{q}, \boxed{\langle a \rangle q} \\
\hline
\boxed{q}, \langle a \rangle \overline{q}
\end{array}
$$

**Fig. 2.** A winning play for player ∃.

**Lemma 1.** *Every game $\Gamma(\Phi_0)$ has a unique winner.*

*Proof.* Note that rules $(\vee)$ – $([\cup])$, $(\langle a \rangle)$ and $([a])$ reduce the size of the formula in focus or the size of the actual configuration. Hence, it is possible for a play to reach a configuration $\boxed{q}, \Phi$. If $\Phi = \emptyset$ then the winner is determined by condition 1 or 3 depending on whether $q = \texttt{ff}$ or not. If $\Phi \neq \emptyset$ and condition 1 does not apply then player ∀ can use rule (FC) to set the focus to a bigger formula. The argument is iterated with this new formula until a configuration is reached that consists of atomic propositions only. Again, conditions 1 and 3 determine the winner uniquely.

This is not necessarily the case if rule $(\langle * \rangle)$ or $([*])$ is played at some point since they increase the size of the actual configuration and possibly the formula in focus. But then eventually there must be a $C_i$ and a $C_n$ s.t. $Sub(\Phi_0) \cap C_i = Sub(\Phi_0) \cap C_n$. Suppose this was not the case. Then there would infinitely many configurations that differ from each other in the set of subformulas of $\Phi_0$. But $|Sub(\Phi_0)| < \infty$.

There are two possibilities for player ∀: either he has used rule (FC) between $C_i$ and $C_n$. Then player ∃ wins with her winning condition 5. Or he has not. But then between $C_i$ and $C_n$ some rules that increase and some rules that decrease the size of the actual configuration must have been played, and a repeat is only possible if this applies to the formula in focus, too. Therefore, this must have been a $\langle \alpha^* \rangle \varphi$ or a $[\alpha^*]\varphi$ and, most importantly, it is a subformula of $\Phi_0$. Note that any configuration $C$ satisfies: if $\varphi \in C - Sub(\Phi_0)$ then $\varphi = \langle a \rangle \psi$ for some $\psi$ and some $a \in \mathcal{A}^+$. This is because only rules $(\langle a \rangle)$ and $([a])$ violate the subformula property but the created formulas are always prefixed by a $\langle a \rangle$.

But then the winner is determined by winning conditions 2 or 4 depending on which formula and its unfoldings stayed in focus. Note that CPDL is non-alternating, i.e. this formula is unique. □

**Theorem 1. (Soundness)** *If player ∃ wins $\Gamma(\Phi_0)$ then $\Phi_0$ is satisfiable.*

*Proof.* Suppose player ∀ uses his best strategy but player ∃ still wins against it. Then there is a successful game tree for player ∃. We use this to construct a tree-like model $\mathcal{T}$ for $\Phi_0$. States of this model are equivalence classes $[C_i]$ of configurations under the following equivalence relation

$$C_i \sim C_j \quad \text{iff} \quad \begin{array}{l} C_i \text{ and } C_j \text{ are on the same path and between} \\ C_i \text{ and } C_j \text{ there is no application of rule } (\langle a \rangle) \text{ or } ([a]) \end{array}$$

Transitions in this model are given for some $a \in \mathcal{A}^+$ by

$$[C_i] \xrightarrow{a} [C_j] \quad \text{iff} \quad \begin{array}{l} C_i \not\sim C_j, \text{ but there is a } C_k \text{ s.t. } C_i \sim C_k \text{ and } C_{k+1} \sim C_j \\ \text{and between } C_k \text{ and } C_{k+1} \text{ rule } (\langle a \rangle) \text{ or } ([a]) \text{ was played} \end{array}$$

Finally, the labellings on the states are given by the atomic propositions that occur in a corresponding configuration:

$$q \in L([C_i]) \quad \text{if} \quad \text{there is a } C_j \text{ with } C_i \sim C_j \text{ and } q \in C_j$$

It remains to be seen that $\mathcal{T}, [C_0] \models \Phi_0$. Indeed, the following stronger fact holds: if $\varphi \in C_i$ then $\mathcal{T}, [C_i] \models \varphi$. We prove this by induction on $\varphi$. For atomic formulas $\varphi = q$ this is true by the construction of the labellings. Note that an inconsistent labelling is not possible because in such a case player ∀ would have won the corresponding play with his winning condition 1 which is excluded by assumption. Moreover, a consistent labelling can easily be extended to a maximal one without changing the truth values of the formulas involved.

For disjunctions and conjunctions it is true because of the way rules (∨) and (∧) are defined. It is trivially true for all constructs for which there is a deterministic rule as they replace formulas by equivalent ones.

The only interesting cases are those of the form $\varphi = \langle \alpha^* \rangle \psi$ and $\varphi = [\alpha^*]\psi$. It locally holds for these cases, too, since there is a deterministic rule which replaces $\varphi$ with its logically equivalent unfolding. However, in the case of $\varphi = \langle \alpha^* \rangle \psi$ one has to ensure that global correctness holds, too. I.e. the least fixed point must eventually get fulfilled.

Suppose this was not the case, i.e. there was no moment in which player ∃ could have chosen the disjunct $\psi$ after $\langle \alpha^* \rangle \psi$ was unfolded. But then player ∀ could have easily won the corresponding play by setting the focus to $\varphi$ at some point and leaving it there. The only reason why he would not have done so would be another $\langle \beta^* \rangle \chi$ which did not get fulfilled and which he left the focus on. In any case, such a play is not possible in player ∃'s game tree. □

**Lemma 2.** *If $\Phi \wedge \langle\alpha^*\rangle\varphi$ is satisfiable then so is $\Phi \wedge (\varphi \vee \langle\alpha\rangle\langle(\neg\Phi?;\alpha)^*\rangle(\varphi \wedge \neg\Phi)$.*

*Proof.* Assume

1. $\Phi \wedge \langle\alpha^*\rangle\varphi$ has a model $\mathcal{T}, s_0$, but
2. $\models \Phi \to (\neg\varphi \wedge [\alpha][(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi))$ is true.

By (1) there is a sequence $s_0, \ldots, s_n$ of states s.t. $s_0 \models \Phi$, $s_n \models \varphi$, and $s_i \xrightarrow{\alpha} s_{i+1}$ for all $i = 0, \ldots, n-1$. Take the least such $n$, i.e. $s_i \not\models \varphi$ for all $i = 0, \ldots, n-1$. We have $n > 0$ because of (2). Again by (2): $s_0 \models [\alpha][(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi)$ and therefore $s_1 \models [(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi)$. Note that

$$\begin{aligned}
[(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi) &\equiv (\neg\varphi \vee \Phi) \wedge [\neg\Phi?;\alpha][(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi) \\
&\equiv (\neg\varphi \vee \Phi) \wedge (\Phi \vee [\alpha][(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi)) \\
&\equiv (\neg\varphi \wedge \Phi) \;\;\vee\;\; \Phi \;\vee \\
&\quad\;\; (\neg\varphi \wedge [\alpha][(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi)) \;\vee \\
&\quad\;\; (\Phi \wedge [\alpha][(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi))
\end{aligned}$$

Regardless of which of the four disjuncts is fulfilled in $s_1$, because of (2) we have $s_1 \not\models \varphi$ and $s_1 \models [\alpha][(\neg\Phi?;\alpha)^*](\neg\varphi \vee \Phi)$. This argument can be iterated down the sequence $s_2, s_3, \ldots$ until $s_n$ is reached to show $s_n \not\models \varphi$ which contradicts the assumption. □

Alternatively, this lemma can be proved by translation into fixed point logic and by using Park's fixed point principle (1) and the fact that a fixed point is equivalent to its unfolding (2).

$$\text{if } \models \varphi\{\psi/Y\} \to \psi \text{ then } \models \mu Y.\varphi \to \psi \tag{1}$$

$$\models \mu Y.\varphi \leftrightarrow \varphi\{\mu Y.\varphi/Y\} \tag{2}$$

This has been done for general fixed point logic, cf. [4, 11].

**Corollary 1.** $\models \psi \to \varphi \wedge [\alpha][(\neg\psi;\alpha)^*](\varphi \vee \psi)$ *implies* $\models \psi \to [\alpha^*]\varphi$

*Proof.* By contraposition of Lemma 2. □

**Lemma 3.** *If $[\alpha]\varphi \wedge \langle\alpha\rangle\psi \wedge \Phi$ is satisfiable then $\varphi \wedge \psi \wedge \Phi^{[\alpha]} \wedge \langle\overline{\alpha}\rangle\Phi$ is satisfiable.*

*Proof.* Suppose there is a transition system with a state $s$ s.t. $s \models [\alpha]\varphi \wedge \langle\alpha\rangle\psi \wedge \Phi$. Then there is another state $t$ with $s \xrightarrow{\alpha} t$ and $t \models \varphi \wedge \psi \wedge \Phi^{[\alpha]}$. Furthermore, because of $t \xrightarrow{\overline{\alpha}} s$ we have $t \models \langle\overline{\alpha}\rangle\Phi$ and, hence, $t \models \varphi \wedge \psi \wedge \Phi^{[\alpha]} \wedge \langle\overline{\alpha}\rangle\Phi$, i.e. this formula is satisfiable. □

**Theorem 2. (Completeness)** *If $\Phi_0$ is satisfiable then player $\exists$ wins the game $\Gamma(\Phi_0)$.*

*Proof.* Assuming that $\Phi_0$ is satisfiable we show what player $\exists$'s winning strategy has to look like. All of player $\forall$'s moves preserve satisfiability. That is trivial for the boolean $\wedge$ and for rule (FC). Preservation of satisfiability in the modality rules is proved in Lemma 3. Player $\exists$ always has the chance to make a choice which preserves satisfiability, i.e. if $\Phi \wedge (\varphi_0 \vee \varphi_1)$ is satisfiable, then so is $\Phi \wedge \varphi_i$ for some $i \in \{0, 1\}$.
If a play reaches a position $\Big[\langle\alpha^*\rangle\varphi\Big], \Phi$ then player $\exists$ takes a note of the context $\Phi$ in the index of the modal formula when it is unfolded to

$$\Big[\varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle_{\neg\Phi}\varphi\Big], \Phi$$

A formula $\langle\alpha^*\rangle_{\neg\Phi}\varphi$ is interpreted as $\langle(\neg\Phi?;\alpha)^*\rangle(\varphi \wedge \neg\Phi)$. Lemma 2 shows that satisfiability is still preserved. This is done for as long as $\langle\alpha^*\rangle\varphi$ is in focus. Subscripting of already subscripted formulas is allowed, i.e. $\langle\alpha^*\rangle_{\neg\Phi_1,\ldots,\neg\Phi_k}\varphi$ is interpreted as

$$\langle(\neg\Phi_1?;\ldots;\neg\Phi_k?;\alpha)^*\rangle(\varphi \wedge \neg\Phi_1 \wedge \ldots \wedge \neg\Phi_k)$$

Once player $\forall$ removes the focus from it, player $\exists$ drops the indices that have been collected so far.

Player $\forall$ cannot win a single play of $\Gamma(\Phi_0)$ with winning condition 1 because this requires him to reach a propositionally unsatisfiable configuration which is excluded by the preservation of satisfiability. He cannot win with condition (2) either because he would enforce a play that ends on $\Big[\langle\alpha^*\rangle_{\neg\Phi,\ldots,\neg\Phi'}\varphi\Big], \Phi$. But such a configuration is also unsatisfiable because of

$$\langle\alpha^*\rangle_{\neg\Phi,\ldots,\neg\Phi'}\varphi \equiv \neg\Phi \wedge \ldots \wedge \neg\Phi' \wedge (\varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle_{\neg\Phi,\ldots,\neg\Phi'}\varphi)$$

Finally, Lemma 1 shows that player $\exists$ must win $\Gamma(\Phi_0)$. □

**Corollary 2.** *CPDL has the finite model property.*

*Proof.* Suppose $\varphi \in$ CPDL is satisfiable. According to Theorem 2, player $\exists$ has a winning strategy for $\Gamma(\varphi)$. The proof of Theorem 1 shows that a finite model can be extracted from this winning startegy. □

We will show that the winner of $\Gamma(\Phi_0)$ can be decided using exponential time. This matches the known lower and upper bounds for deciding satisfiability of CPDL formulas, [9]. Before that, we need to prove a technical lemma.

**Lemma 4.** *It suffices to explore the part of the game $\Gamma(\Phi_0)$ that consists of subformulas of $\Phi_0$ only.*

*Proof.* Assume player $p$ wins $\Gamma(\Phi_0)$. Take their game tree $T$. First, let $p = \exists$. If every application of rules ($\langle a\rangle$) or ($[a]$) is replaced by

$$\frac{\Big[\langle a\rangle\varphi\Big], \Phi}{\Big[\varphi\Big], \Phi^{[a]}} \qquad\qquad \frac{\Big[[a]\varphi\Big], \Phi}{\Big[\varphi\Big], \psi, \Phi^{[a]}} \;\; \forall\langle a\rangle\psi \in \Phi$$

then $T$ can be transformed into another game tree $T'$ for player $\exists$. Note that her winning conditions 3 and 4 are not effected by the removal of formulas of the form $\langle\overline{a}\rangle\psi$. Now take a play in $T$ which is won with condition 5. I.e. there are configurations $C_i$ and $C_n$ s.t. $Sub(\Phi_0) \cap C_i = Sub(\Phi_0) \cap C_n$ and $\left[\varphi\right] \in C_i \cap C_n$. Moreover, player $\forall$ has changed focus between $C_i$ and $C_n$. If $\varphi \in Sub(\Phi_0)$ then removing formulas not in $Sub(\Phi_0)$ results in a shorter play because there are fewer possibilities for player $\forall$ to set the focus to. If $\varphi \notin Sub(\Phi_0)$ then, with the new game rules, the focus must have been on a different formula. But then player $\forall$ cannot win the new play either since removing formulas does not give him new chances to win. According to Lemma 1, player $\exists$ still wins the new play.

Now let $p = \forall$. Note that all he does is position the focus and choose formulas of the form $\langle a\rangle\psi$ with rule ([$a$]). We describe an optimal strategy for player $\forall$, i.e. if he can win and he uses this strategy then he will win. Regarding the focus, this strategy will only make use of subformulas of $\Phi_0$.

Note that in a game tree for player $\forall$, all occurring configurations are unsatisfiable. But the converse holds, too. Thus, a significant part of his strategy, namely what he does in rule ([$a$]), is to preserve unsatisfiability with his choices. But with the new rule above where no $\langle\overline{a}\rangle(\Phi - \Phi^{[a]})$ is included, he can still preserve unsatisfiability.

It remains to be seen what he does with the position of the focus. He maintains a list of all formulas of the form $\langle\alpha^*\rangle\psi$ in decreasing order of size. At the beginning he sets the focus to the $\langle\alpha^*\rangle\psi$ which is earliest in the list or a superformula of it, and keeps it there until player $\exists$ fulfils it after it has been unfolded. Then he deletes it from the list, adds $\psi$ to its end and changes focus to the next formula which is present and earliest in the list. At any moment he checks whether he can win with condition 1 by changing focus to an atomic proposition.

This strategy guarantees him to win if he can because he will not miss out atomic propositions and if there is a $\langle\alpha^*\rangle\psi$ that does not get fulfilled, he will eventually set the focus to it. Note that by adding formulas to the end of the list he avoids creating a repeat for as long as possible.

Most importantly, player $\forall$ never needs to put the focus onto a formula which is not a subformula of $\Phi_0$. Thus, he can also win $\Gamma(\Phi_0)$ with the amended rules. □

**Theorem 3.** *Deciding the winner of $\Gamma(\Phi_0)$ is in EXPTIME.*

*Proof.* An alternating algorithm can be used to decide the winner of $\Gamma(\Phi_0)$. Lemma 4 shows that only subformulas of $\Phi_0$ need to be taken into consideration when player $\forall$'s strategy is partially determinised using a priority list to establish the position of the focus.

A single play can easily be played using polynomial space only. The algorithm needs to store the actual configuration and one that player $\forall$ thinks will occur again. The actual one gets overwritten each time a rule is played. If the focus is changed then player $\forall$'s configuration gets deleted. To validate the guesses and to disable infinite plays, the algorithm also needs to store a counter to measure the length of the play which is restarted with a different stored configuration when no repeat has been found. The size of the counter is $O(|\Phi_0| + \log |\Phi_0|)$ because there are only $|\Phi_0| \cdot 2^{|\Phi_0|}$ many different configurations when subformulas of $\Phi_0$ are considered only.

Finally, alternating polynomial space is the same as deterministic exponential time according to [1]. □

## 4 A Sound and Complete Axiomatisation

Using the same technique as in the completeness proof of the satisfiability games it is easy to prove completeness of an axiom system that can be extracted from the games.

**Definition 1.** An axiom system $\mathtt{A}$ is a finite set of axioms of the form $\vdash \varphi$ and rules of the form "if $\vdash \varphi \ldots$ then $\vdash \psi$". A proof is a finite sequence of formulas s.t. every member of this sequence is an instance of an axiom in $\mathtt{A}$ or follows from earlier ones by an application of a rule in $\mathtt{A}$. If there is a proof of $\varphi$ in $\mathtt{A}$ we write $\vdash_{\mathtt{A}} \varphi$ and often for short just $\vdash \varphi$.

Given an axiom system $\mathtt{A}$, a formula $\varphi$ is called $\mathtt{A}$-*consistent* if its negation is not derivable, i.e. $\not\vdash_{\mathtt{A}} \neg\varphi$. $\mathtt{A}$ is called *sound* if $\vdash_{\mathtt{A}} \varphi$ implies $\models \varphi$ for any $\varphi$, and *complete* if the converse is true, i.e. $\models \varphi$ implies $\vdash_{\mathtt{A}} \varphi$. Completeness can be reformulated as: if $\varphi$ is $\mathtt{A}$-consistent then $\varphi$ is satisfiable.

Completeness of an axiom system can be shown by the help of the satisfiability games in the following way.

**Proposition 1.** *If for any* $\mathtt{A}$*-consistent* $\varphi$ *player* $\exists$ *wins* $\Gamma(\varphi)$ *then* $\mathtt{A}$ *is complete for CPDL.*

The axiom system $\mathtt{A}$ that has been constructed with respect to the satisfiability games is shown in Figure 3. Note that the axioms and rules are to be taken as schemes where $\varphi$ and $\psi$ can be any formula of CPDL, $\alpha$ and $\beta$ can be any program, and $a$ can be any atomic program or the complement thereof. Remember that $\overline{\overline{a}} \equiv a$ for any $a \in \mathcal{A}^+$.

**Lemma 5.** *The rules of the satisfiability games preserve* $\mathtt{A}$*-consistency.*

*Proof.* Suppose $\Phi$ is $\mathtt{A}$-consistent. We show that every move taken by player $\forall$ results in a configuration $\Phi'$ that is also $\mathtt{A}$-consistent, and that player $\exists$ can always make a choice that preserves consistency.

The rules for conjuncts and (FC) obviously preserve consistency. If $\Phi, \varphi_0 \vee \varphi_1$ is $\mathtt{A}$-consistent then $\Phi, \varphi_i$ is $\mathtt{A}$-consistent for some $i \in \{0, 1\}$ by axiom 1 and rule $\mathtt{MP}$. Player $\exists$ can select this $\varphi_i$.

Axioms 2–6 show that the game rules for modalities and non-atomic programs preserve consistency. Axioms 7–10 do the same for the equivalences for programs which we could have formulated as game rules instead of requiring complementation to be pushed inwards in the first place.

**Axioms**

1. any tautology of propositional logic
2. $\neg\langle\alpha\rangle\varphi \leftrightarrow [\alpha]\neg\varphi$
3. $\langle\alpha \cup \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi$
4. $\langle\alpha;\beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\varphi$
5. $\langle\alpha^*\rangle\varphi \leftrightarrow \varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle\varphi$
6. $\langle\psi?\rangle\varphi \leftrightarrow \psi \wedge \varphi$
7. $\langle\overline{\alpha \cup \beta}\rangle\varphi \leftrightarrow \langle\overline{\alpha} \cup \overline{\beta}\rangle\varphi$
8. $\langle\overline{\alpha;\beta}\rangle\varphi \leftrightarrow \langle\overline{\beta};\overline{\alpha}\rangle\varphi$
9. $\langle\overline{\alpha^*}\rangle\varphi \leftrightarrow \langle\overline{\alpha}^*\rangle\varphi$
10. $\langle\overline{\psi?}\rangle\varphi \leftrightarrow \langle\psi?\rangle\varphi$
11. $[a]\varphi \wedge [a]\psi \rightarrow [a](\varphi \wedge \psi)$
12. $[a](\varphi \rightarrow \psi) \rightarrow ([a]\varphi \rightarrow [a]\psi)$
13. $\varphi \rightarrow [a]\langle\overline{a}\rangle\varphi$

**Rules**

`MP` if $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$ then $\vdash \psi$
`Gen` if $\vdash \varphi$ then $\vdash [a]\varphi$ for any $a \in \mathcal{A}^+$
`Rel` if $\vdash \psi \rightarrow \varphi \wedge [\alpha][\alpha^*](\varphi \vee \psi)$ then $\vdash \psi \rightarrow [\alpha^*]\varphi$

**Fig. 3.** A complete axiomatisation for CPDL.

The case of subscripting a $\langle\alpha^*\rangle\varphi$ is dealt with by rule `REL`.

The interesting case is rule ($\langle a\rangle$) or ($[a]$). Note that they only differ in the position of the focus, which does not effect consistency. Thus, they can be dealt with in one case. Suppose $\varphi \wedge \Phi \wedge \langle\overline{a}\rangle\Phi'$ is inconsistent, i.e. $\vdash \Phi \wedge \langle\overline{a}\rangle\Phi' \rightarrow \neg\varphi$. By axioms 11,12,2 and rule `MP` we have $\vdash [a]\Phi \wedge [a]\langle\overline{a}\rangle\Phi' \rightarrow \neg\langle a\rangle\varphi$. By axiom 1 and rule `MP`, this can be transformed into $\vdash \neg[a]\Phi \vee \neg[a]\langle\overline{a}\rangle\Phi' \vee \neg\langle a\rangle\varphi$.

Axiom 13 can be instantiated and inverted by 1 and `MP` to $\vdash \neg[a]\langle\overline{a}\rangle\Phi' \rightarrow \neg\Phi'$. Again, by 1 and `MP` this can be used to obtain $\vdash \neg[a]\Phi \vee \neg\Phi' \vee \neg\langle a\rangle\varphi$, resp. $\vdash [a]\Phi \wedge \Phi' \rightarrow \neg\langle a\rangle\varphi$ which shows that the conclusion of rule ($\langle a\rangle$) or ($[a]$) would be inconsistent as well. ☐

**Theorem 4.** *The axiom system* `A` *is sound and complete for* CPDL*.*

*Proof.* Soundness of `A` is straightforward since all the axioms are valid and the rules preserve validity. The only interesting case is the rule `Rel` whose correctness is proved in Corollary 1.

The proof of completeness of `A` is similar to the one of Theorem 2. Suppose $\Phi_0$ is `A`-consistent. Player $\forall$ is not able to win a play of $\Gamma(\Phi_0)$ with winning condition 1 since this would contradict Lemma 5. He is also unable to win a play with condition 2 because a configuration $\langle\alpha^*\rangle_{\neg\Phi,\ldots,\neg\Phi'}\varphi, \Phi$ is inconsistent by propositional reasoning.

By Lemma 1 player $\exists$ must win $\Gamma(\Phi_0)$ and by Proposition 1 the axiom system `A` is complete. ☐

A differs from the Segerberg axioms S (cf. [5, 10]) for PDL in the use of rule REL instead of the induction axiom I:

$$\vdash \varphi \wedge [\alpha^*](\varphi \rightarrow [\alpha]\varphi) \rightarrow [\alpha^*]\varphi$$

To show that REL really replaces I in A, one can consider its negation ¬I and the way player ∀ wins $\Gamma(\neg\mathtt{I})$. Depending on the exact structure of $\varphi$ and $\alpha$, the resulting play looks like

$$\begin{array}{c}
\varphi, [\alpha^*](\neg\varphi \vee [\alpha]\varphi), \boxed{\langle\alpha^*\rangle\neg\varphi} \\
\hline
\varphi, (\neg\varphi \vee [\alpha]\varphi) \wedge [\alpha][\alpha^*](\neg\varphi \vee [\alpha]\varphi), \boxed{\neg\varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle_{\neg\Phi}\neg\varphi} \\
\hline
\varphi, [\alpha]\varphi, [\alpha][\alpha^*](\neg\varphi \vee [\alpha]\varphi), \boxed{\langle\alpha\rangle\langle\alpha^*\rangle_{\neg\Phi}\neg\varphi} \\
\hline
\varphi, [\alpha^*](\neg\varphi \vee [\alpha]\varphi), \boxed{\langle\alpha^*\rangle_{\neg\Phi}\neg\varphi}
\end{array}$$

where $\Phi = \varphi \wedge [\alpha]\varphi \wedge [\alpha][\alpha^*](\neg\varphi \vee [\alpha]\varphi)$. Player ∀ wins with his winning condition 2 since he is able to keep the focus on $\langle\alpha^*\rangle\neg\varphi$ which gets subscripted with the context. Subscripting is captured by rule REL.

## References


1. A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, January 1981.
2. M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, April 1979.
3. G. de Giacomo and F. Massacci. Combining deduction and model checking into tableaux and algorithms for Converse-PDL. *Information and Computation*, 162:117–137, 2000.
4. D. Kozen. Results on the propositional $\mu$-calculus. *TCS*, 27:333–354, December 1983.
5. D. Kozen and R. Parikh. An elementary proof of the completeness of PDL (note). *TCS*, 14:113 – 118, 1981.
6. M. Lange and C. Stirling. Model checking games for CTL$^*$. In *Proc. Conf. on Temporal Logic, ICTL'00*, pages 115–125, Leipzig, Germany, October 2000.
7. M. Lange and C. Stirling. Focus games for satisfiability and completeness of temporal logic. In *Proc. 16th Symp. on Logic in Computer Science, LICS'01*, Boston, MA, USA, June 2001. IEEE.
8. V. R. Pratt. A practical decision method for propositional dynamic logic. In *Proc. 10th Symp. on Theory of Computing, STOC'78*, pages 326–337, San Diego, California, May 1978.
9. V. R. Pratt. Models of program logics. In *Proc. 20th Symp. on Foundations of Computer Science, FOCS'79*, pages 115 – 122. IEEE, 1979.
10. K. Segerberg. A completeness theorem in the modal logic of programs. *Notices of the AMS*, 24(6):A–552, October 1977.
11. C. Stirling. Modal and temporal logics. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2 (Background: Computational Structures), pages 477–563. Clarendon Press, Oxford, 1992.

12. R. S. Streett. Propositional dynamic logic of looping and converse is elementarily decidable. *Information and Control*, 54(1/2):121–141, July 1982.
13. M. Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logic of programs. *Journal of Computer and System Sciences*, 32:183–221, 1986.
14. W. A. Woods and J. G. Schmolze. The KL-ONE family. In Fritz Lehmann, editor, *Semantic Networks in Artificial Intelligence*, pages 133–177. Pergamon Press, Oxford, 1992.