Propositional Dynamic Logic of Context-Free Programs and Fixpoint Logic with Chop

Martin Lange

University of Munich, Institut für Informatik, Oettingenstr. 67, D-80538 München, Germany

Rafał Somla

University of Uppsala, IT Department, Box 337, SE-751 05 Uppsala, Sweden

Abstract

This paper compares Propositional Dynamic Logic of Non-Regular Programs and Fixpoint Logic with Chop. It identifies a fragment of the latter which is equi-expressive to the former. This relationship transfers several decidability and complexity results between the two logics.

Key words: Program Logics, Context-Free Languages, μ-Calculi

1 Introduction

Propositional Dynamic Logic (PDL) extends multi-modal logic with an infinite set of modal relations. In ordinary PDL – as defined by Fischer and Ladner [1] and building on a proposal by Pratt [10] – these relations form a Kleene Algebra. PDL can therefore be used to reason about programs involving non-deterministic choice, concatenation and iteration.

An extension of PDL is *Propositional Dynamic Logic of Non-Regular Programs* (PDL[CFG]) as defined by Harel, Pnueli and Stavi [2]. There, modal relations are constructed using the full power of context-free grammars rather than regular expressions only.

Let $\mathcal{P} = \{p, q, \ldots\}$ be a set of propositional constants and Σ be a finite alphabet. PDL[CFG] formulas in positive normal form are defined by

$$\varphi \quad ::= \quad q \quad | \quad \neg q \quad | \quad \varphi \vee \varphi \quad | \quad \varphi \wedge \varphi \quad | \quad \langle G \rangle \varphi \quad | \quad [G]\varphi$$

Preprint submitted to Elsevier Science *9 February 2006*

where $q \in \mathcal{P}$ and G is a context-free grammar (N, Σ, S, P) with N a finite set of non-terminals, Σ a finite alphabet, $S \in N$ a designated starting symbol, and $P \subseteq N \times (N \cup \Sigma)^*$ a finite set of production rules [4]. The language generated by grammar G is $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$.

A Kripke structure is a tuple $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \Sigma\}, \lambda)$ where $\mathcal{S}$ is a set of states, each $\xrightarrow{a} : \mathcal{S} \times \mathcal{S}$ a binary relation between states, and $\lambda : \mathcal{S} \to 2^{\mathcal{P}}$ a function that assigns to each state the set of propositions that hold true in it.

The accessibility relations $\xrightarrow{a}$ can naturally be extended to finite words over Σ via $\xrightarrow{\epsilon} := \{(s, s) \mid s \in \mathcal{S}\}$ and $\xrightarrow{aw} := \xrightarrow{a} \circ \xrightarrow{w}$ for any $w \in \Sigma^*$.

The semantics of PDL[CFG] w.r.t. a Kripke structure $\mathcal{T}$ is explained inductively. Let $s, t \in \mathcal{S}$.

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

It was shown that PDL[CFG] does not retain nice model-theoretic properties of PDL. It is undecidable and does not have the finite model property. It is also strictly more expressive than PDL, it can even express properties that are not definable by finite Rabin tree automata, e.g. $\langle a^n b^n \rangle \mathtt{tt}$.

Another modal logic that is capable of expressing non-regular properties is Müller-Olm's *Fixpoint Logic with Chop* [9]. It extends Kozen's modal μ-calculus [6] with a sequential composition operator.

Let $\mathcal{P}$ and Σ be as introduced above, $\mathcal{V} = \{Z, Y, \ldots\}$ be a set of propositional variables. Formulas of FLC are given by the following grammar.

$$\varphi \quad ::= \quad q \mid \neg q \mid Z \mid \tau \mid \langle a \rangle \mid [a] \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mu Z.\varphi \mid \nu Z.\varphi \mid \varphi ; \varphi$$

where $q \in \mathcal{P}$, $Z \in \mathcal{V}$, and $a \in \Sigma$. We will write σ for μ or ν, and use the following abbreviations: $\mathtt{tt} := q \vee \neg q$, $\mathtt{ff} := q \wedge \neg q$ for some $q \in \mathcal{P}$.

The function space $2^{\mathcal{S}} \to 2^{\mathcal{S}}$ together with the partial order given by

$$f \sqsubseteq g \text{ iff } \forall X \subseteq \mathcal{S} : f(X) \subseteq g(X)$$

forms a complete lattice with joins $\sqcup$ and meets $\sqcap$. By the Knaster-Tarski

Theorem [11] the least and greatest fixpoints of monotonic functionals $F : (2^{\mathcal{S}} \to 2^{\mathcal{S}}) \to (2^{\mathcal{S}} \to 2^{\mathcal{S}})$ exist. They are used to interpret fixpoint formulas of FLC. We call an element of $2^{\mathcal{S}} \to 2^{\mathcal{S}}$ a predicate transformer. A function $f : 2^{\mathcal{S}} \to 2^{\mathcal{S}}$ that is constant, i.e. $f(X) = f(Y)$ for all $X, Y \subseteq \mathcal{S}$, is called a *predicate*. The next lemma is straight-forward.

Lemma 1 *If f and g are predicates then so are $f \sqcup g$ and $f \sqcap g$. If g is a predicate and f is any predicate transformer then $f \circ g$ is a predicate.*

An environment is a $\rho : \mathcal{V} \to (2^{\mathcal{S}} \to 2^{\mathcal{S}})$ that maps variables to predicate transformers. The environment $\rho[Z \mapsto f]$ maps Z to f and agrees with ρ on all other arguments. Given a Kripke structure $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \Sigma\}, \lambda)$, the semantics $[\![\cdot]\!]^{\mathcal{T}}_{\rho} : 2^{\mathcal{S}} \to 2^{\mathcal{S}}$ of an FLC formula, relative to $\mathcal{T}$ and ρ, is a monotone predicate transformer w.r.t. $\subseteq$.

$$\begin{array}{rcl}
[\![q]\!]^{\mathcal{T}}_{\rho} & := & \lambda T.\{t \mid q \in L(t)\} \\
[\![Z]\!]^{\mathcal{T}}_{\rho} & := & \rho(Z) \\
[\![\tau]\!]^{\mathcal{T}}_{\rho} & := & \lambda T.T \\
[\![\varphi \vee \psi]\!]^{\mathcal{T}}_{\rho} & := & [\![\varphi]\!]^{\mathcal{T}}_{\rho} \sqcup [\![\psi]\!]^{\mathcal{T}}_{\rho} \\
[\![\varphi \wedge \psi]\!]^{\mathcal{T}}_{\rho} & := & [\![\varphi]\!]^{\mathcal{T}}_{\rho} \sqcap [\![\psi]\!]^{\mathcal{T}}_{\rho} \\
[\![\langle a\rangle]\!]^{\mathcal{T}}_{\rho} & := & \lambda T.\{s \in \mathcal{S} \mid \exists t \in T, \text{ s.t. } s \xrightarrow{a} t\} \\
[\![[a]]\!]^{\mathcal{T}}_{\rho} & := & \lambda T.\{s \in \mathcal{S} \mid \forall t \in \mathcal{S}, \text{if } s \xrightarrow{a} t \text{ then } t \in T\} \\
[\![\mu Z.\varphi]\!]^{\mathcal{T}}_{\rho} & := & \bigsqcap\{f : 2^{\mathcal{S}} \to 2^{\mathcal{S}} \mid f \text{ monotone, } [\![\varphi]\!]^{\mathcal{T}}_{\rho[Z\mapsto f]} \sqsubseteq f\} \\
[\![\nu Z.\varphi]\!]^{\mathcal{T}}_{\rho} & := & \bigsqcup\{f : 2^{\mathcal{S}} \to 2^{\mathcal{S}} \mid f \text{ monotone, } f \sqsubseteq [\![\varphi]\!]^{\mathcal{T}}_{\rho[Z\mapsto f]}\} \\
[\![\varphi;\psi]\!]^{\mathcal{T}}_{\rho} & := & [\![\varphi]\!]^{\mathcal{T}}_{\rho} \circ [\![\psi]\!]^{\mathcal{T}}_{\rho}
\end{array}$$

A state s of a Kripke structure $\mathcal{T}$ satisfies a formula φ under ρ, written $\mathcal{T}, s \models_{\rho} \varphi$, iff $s \in [\![\varphi]\!]^{\mathcal{T}}_{\rho}(\mathcal{S})$. If φ is a closed formula then ρ can be omitted and we write $[\![\varphi]\!]^{\mathcal{T}}(\mathcal{S})$ as well as $\mathcal{T}, s \models \varphi$. Given a $\varphi \in$ PDL[CFG] and a closed $\psi \in$ FLC, we write $\varphi \equiv \psi$ iff for any Kripke structure $\mathcal{T}$ with state s: $\mathcal{T}, s \models \varphi$ iff $\mathcal{T}, s \models \psi$.

In order to simplify notation we will extend the syntax of FLC by simultaneous fixpoint definitions: if $X_1, \ldots, X_n$ are variables and $\varphi_1, \ldots, \varphi_n$ are FLC formulas then $\sigma_i(X_1, \ldots, X_n), (\varphi_1, \ldots, \varphi_n)$ is also an FLC formula. Its semantics is the simultaneously defined fixpoint of the corresponding system of equations projected onto the solution for X_i. According to the Békič principle, this formula simply abbreviates an iterative nesting, e.g.

$$\sigma_1(X_1, \ldots, X_n).(\varphi_1, \ldots, \varphi_n) \equiv \sigma X_1.\varphi_1(X_1, \sigma X_2.\varphi_2(X_1, X_2, \ldots) \ldots)$$

A formula is called *disjunctive* if it only consists of variables, the atoms τ and $\langle a\rangle$, disjunctions, sequential compositions, and the μ-operator. It is called *conjunctive* if it only consists of variables, the atoms τ and $\langle a\rangle$, conjunctions, sequential compositions, and the ν-operator.

Homogenous FLC, HFLC, is the least fragment that contains atomic propositions, and is closed under the boolean operators and left-composition with closed disjunctive or conjunctive formulas. For instance, the following is an HFLC-formula: $(\mu X.\tau \vee \langle a\rangle; X; \langle b\rangle); (\nu Y.\tau \wedge [c]; Y); q$. It says that there is a path leading – under a word of the form $a^n b^n$ – to some state from which no other state is reachable via an arbitrary number of c-relations that does not satisfy q.

We define the complement $\overline{\varphi}$ of an FLC formula φ inductively as $\overline{q} := \neg q$, $\overline{\neg q} := q$, $\overline{\tau} := \tau$, $\overline{\langle a\rangle} := [a]$, $\overline{[a]} := \langle a\rangle$, $\overline{\varphi \vee \psi} := \overline{\varphi} \wedge \overline{\psi}$, $\overline{\varphi \wedge \psi} := \overline{\varphi} \vee \overline{\psi}$, $\overline{\varphi; \psi} := \overline{\varphi}; \overline{\psi}$, $\overline{X} := X$, $\overline{\mu X.\varphi} := \nu X.\overline{\varphi}$, and $\overline{\nu X.\varphi} := \mu X.\overline{\varphi}$.

The following lemma is then easy to prove by a straight-forward induction on the formula structure.

Lemma 2 *Let $\mathcal{T}$ be a Kripke structure with state set $\mathcal{S}$. For all $T \subseteq \mathcal{S}$, all $\varphi \in FLC$ and all environments ρ we have: $[\![\overline{\varphi}]\!]_\rho(T) = \mathcal{S} \setminus [\![\varphi]\!]_\rho(\mathcal{S} \setminus T)$.*

2 Equi-Expressiveness of HFLC and PDL[CFG]

Let $G = (N, \Sigma, S, P)$ be a CFG with $N := \{A_1, \ldots, A_n\}$, $S = A_1$ and $A_i \rightarrow \alpha_{i,1} \mid \ldots \mid \alpha_{i,m_i}$ for each $i = 1, \ldots, n$. We define two FLC formulas $\|G\|^\exists$ and $\|G\|^\forall$ in the following way.

$$\begin{array}{rcl} \|G\|^\exists & := & \mu_1(A_1, \ldots, A_n).(\bigvee_{i=1}^{m_1} \|\alpha_{1,i}\|^\exists, \ldots, \bigvee_{i=1}^{m_n} \|\alpha_{n,i}\|^\exists) \\ \|G\|^\forall & := & \nu_1(A_1, \ldots, A_n).(\bigwedge_{i=1}^{m_1} \|\alpha_{1,i}\|^\forall, \ldots, \bigwedge_{i=1}^{m_n} \|\alpha_{n,i}\|^\forall) \end{array}$$

where $\|\epsilon\|^\exists = \|\epsilon\|^\forall := \tau$, $\|A_i\|^\exists = \|A_i\|^\forall := A_i$, $\|aw\|^\exists := \langle a\rangle; \|w\|^\exists$, and $\|aw\|^\forall := [a]; \|w\|^\forall$.

Lemma 3 *Let $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \Sigma\}, \lambda)$ be a Kripke structure, and G be a CFG. For all $s \in \mathcal{S}$ and all $T \subseteq \mathcal{S}$ we have*

a) $s \in [\![\|G\|^\exists]\!](T)$ iff $\exists t \in T$ with $s \xrightarrow{G} t$;
b) $s \in [\![\|G\|^\forall]\!](T)$ iff $\forall t \in \mathcal{S}$: $s \xrightarrow{G} t$ implies $t \in T$.

PROOF. (a) Let $G = (N, \Sigma, P, S)$. Note that P defines a system of equations for each $A \in N$, and it is well-known that $L(G)$ is the simultaneously defined least fixpoint of this system, projected onto S. This is what $\|G\|^\exists$ defines explicitly, from which the claim follows immediately.

(b) This follows immediately from part (a) and Lemma 2. Note that $\|G\|^\forall \equiv \|G\|^\exists$. □

Theorem 4 *For any $\varphi \in$ PDL[CFG] there is a $\varphi' \in$ HFLC s.t. $\varphi \equiv \varphi'$.*

PROOF. Let $\varphi \in$ PDL[CFG]. Define a $\varphi' \in$ FLC inductively as follows.

$$\begin{array}{llll}
q' & := q & (\neg q)' & := \neg q \\
(\psi_0 \vee \psi_1)' & := \psi_0' \vee \psi_1' & (\psi_0 \wedge \psi_1)' & := \psi_0' \wedge \psi_1' \\
(\langle G \rangle \psi)' & := \|G\|^\exists ; \psi' & ([G]\varphi)' & := \|G\|^\forall ; \psi'
\end{array}$$

A straight-forward induction on the structure of φ shows $\varphi \equiv \varphi'$. The cases of modal formulas follow from Lemma 3. Note that ψ' is a predicate for any $\psi \in$ PDL[CFG] according to Lemma 1. Furthermore, for any CFG G, $\|G\|^\exists$ is disjunctive and $\|G\|^\forall$ is conjunctive. Hence, $\varphi' \in$ HFLC for any $\varphi \in$ PDL[CFG]. □

This shows PDL[CFG] $\leq$ HFLC. As an example, take the CFG G given by $S \to b \mid aTS$, $T \to b \mid aTT$. Note that $L(G) = \{w \in \{a,b\}^* \mid$ there is a prefix v of w s.t. $|v|_b > |v|_a\}$. Consequently, the PDL[CFG] formula $[G]\mathtt{ff}$ as well as the HFLC formula $(\nu S.[b] \wedge [a];(\nu T.[b] \wedge [a];T;T);S);\mathtt{ff}$ both express: "on all paths the number of b-actions never exceeds the number of a-actions".

For the converse translation from HFLC to PDL[CFG] we first observe that, with the help of simultaneous fixpoint formulas, it is possible to normalise every HFLC formula.

A disjunctive FLC formula is said to be normalised if it is of the form $\mu_i(X_1, \ldots, X_n).(\varphi_1, \ldots, \varphi_n)$ s.t. for all $i = 1, \ldots, n$: φ_i is a disjunction of sequential compositions of the variables X_i and atomic formulas τ and $\langle a \rangle$. A conjunctive formula is said to be normalised if it is of the form $\nu_i(X_1, \ldots, X_n).(\varphi_1, \ldots, \varphi_n)$ s.t. for all $i = 1, \ldots, n$: φ_i is a conjunction of sequential compositions of the variables X_i and atomic formulas τ and $[a]$. For instance, the formulas produced by the translations $\| \cdot \|^\exists$ and $\| \cdot \|^\forall$ are normalised.

Lemma 5 *Every disjunctive and every conjunctive formula $\varphi \in$ FLC is equivalent to a normalised φ'.*

PROOF. Let φ be disjunctive. W.l.o.g. we can assume φ to start with a μ-operator because of the equivalence $\varphi \equiv \mu X.\varphi$ where X does not occur in φ. Since φ does not contain greatest fixpoint quantifiers we can apply the Békič principle and rewrite it into the form $\mu_i(X_1, \ldots, X_n).(\varphi_1, \ldots, \varphi_n)$. However, each φ_i may still contain disjunctions under the scope of sequential compositions.

Suppose there is a subformula of the form $(\psi_1 \vee \psi_2); \psi_3$. This can equivalently be replaced by $\psi_1; \psi_3 \vee \psi_2; \psi_3$. However, for subformulas of the form $\psi_3; (\psi_1 \vee \psi_2)$ such a general equivalence does not hold. However, since ψ_3 is disjunctive, the following transformation can be applied. Replace this subformula by $\psi_3; X_{n+1}$ where X_{n+1} is a new variable, and rewrite φ to

$$\mu_i(X_1, \ldots, X_n, X_{n+1}).(\varphi_1[X_{n+1}/\psi_1 \vee \psi_2], \ldots, \varphi_n[X_{n+1}/\psi_1 \vee \psi_2], \psi_1 \vee \psi_2)$$

Iterating this process will eventually create a formula of the desired form.

The normalisation process for conjunctive formulas is entirely analogous. □

Theorem 6 *For every $\varphi \in$ HFLC there is a $\varphi' \in$ PDL[CFG] s.t. $\varphi' \equiv \varphi$.*

PROOF. By induction on the structure of φ. Atomic propositions as well as boolean connectives trivially translate into PDL[CFG]. The only interesting cases are those of $\varphi = \psi'; \chi$. By hypothesis, there is a $\chi' \in$ PDL[CFG] with $\chi' \equiv \chi$. Furthermore, ψ' is closed and either disjunctive or conjunctive. According to Lemma 5, ψ' is equivalent to a normalised ψ in FLC.

Now note that being normalised coincides with being of the form $\|G\|^{\exists}$, resp. $\|G\|^{\forall}$ for some CFG G. It is easy to reverse the translations of normalised FLC formulas into CFGs. Correctness of these reversed translations is proved in the same way as it is done in Thm. 4. Hence, there is a CFG G_ψ s.t. $\varphi \equiv \langle G_\psi \rangle; \chi$ if ψ' is disjunctive, and $\varphi \equiv [G_\psi]; \chi$ if ψ' is conjunctive. □

This shows HFLC $\leq$ PDL[CFG]. For example, take the HFLC formula $(\mu X.\tau \vee \langle a \rangle; X; \langle b \rangle); (\nu Y.\tau \wedge [c]; Y); q$ mentioned above. It is already normalised. Hence, it can directly be translated into PDL[CFG] and the result is $\langle G_1 \rangle [G_2] q$ where G_1 is given by $X \to \epsilon \mid aXb$, and G_2 is given by $Y \to \epsilon \mid cY$.

3 Conclusions

Thms. 4 and 6 not only show that PDL[CFG] and HFLC are equi-expressive, but also equi-succinct. Note that both translations are linear in the size of the input formula, i.e. the number of its subformulas. This transfers several results from one logic to another. For example, satisfiability in HFLC is already Σ_1^1-complete in the analytical hierarchy since so is satisfiability for PDL[CFG] [2]. So far, it was only known that FLC is undecidable [9], but the degree of undecidability had not been measured. On the other hand, non-trivial fragments of PDL[CFG] have been found to be decidable [5]. Hence, there also are decidable and non-trivial fragments of FLC.

Related to the question of undecidability is the finite model property. It is known that the full logic FLC does not have the finite model property. However, many fragments of PDL[CFG] with one non-regular program only do not admit finite models [3]. Thus, already fragments of HFLC do not possess this property.

It is known that the model checking problem for FLC is EXPTIME-hard for certain fixed, alternation-free formulas already [8]. On the other hand, PDL[CFG]'s model checking problem is in P [7]. Hence, the equivalence result above reveals a non-trivial fragment of (alternation-free) FLC for which model checking is tractable.

References

[1] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[2] D. Harel, A. Pnueli, and J. Stavi. Propositional dynamic logic of nonregular programs. *Journal of Computer and System Sciences*, 26(2):222–243, 1983.

[3] D. Harel and E. Singerman. More on nonregular PDL: Finite models and Fibonacci-like programs. *Information and Computation*, 128(2):109–118, 1996.

[4] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, N. Reading, MA, 1980.

[5] T. Koren and A. Pnueli. There exist decidable context free propositional dynamic logics. In E. Clarke and D. Kozen, editors, *Proc. of the Workshop on Logics of Programs*, volume 164 of *LNCS*, pages 290–312, Pittsburgh, PA, 1983. Springer.

[6] D. Kozen. Results on the propositional μ-calculus. *TCS*, 27:333–354, December 1983.

[7] M. Lange. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4(1):39–49, 2005.

[8] M. Lange. Three notes on the complexity of model checking fixpoint logic with chop, 2005. (submitted).

[9] M. Müller-Olm. A modal fixpoint logic with chop. In C. Meinel and S. Tison, editors, *Proc. 16th Symp. on Theoretical Aspects of Computer Science, STACS'99*, volume 1563 of *LNCS*, pages 510–520, Trier, Germany, 1999. Springer.

[10] V. R. Pratt. Semantical considerations on Floyd-Hoare logic. In *17th Annual Symp. on Foundations of Computer Science, FOCS'76*, pages 109–121, Houston, Texas, 1976. IEEE.

[11] A. Tarski. A lattice-theoretical fixpoint theorem and its application. *Pacific Journal of Mathematics*, 5:285–309, 1955.