HAL
open science

Logical Analysis of Demonic NonDeterministic Programs

Stéphane Demri, Ewa Orlowska

To cite this version:

Stéphane Demri, Ewa Orlowska. Logical Analysis of Demonic NonDeterministic Programs. Theoretical Computer Science, 1996, 166 (1-2), pp.173-202. 10.1016/0304-3975(95)00190-5 . hal-03193705

HAL Id: hal-03193705

https://hal.science/hal-03193705v1

Submitted on 9 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Logical Analysis of Demonic NonDeterministic Programs

Stéphane Demri[1] Ewa Orłowska[2*]

[1] Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle
Grenoble, France
e-mail: demri@lifia.imag.fr

[2] Institute of Theoretical and Applied Computer Science
Polish Academy of Sciences, Warsaw, Poland
e-mail: orlowska@plearn.bitnet

December 17, 2004

Abstract

A logical framework is presented for representing and reasoning about nondeterministic programs that may not terminate. We propose a logic PDL(; ;, ||, $^{d(*)}$) which is an extension of dynamic logic such that the program constructors related to demonic operations are introduced in its language. A complete and sound Hilbert-style proof system is given and it is shown that PDL(; ;, ||, $^{d(*)}$) is decidable. In the second part of this paper, a translation is defined between PDL(; ;, ||, $^{d(*)}$) and a relational logic. A sound and complete Rasiowa-Sikorski-style proof system for the relational logic is given. It provides a natural deduction-style method of reasoning for PDL(; ;, ||, $^{d(*)}$).

1 Introduction

The logic-based methodology of the theory of programs originated in [Sal70, Mir77a, Mir77b, Pra76] is well established in the literature. In particular, a variety of modal-style logical systems have been developed and a number of papers have been devoted to the study of the underlying relational semantics of programs. In modal logics of programs a program α is represented by means of a binary relation R_α over a state space, with the intuition that $(s,t) \in R_\alpha$ iff the program α executed from initial state s terminates at state t. The programs represented by relations are nondeterministic, that is for a given input state, the output state obtained by executing a program is not necessarily unique. The central problem in defining a formal semantics of programs is the treatment of nontermination. The approach developed in [HHJ$^+$87] is based on the proposal of introducing a fictitious state, say $s_\perp$, such that if program α does not terminate when starting from state s, then $(s, s_\perp) \in R_\alpha$. An alternative approach is based on the postulate that if program α does not terminate when executed from state s, then the set $\{t \mid (s,t) \in R_\alpha\}$ of R_α-successors of s is empty. In this paper we follow the latter approach.

The main motivation for introducing demonic program constructors can be expressed as a principle that possible nontermination implies definite nontermination (see [Ngu91]).

*The mailing address: Azaliowa 29, 04-539 Warsaw, Poland

To incorporate this principle in the relational semantics of programs, the classical operators of nondeterministic choice and sequential composition should be modified appropriately:

- If two commands of a nondeterministic program α can be executed nondeterministically, and if an execution of one of them does not terminate, then the execution of α does not terminate,
- If a sequence of commands of a nondeterministic program α is executed sequentially, and if an execution of one of them does not terminate, then the execution of α does not terminate.

The above postulates reflect the well-known Murphy's law: 'If it can go wrong, it will'. Nondeterminism modelled according to these postulates is referred to as demonic nondeterminism, as opposed to angelic and erratic nondeterminism [BZ86].

The calculus of binary relations with operators of demonic union ($||$) and demonic composition ($;;$) of relations have been studied in [Ngu91]. Our proposal is to extend the calculus with a demonic iteration operator ($d(*)$) defined in a natural way by means of the respective binary demonic operators. Demonic iteration is motivated by the following postulate:

- If a command of a nondeterministic program α is executed nondeterministically any finite number of times and if one of these executions does not terminate, then the execution of α does not terminate.

In this paper we develop a logical framework for the analysis of demonic nondeterminism. In the first part of the paper a propositional program logic PDL($;$, $;;$, $||$, $^{d(*)}$) is introduced such that the language of the logic enables us to specify demonic nondeterministic programs. We admit in the language the standard dynamic logic program constructors as well as demonic union, demonic composition, and demonic iteration of programs. We present a complete Hilbert-style axiomatization of the logic and we prove that the logic possesses the finite model property. We discuss demonic iteration and we point out various ways of defining the respective relational operators. In the second part of the paper a relational semantics is developed as well as a relational proof system for the logic. Under the relational semantics both expressions that represent programs and formulae of PDL($;$, $;;$, $||$, $^{d(*)}$) are interpreted as binary relations. It enables us to interpret our logic in a relational logic and to define a relational proof system for it, following the method suggested in [Orło88].

2 Syntax and Semantics of Program Logics

For a given binary relation $R \subseteq U \times U$ and for $x \in U$ we denote by $R(x)$ the set $\{y \mid (x, y) \in R\}$ of R-successors of x .

2.1 Syntax

A (propositional) *program modal language* is determined by four sets which are supposed to be pairwise disjoint, viz,

(i) a set Φ_0 of propositional variables,

(ii) a set Π_0 of program constants,

(iii) a set of propositional operators,

(iv) a set of program operators.

The set Π of *program expressions* is the smallest set that satisfies the following conditions:

(i) $\Pi_0 \subseteq \Pi$

(ii) if ϕ is an n-ary program operator and $a_0, \ldots, a_{n-1} \in \Pi$ then $\phi(a_0, \ldots, a_{n-1}) \in \Pi$,

The set Σ of formulae is the smallest set that satisfies the following conditions:

(i) $\Phi_0 \cup \{false, true\} \subseteq \Sigma$

(ii) if o is any n-ary propositional operator and $F_0, \ldots, F_{n-1} \in \Sigma$ then $o(F_0, \ldots, F_{n-1}) \in \Sigma$.

(iii) If $\alpha \in \Pi$ and $F \in \Sigma$ then $[\alpha]F \in \Sigma$ and $\langle\alpha\rangle F \in \Sigma$.

We assume throughout the paper that a fixed program modal language is given such that it satisfies the following conditions:

(i) the set ϕ_0 of propositional variables is an infinite denumerable set

(ii) the set of program constants is finite or infinite denumerable

(iii) the propositional operators are the unary $\neg$, the binary $\Leftrightarrow$, $\Rightarrow$, $\vee$, $\wedge$.

(iv) the program operators are the binary $\cup, ||, ;, ;;$ and the unary $^*, ^{d(*)}$.

2.2 Semantics

For the sake of simplicity, the same symbol is used for a relational operation and the respective program operator. We use the symbol ; for composition, that is if R and S are binary relations then

$$R; S = \{(x, y) : \exists z(x, z) \in R \text{ and } (z, y) \in S\}$$

We recall that if R denotes a binary relation on the set U, the iteration operator is defined as follows

$$R^* = \{R^n \mid n \in \omega\}$$

with $R^0 = \{(x, x) \mid x \in U\}$ and $R^{n+1} = R; R^n$ where ω denotes the set of natural numbers. We use the symbol ;; for demonic composition (e.g. [Ngu91]), that is if R and S are binary relations then

$$R;; S = \{(x, y) : \forall z, \text{ if } (x, z) \in R \text{ then } \exists t(z, t) \in S \text{ and } (x, y) \in R; S\}$$

The set $(R;; S)(x)$ is empty if either $(R; S)(x)$ is empty or there exists z_0 such that $(x, z_0) \in R$ and $S(z_0) = \emptyset$. We use the symbol $||$ for demonic union (e.g. [Ngu91]), that is if R and S are binary relations then

$$R||S = \{(x, y) : \exists t(x, t) \in R \text{ and } \exists t'(x, t') \in S \text{ and } (x, y) \in R \cup S\}$$

If the relations are serial then the demonic union (respectively composition) collapses to the classical union (respectively composition). Let R and S be two relations on the set U. As mentioned in [Ngu93], it is a routine matter to check

- $R;; S = (R; S) \cap -(R; -(S; (U \times U)))$

- $R||S = (R \cup S) \cap (R; (U \times U)) \cap (S; (U \times U))$

Observe that although the intersection and complement operations are used in the above characterization of demonic operations, no program operator is associated to these relational operations.

A new demonic iteration operation $^{d(*)}$ is defined as follows:

$$R^{d(*)} = ||_{i \in \omega} R^{d(i)}$$

with $R^{d(i+1)} = R;; R^{d(i)}$ for $i \geq 1$, and $R^{d(0)} = \{(x, x) \mid x \in U\}$. Since the demonic composition is associative [Ngu91], for $i \geq 0$, $R^{d(i+1)} = R^{d(i)};; R$. As for the demonic composition, the demonic iteration operation is locally an 'all or nothing iteration'.

Lemma 2.1. Let R be a binary relation on the set U and $x \in U$.

(i) For all $i \in \omega$, if $R^{d(i)}(x) \neq \emptyset$ then $R^{d(i)}(x) = R^i(x)$

(ii) For all $i \in \omega$, if $R^{d(i)}(x) = \emptyset$ then $R^{d(i+1)}(x) = \emptyset$

The straightforward inductive proof is omitted. In order to show that $R^{d(*)}(x) \neq \emptyset$ only if x cannot reach through R an element y having no R-successor, we introduce the standard notion of *maximal R-chain*.

Definition 2.2. (Maximal R-chain) Let R be a binary relation on U and $x \in U$. A maximal R-chain from x is either a finite sequence $(x_0, \ldots, x_N)$ such that $x_0 = x$, $(x_i, x_{i+1}) \in R$ for all $i \in \{0, \ldots, (N-1)\}$ and $R(x_N) = \emptyset$ (N possibly equal to 0) or an infinite sequence $(x_0, x_1, \ldots)$ such that $x_0 = x$, $(x_i, x_{i+1}) \in R$ for all $i \in \omega$. We write $x \uparrow R$ iff every maximal R-chain from x is infinite. The *length* of a finite maximal R-chain σ, written $|\sigma|$, is the number of elements of the sequence. ∇

Lemma 2.3. Let R be a binary relation on the set U and $x \in U$.

(i) If $x \uparrow R$ then $R^{d(*)}(x) = R^*(x)$, otherwise $R^{d(*)}(x) = \emptyset$.

(ii) $(x, y) \in R^{d(*)}$ iff $(x, y) \in R^*$ and for all $i \in \omega$, $R^{d(i)}(x) \neq \emptyset$.

Proof: (i) Assume not $x \uparrow R$. Let $\mathcal{S}$ be the non-empty set of all the finite maximal R-chains from x. We write M to denote the minimal element of the set $\{|\sigma| \mid \sigma \in \mathcal{S}\}$ and $(X_0, \ldots, X_{M-1})$ an element of $\mathcal{S}$ ($M \geq 1$). It shall be shown that for $i \in \{0, \ldots, (M-1)\}$, $R^{d(i)}(x) \neq \emptyset$. The proof is by induction. By definition $R^{d(0)}(x) \neq \emptyset$. Now suppose $R^{d(j)}(x) = \emptyset$ for some $j \in \{1, \ldots, (M-1)\}$ and for all $k \in \{0, \ldots, j-1\}$, $R^{d(k)}(x) \neq \emptyset$. Hence $R^{d(j)}(x) = (R^{d(j-1)};; R)(x) = (R^{j-1};; R)(x)$ (see Lemma 2.1(i)).Since $X_j \in R^j(x)$, $R^j(x) \neq \emptyset$. Moreover suppose there is z_0 such that $(x, z_0) \in R^{j-1}$ and $R(z_0) = \emptyset$. It leads to a contradiction, since M is minimal. So for all $i \in \{0, \ldots, (M-1)\}$, $R^{d(i)}(x) \neq \emptyset$ and therefore from Lemma 2.1 $R^{d(i)}(x) = R^i(x)$ for all $i \in \{0, \ldots, (M-1)\}$. We have $R^{d(M)}(x) = R^{M-1};; R(x)$. Since $(x, X^{M-1}) \in R^{M-1}$ and $R(X_{M-1}) = \emptyset$, then $R^{d(M)}(x) = \emptyset$. It follows that $R^{d(*)}(x) = \emptyset$.
Now assume $x \uparrow R$ and suppose that there exists $i \in \omega$ such that $R^{d(i)}(x) = \emptyset$. There exists $j \geq 1$ such that $R^{d(j)}(x) = \emptyset$ and $R^{d(j-1)}(x) \neq \emptyset$ ($R^{d(j)}(x) = (R^{j-1};; R)(x)$). Suppose $R^j(x) = \emptyset$. This leads to the existence of a finite maximal R-chain from x which is a contradiction. Now suppose there exists z_0 such that $(x, z_0) \in R^{j-1}$ and $R(z_0) = \emptyset$. This also leads to the existence of a finite maximal R-chain from x. As a consequence for all $i \in \omega$, $R^{d(i)}(x) \neq \emptyset$. So $R^{d(*)}(x) = ||_{i \in \omega} R^i(x)$ from Lemma 2.1(i). Since each $R^i(x)$ is non-empty, $R^{d(*)}(x) = \bigcup_{i \in \omega} R^i(x) = R^*(x)$.

The condition (ii) is a consequence of the proof of (i). Q.E.D.

It follows that $R^{d(*)}(x)$ is empty if either not $x \uparrow R$ or $R^*(x)$ is empty. Kripke-style semantics for dynamic logic with the demonic operators is defined in the standard way (see e.g. [Seg82]).

Definition 2.4. (Frame, Program Frame, Model, Program Model) By a frame we understand a pair $(U, \mathcal{R})$ such that U is a non-empty set and $\mathcal{R} = \{R_\alpha\}_{\alpha \in \Pi}$ is a family of binary relations on U. A program frame is a frame such that the following conditions are satisfied:

(i) $R_{\alpha \cup \beta} = R_\alpha \cup R_\beta$ (ii) $R_{\alpha || \beta} = R_\alpha || R_\beta$ (iii) $R_{\alpha ; \beta} = R_\alpha ; R_\beta$

(iv) $R_{\alpha ;; \beta} = R_\alpha ;; R_\beta$ (v) $(R_\alpha)^* = R_{\alpha^*}$ (vi) $(R_\alpha)^{d(*)} = R_{\alpha^{d(*)}}$

By a model $\mathcal{M}$, we understand a triple $(U, \mathcal{R}, m)$ such that $(U, \mathcal{R})$ is a frame and m is a function from ϕ_0 to $\mathcal{P}(U)$, the power set of U. We say that the model $\mathcal{M}$ is based on the frame $(U, \mathcal{R})$. A program model is a model that is based on a program frame. ∇

Observe that for any program frame $\mathcal{F}$, the set $\{R_\alpha\}_{\alpha \in \Pi_0}$ determines in a unique way the set $\{R_\alpha\}_{\alpha \in \Pi}$. Let $\mathcal{M} = (U, \mathcal{R}, m)$ be any given model (not necessarily a program model). The concept of *satisfiability at a point in* $\mathcal{M}$ is recursively defined as follows. Let $u \in U$.

(i) $\mathcal{M}, u$ sat P iff $u \in m(P)$, for $P \in \phi_0$; (ii) $\mathcal{M}, u$ sat $\neg F$ iff not $\mathcal{M}, u$ sat F

(iii) $\mathcal{M}, u$ sat $F \Rightarrow G$ iff $\mathcal{M}, u$ sat F only if $\mathcal{M}, u$ sat G,

(iv) $\mathcal{M}, u$ sat $[\alpha]F$ iff, for all v, if $uR_\alpha v$ then $\mathcal{M}, v$ sat F.

(v) $\mathcal{M}, u$ sat *true* and not $\mathcal{M}, u$ sat $false$

We omit the standard definitions of satisfiability for the other logical operators. A formula F is *true* in a model $\mathcal{M}$ (written $\mathcal{M} \models F$) iff for all $x \in U$, $\mathcal{M}, x$ sat F. A formula F is *true* in a frame $\mathcal{F}$ (written $\mathcal{F} \models F$) iff F is true in every model based on $\mathcal{F}$.

3 Hilbert-style Proof System for PDL($;$;, $||$, $^{d(*)}$)

By a *normal logic* we understand any set L of formulae that satisfies the following conditions:

(i) L contains every tautology of the classical two-valued propositional calculus

(ii) L is closed under *modus ponens* ; (iii) L is closed under substitution

(iv) L contains every formula of the form

(D1) $[\alpha](F \Rightarrow G) \Rightarrow ([\alpha]F \Rightarrow [\alpha]G)$ for $\alpha \in \Pi$, $F, G \in \Sigma$,

(v) for every $\alpha \in \Pi$ and $F \in L$, $[\alpha]F \in L$.

Finally by a *program logic with demonic operators* we understand a normal logic that contains all formulae of the following form:

(D2) $[\alpha \cup \beta]F \Leftrightarrow [\alpha]F \wedge [\beta]F$

(D3) $[\alpha ; \beta]F \Leftrightarrow [\alpha][\beta]F$

(D4) $[\alpha]F \wedge [\beta]F \Rightarrow [\alpha||\beta]F$

(D5) $[\alpha||\beta]F \wedge \langle\alpha||\beta\rangle true \Rightarrow \langle\alpha\rangle true \wedge \langle\beta\rangle true \wedge [\alpha]F \wedge [\beta]F$

(D6) $[\alpha||\beta]false \Rightarrow [\alpha]false \vee [\beta]false$

(D7) $[\alpha][\beta]F \Rightarrow [\alpha;;\beta]F$

(D8) $[\alpha;;\beta]F \wedge \langle\alpha;;\beta\rangle true \Rightarrow [\alpha](\langle\beta\rangle true \wedge [\beta]F)$

(D9) $[\alpha;;\beta]false \Rightarrow [\alpha][\beta]false \vee \langle\alpha\rangle[\beta]false$

(D10) $[\alpha^*]F \Rightarrow F \wedge [\alpha][\alpha^*]F$

(D11) $F \Rightarrow ([\alpha^*](F \Rightarrow [\alpha]F) \Rightarrow [\alpha^*]F)$

(D12) $[\alpha^*]F \Rightarrow [\alpha^{d(*)}]F$

(D13) $[\alpha^{d(*)}]false \Rightarrow \langle\alpha^*\rangle[\alpha]false$

(D14) $[\alpha^{d(*)}]F \wedge \langle\alpha^{d(*)}\rangle true \Rightarrow [\alpha^*]\langle\alpha\rangle true \wedge [\alpha^*]F$

Axioms (D2) and (D3) are standard for the union and the composition operators (see e.g. [Seg82]). Axioms (D10) and (D11) are the Segerberg axioms for the iteration operation [Seg82]. Axiomatisation of the operator $||$ (respectively $;;$) is provided by the axioms (D4),(D5) and (D6) (respectively (D7),(D8) and (D9)). Observe that if $\langle\alpha\rangle true$ is added to the system (seriality axiom) then $[\alpha||\beta]F \Leftrightarrow [\alpha \cup \beta]F$, $[\alpha;;\beta]F \Leftrightarrow [\alpha;\beta]F$ and $[\alpha^{d(*)}]F \Leftrightarrow [\alpha^*]F$ can be deduced in the system. In a standard way, we define the notions of theoremhood in L, and deducibility. Let PDL($;;$, $||$, $^{d(*)}$) be the smallest program logic with demonic operators.

The lemmas 3.1, 3.2 and 3.3 express correspondences between modal formulae and properties of relations in the frames. A survey of correspondence theory can be found in [vB84].

Lemma 3.1. The axioms (D4),(D5),(D6) are true in a frame $\mathcal{F} = (U, \{R_\gamma\}_{\gamma\in\Pi})$ iff for all $\alpha, \beta \in \Pi$, $R_{\alpha||\beta} = R_\alpha||R_\beta$.

Proof: Let $\mathcal{F} = (U, \{R_\gamma\}_{\gamma\in\Pi})$ be a frame.

(I) Suppose there exists $x, y \in U$ such that $(x, y) \in R_{\alpha||\beta}$ and $(x, y) \notin R_\alpha||R_\beta$.

Suppose $(x, y) \notin R_\alpha \cup R_\beta$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in R_\alpha \cup R_\beta\}$. Since (D4) is true in $\mathcal{F}$, $\mathcal{M}_0, x$ sat $[\alpha]p \wedge [\beta]p \Rightarrow [\alpha||\beta]p$. By construction of m_0, $\mathcal{M}_0, x$ sat $[\alpha]p \wedge [\beta]p$ and therefore $\mathcal{M}_0, x$ sat $[\alpha||\beta]p$. Since $(x, y) \in R_{\alpha||\beta}$, we have $\mathcal{M}_0, y$ sat p. By construction of m_0, $\mathcal{M}_0, y$ sat $\neg p$ which leads to a contradiction.

Now suppose $R_\alpha(x) = \emptyset$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in R_{\alpha||\beta}\}$. Since (D5) is true in $\mathcal{F}$, $\mathcal{M}_0, x$ sat $[\alpha||\beta]p \wedge \langle\alpha||\beta\rangle true \Rightarrow \langle\alpha\rangle true \wedge \langle\beta\rangle true \wedge [\alpha]p \wedge [\beta]p$. By construction of m_0, we have $\mathcal{M}_0, x$ sat $[\alpha||\beta]p$. Since $(x, y) \in R_{\alpha||\beta}$, we have $\mathcal{M}_0, x$ sat $\langle\alpha||\beta\rangle true$. Hence $\mathcal{M}_0, x$ sat $\langle\alpha\rangle true \wedge \langle\beta\rangle true \wedge [\alpha]p \wedge [\beta]p$. There exists x_0 such that $(x, x_0) \in R_\alpha$ which leads to a contradiction. If $R_\beta(x) = \emptyset$, then a contradiction can be found in a similar way. This proves that if the axioms (D4),(D5),(D6) are true in $\mathcal{F}$ then for all $\alpha, \beta \in \Pi$, $R_{\alpha||\beta} \subseteq R_\alpha||R_\beta$.

(II) Suppose there exists $x, y \in U$ such that $(x, y) \notin R_{\alpha||\beta}$ and $(x, y) \in R_\alpha||R_\beta$.

First suppose $R_{\alpha||\beta}(x) = \emptyset$. Consider a model $\mathcal{M} = (U, \{R_\gamma\}_{\gamma\in\Pi}, m)$ based on $\mathcal{F}$. Since (D6) is true in $\mathcal{F}$, $\mathcal{M}, x$ sat $[\alpha||\beta]false \Rightarrow [\alpha]false \vee [\beta]false$. Since $R_{\alpha||\beta}(x) = \emptyset$, $\mathcal{M}, x$ sat $[\alpha||\beta]false$ and therefore $\mathcal{M}, x$ sat $[\alpha]false \vee [\beta]false$. Since $(x, y) \in R_\alpha||R_\beta$, there exist x_1, x_2 such that $(x, x_1) \in R_\alpha$ and $(x, x_2) \in R_\beta$. It follows that neither $\mathcal{M}, x$ sat $[\alpha]false$ nor $\mathcal{M}, x$ sat $[\beta]false$ can hold, which leads to a contradiction.

Now suppose $R_{\alpha||\beta}(x) \neq \emptyset$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in R_{\alpha||\beta}\}$. Since (D5) is true in $\mathcal{F}$, $\mathcal{M}_0, x$ sat $[\alpha||\beta]p \wedge \langle\alpha||\beta\rangle true \Rightarrow \langle\alpha\rangle true \wedge \langle\beta\rangle true \wedge [\alpha]p \wedge [\beta]p$. By construction of m_0, $\mathcal{M}_0, x$ sat $[\alpha||\beta]p$. Since $R_{\alpha||\beta}(x) \neq \emptyset$, $\mathcal{M}_0, x$ sat $\langle\alpha||\beta\rangle true$ and therefore $\mathcal{M}_0, x$ sat $\langle\alpha\rangle true \wedge \langle\beta\rangle true \wedge [\alpha]p \wedge [\beta]p$. Since $(x, y) \in R_\alpha \cup R_\beta$, we have $\mathcal{M}_0, y$ sat p. $\mathcal{M}_0, y$ sat $\neg p$ ($(x, y) \notin R_{\alpha||\beta}$), which leads to a contradiction. This proves that if the axioms (D4),(D5),(D6) are true in $\mathcal{F}$ then for all $\alpha, \beta \in \Pi$, $R_\alpha||R_\beta \subseteq R_{\alpha||\beta}$.

(III) Suppose that in the frame $\mathcal{F} = (U, \{R_\gamma\}_{\gamma\in\Pi})$, for all $\alpha, \beta \in \Pi$, $R_{\alpha||\beta} = R_\alpha||R_\beta$. Let $\mathcal{M} = (U, \{R_\gamma\}_{\gamma\in\Pi}, m)$ be a model based on $\mathcal{F}$ and $x \in U$.
(D4) Assume that $\mathcal{M}, x$ sat $[\alpha]F \wedge [\beta]F$. It follows that for $y \in U$ if either $(x, y) \in R_\alpha$ or $(x, y) \in R_\beta$ then $\mathcal{M}, y$ sat F. So for all $y \in (R_\alpha \cup R_\beta)(x)$, we have $\mathcal{M}, y$ sat F. *A fortiori*, for all $y \in (R_\alpha||R_\beta)(x)$, we have $\mathcal{M}, y$ sat F, which entails that $\mathcal{M}, x$ sat $[\alpha||\beta]F$.
(D5) Assume that $\mathcal{M}, x$ sat $[\alpha||\beta]F \wedge \langle\alpha||\beta\rangle true$. It follows that there exists $x_0 \in U$ such that $(x, x_0) \in R_\alpha||R_\beta = R_{\alpha||\beta}$. So there exists $x_1, x_2 \in U$ such that $(x, x_1) \in R_\alpha$ and $(x, x_2) \in R_\beta$. It follows that $\mathcal{M}, x$ sat $\langle\alpha\rangle true \wedge \langle\beta\rangle true$. Moreover, since $(R_\alpha||R_\beta)(x) \neq \emptyset$ it follows that $(R_{\alpha||\beta})(x) = (R_\alpha \cup R_\beta)(x)$. Hence we also have $\mathcal{M}, x$ sat $[\alpha]F \wedge [\beta]F$.
(D6) Assume that $\mathcal{M}, x$ sat $[\alpha||\beta]false$. It follows that $(R_\alpha||R_\beta)(x) = \emptyset$. In case $R_\alpha(x) = \emptyset$ or $R_\beta(x) = \emptyset$, we have either $\mathcal{M}, x$ sat $[\alpha]false$ or $\mathcal{M}, x$ sat $[\beta]false$. In case for all $z \in U$, $(x, z) \notin R_\alpha \cup R_\beta$, we have $\mathcal{M}, x$ sat $[\alpha]false$ and $\mathcal{M}, x$ sat $[\beta]false$.

This proves that if for all $\alpha, \beta \in \Pi$, $R_{\alpha||\beta} = R_\alpha||R_\beta$, then the axioms (D4),(D5),(D6) are true in $\mathcal{F}$. Q.E.D.

Lemma 3.2. The axioms (D7),(D8),(D9) are true in a frame $\mathcal{F} = (U, \{R_\gamma\}_{\gamma\in\Pi})$ iff for all $\alpha, \beta \in \Pi$, $R_{\alpha;;\beta} = R_\alpha;;R_\beta$.

Proof: Let $\mathcal{F} = (U, \{R_\gamma\}_{\gamma\in\Pi})$ be a frame.

(I) Suppose there exists $x, y \in U$ such that $(x, y) \in R_{\alpha;;\beta}$ and $(x, y) \notin R_\alpha;;R_\beta$.

Suppose $(x, y) \notin R_\alpha;R_\beta$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in R_\alpha;R_\beta\}$. Since (D7) is true in $\mathcal{F}$, $\mathcal{M}_0, x$ sat $[\alpha][\beta]p \Rightarrow [\alpha;;\beta]p$. By construction of m_0, $\mathcal{M}_0, x$ sat $[\alpha][\beta]p$. Hence we get $\mathcal{M}_0, x$ sat $[\alpha;;\beta]p$. Since $(x, y) \in R_{\alpha;;\beta}$, we have $\mathcal{M}_0, y$ sat p. By construction of m_0, $\mathcal{M}_0, y$ sat $\neg p$ which leads to a contradiction.

Now suppose there exists z_0 such that $(x, z_0) \in R_\alpha$ and $R_\beta(z_0) = \emptyset$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in R_{\alpha;;\beta}\}$. Since (D8) is true in $\mathcal{F}$, $\mathcal{M}_0, x$ sat $[\alpha;;\beta]p \wedge \langle\alpha;;\beta\rangle true \Rightarrow [\alpha](\langle\beta\rangle true \wedge [\beta]p)$. By construction of m_0, we get $\mathcal{M}_0, x$ sat $[\alpha;;\beta]p$. Since $(x, y) \in R_{\alpha;;\beta}$, we have $\mathcal{M}_0, x$ sat $\langle\alpha;;\beta\rangle true$. and therefore $\mathcal{M}_0, x$ sat $[\alpha](\langle\beta\rangle true \wedge [\beta]p)$. Since $(x, z_0) \in R_\alpha$, $\mathcal{M}_0, z_0$ sat $\langle\beta\rangle true \wedge [\beta]p$. We conclude that there exists t_0 such that $(z_0, t_0) \in R_\beta$ which leads to a contradiction. This proves that if the axioms (D7),(D8),(D9) are true in $\mathcal{F}$ then for all $\alpha, \beta \in \Pi$, $R_{\alpha;;\beta} \subseteq R_\alpha;;R_\beta$.

(II) Suppose there exists $x, y \in U$ such that $(x, y) \notin R_{\alpha;;\beta}$ and $(x, y) \in R_\alpha;;R_\beta$.

Suppose $R_{\alpha;;\beta}(x) = \emptyset$. Consider a model $\mathcal{M} = (U, \{R_\gamma\}_{\gamma\in\Pi}, m)$. Since (D9) is true in $\mathcal{F}$, we obtain $\mathcal{M}, x$ sat $[\alpha;;\beta]false \Rightarrow [\alpha][\beta]false \vee \langle\alpha\rangle[\beta]false$. Since $R_{\alpha;;\beta}(x) = \emptyset$, we have $\mathcal{M}, x$ sat $[\alpha;;\beta]false$. Hence we get $\mathcal{M}, x$ sat $[\alpha][\beta]false \vee \langle\alpha\rangle[\beta]false$. In case $\mathcal{M}, x$ sat $[\alpha][\beta]false$, considering that $(x, y) \in R_\alpha;;R_\beta$ it entails that $(x, y) \in R_\alpha;R_\beta$ and

therefore $\mathcal{M}, y$ sat $false$, which leads to a contradiction. In case $\mathcal{M}, x$ sat $\langle\alpha\rangle[\beta]false$, considering that $(x, y) \in R_\alpha;;R_\beta$, it follows that for all $z \in U$, if $(x, z) \in R_\alpha$ then there exists $t \in U$ such that $(z, t) \in R_\beta$. However, there exists $z_0 \in U$ such that $(x, z_0) \in R_\alpha$ and $\mathcal{M}, z_0$ sat $[\beta]false$. Hence there exists $t_0 \in U$ such that $(z_0, t_0) \in R_\beta$, which leads to a contradiction.

Now suppose $R_{\alpha;;\beta}(x) \neq \emptyset$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in R_{\alpha;;\beta}\}$. Since (D8) is true in $\mathcal{F}$, we obtain $\mathcal{M}_0, x$ sat $[\alpha;;\beta]p \wedge \langle\alpha;;\beta\rangle true \Rightarrow [\alpha](\langle\beta\rangle true \wedge [\beta]p)$. By construction of m_0, we have $\mathcal{M}_0, x$ sat $[\alpha;;\beta]p$. Considering that $R_{\alpha;;\beta}(x) \neq \emptyset$, it follows that $\mathcal{M}_0, x$ sat $\langle\alpha;;\beta\rangle true$. Hence, we deduce $\mathcal{M}_0, x$ sat $[\alpha](\langle\beta\rangle true \Rightarrow [\beta]p)$. Since $(x, y) \in R_\alpha;;R_\beta$ then there exists $z_1 \in U$ such that $(x, z_1) \in R_\alpha$ and $(z_1, y) \in R_\beta$. So $\mathcal{M}_0, z_1$ sat $\langle\beta\rangle true \wedge [\beta]p$ and $\mathcal{M}_0, y$ sat p, which is in contradiction with the fact that not $\mathcal{M}_0, y$ sat p since $(x, y) \notin R_{\alpha;;\beta}$. This proves that if the axioms (D7),(D8),(D9) are true in $\mathcal{F}$ then for all $\alpha, \beta \in \Pi$, $R_\alpha;;R_\beta \subseteq R_{\alpha;;\beta}$.

As in the proof of Lemma 3.1 it can be easily proved that if for all $\alpha, \beta \in \Pi$, $R_{\alpha;;\beta} = R_\alpha;;R_\beta$ then the axioms (D7),(D8),(D9) are true in $\mathcal{F}$. Q.E.D.

Lemma 3.3. Given a frame $\mathcal{F} = (U, \{R_\gamma\}_{\gamma\in\Pi})$ satisfying for all $\alpha \in \Pi$, $R_{\alpha^*} = (R_\alpha)^*$, the axioms (D12), (D13), (D14) are true in the frame $\mathcal{F}$ iff for all $\alpha \in \Pi$, $(R_\alpha)^{d(*)} = R_{\alpha^{d(*)}}$.

Proof: Let $\mathcal{F} = (U, \{R_\gamma\}_{\gamma\in\Pi})$ be a frame satisfying for all $\alpha \in \Pi$, $R_{\alpha^*} = (R_\alpha)^*$.

(I) Suppose there exists $x, y \in U$ such that $(x, y) \in R_{\alpha^{d(*)}}$ and $(x, y) \notin (R_\alpha)^{d(*)}$.

Suppose $(x, y) \notin R_{\alpha^*}$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in (R_\alpha)^*\}$. Since (D12) is true in $\mathcal{F}$, it follows $\mathcal{M}_0, x$ sat $[\alpha^*]p \Rightarrow [\alpha^{d(*)}]p$. By construction of m_0, $\mathcal{M}_0, x$ sat $[\alpha^*]p$ and therefore $\mathcal{M}_0, x$ sat $[\alpha^{d(*)}]p$. Since $(x, y) \in R_{\alpha^{d(*)}}$, we have $\mathcal{M}_0, y$ sat p. By construction of m_0, $\mathcal{M}_0, y$ sat $\neg p$ which leads to a contradiction. Now suppose that $x \uparrow R_\alpha$ does not hold. There exists a finite maximal R_α-chain from x, namely $(x_0, \ldots, x_N)$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in R_{\alpha^{d(*)}}\}$. Assuming that (D14) is true in $\mathcal{F}$, we get $\mathcal{M}_0, x$ sat $[\alpha^{d(*)}]p \wedge \langle\alpha^{d(*)}\rangle true \Rightarrow [\alpha^*]\langle\alpha\rangle true \wedge [\alpha^*]p$. By construction of m_0, $\mathcal{M}_0, x$ sat $[\alpha^{d(*)}]p$. Since $(x, y) \in R_{\alpha^{d(*)}}$, we have $\mathcal{M}_0, x$ sat $\langle\alpha^{d(*)}\rangle true$. Hence, $\mathcal{M}_0, x$ sat $[\alpha^*]\langle\alpha\rangle true \wedge [\alpha^*]p$. Since $(x, x_N) \in R_{\alpha^*}$, we have $\mathcal{M}_0, x_N$ sat $\langle\alpha\rangle true$ which leads to a contradiction. This proves that if the axioms (D12),(D13),(D14) are true in $\mathcal{F}$ then for all $\alpha \in \Pi$, $R_{\alpha^{d(*)}} \subseteq (R_\alpha)^{d(*)}$.

(II) Suppose there exists $x, y \in U$ such that $(x, y) \notin R_{\alpha^{d(*)}}$ and $(x, y) \in (R_\alpha)^{d(*)}$.

Suppose $R_{\alpha^{d(*)}}(x) = \emptyset$. Consider a model $\mathcal{M} = (U, \{R_\gamma\}_{\gamma\in\Pi}, m)$. Since (D13) is true in $\mathcal{F}$, we have $\mathcal{M}, x$ sat $[\alpha^{d(*)}]false \Rightarrow \langle\alpha^*\rangle[\alpha]false$. Since $R_{\alpha^{d(*)}}(x) = \emptyset$, we obtain $\mathcal{M}, x$ sat $[\alpha^{d(*)}]false$. Hence $\mathcal{M}, x$ sat $\langle\alpha^*\rangle[\alpha]false$. There exists $x_0 \in U$ such that $(x, x_0) \in R_{\alpha^*}$ and $R_\alpha(x_0) = \emptyset$. It follows that not $x \uparrow R_\alpha$ which leads to a contradiction since $(x, y) \in (R_\alpha)^{d(*)}$.

Now suppose $R_{\alpha^{d(*)}}(x) \neq \emptyset$. Consider the model $\mathcal{M}_0 = (U, \{R_\gamma\}_{\gamma\in\Pi}, m_0)$ such that for a certain $p \in \phi_0$, $m_0(p) = \{u \in U \mid (x, u) \in R_{\alpha^{d(*)}}\}$. Since (D14) is true in $\mathcal{F}$, $\mathcal{M}_0, x$ sat $[\alpha^{d(*)}]p \wedge \langle\alpha^{d(*)}\rangle true \Rightarrow [\alpha^*]\langle\alpha\rangle true \wedge [\alpha^*]p$. By construction of m_0, we have $\mathcal{M}_0, x$ sat $[\alpha^{d(*)}]p$. Since $R_{\alpha^{d(*)}}(x) \neq \emptyset$, we have $\mathcal{M}_0, x$ sat $\langle\alpha^{d(*)}\rangle true$. Hence, we obtain $\mathcal{M}_0, x$ sat $[\alpha^*]\langle\alpha\rangle true \wedge [\alpha^*]p$. Since $(x, y) \in (R_\alpha)^*$ then $\mathcal{M}_0, y$ sat p, which is in contradiction with the fact that not $\mathcal{M}_0, y$ sat p since $(x, y) \notin R_{\alpha^{d(*)}}$. This proves that if the axioms (D12),(D13),(D14) are true in $\mathcal{F}$, then for all $\alpha \in \Pi$, $(R_\alpha)^{d(*)} \subseteq R_{\alpha^{d(*)}}$.

As in the proof of Lemma 3.1 it can be easily proved that if for all $\alpha \in \Pi$, $R_{\alpha^{d(*)}} = (R_\alpha)^{d(*)}$ then the axioms (D12),(D13),(D14) are true in $\mathcal{F}$. Q.E.D.

Theorem 3.4. (Soundness of PDL(; ;, ||, $^{d(*)}$))
If F is a theorem of PDL(; ;, ||, $^{d(*)}$), then for every program model $\mathcal{M}$ we have $\mathcal{M} \models F$.

Proof: The proof consists in showing that the axioms are valid and the rules preserve validity. Validity of specific axioms (D4),. . . ,(D9) and (D12) . . . (D14) follows from Lemma 3.1, Lemma 3.2, and Lemma 3.3. Q.E.D.

4 Completeness of PDL(; ;, ||, $^{d(*)}$)

We use the standard construction of the canonical structure (e.g., [Mak66, Seg82]).

Definition 4.1. A canonical structure for PDL(; ;, ||, $^{d(*)}$) is the system $\mathcal{M}^c = (U^c, \{R^c_\alpha\}_{\alpha\in\Pi}, m^c)$ where

(i) U^c is the family of all the maximal consistent sets of formulae.

(ii) For all $\alpha \in \Pi$, the relations R^c_α over U^c are defined by $(Z, T) \in R^c_\alpha$ iff $\{F \in \Sigma \mid [\alpha]F \in Z\} \subseteq T$.

(iii) Valuation m^c is constructed by taking $m^c(p) = \{Z \in U^c \mid p \in Z\}$ for every formula $p \in \phi_0$.

∇

We show that $\mathcal{M}^c$ is a model but not necessarily a program model.

Lemma 4.2. For all $u \in U^c$, $F \in \Sigma$ and $\alpha, \beta \in \Pi$,

(i) $\mathcal{M}^c, u$ sat F iff $F \in u$

(ii) F is a theorem of PDL(; ;, ||, $^{d(*)}$) iff F is true in $\mathcal{M}^c$.

(iii) $R^c_{\alpha\cup\beta} = R^c_\alpha \cup R^c_\beta$; (iv) $R^c_{\alpha;\beta} = R^c_\alpha; R^c_\beta$; (v) $(R^c_\alpha)^* \subseteq R^c_{\alpha^*}$; (vi) $R^c_{\alpha^{d(*)}} \subseteq R^c_{\alpha^*}$

Proof(sketch): From Lemma 4.1 (respectively Corollary 4.2, Lemma 4.3A, Lemma 4.3B, Lemma4.3C) in [Seg82] we can easily deduce (i) (respectively (ii), (iii), (iv), (v)) holds. (vi) follows from axiom (D12). Q.E.D.

Lemma 4.3. If S is a maximal consistent set of formulae of PDL(; ;, ||, $^{d(*)}$), then

(i) If $[\alpha]F, [\beta]F \in S$ then $[\alpha||\beta]F \in S$; (ii) If $[\alpha][\beta]F \in S$ then $[\alpha;;\beta]F \in S$

(iii) If $[\alpha^*]F \in S$ then $[\alpha^{d(*)}]F \in S$

(iv) $[\alpha||\beta]false \in S$ iff either $[\alpha]false \in S$ or $[\beta]false \in S$

(v) $[\alpha;;\beta]false \in S$ iff either $[\alpha][\beta]false \in S$ or $\langle\alpha\rangle[\beta]false \in S$

(vi) If $[\alpha^{d(*)}]false \in S$ then $\langle\alpha^*\rangle[\alpha]false \in S$

(vii) If $[\alpha||\beta]F, \langle\alpha||\beta\rangle true \in S$ then $\langle\alpha\rangle true, \langle\beta\rangle true, [\alpha]F, [\beta]F \in S$

(viii) If $[\alpha;;\beta]F, \langle\alpha;;\beta\rangle true \in S$ then $[\alpha](\langle\beta\rangle true \wedge [\beta]F) \in S$.

(ix) If $[\alpha^{d(*)}]F, \langle\alpha^{d(*)}\rangle true \in S$ then $[\alpha^*]F, [\alpha^*]\langle\alpha\rangle true \in S$

The proof of Lemma 4.3 is by an easy verification knowing that any maximal consistent set is closed on *modus ponens* and contains all the theorems of PDL($;;, ||, {}^{d(*)}$). In the canonical model the demonic union and the demonic composition satisfy the conditions of Definition 2.4, namely we have the following lemma.

Lemma 4.4. For all $\alpha, \beta \in \Pi$,

(i) $R^c_{\alpha||\beta} = R^c_\alpha || R^c_\beta$

(ii) $R^c_{\alpha;;\beta} = R^c_\alpha ;; R^c_\beta$.

Proof: By way of example, the proof of (ii) is presented below.

(ii) Suppose that $R^c_{\alpha;;\beta} \not\subseteq R^c_\alpha ;; R^c_\beta$. Then there exist $x, y \in U^c$ such that $(x, y) \in R^c_{\alpha;;\beta}$ and either $(x, y) \notin R^c_\alpha ; R^c_\beta$, or there exists $z_0 \in U$ such that $(x, z_0) \in R^c_\alpha$ and $R^c_\beta(z_0) = \emptyset$. First suppose $(x, y) \notin R^c_\alpha ; R^c_\beta$. There exists $F_1 \in \Sigma$ such that $[\alpha][\beta]F_1 \in x$ and $F_1 \notin y$. From Lemma 4.3(ii) it follows that $[\alpha;;\beta]F_1 \in x$. Since $(x, y) \in R^c_{\alpha;;\beta}$, we have $F_1 \in y$, which leads to a contradiction. Now suppose that there exists $z_0 \in U$ such that $(x, z_0) \in R^c_\alpha$ and $R^c_\beta(z_0) = \emptyset$. It follows $\langle\alpha\rangle[\beta]false \in x$. From Lemma 4.3(v) $[\alpha;;\beta]false \in x$. Hence $false \in y$ since $(x, y) \in R^c_{\alpha;;\beta}$, which leads to a contradiction.

Now suppose that $R^c_\alpha ;; R^c_\beta \not\subseteq R^c_{\alpha;;\beta}$. Then there exist $x, y \in U^c$ such that $(x, y) \in R^c_\alpha ;; R^c_\beta$ and $(x, y) \notin R^c_{\alpha;;\beta}$. Suppose that $R^c_{\alpha;;\beta}(x) = \emptyset$. It follows that $[\alpha;;\beta]false \in x$. From Lemma 4.3(v) we have $[\alpha][\beta]false \vee \langle\alpha\rangle[\beta]false \in x$. Since $(x, y) \in R^c_\alpha ;; R^c_\beta$ (and therefore $(x, y) \in R^c_\alpha ; R^c_\beta$), for all $z \in U$, if $(x, z) \in R^c_\alpha$, then there exists $t \in U$ such that $(z, t) \in R^c_\beta$. Hence neither $[\alpha][\beta]false \in x$ nor $\langle\alpha\rangle[\beta]false \in x$, which leads to a contradiction. Now suppose that $R^c_{\alpha;;\beta}(x) \neq \emptyset$. It follows that $\langle\alpha;;\beta\rangle true \in x$. Since $(x, y) \notin R^c_{\alpha;;\beta}$, there exists $[\alpha;;\beta]F_1 \in x$ such that $F_1 \notin y$. From Lemma 4.3(viii), we have $[\alpha]\langle\beta\rangle true, [\alpha][\beta]F_1 \in x$. Since $(x, y) \in R^c_\alpha ; R^c_\beta$ it follows that $F_1 \in y$, which leads to a contradiction. Q.E.D.

To prove completeness, we use the filtration method developed in [Seg82] (see also [Gab72, Gol92]) and we show that the demonic operators behave adequately. Let $Sub(F)$ be the set of *subformulae* of a formula F. Let $\Gamma(F)$ be the smallest set such that

(i) $Sub(F) \cup \{false, true\} \subseteq \Gamma(F)$

(ii) $\Gamma(F)$ is closed under subformulae

(iii) if $[\alpha \cup \beta]G \in \Gamma(F)$ then $[\alpha]G, [\beta]G \in \Gamma(F)$

(iv) if $[\alpha;\beta]G \in \Gamma(F)$ then $[\alpha][\beta]G \in \Gamma(F)$

(v) if $[\alpha||\beta]G \in \Gamma(F)$ then $[\alpha]G, [\beta]G \in \Gamma(F)$

(vi) if $[\alpha;;\beta]G \in \Gamma(F)$ then $[\alpha][\beta]G, \langle\alpha\rangle[\beta]false \in \Gamma(F)$

(vii) if $[\alpha]G \in \Gamma(F)$ then $[\alpha]false \in \Gamma(F)$

(viii) if $[\alpha^*]G \in \Gamma(F)$ then $[\alpha][\alpha^*]G \in \Gamma(F)$

(ix) if $[\alpha^{d(*)}]G \in \Gamma(F)$ then $[\alpha^*]G, [\alpha^*]\langle\alpha\rangle true \in \Gamma(F)$

By the set of program terms of a formula F, denoted by $\Pi(F)$, we understand the smallest set such that $\{\alpha \mid [\alpha]F \in \Gamma(F)\} \subseteq \Pi(F)$ and $\Pi(F)$ is closed under subterms. By the set of program letters of a formula F, denoted by $\Pi_0(F)$ we understand the set $\Pi(F) \cap \Pi_0$. The set $\Gamma(F)$ is an extension of the Fischer-Ladner closure of the set $\{F\}$ [FL79]. Using the standard techniques, it can be shown that the set $\Gamma(F)$ is finite.

For any formula F and model $\mathcal{M}$, we define a structure $\mathcal{M}^F$ which is a filtration of the canonical model through the set $\Gamma(F)$. Let $\mathcal{M} = (U, \{R_\alpha\}_{\alpha\in\Pi}, m)$ be a program model. Define an equivalence relation $\equiv_{\Gamma(F)}$ on U by identifying states which satisfy the same formulae from $\Gamma(F)$, that is

$$x \equiv_{\Gamma(F)} y \text{ iff for all } G \in \Gamma(F),\ \mathcal{M}, x \text{ sat } G \text{ iff } \mathcal{M}, y \text{ sat } G$$

We denote by $|x|_{\Gamma(F)}$ the set $\{y \mid y \in U, x \equiv_{\Gamma(F)} y\}$ of equivalence classes of $\equiv_{\Gamma(F)}$. The subscript $\Gamma(F)$ may be omitted in $|x|_{\Gamma(F)}$ when the context is not ambiguous.

Definition 4.5. Let F be a formula and $\mathcal{M}^c = (U^c, \{R^c_\alpha\}_{\alpha\in\Pi}, m^c)$ be the canonical model. Define the model $\mathcal{M}^F = (U', \{R'_\alpha\}_{\alpha\in\Pi}, m')$ as follows:

(i) $U' = \{|x|_{\Gamma(F)} \mid x \in U^c\}$

(ii) For $\alpha \in \Pi_0$, $(|x|, |y|) \in R'_\alpha$ iff there exists $x_0, y_0 \in U^c$ such that $x_0 \equiv_{\Gamma(F)} x$, $y_0 \equiv_{\Gamma(F)} y$ and $x_0 R^c_\alpha y_0$.

(iii) $m'(p) = \{|x|_{\Gamma(F)} \mid x \in m^c(p)\}$ for $p \in \phi_0$

(iv) For $\alpha \in \Pi \setminus \Pi_0$, R'_α is defined inductively with respect to the complexity of the program operators in α

∇

By construction, $\mathcal{M}^F$ is a program model.

Lemma 4.6. Let F be a formula and $\mathcal{M}^F = (U', \{R'_\alpha\}_{\alpha\in\Pi}, m')$. For any $\gamma \in \Pi(F)$,

(i) if $(x, y) \in R^c_\gamma$ then $(|x|_{\Gamma(F)}, |y|_{\Gamma(F)}) \in R'_\gamma$.

(ii) if $(|x|_{\Gamma(F)}, |y|_{\Gamma(F)}) \in R'_\gamma$ then for all $[\gamma]G \in \Gamma(F)$ if $\mathcal{M}^c, x$ sat $[\gamma]G$ then $\mathcal{M}^c, y$ sat G.

Proof: The proof is by induction on the complexity of γ. The basic step follows from the definition of $\mathcal{M}^F$. For γ of the form $\alpha \cup \beta$, $\alpha;\beta$ and α^* the proof can be found in [Seg82]. We prove the induction step for demonic operators.

(I) Assume $(x, y) \in R^c_{\alpha||\beta}$. From Lemma 4.4(i), there exist $u_1, u_2 \in U$ such that $(x, u_1) \in R^c_\alpha$ and $(x, u_2) \in R^c_\beta$. Moreover either $(x, y) \in R^c_\alpha$ or $(x, y) \in R^c_\beta$. By the induction hypothesis we get $(|x|, |u_1|) \in R'_\alpha$ and $(|x|, |u_2|) \in R'_\beta$. Moreover either $(|x|, |y|) \in R'_\alpha$ or $(|x|, |y|) \in R'_\beta$. Since $\mathcal{M}^F$ is a program model, we get $(|x|, |y|) \in R'_{\alpha||\beta}$.

Assume $(|x|, |y|) \in R'_{\alpha||\beta}$. Take any $[\alpha||\beta]F_1 \in x \cap \Gamma(F)$. Suppose that $[\alpha||\beta]false \in x$. From Lemma 4.3(iv) we get $[\alpha]false \vee [\beta]false \in x$. Since either $[\alpha]false \in \Gamma(F) \cap x$ or $[\beta]false \in \Gamma(F) \cap x$, by the induction hypothesis $false \in y$, which naturally leads to a contradiction. So $\langle\alpha||\beta\rangle true \in x$. From Lemma 4.3(vii) we get $[\alpha]F_1 \wedge [\beta]F_1 \in x$. Since $(|x|, |y|) \in R'_\alpha \cup R'_\beta$, by the induction hypothesis we get $F_1 \in y$.

(II) Assume $(x, y) \in R^c_{\alpha;;\beta}$. From Lemma 4.4(ii), there exists $t \in U^c$ such that $(x, t) \in R^c_\alpha$ and $(t, y) \in R^c_\beta$. First, suppose $(|x|, |y|) \notin R'_\alpha; R'_\beta$. By the induction hypothesis, we get $(|x|, |t|) \in R'_\alpha$ and $(|t|, |y|) \in R'_\beta$, which leads to a contradiction. Now suppose there exists $|z_0| \in U'$ such that $(|x|, |z_0|) \in R'_\alpha$ and $R'_\beta(|z_0|) = \emptyset$. Since $\langle\alpha;;\beta\rangle true, [\alpha;;\beta]true \in x$, from Lemma 4.3(viii) we have $[\alpha]\langle\beta\rangle true \in x$. By the induction hypothesis, we have $\langle\beta\rangle true \in z_0$. There exists $z_1 \in U^c$ such that $(z_0, z_1) \in R^c_\beta$. By the induction hypothesis it follows that $(|z_0|, |z_1|) \in R'_\beta$ which leads to a contradiction.

Assume $(|x|, |y|) \in R'_{\alpha;;\beta}$. Take any $[\alpha;;\beta]F_1 \in x \cap \Gamma(F)$. Suppose $[\alpha;;\beta]false \in x$. By Lemma 4.3(v), we get $[\alpha][\beta]false \vee \langle\alpha\rangle[\beta]false \in x$. In case $[\alpha][\beta]false \in x$,

by the induction hypothesis we get $false \in y$, which leads to a contradiction. In case $\langle\alpha\rangle[\beta]false \in x$, there exists $z_0 \in U$ such that $(x, z_0) \in R^c_\alpha$ and $[\beta]false \in z_0$. So by the induction hypothesis, $(|x|, |z_0|) \in R'_\alpha$ and there exists $|t_0| \in U'$ such that $(|z_0|, |t_0|) \in R'_\beta$ since $(|x|, |y|) \in R'_{\alpha;;\beta}$. By the induction hypothesis, $false \in t_0$ which leads to a contradiction. It follows that $\langle\alpha;;\beta\rangle true \in x$. By Lemma 4.3(viii), $[\alpha;\beta]F_1 \in x$ and therefore $[\alpha][\beta]F_1 \in x$. There exists $|t| \in U'$ such that $(|x|, |t|) \in R'_\alpha$ and $(|t|, |y|) \in R'_\beta$. By the induction hypothesis, $[\beta]F_1 \in t$ and $F_1 \in y$.

(III) Assume $(x, y) \in R^c_{\alpha^{d(*)}}$. Suppose that $(|x|, |y|) \notin R'_{\alpha^{d(*)}}$. Since $R^c_{\alpha^{d(*)}} \subseteq R^c_{\alpha^*}$ we have $(x, y) \in R^c_{\alpha^*}$. By the induction hypothesis we have $(|x|, |y|) \in R'_{\alpha^*}$. Since $R'_{\alpha^{d(*)}} = (R'_\alpha)^{d(*)}$ there is a sequence $(|x_0|, \ldots, |x_N|)$ $(N \geq 0)$ such that $|x_0| = |x|$ and for $i \in \{0, \ldots, (N-1)\}$ we have $(|x_i|, |x_{i+1}|) \in R'_\alpha$ and $R'_\alpha(|x_N|) = \emptyset$. It follows that $R^c_\alpha(x_N) = \emptyset$. Since $(x, y) \in R^c_{\alpha^{d(*)}}$ it follows that $\langle\alpha^{d(*)}\rangle true \in x$. Moreover, since $[\alpha^{d(*)}]true \in x$, we have $[\alpha^*]\langle\alpha\rangle true \in x$ (see Lemma 4.3(ix)). Then $[\alpha][\alpha^*]\langle\alpha\rangle true \in x \cap \Gamma(F)$ (see the condition (viii) in the definition of $\Gamma(F)$) and by the induction hypothesis it follows that $[\alpha^*]\langle\alpha\rangle true \in x_1$ since $(|x_0|, |x_1|) \in R'_\alpha$. By continuing this process we get $[\alpha^*]\langle\alpha\rangle true \in x_N$, which is in contradiction with the fact that $R^c_\alpha(x_N) = \emptyset$.

Assume $(|x|, |y|) \in R'_{\alpha^{d(*)}}$. Take any $[\alpha^{d(*)}]F_1 \in x \cap \Gamma(F)$. Suppose $F_1 \notin y$. First suppose $\langle\alpha^{d(*)}\rangle true \in x$. From Lemma 4.3(ix), we get $[\alpha^*]F_1 \in x$. Since $(|x|, |y|) \in R'_{\alpha^*}$, by the induction hypothesis we have $F_1 \in y$, which leads to a contradiction. Now suppose $\langle\alpha^{d(*)}\rangle true \notin x$. From Lemma 4.3(vi), we get $\langle\alpha^*\rangle[\alpha]false \in x$. There exists $y_0 \in U^c$ such that $(x, y_0) \in R^c_{\alpha^*}$ and $R^c_\alpha(y_0) = \emptyset$. By the induction hypothesis, we deduce $(|x|, |y_0|) \in R'_{\alpha^*}$. Since $R'_{\alpha^{d(*)}}(|x|) \neq \emptyset$, there exists $t_0 \in U^c$ such that $(|y_0|, |t_0|) \in R'_\alpha$. Since $[\alpha]false \in y_0 \cap \Gamma(F)$, by the induction hypothesis it follows that $false \in t_0$ which leads to a contradiction. Q.E.D.

We can now prove the following lemmas.

Lemma 4.7. Let F be a formula. For each state x of U^c, $G \in \Gamma(F)$, $\mathcal{M}^F, |x|_{\Gamma(F)}$ sat G iff $\mathcal{M}^c, x$ sat G.

The standard proof is by the induction with respect to the complexity of G.

Theorem 4.8. (Completeness of PDL(;;, ||, $^{d(*)}$)) If F is true in every program model, then F is a theorem of PDL(;;, ||, $^{d(*)}$).

The standard proof is based on Lemma 4.7.

Theorem 4.9. The logic PDL(;;, ||, $^{d(*)}$) has the finite model property.

Proof: It is sufficient to show the following statement: for every satisfiable formula F, there exists a model $\mathcal{M} = (U, \{R_\alpha\}_{\alpha\in\Pi}, m)$ with a finite set U of states and there is $x \in U$ such that $\mathcal{M}, x$ sat F. Assume G is a satisfiable formula. By Theorem 4.8, there exists a state $x \in U^c$ (from the canonical model) such that $\mathcal{M}^c, x$ sat G. From Lemma 4.7, $\mathcal{M}^G, |x|_{\Gamma(G)}$ sat G. The set of states of $\mathcal{M}^G$ has less than $2^{Card(\Gamma(G))}$ elements. Q.E.D.

Since PDL(;;, ||, $^{d(*)}$) has the finite model property and PDL(;;, ||, $^{d(*)}$) has a complete finite axiomatization, then it is decidable (see Theorem 8.15 in [HC84]).

Corollary 4.10. The logic PDL(;;, ||, $^{d(*)}$) is decidable.

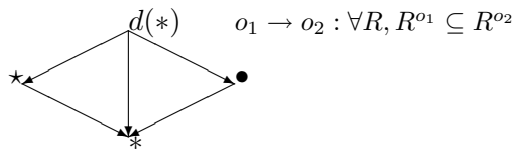

Figure 1: The lattice of demonic iteration operations

Alternative demonic iteration operations The demonic operator $^{d(*)}$ is defined in terms of the demonic union and the demonic composition. Two other demonic iteration operations can be defined:

$$R^{\bullet}(x) = (||_{i\in\omega} R^i)(x) \qquad R^{\star}(x) = (\bigcup_{i\in\omega} R^{d(i)})(x)$$

Properties of these two operators are stated in the following lemmas.

Lemma 4.11. Let R be a binary relation on U and $x \in U$. If for all $i \in \omega$, $R^i(x) \neq \emptyset$ then $R^{\bullet}(x) = R^*(x)$, otherwise $R^{\bullet}(x) = \emptyset$.

The iteration operation $^{\bullet}$ is also an 'all or nothing iteration'. However it is different from $^{d(*)}$. It can be easily shown that $R^{d(*)} \subseteq R^{\bullet}$ but the converse does not always hold. The iteration operator * can be characterized as follows.

Lemma 4.12. Let R be a binary relation on U and $x \in U$. If the set $\mathcal{S}_x = \{\sigma \mid \sigma$ is a finite maximal R-chain from $x\}$ is non-empty -i.e., not $x \uparrow R$-, then $R^{\star}(x) = \bigcup_{0\leq i\leq M-1} R^i(x)$ with $M = min\{|\sigma| \mid \sigma \in \mathcal{S}_x\}$, otherwise $R^{\star}(x) = R^*(x)$.

The proof of this lemma is similar to the proof of Lemma 2.3. The operation * is not an 'all or nothing iteration'. However, it can be easily shown that $R^{d(*)} \subseteq R^{\star}$. Figure 1 presents relationships between the different demonic iteration operations. Axiomatisation of a dynamic logic with the operators * or $^{\bullet}$ is an open problem.

5 Relational Formalization of PDL(; ;, ||, $^{d(*)}$)

In sections 5 and 6 we develop a relational formalization of logic PDL(; ;, ||, $^{d(*)}$) based on the method developed in [Orło90, Orło91, Orło92]. The method consists in defining a relational logic RelPDL(; ;, ||, $^{d(*)}$) for PDL(; ;, ||, $^{d(*)}$), and next in providing a validity preserving embedding of PDL(; ;, ||, $^{d(*)}$) into RelPDL(; ;, ||, $^{d(*)}$). Then the deduction system of RelPDL(; ;, ||, $^{d(*)}$) provides a means of natural deduction for PDL(; ;, ||, $^{d(*)}$).

The syntax of RelPDL(; ;, ||, $^{d(*)}$) consists of terms and formulae. The language of RelPDL(; ;, ||, $^{d(*)}$) is determined by five sets of symbols which are supposed to be pairwise disjoint:

(i) a denumerable set V_R of relational variables

(ii) a denumerable set V_I of individual variables

(iii) the set of relational operators $\{-, \cup, \cap, ;, *, ||, ;;, d(*)\}$ denoting the relational operations of complement, union, intersection, composition, iteration, demonic union, demonic composition, demonic iteration, respectively.

(iv) the set Π_0 of relational constants denoting atomic programs

(v) the set of relational constants $\{1, I\}$ denoting the universal relation and the identity relation, respectively.

The set of terms of RelPDL(;;, ||, $^{d(*)}$) is the smallest set S that satisfies the following conditions: (i) $\Pi_0 \cup \{1, I\} \subseteq S$, and (ii) if ϕ is a n-ary relational operator and $a_1, \ldots, a_n \in S$ then $\phi(a_1, \ldots, a_n) \in S$. Formulae of the relational logic are of the form xAy where A is a term and $\{x, y\} \subseteq V_I$.

Definition 5.1. A model of RelPDL(;;, ||, $^{d(*)}$) is a system of the form $\mathcal{M} = (U, \{R_\alpha \mid \alpha \in \Pi_0\}, R_1, R_I, m)$ where U is a nonempty set, R_α are binary relations in U and R_1, R_I are relations such that

(i) $R_1 = U \times U$ and $R_I = \{(x, x) \mid x \in U\}$

m is a meaning function such that

(ii) $m(\alpha) = R_\alpha$ for all $\alpha \in \Pi_0$, $m(1) = R_1$ and $m(I) = R_I$

(iii) if A is a relational variable then $m(A) = X \times U$ for some $X \subseteq U$

(iv) m preserves the relational operations.

∇

Relations of the form $X \times U$ are called *right ideal relations*. Observe that if A, B are right ideal relations then (i) $A; 1 = A$, (ii) $-A$, $A \cup B$, $A \cap B$ are right ideal relations and (iii) $P; A$ is a right ideal relation for any relation P.

By a *valuation* in $\mathcal{M}$ we mean an assignment $v : V_I \to U$ of states from U to individual variables. We say that in model $\mathcal{M}$ a valuation v satisfies a relational formula xAy ($\mathcal{M}, v$ *sat* xAy) whenever it holds $(v(x), v(y)) \in m(A)$. A formula xAy is *true* in a model $\mathcal{M}$ iff $\mathcal{M}, v$ *sat* xAy for all valuations v in $\mathcal{M}$. A formula is *valid* in RelPDL(;;, ||, $^{d(*)}$) iff it is true in all models.

Lemma 5.2. A formula xAy is true in a model $\mathcal{M} = (U, \{R_\alpha \mid \alpha \in \Pi_0\}, R_1, R_I, m)$ iff $m(A) = m(1)$.

Following [Orło92] we define a translation t of formulae from PDL(;;, ||, $^{d(*)}$) into formulae of RelPDL(;;, ||, $^{d(*)}$). Let t' be a bijection from the set ϕ_0 of propositional variables into set V_R of relational variables. Then we define:

$t(p) = t'(p)$ for any propositional variable p, $t(true) = 1$
$t(\alpha) = \alpha$ for $\alpha \in \Pi_0$

and t is a homomorphism with respect to the Booleans; while for any formula F of PDL(;;, ||, $^{d(*)}$) and program expressions α, β:

$t(\langle\alpha\rangle F) = t(\alpha); t(F)$
$t([\alpha]F) = -(t(\alpha); -t(F))$.
$t(\alpha o \beta) = t(\alpha) o (\beta)$ for $o \in \{\cup, ;, ||, ;;\}$
$t(\alpha^o) = (t(\alpha))^o$ for $o \in \{^*, ^{d(*)}\}$

In this way, every formula of PDL(;;, ||, $^{d(*)}$) is translated into a term that represents a right ideal relation. Semantical relationship between logic PDL(;;, ||, $^{d(*)}$) and relational logic RelPDL(;;, ||, $^{d(*)}$) is provided by the following lemma.

Lemma 5.3. A formula F of logic PDL(; ;, ||, $^{d(*)}$) is valid iff $xt(F)y$ is valid in RelPDL(; ;, ||, $^{d(*)}$).

Proof: The proof consists of the following two parts. (I) For every model $\mathcal{M} = (U,(R_\alpha)_{\alpha\in\Pi}, m)$ of PDL(; ;, ||, $^{d(*)}$) there is a model $\mathcal{M}' = (U, \{R'_\alpha \mid \alpha \in \Pi_0\}, R_1, R_I, m')$ of RelPDL(; ;, ||, $^{d(*)}$) such that for every formula F of PDL(; ;, ||, $^{d(*)}$)we have (i) $\mathcal{M}, x$ *sat* F iff $(x, z) \in m'(t(F))$ for all z in U. We define the model $\mathcal{M}'$ as follows. Its universe coincides with the universe U of $\mathcal{M}$. If $P \in V_R$ and $P = t'(p)$ for a propositional variable p, then we put $m'(P) = m(p) \times U$. We put $m'(1) = U \times U$, $m'(I) = \{(x, x) \mid x \in U\}$, $m'(\alpha) = R_\alpha$ for $\alpha \in \Pi_0$ and we extend m' to all the relational terms according to Definition 5.1 (iv). Observe that relational variables are interpreted as right ideal relations. It is easy to see that $\mathcal{M}'$ satisfies conditions 5.1(i)...(iv). The proof of the required condition (i) is by induction on the complexity of F.

(II) For every model $\mathcal{M}' = (U, \{R'_\alpha \mid \alpha \in \Pi_0\}, R_1, R_I, m')$ of the relational logic RelPDL(; ;, ||, $^{d(*)}$) there is a model $\mathcal{M} = (U, (R_\alpha)_{\alpha\in\Pi}, m)$ of PDL(; ;, ||, $^{d(*)}$) such that condition (i) is satisfied. We define the model $\mathcal{M}$ as follows. Its universe coincides with the universe U of $\mathcal{M}'$. For any propositional variable p we put $m(p)$ = domain of $m'(P)$ where $P = t'(p)$. By induction on the complexity of a formula F one can show that condition (i) is satisfied. Q.E.D.

6 A Relational Proof System for RelPDL(; ;, ||, $^{d(*)}$)

Proof systems for relational logics are Rasiowa-Sikorski style systems [RS63]. They consist of rules that apply to finite sequences of relational formulae in a top-down manner. There are the two groups of rules: decomposition rules and specific rules. Decomposition rules enable us to decompose formulae in a sequence into some simpler formulae (see Figures 2 and 3). Specific rules characterize relational constants from the language. In the figures, K and H denote finite, possibly empty, sequences of formulae of the relational logic RelPDL(; ;, ||, $^{d(*)}$). A variable is said to be restricted in a rule whenever it does not appear in any formula of the upper sequence in this rule. The rules of Figure 2 have been defined in [Orło92] for the standard propositional dynamic logic PDL.

The specific rules enable us to modify a sequence to which they are applied, they have a status of structural rules (see Figure 4).

The role of axioms is played by *fundamental sequences*. A sequence of formulae is said to be fundamental whenever it contains formulae of the following form.
($\mathcal{F}$) Fundamental sequences:

(f1) $xAy, x - Ay$ for any relational term A and for any $x, y \in V_I$

(f2) $x1y$ for any $x, y \in V_I$; (f3) xIx for any $x \in V_I$

A sequence K of relational formulae is *true in a model* $\mathcal{M}$ of the relational logic RelPDL(; ;, ||, $^{d(*)}$) if for every valuation v over $\mathcal{M}$ there is a formula in K which is satisfied by v in $\mathcal{M}$. Sequence K is *valid in RelPDL(; ;, ||,* $^{d(*)}$*)* iff it is true in all models. It follows that sequences of formulae are interpreted as (metalevel) disjunction of their elements. A relational rule of the form $\frac{K}{\{H_i : i \in \mathcal{I}\}}$ is *admissible* whenever the sequence K is valid iff for all $i \in \mathcal{I}$ the sequence H_i is valid.

Lemma 6.1. (i) All the rules in $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$ are admissible.

(ii) All the sequences in ($\mathcal{F}$) are valid.

$(\cup)\frac{K,xA\cup By,H}{K,xAy,xBy,H}$

$(-\cup)\frac{K,x-(A\cup B)y,H}{K,x-Ay,H \quad K,x-By,H}$

$(\cap)\frac{K,xA\cap By,H}{K,xAy,H \quad K,xBy,H}$

$(-\cap)\frac{K,x-(A\cap B)y,H}{K,x-Ay,x-By,H}$

$(--)\frac{K,x--Ay,H}{K,xAy,H}$

$(;)\frac{K,xA;By,H}{K,xAz,H,xA;By \quad K,zBy,H,xA;By}$ z is a variable

$(-;)\frac{K,x-(A;B)y,H}{K,x-Az,z-By,H}$ z is a restricted variable.

$(^*)\frac{K,xA^*y,H}{K,xA^iy,H,xA^*y}$ for any natural number i where $A^0 = I$, $A^{i+1} = A; A^i$

$(-^*)\frac{K,x-A^*y,H}{\{K,x-A^iy,H\}_{i\geq 0}}$

Figure 2: $(\mathcal{R}_1)$ Decomposition rules for the standard relational operations

$(||)\frac{K,xA||By,H}{K,xAz,H,xA||By \quad K,xBt,H,xA||By \quad K,xAy,xBy,H}$ z, t are arbitrary variables

$(-||)\frac{K,x-(A||B)y,H}{K,x-Az,x-Bt,x-Ay,H \quad K,x-Az,x-Bt,x-By,H}$ z, t are restricted variables

$(;;)\frac{K,xA;;By,H}{K,x-Az,H,zB;1y \quad K,xAt,H,xA;;By \quad K,tBy,H,xA;;By}$ z is restricted variable and t is an arbitrary variable

$(-;;)\frac{K,x-(A;;B)y,H}{K,x-Az,z-By,xAu,H,x-(A;;B)y \quad K,x-Az,z-By,u-Bt,H,x-(A;;B)y}$ with z restricted variables and t, u are arbitrary variables

$(^{d(*)})\frac{K,xA^{d(*)}y,H}{K,xA^*y,H \quad \{K,xA^{d(i)};1y,H\}_{i\geq 0}}$ where $A^{d(0)} = I$, $A^{d(i+1)} = A;; A^{d(i)}$

$(-^{d(*)})\frac{K,x-A^{d(*)}y,H}{K,x-A^*y,x-(A^{d(i)};1)y,H,x-A^{d(*)}y}$ for any natural number i

Figure 3: $(\mathcal{R}_2)$ Decomposition rules for the demonic operations

$(I1)\frac{K,xAy,H}{K,xIz,H,xAy \quad K,zAy,H,xAy}$ z is a variable, $A \in V_R \cup \Pi_0$

$(I2)\frac{K,xAy,H}{K,xAz,H,xAy \quad K,zIy,H,xAy}$ z is a variable, $A \in V_R \cup \Pi_0$

$(SymI)\frac{K,xIy,H}{K,yIx,H,xIy}$

$(TranI)\frac{K,xIy,H}{K,xIz,H,xIy \quad K,zIy,H,xIy}$ z is a variable

$(ideal)\frac{K,xAy,H}{K,xAz,H,xAy}$ z is a variable, $A \in V_R$

Figure 4: $(\mathcal{R}_3)$ Specific rules

Proof: Admissibility of the decomposition rules follows from definitions of the respective relational operations, and admissibility of the specific rules follows from the properties of relational constants reflected by those rules [Orło92]. By way of example, the proof for the rules $(||)$, $(-;;)$ and $({}^{d(*)})$ is given below.

$(||)$ Assume $K, xA||By, H$ is a valid sequence. It is immediate that the sequences K, xAz, $H, xA||By$ and $K, xBt, H, xA||By$ are also valid. Now suppose that K, xAy, xBy, H is not valid. There exists a model $\mathcal{M}_0 = (U, \{R_\alpha\}_{\alpha\in\Pi_0}, R_1, R_I, m)$ and a valuation v_0 such that for every formula $x_1A_1y_1$ of K, xAy, xBy, H , not $\mathcal{M}_0, v_0$ sat $x_1A_1y_1$. It follows that $(v_0(x), v_0(y)) \notin m(A) \cup m(B)$. Hence $(v_0(x), v_0(y)) \notin m(A||B)$. It follows that the sequence $K, xA||By, H$ is not valid which leads to a contradiction.

Now assume that the lower sequences of the rule $(||)$ are valid. Let $\mathcal{M}$ be any model and v a valuation in $\mathcal{M}$. There exists $x_1A_1y_1 \in K, xAz, H, xA||By$, $x_2A_2y_2 \in K, xBt, H, xA||By$ and $x_3A_3y_3 \in K, xAy, xBy, H$ such that $\mathcal{M}, v$ sat $x_1A_1y_1$, $\mathcal{M}, v$ sat $x_2A_2y_2$, and $\mathcal{M}, v$ sat $x_3A_3y_3$. If $x_1A_1y_1 \in K, xA||By, H$, or $x_2A_2y_2 \in K, xA||By, H$ or $x_3A_3y_3 \in K, H$ then it is immediate that there is a formula $x_4A_4y_4 \in K, xA||By, H$ such that $\mathcal{M}, v$ sat $x_4A_4y_4$. Now assume $x_1A_1y_1 = xAz$ and $x_2A_2y_2 = xBt$. First, assume $x_3A_3y_3 = xAy$. Since $(v(x), v(y)) \in m(A)$, $(v(x), v(t)) \in m(B)$ and $(v(x), v(y)) \in m(A)$ it follows that $(v(x), v(y))$ belongs to $m(A)||m(B) = m(A||B)$. Hence $\mathcal{M}, v$ sat $xA||By$. Now assume $x_3A_3y_3 = xBy$. Since $(v(x), v(y)) \in m(A)$, $(v(x), v(t)) \in m(B)$ and $(v(x), v(y)) \in m(B)$ it follows that $(v(x), v(y))$ belongs to $m(A)||m(B) = m(A||B)$. Hence $\mathcal{M}, v$ sat $xA||By$.

$(-;;)$ Assume $K, x-(A;;B)y, H$ is a valid sequence. It is immediate that the sequences $K, x - Az, z - By, xAu, H, x - A;;By$ and $K, x - Az, z - By, u - Bt, H, x - (A;;B)y$ are also valid.

Now assume that $K, x - (A;;B)y, H$ is not valid. There exists a model $\mathcal{M}_0 = (U, \{R_\alpha\}_{\alpha\in\Pi_0}, R_1, R_I, m)$ and a valuation v_0 such that for all $x_1A_1y_1 \in K, x-(A;;B)y, H$, not $\mathcal{M}_0, v_0$ sat $x_1A_1y_1$. In particular, $(v_0(x), v_0(y)) \in m(A;;B) = m(A);;m(B)$. There exists $z_0 \in U$ such that $(v_0(x), z_0) \in m(A)$ and $(z_0, v_0(y)) \in m(B)$. Let v_0' be the valuation such that $v_0'(z) = z_0$ and $v_0'(w) = v_0(w)$ for all $w \in V_I \setminus \{z\}$. It can be easily shown that neither $\mathcal{M}_0, v_0'$ sat $x-Az$ nor $\mathcal{M}_0, v_0'$ sat $z-By$. Moreover for all $s \in U$ either $(v_0'(x), s) \notin m(A)$ or for $s' \in U$, $(s, s') \in m(B)$. For $u \in V_I$, if $(v_0'(x), v_0'(u)) \notin m(A)$ then not $\mathcal{M}_0, v_0'$ sat xAu. Otherwise for $t \in V_I$, $(v_0'(u), v_0'(t)) \in m(B)$ and therefore not $\mathcal{M}_0, v_0'$ *sat* $u - Bt$.

$({}^{d(*)})$ We recall that according to Lemma 2.3(ii) we have $(x, y) \in R^{d(*)}$ iff $(x, y) \in R^*$ and for all $i \in \omega$, $(x, y) \in R^{d(i)}; 1$.

Assume that $K, xA^{d(*)}y, H$ is a valid sequence. Now suppose that K, xA^*y, H is not valid. There exists a model $\mathcal{M}_0 = (U, \{R_\alpha\}_{\alpha\in\Pi_0}, R_1, R_I, m)$ and a valuation v_0 such that for every formula $x_1A_1y_1$ of K, xA^*y, H, not $\mathcal{M}_0, v_0$ sat $x_1A_1y_1$. It follows that $(v_0(x), v_0(y)) \notin m(A^*)$. Hence $(v_0(x), v_0(y)) \notin m(A^{d(*)})$. It follows that the sequence $K, xA^{d(*)}y, H$ is not valid which leads to a contradiction. Suppose that there exists $i \in \omega$ such that $K, xA^{d(i)}; 1y, H$ is not valid. There exists a model $\mathcal{M}_0 = (U, \{R_\alpha\}_{\alpha\in\Pi_0}, R_1, R_I, m)$ and a valuation v_0 such that for all $x_1A_1y_1 \in K, xA^{d(i)}; 1y, H$, not $\mathcal{M}_0, v_0$ sat $x_1A_1y_1$. In particular it follows that $(v_0(x), v_0(y)) \notin m(A)^{d(i)}; 1$ and therefore $(v_0(x), v_0(y)) \notin m(A^{d(*)})$, which leads to a contradiction.

Now assume that all the lower sequences of the rule $({}^{d(*)})$ are valid. Let $\mathcal{M}$ be any model and v a valuation in $\mathcal{M}$. There exists $x'A'y' \in K, xA^*y, H$, and $x_iA_iy_i \in K, xA^{d(i)}; 1y, H$ for all $i \in \omega$ such that $\mathcal{M}, v$ sat $x'A'y'$ and for all $i \in \omega$, $\mathcal{M}, v$ sat $x_iA_iy_i$. If $x'A'y' \in K, H$, or if there exists $k \in \omega$ such that $x_kA_ky_k \in K, H$ then it is immediate that there is a formula $x''A''y'' \in K, xA^{d(*)}y, H$ such that $\mathcal{M}, v$ sat $x''A''y''$. Now assume $x'A'y' = xA^*y$ and for all $i \in \omega$ $x_iA_iy_i = xA^{d(i)}; 1y$. It follows that $(v(x), v(y)) \in m(A)^*$ and for all $i \in \omega$, $(v(x), v(y)) \in m(A)^{d(i)}; 1$. By definition of ${}^{d(*)}$

it follows that $\mathcal{M}, v$ sat $xA^{d(*)}y$. Q.E.D.

Relational proofs have the form of trees. Given a relational formula xAy, where A might be a compound relational expression, we successively apply decomposition or specific rules. In this way we form a tree whose root consists of xAy and whose nodes consist of finite sequences of relational formulae. We stop applying rules to the formulae in a node after obtaining a fundamental sequence, or when none of the rules is applicable to the formulae in this node. A branch of a proof tree is said to be *closed* whenever it contains a node with a fundamental sequence of formulae. A tree is closed iff all of its branches are closed.

Definition 6.2. (Complete branch, Complete tree) Let T be a tree whose root is labeled by the formula xAy and b a branch of T. The branch b is said to be complete iff either b is closed or b satisfies the following conditions. In what follows we write $G \in b$ whenever a formula G is a member of a sequence of formulae in a certain node of branch b.

(b1) $xAy \in b$.

(b2) If $x(B \cup C)y$ $(x - (B \cap C)y) \in b$, then both xBy $(x - By) \in b$ and xCy $(x - Cy) \in b$ obtained by application of rule $(\cup)$ (resp. $(-\cap)$).

(b3) If $x - (B \cup C)y$ $(x(B \cap C)y) \in b$, then either $x - By$ $(xBy) \in b$ or $x - Cy$ $(xCy) \in b$ obtained by application of rule $(-\cup)$ (resp. $(\cap)$).

(b4) If $x(B; C)y \in b$, then for every $z \in V_I$ either $xBz \in b$ or $zCy \in b$ obtained by application of rule $(;)$.

(b5) If $x - (B; C)y \in b$, then for some $z \in V_I$ both $x - Bz \in b$ and $z - Cy \in b$ obtained by application of rule $(-;)$.

(b6) If $x - -By \in b$, then $xBy \in b$ obtained by application of rule $(--)$.

(b7) If $xB^*y \in b$, then for all $i \in \omega$, $xB^iy \in b$ obtained by application of rule $(*)$

(b8) If $x - B^*y \in b$, then for some $i \in \omega$, $x - B^iy \in b$ obtained by application of rule $(-*)$

(b9) If $xBy \in b$ and $B \in V_R \cup \Pi_0$, then for all $z \in V_I$ either $xIz \in b$ or $zBy \in b$ obtained by application of rule (I1)

(b10) If $xBy \in b$ and $B \in V_R \cup \Pi_0$, then for all $z \in V_I$ either $xBz \in b$ or $zIy \in b$ obtained by application of rule (I2)

(b11) If $xIy \in b$, then $yIx \in b$ obtained by application of the rule (SymI)

(b12) If $xIy \in b$, then for all $z \in V_I$ either $xIz \in b$ or $zIy \in b$ obtained by application of rule (TranI)

(b13) If $xBy \in b$ and $B \in V_R$, then for every $z \in V_I$ we have $xBz \in b$ obtained by application of rule (ideal)

(b14) If $xB||Cy \in b$, then for all $z, t \in V_I$ either $xBz \in b$ or $xCt \in b$ or $xBy \in b$ and $xCy \in b$ obtained by application of the rule $(||)$

(b15) If $x - (B||C)y \in b$, then for some $z, t \in V_I$ either $x - Bz, x - Ct, x - By \in b$ or $x - Bz, x - Ct, x - Cy \in b$ obtained by application of rule $(-||)$

(b16) If $xB;;Cy \in b$, then for all $t \in V_I$ and for some $z \in V_I$ either $x - Bz, zC; 1y \in b$ or $xBt \in b$ or $tCy \in b$ obtained by application of rule $(;;)$

(b17) If $x - (B;;C)y \in b$, then for all $t, u \in V_I$ and for some $z \in V_I$ $x - Bz, z - Cy \in b$ and either $xBu \in b$ or $u - Ct \in b$ obtained by application of rule $(-;;)$

(b18) If $xB^{d(*)}y \in b$, then either $xB^*y \in b$ or for some $i \in \omega$ $xB^{d(i)}; 1y \in b$ obtained by application of rule $(d(*))$

(b19) If $x - B^{d(*)}y \in b$ then for all $j \in \omega$ $x - (B^{d(j)}; 1)y \in b$ and $x - B^*y \in b$ obtained by application of rule $(-d(*))$

A tree is said to be complete iff all of its branches are complete. ∇

Lemma 6.3. For any finite sequence S of relational formulae there is a complete tree T such that its root is labeled by the sequence S.

The full technical development of the proof of Lemma 6.3 is quite tedious and is omitted here. The basic idea is to define a procedure which guarantees that any tree constructed according to the procedure is such that if the procedure runs infinitely, then every branch will be complete. The application of the rules is made in a fair way so that it is not the case that when two rules can be applied in a sequence, one of them is applied infinitely many times. Furthermore, a particular treatment is provided for the infinitary rules.

Theorem 6.4. (Completeness of RelPDL$(;;, ||, ^{d(*)})$)
A relational formula $xA'y$ is valid iff there is a closed proof tree with the root $xA'y$.

Proof: Observe that there exists a complete tree T such that the root is labeled by $xA'y$ (Lemma 6.3). Suppose that a complete branch b of T is non-closed. We define the structure $\mathcal{M}^b = (W^b, m^b)$ such that $W^b = V_I$ and $m^b(P) = \{(x, y) \in W^b \times W^b : xPy \notin b\}$ for $P \in V_R \cup \Pi_0 \cup \{1, I\}$. We extend m^b in a homomorphic way to all the relational terms. Observe that:

(i) $m^b(1)$ is the universal relation on W^b (see (f2)).

(ii) $m^b(P)$ is an ideal relation for any $P \in V_R$ (see (b13)).

(iii) $m^b(I)$ is an equivalence relation on W^b (see (f3), (b11), (b12)).

Define the quotient structure $\mathcal{M}'^b = \mathcal{M}^b/m^b(I) = (W'^b, \{R'_\alpha : \alpha \in \Pi_0\}, R'_1, R'_I, m')$ such that

- $W'^b = \{|x|_{m^b(I)} : x \in V_I\}$ is the set of equivalence classes of $m^b(I)$
- $(|x|_{m^b(I)}, |y|_{m^b(I)}) \in R'_\alpha$ iff $(x, y) \in m^b(\alpha)$ for $\alpha \in \Pi_0 \cup V_R \cup \{1, I\}$
- m'^b is defined as in Definition 5.1

Definition of R'_α is correct (does not depend on the choice of elements from the respective equivalence classes), since due to (b9) and (b10) we have $m^b(I); m^b(\alpha) = m^b(\alpha) = m^b(\alpha); m^b(I)$. In $\mathcal{M}'^b$ the constant I is interpreted as the identity relation since we have $|x|_{m^b(I)} = |y|_{m^b(I)}$ iff $(x, y) \in m^b(I)$ iff $(|x|_{m^b(I)}, |y|_{m^b(I)}) \in R'_I$. We conclude that $\mathcal{M}'^b$ is a model of RelPDL$(;;, ||, ^{d(*)})$. Moreover for every term A we have $(|x|_{m^b(I)}, |y|_{m^b(I)}) \in m'^b(A)$ iff $(x, y) \in m^b(A)$. Hence the structures $\mathcal{M}^b$ and $\mathcal{M}'^b$ are elementary equivalent, and we can treat $\mathcal{M}^b$ as a model of RelPDL$(;;, ||, ^{d(*)})$.

Let v^b be a valuation in $\mathcal{M}^b$ such that $v^b(x) = x$ for every individual variable x. We define a well-founded ordering relation $\prec$ in the set of relational terms. For each relational term A we denote by $size(A)$ the number of relational operators occurring in A. We also write $subterm(A)$ the set of subterms of the relational term A recursively defined as follows:

- For $A_1 \in V_R \cup \Pi \cup \{I, 1\}$, $subterm(A_1) = \{A_1\}$
- For $\sharp \in \{\cup, ;, ||, ;;\}$, $subterm(A_1 \sharp A_2) = \{A_1 \sharp A_2\} \cup subterm(A_1) \cup subterm(A_2)$
- $subterm(-A_1) = \{-A_1\} \cup subterm(A_1)$; For $\sharp \in \{*, d(*)\}$, $subterm(A_1^\sharp) = \{A_1^\sharp\} \cup subterm(A_1)$

For any relational term A we define $N_1(A), N_2(A), N_3(A)$ and $\Delta(A)$ as follows:

- $N_1(A) = max\{size(A'^{d(*)}) \mid A'^{d(*)} \in subterm(A)\}$
- $N_2(A) = max\{size(A'^{*}) \mid A'^{*} \in subterm(A)\}$
- $N_3(A) = max\{size(A_1;;A_2) \mid A_1;;A_2 \in subterm(A)\}$
- $\Delta(A) = (N_1(A), N_2(A), size(A), N_3(A))$

We write $<_4$ to denote the left-right lexicographical order of ω^4 that is a well-founded order. We define for any relational terms A, B, $A \prec B$ iff $\Delta(A) <_4 \Delta(B)$. It follows that $\prec$ is well-founded. As a consequence, there are no terms A, B such that $A \prec B$ and $B \prec A$, otherwise there would exist an infinite chain $A \succ B \succ A \succ B \succ \ldots$.

We will show that:

not $\mathcal{M}^b, v^b$ sat $xA'y$

For suppose conversely, and let X^b be the set of formulae zBt on b such that $\mathcal{M}^b, v^b$ sat zBt. X^b is non-empty since $xA'y \in X^b$. Let C be a term of a minimal order such that $uCw \in X^b$ for some variables u, w. We show that C must belong to $V_R \cup \Pi_0 \cup \{1, I\}$.

- Suppose that C is of the form $u - Pw$ with $P \in V_R \cup \Pi_0 \cup \{1, I\}$. So $\mathcal{M}^b, v^b$ sat $u - Pw$ and from the definition of m^b $uPw \in b$. It follows that there is a node in b such that both $u - Pw$ and uPw occur among the formulae in this node which is in contradiction with the fact that b is not closed. Indeed if uPw (resp. $u - Pw$) occurs in a sequence of formulae in a node, then all the sequences occurring in the successors of this node contain uPw (resp. $u - Pw$) as well.
- Suppose that C is of the form $u - (A \cup B)w$. So $\mathcal{M}^b, v^b$ sat $u - (A \cup B)w$ and hence $\mathcal{M}^b, v^b$ sat $u - Aw$ and $\mathcal{M}^b, v^b$ sat $u - Bw$. From (b3) either $u - Aw \in b$ or $u - Bw \in b$. Since $-A \prec -(A \cup B)$ and $-B \prec -(A \cup B)$ it leads to a contradiction with the minimality of C.
- Suppose that C is of the form $u - (A||B)w$. So $\mathcal{M}^b, v^b$ sat $u - (A||B)w$ and hence either for all $z' \in W^b$ $\mathcal{M}^b, v^b$ sat $u - Az'$, or for all $z' \in W^b$ $\mathcal{M}^b, v^b$ sat $u - Bz'$. From (b15), for some $z, t \in W^b$, $u - Az, u - Bt \in b$. In particular $\mathcal{M}^b, v^b$ sat $u - Az$, or $\mathcal{M}^b, v^b$ sat $u - Bt$. Hence either $u - Az \in X^b$ or $u - Bt \in X^b$. Since $-A \prec -(A||B)$ and $-B \prec -(A||B)$ it leads to a contradiction from the minimality of C.

- Suppose that C is of the form $u-(A;;B)w$. So $\mathcal{M}^b, v^b$ sat $u-(A;;B)w$ and either $\mathcal{M}^b, v^b$ sat $u-(A;B)w$ (C1), or for some $z_0 \in W^b$ $\mathcal{M}^b, v^b$ sat uAz_0 and for all $t \in W^b$ $\mathcal{M}^b, v^b$ sat z_0-Bt (C2). (C1) is equivalent to for all $t' \in W^b$ either $\mathcal{M}^b, v^b$ sat $u-At'$ or $\mathcal{M}^b, v^b$ sat $t'-Bw$. From (b17), for all $t_1, t' \in W^b$ and for some $z_1 \in V_I$ either $u-Az_1, z_1-Bw, uAt' \in b$ or $u-Az_1, z_1-Bw, t'-Bt_1 \in b$.

  Suppose (C1) holds. There is $z_1 \in W^b$ such that $u-Az_1, z_1-Bw \in b$ -(b5)- and either $\mathcal{M}^b, v^b$ sat $u-Az_1$ or $\mathcal{M}^b, v^b$ sat z_1-Bw. Since $-A \prec -(A;;B)$ and $-B \prec -(A;;B)$ it leads to a contradiction from the minimality of C.

  Now suppose (C2) holds. For $t_1 \in W^b$, there exists $z_1 \in W^b$, such that either $u-Az_1, z_1-Bw, uAz_0 \in b$ (C3) or $u-Az_1, z_1-Bw, z_0-Bt_1 \in b$ (C4) -t' is instanciated with z_0. Suppose (C3) holds. We have both $uAz_0 \in b$ and $\mathcal{M}^b, v^b$ sat uAz_0. Since $A \prec -(A;;B)$ it leads to a contradiction from the minimality of C. Now suppose (C4) holds. We have both $z_0-Bt_1 \in b$ and $\mathcal{M}^b, v^b$ sat z_0-Bt_1 -from (C2). Since $-B \prec -(A;;B)$ it leads to a contradiction from the minimality of C.

- Suppose that C is of the form $u-A^{d(*)}w$. So $\mathcal{M}^b, v^b$ sat $u-A^{d(*)}w$ and hence either there is some $j \in \omega$ such that $\mathcal{M}^b, v^b$ sat $u-(A^{d(j)};1)w$ or $\mathcal{M}^b, v^b$ sat $u-A^*w$. From (b19), for all $k \in \omega$ $u-A^{d(k)};1w \in b$ and $x-A^*y \in b$. In particular, $u-A^{d(j)};1w \in b$. Since $-A^{d(j)};1 \prec -A^{d(*)}$ and $-A^* \prec -A^{d(*)}$ it leads to a contradiction from the minimality of C.

- Suppose that C is of the form $u(A||B)w$. So $\mathcal{M}^b, v^b$ sat $u(A||B)w$ and for some $t_1 \in W^b$ $\mathcal{M}^b, v^b$ sat uAt_1, for some $t_2 \in W^b$ $\mathcal{M}^b, v^b$ sat uBt_2 and either $\mathcal{M}^b, v^b$ sat uAw or $\mathcal{M}^b, v^b$ sat uBw. From (b14), for all $z, t \in V_I$ either $uAz \in b$ or $uBt \in b$ or $uAw \in b$ and $uBw \in b$. In particular, either $uAt_1 \in b$ or $uBt_2 \in b$. Since $A \prec (A||B)$ and $B \prec (A||B)$ it leads to a contradiction from the minimality of C.

- Suppose C is of the form $uA^{d(*)}w$ ($\in X^b$). So $\mathcal{M}^b, v^b$ sat $uA^{d(*)}w$ and hence $\mathcal{M}^b, v^b$ sat uA^*w and for all $i \in \omega$, $\mathcal{M}^b, v^b$sat $uA^{d(i)};1w$. As a consequence for all $i \in \omega$ there is z_0 such that $\mathcal{M}^b, v^b$sat $uA^{d(i)}z_0$ and $\mathcal{M}^b, v^b$sat $z_0 1w$. From (b18) either (a) $uA^*w \in b$ or (a$'$) for some l, $uA^{d(l)};1w \in b$. When (a$'$) holds, from (b4) for every $z \in V_I$ either (b) $uA^{d(l)}z \in b$ or (c) $z1w \in b$. If (a) holds then since $A^* \prec A^{d(*)}$ we obtain a contradiction with the minimality of C. If (b) holds, then since $A^{d(l)} \prec A^{d(*)}$ we again obtain a contradiction. If (c) holds then branch b would be closed, a contradiction.

In the remaining cases the proof is similar. It follows that C is of the form uPw with $P \in V_R \cup \Pi \cup \{1, I\}$. So $\mathcal{M}^b, v^b$ sat uPw by definition of X^b and not $\mathcal{M}^b, v^b$ sat uPw by definition of m^b which obviously leads to a contradiction.

A verification of part ($\leftarrow$) can be easily obtained from Lemma 6.1. Q.E.D.

Example In Figure 5 we give a relational proof of axiom (D12). The corresponding relational term is $t([\alpha^*]A \Rightarrow [\alpha^{d(*)}]A) = --(\alpha^*;-A) \cup -(\alpha^{d(*)};-A)$. For the sake of simplicity we denote $t(A)$ by A.

7 Conclusion

A logic of programs PDL($;;, ||, {}^{d(*)}$) has been defined admitting demonic operators as program constructors. These operators are of special interest when nontermination of a program α is represented by the local nonseriality of the corresponding binary relation R_α.

$$
\begin{array}{l}
x - -(\alpha^*; -A) \cup -(\alpha^{d(*)}; -A)y \\
\hline
x - -(\alpha^*; -A)y, x - (\alpha^{d(*)}; -A)y \quad (\cup) \\
\hline
x(\alpha^*; -A)y, x - (\alpha^{d(*)}; -A)y \quad (--) \\
\hline
x(\alpha^*; -A)y, x - \alpha^{d(*)} z_1, z_1 - -Ay \quad (-;) \text{ with the restricted variable } z_1 \\
\hline
x(\alpha^*; -A)y, x - \alpha^{d(*)} z_1, z_1 Ay \quad (--)
\end{array}
$$

$$
\begin{array}{l}
\hline
x\alpha^* z_1, x - \alpha^{d(*)} z_1, z_1 Ay \quad (;) \text{ with the variable } z_1 \\
\hline
x\alpha^* z_1, x - \alpha^* z_1, x - (I;1) z_1, z_1 Ay \quad (-d(*)) \text{ with } i = 0 \\
\text{closed}
\end{array}
\qquad
\begin{array}{l}
z_1 - Ay, x - \alpha^{d(*)} z_1, z_1 Ay \\
\text{closed}
\end{array}
$$

Figure 5: Relational proof of (D12)

A demonic iteration operator has been included in the language of the logic. It has been motivated by the following assumption. If a command α in a nondeterministic program P is executed a nondeterministic number of times and if one of its executions does not terminate then the whole program does not terminate. Other demonic iteration operators have been also discussed.

A Hilbert-style proof system has been defined and proved to be complete and sound for the logic PDL(; ;, ||, $^{d(*)}$). Decidability of PDL(; ;, ||, $^{d(*)}$) has also been proved using the filtration construction. In the second part of our work, we have applied the methodology developed in [Orło88, Orło92] in order to define the underlying relational logic for PDL(; ;, ||, $^{d(*)}$). A sound and complete Rasiowa-Sikorski proof system has been defined for the relational logic RelPDL(; ;, ||, $^{d(*)}$).

Some open problems are the following:

- To find a complete axiomatization of $^{d(*)}$ without using *
- To find a complete axiomatization of the iteration operators $^\bullet$ and *

References

[BZ86] R. Berghammer and H. Zierer. Relational algebraic semantics of deterministic and nondeterministic programs. *Theoretical Computer Science*, 43:123–147, 1986.

[FL79] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.

[Gab72] D. Gabbay. A general filtration method for modal logics. *Journal of Philosophical Logic*, 1:29–34, 1972.

[Gol92] R. I. Goldblatt. *Logics of Time and Computation.* Lecture Notes 7, CSLI Standford, 2d edition, 1992.

[HC84] G. Hughes and M. Cresswell. *A companion to modal logic.* Methuen, London and New York, 1984.

[HHJ+87] C. Hoare, I. J. Hayes, He Jifeng, C. C. Morgan, A. W. Roscoe, J. W. Sanders, I. H. Sorensen, J. M. Spivey, and B. A. Sufrin. Laws of programming. *Communications of the ACM*, 30:672–686, 1987.

[Mak66] D. Makinson. On some completeness theorems in modal logic. *The Journal of Symbolic Logic*, 12:379–384, 1966.

[Mir77a] G. Mirkowska. Algorithmic logic and its application in the theory of programs (I). *Annales Societatis Mathematicae Polonae. Series IV: Fundamenta Informaticae I.*, 1–17, 1977.

[Mir77b] G. Mirkowska. Algorithmic logic and its application in the theory of programs (II). *Annales Societatis Mathematicae Polonae. Series IV: Fundamenta Informaticae I.*, 147–165, 1977.

[Ngu91] T. Nguyen. A relational model of demonic nondeterministic programs. *International Journal of Foundation of Computer Science*, 2(2):101–131, 1991.

[Ngu93] T. Nguyen. The connection between predicate logic and demonic relation calculus. In *Schloss Dagstuhl International Seminar 9403 on Relational Methods in Computer Science*, 1993.

[Orło88] E. Orłowska. Relational interpretation of modal logics. In H. Andreka, D. Monk, and I. Nemeti, editors, *Algebraic logic. Colloquia Mathematica Societatis Janos Bolyai 54*, 443–471, Amsterdam, 1988. North Holland.

[Orło90] E. Orłowska. Algebraic aspects of the relational knowledge representation: modal relation algebras. In D. Pearce and H. Wansing, editors, *Nonclassical Logics and Information Processing*. Springer-Verlag, LNAI 619, 1990.

[Orło91] E. Orłowska. Relational proof systems for some AI logics. In Ph. Jorrand and J. Kelemen, editors, *FAIR'91*, 33–47. Springer-Verlag, LNAI 535, 1991.

[Orło92] E. Orłowska. Dynamic logic with program specifications and its relational proof system. *International Journal of Applied Non-Classical Logics*, 1992.

[Pra76] V. R. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings of the 17th IEEE Symposium on Foundations of Computer Science*, 109–121, 1976.

[RS63] H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*. PWN-Polish Scientific Publishers, Warsaw, 1963.

[Sal70] A. Salwicki. Formalized algorithmic languages. *Bulletin of the PAS 18, Ser. Math.*, 227–232, 1970.

[Seg82] K. Segerberg. A completeness theorem in the modal logic of programs. In T. Traczyk, editor, *Universal algebra and applications*, 31–46. Banach Center Publications, Volume 9 PWN - Polish Scientific Publishers, Warsaw, 1982.

[vB84] J. van Benthem. Correspondence Theory. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Volume II*, 167–247. Reidel, Dordrecht, 1984.

APPENDIX

Proof of Lemma 6.3

The key problem is to find a systematic procedure which will guarantee that any tree T constructed according to the procedure is such that if the procedure runs infinitely, every branch will have to be complete. At the initial step of the procedure, the tree T is composed of a unique node labeled by the sequence S. At each step a rule is applied to a leaf of T such that the corresponding sequence is not fundamental. At any step the height of the tree is finite and the number of leaves is finite as well. To each node v of the tree T we associate a unique sequence of natural numbers, noted $\sigma(v)$, such that

1. The sequence for the root of T is the empty sequence Λ.
2. If $\{v_0, v_1, \ldots, v_i, \ldots\}$ are the children of the node v in T then $\sigma(v_j) = \sigma(v), j$.

Considering that V_I is denumerable, we shall use the following 1-1 functions:

1. $\phi : \omega \to \omega^*$ (ω^* is the set of all the finite sequences of natural numbers)
2. $\phi_1 : \omega \to V_I$
3. $\phi_2 : \omega \to V_I \times V_I$

These functions enable us to enumerate leaves of T, individual variables and pairs of individual variables. If all the leaves of T are closed, then the procedure stops. Otherwise, let N be the smallest natural number such that $\phi(N)$ is the sequence that is associated with a leaf of T that is not closed. Such a leaf is unique and a rule is then applied to $\phi(N)$. The application of the rules is made in a fair way so that it is not the case that when two rules can be applied in a sequence, one of them is applied infinitely many times.

We build a tree $\mathcal{T} = (\mathcal{V}, \mathcal{R}, r)$ step by step. $\mathcal{V}$ is a set of nodes, $\mathcal{R}$ is a binary relation on $\mathcal{V}$ and $r \in \mathcal{V}$ is the root. To guide the application of the rules, with each node $v \in \mathcal{V}$ we associate the following information:

- $\Sigma(v)$: the sequence of formulas occurring in the node v
- $\sigma(v)$: the sequence of natural numbers which can be seen as the *address* of v in $\mathcal{T}$
- $Var(v)$: the set of variables from V_I that occur in the formulas that appear in the nodes between the root and the node v
- $Ind(v)$: the index of the next rule to be applied. To each rule r_i corresponds an index i for $i \in \{1, \ldots, 20\}$.
- $L_{\cup}(v)$ (respectively $L_{-\cup}(v)$,$L_{\cap}(v)$,$L_{-\cap}(v)$,$L_{--}(v)$, $L_{-;}(v)$,$L_{-*}(v)$,$L_{d(*)}(v)$,$L_{-||}(v)$) is a finite list $(e_1, \ldots, e_n)$ such that each e_i is a formula of the form $x(A \cup B)y$ (respectively $x{-}(A{\cup}B)y$,$x(A{\cap}B)y$,$x{-}(A{\cap}B)y$,$x{--}Ay$, $x{-}(A; B)y$,$x{-}A^*y$,$xA^{d(*)}y$,$x{-}(A \mid\mid B)y$).
- $L_{;}(v)$ (respectively $L_{ideal}(v)$,L_{I1},L_{I2},$L_*(v)$,$L_{-d(*)}(v)$,$L_{||}(v)$, $L_{;;}(v)$,$L_{-;;}(v)$) is a finite list $(e_1, \ldots, e_n)$ such that each e_i is a pair (f, j) such that $j \in \omega$ and f is a formula of the form $x(A; B)y$ (respectively xAy with $A \in V_R$, xAy with $A \in V_R$,xAy with $A \in V_R$, xA^*y, $x - A^{d(*)}y$, $x(A \mid\mid B)y$, $x(A;;B)y$,$x - (A;;B)y$).
- L_{SymI} and L_{TranI} are lists of formulas of the form xIy.

- $App(v)$ is either NIL or a name of the rule that is applied to the node v

The set of all the sets of the form $L_\epsilon(v)$ is denoted by $L(v)$ where

$$\epsilon \in \{-\cup, -\cap, --, -;, -*, d(*), -||, ideal, *, -d(*), ||, ;;, -;;, I1, I2, SymI, TranI\}$$

Let S be a sequence of formulas of the following form: $x_1 t_1 y_1, \ldots, x_K t_K y_K$. The function $build-complete-tree$ builds a complete tree (it may run infinitely) for the initial sequence S. For a list $l = (a_1, \ldots, a_n)$ we denote by $cdr(l)$ the sublist $(a_2, \ldots, a_n)$ (if $n = 0$ then it is the empty list NIL). Moreover, we denote by $queue(l, a_{n+1})$ the list $(a_1, \ldots, a_n, a_{n+1})$. According to the standard notation, $car(l) = a_1$. The figures 6 and 7 contain auxiliary functions.

FUNCTION $Update\ (L', (xty))$
INPUT: L': set of lists, xty: formula
OUTPUT: L the updated set of lists %

BEGIN
$L \leftarrow copy(L')$;
IF t has the form $A \cup B$ THEN $L_\cup \leftarrow queue(L_\cup, x\ t\ y)$ ENDIF;
IF t has the form $-(A \cup B)$ THEN $L_{-\cup} \leftarrow queue(L_{-\cup}, x\ t\ y)$ ENDIF;
IF t has the form $A \cap B$ THEN $L_\cap \leftarrow queue(L_\cap, x\ t\ y)$ ENDIF;
IF t has the form $-(A \cap B)$ THEN $L_{-\cap} \leftarrow queue(L_{-\cap}, x\ t\ y)$ ENDIF;
IF t has the form $--A$ THEN $L_{--} \leftarrow queue(L_{--}, x\ t\ y)$ ENDIF;
IF t has the form $-(A; B)$ THEN $L_{-;} \leftarrow queue(L_{-;}, x\ t\ y)$ ENDIF;
IF t has the form $-A^*$ THEN $L_{-*} \leftarrow queue(L_{-*}, x\ t\ y)$ ENDIF;
IF t has the form $A^{d(*)}$ THEN $L_{d(*)} \leftarrow queue(L_{d(*)}, x\ t\ y)$ ENDIF;
IF t has the form $-A\ ||\ B$ THEN $L_{-||} \leftarrow queue(L_{-||}, x\ t\ y)$ ENDIF;
IF t has the form $A; B$ THEN $L_; \leftarrow queue(L_;, ((x\ t\ y), 0))$ ENDIF;
IF $t \in V_R$ THEN
$L_{ideal} \leftarrow queue(L_{ideal}, ((x\ t\ y), 0))$; $L_{I1} \leftarrow queue(L_{I1}, ((x\ t\ y), 0))$;
$L_{I2} \leftarrow queue(L_{I2}, ((x\ t\ y), 0))$; ENDIF;
IF t has the form A^* THEN $L_* \leftarrow queue(L_*, ((x\ t\ y), 0)$ ENDIF;
IF t has the form $-A^{d(*)}$ THEN $L_{-d(*)} \leftarrow queue(L_{-d(*)}, ((x\ t\ y), 0)$ ENDIF;
IF t has the form $A\ ||\ B$ THEN $L_{||} \leftarrow queue(L_{||}, ((x\ t\ y), 0))$ ENDIF;
IF t has the form $(A; ; B)$ THEN $L_{;;} \leftarrow queue(L_{;;}, ((x\ t\ y), 0))$ ENDIF;
IF t has the form $-(A; ; B)$ THEN $L_{-;;} \leftarrow queue(L_{-;;}, ((x\ t\ y), 0))$ ENDIF;
IF $t = I$ THEN $L_{TranI} \leftarrow queue(L_{TranI}, xIy)$; $L_{SymI} \leftarrow queue(L_{SymI}, xIy)$ ENDIF;
RETURN (L)
END

Figure 6: Procedure *Update*

FUNCTION $build-complete-tree\ (S)$
INPUT: S: a formula sequence
OUTPUT: $\mathcal{T}$ a tree %

BEGIN
Generate a new node r; $\mathcal{T} \leftarrow (\{r\}, \emptyset, r)$; $L \leftarrow \{r\}$; $\Sigma(r) \leftarrow S$; $\sigma(r) \leftarrow \Lambda$; $Var(r) \leftarrow \{x_1, y_1, \ldots, x_K, y_K\}$;
$Ind(v) \leftarrow 1$;
All the $L_\epsilon(r)$ are initialized to NIL; $Auxi \leftarrow 1$;
WHILE $(Auxi < K)$

Do $L(r) \leftarrow Update(L(r), x_{Auxi} t_{Auxi} y_{Auxi})$; $Auxi \leftarrow Auxi + 1$
Done
While $[L \neq \emptyset$ Or $\exists v \in \mathcal{V}, App(v) \in \{(d(*)), (-(*))\}]$
Do $N_1 \leftarrow \{(n, v) \mid \exists v \in L, \sigma(v) = \phi(n), n \in \omega\}$; $N_2 \leftarrow \{(n, v) \mid v \in \mathcal{V}, Condition1(T, v, n)\}$;
  (N, V) is defined as follows, $N = min(\{n \mid (n, v) \in N_1 \cup N_2\})$ and $(N, V) \in (N_1 \cup N_2)$;
  % Rules without the introduction of variables
  For ϵ In $\{(\cup), (\neg\cup), (\cap), (\neg\cup), (--), (SymI)\}$ Do If $r_{Ind(V)} = \epsilon$ And $L_\epsilon(V) \neq$ Nil Then
    $Children \leftarrow 1$;
    For Each sequence S' generated by applying ϵ on $\Sigma(V)$ with $car(L_\epsilon(V))$
    Do Generate a new node: W;
      $\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = S'$;
      $\sigma(W) \leftarrow \sigma(V), Children$; $Children \leftarrow Children + 1$;
      $L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V)$; $L_\epsilon(W) \leftarrow cdr(L_\epsilon(W))$;
      For Each $x'A'y' \in S'$ and $x'A'y' \notin \Sigma(V)$
      Do $L(W) \leftarrow Update(L(W), x'A'y')$;
      Done
      $App(W) \leftarrow \emptyset$; $App(V) \leftarrow \epsilon$; $Ind(W) \leftarrow 1 + Ind(V)$ mod 20;
      If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
    Done
    $L \leftarrow (L \setminus \{V\})$;
  Done
  % Rules with the introduction of a unique arbitrary variable
  If $r_{Ind(V)} = (ideal)$ And $L_{ideal}(V) \neq$ Nil
  Then $F \leftarrow car(car(L_{ideal}(V)))$; $z \leftarrow \phi_1(car(cdr(car(L_{ideal}(V)))))$;
    (F is of the form xAy with $A \in V_R$)
    Let S' be the sequence generated from $\Sigma(V)$ by applying (ideal) with F and z;
    Generate a new node: W;
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = S'$; $\sigma(W) \leftarrow \sigma(V), \phi_1^{-1}(z)$;
    $L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V) \cup \{z\}$;
    $L_{ideal}(W) \leftarrow queue(cdr(L_{ideal}(W)), (F, \phi_1^{-1}(z) + 1))$; $L(W) \leftarrow Update(L(W), xAz)$;
    $App(W) \leftarrow \emptyset$; $App(V) \leftarrow (ideal)$; $Ind(W) \leftarrow 1 + Ind(V)$ mod 20;
    If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
    $L \leftarrow (L \setminus \{V\})$;
  Endif

  The cases for $(;)$, $(*)$, (I1), (I2), (TranI) and $(-d(*))$ are very similar to the previous one.
  In particular with $(;)$ two new nodes are introduced.
  With $(*)$ and $(-d(*))$, the index are related to the number of (possibly demonic) compositions.

  % Rule with the introduction of a unique restricted variable
  If $r_{Ind(V)} = (-;)$ And $L_{-;}(V) \neq$ Nil
  Then $F \leftarrow car(L_{ideal}(V))$; $z \leftarrow \phi_1(min(\{k' \mid \phi_1(k') \notin Var(V)\}))$;
    Let S' be the sequence generated from $\Sigma(V)$ by applying $(-;)$ with the formula F and the variable z;
    Generate a new node: W; $\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = S'$;
    $\sigma(W) \leftarrow \sigma(V), 1$; $L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V) \cup \{z\}$;
    $L_{-;}(W) \leftarrow cdr(L_{-;}(W))$;
    For Each $x'A'y' \in S'$ and $x'A'y' \notin \Sigma(V)$
    Do $L(W) \leftarrow Update(L(W), x'A'y')$;
    Done
    $App(W) \leftarrow \emptyset$; $App(V) \leftarrow (-;)$; $Ind(W) \leftarrow 1 + Ind(V)$ mod 20;
    If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
    $L \leftarrow (L \setminus \{V\})$;
  Endif
  % Rule with the introduction of 2 arbitrary variables
  If $r_{Ind(V)} = (||)$ And $L_{||}(V) \neq$ Nil
  Then $F \leftarrow (car(L_{||}(V)))$; $(z, t) \leftarrow \phi_2(car(cdr(car(L_{||}(V)))))$;
    $Children \leftarrow 1$;
    For Each sequence S' generated by applying $(||)$ with F, z and t
    Do Generate a new node: W;

$\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = S'$;
$\sigma(W) \leftarrow \sigma(V), Children$; $Children = Children + 1$;
$L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V) \cup \{z, t\}$;
If F is not reintroduced in the sequence S'
Then $L_{||}(W) \leftarrow cdr(L_{||}(W))$
Else $L_{||}(W) \leftarrow queue(cdr(L_{||}(W)), (F, \phi_2^{-1}(z, t) + 1))$
Endif
For Each $x'A'y' \in S'$ and $x'A'y' \notin \Sigma(V)$
Do $L(W) \leftarrow Update(L(W), x'A'y')$;
Done
$App(W) \leftarrow \emptyset$; $App(V) \leftarrow (||)$; $Ind(W) \leftarrow 1 + Ind(V)\ mod\ 20$;
If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
Done
$L \leftarrow L \setminus \{V\}$;
Endif
% Rule with the introduction of 2 restricted variables
If $r_{Ind(V)} = (- ||)$ And $L_{-||}(V) \neq$ Nil
Then $F \leftarrow (car(L_{-||}(V)))$;
$z \leftarrow \phi_1(min(\{k' \mid \phi_1(k') \notin Var(V)\}))$; $t \leftarrow \phi_1(min(\{k' \mid \phi_1(k') \notin Var(V)$ and $k' > \phi_1^{-1}(z)\}))$;
$Children \leftarrow 1$;
For Each sequence S' generated by applying $(- ||)$ with F, z and t
Do Generate a new node: W;
$\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = S'$;
$\sigma(W) \leftarrow \sigma(V), Children$; $Children = Children + 1$;
$L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V) \cup \{z, t\}$; $L_{-||}(W) \leftarrow queue(cdr(L_{-||}(W)), F)$;
For Each $x'A'y' \in S'$ and $x'A'y' \notin \Sigma(V)$
Do $L(W) \leftarrow Update(L(W), x'A'y')$;
Done
$App(W) \leftarrow \emptyset$; $App(V) \leftarrow (- ||)$; $Ind(W) \leftarrow 1 + Ind(V)\ mod\ 20$;
If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
Done
$L \leftarrow L \setminus \{V\}$;
Endif
% Rule with the introduction of one arbitrary variable and one restricted variable
If $r_{Ind(V)} = (;;)$ And $L_{;;}(V) \neq$ Nil
Then $F \leftarrow car(car(L_{||}(V)))$;
$t \leftarrow \phi_1(car(cdr(car(L_{;;}(V)))))$; $z \leftarrow \phi_1(min(\{k' \mid \phi_1(k') \notin Var(V), k' > \phi_1^{-1}(t)\}))$; $Children \leftarrow 1$;
For Each S' generated by applying $(;;)$ with F, z and t
Do Generate a new node: W;
$\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = S'$;
$\sigma(W) \leftarrow \sigma(V), Children$; $Children = Children + 1$;
$L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V) \cup \{z, t\}$;
$L_{;;}(W) \leftarrow queue(cdr(L_{;;}(W)), (F, \phi_1^{-1}(t) + 1))$;
For Each $x'A'y' \in S'$ and $x'A'y' \notin \Sigma(V)$
Do $L(W) \leftarrow Update(L(W), x'A'y')$;
Done
$App(W) \leftarrow \emptyset$; $App(V) \leftarrow (;;)$; $Ind(W) \leftarrow 1 + Ind(V)\ mod\ 20$;
If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
Done
$L \leftarrow L \setminus \{V\}$;
Endif
% Rule with the introduction of one restricted variable and two arbitrary variables
If $r_{Ind(V)} = (-;;)$ And $L_{-;;}(V) \neq$ Nil
Then $F \leftarrow car(car(L_{-;;}(V)))$;
$(t, u) \leftarrow \phi_2(car(cdr(car(L_{;;}(V)))))$; $z \leftarrow \phi_1(min(\{k' \mid \phi_1(k') \notin Var(V), k' > \phi_2^{-1}((t, u))\}))$;
$Children \leftarrow 1$;
For Each sequence S' generated by applying $(-;;)$ with F, z, u and t
Do Generate a new node: W;
$\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = S'$;

$\sigma(W) \leftarrow \sigma(V), Children$; $Children \leftarrow Children + 1$;
$L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V) \cup \{z, t, u\}$;
$L_{-;;}(W) \leftarrow queue(cdr(L_{-;;}(W)), (F, \phi_2^{-1}((t, u)) + 1))$;
For Each $x'A'y' \in S'$ and $x'A'y' \notin \Sigma(V)$
Do $L(W) \leftarrow Update(L(W), x'A'y')$;
Done
$App(W) \leftarrow \emptyset$; $App - rule(V) \leftarrow (-;;)$; $Ind(W) \leftarrow 1 + Ind(V)\ mod\ 20$;
If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
Done
$L \leftarrow L \setminus \{V\}$;
Endif
% Infinitary rules
If $r_{Ind(V)} = (d(*))$ And $L_{d(*)}(V) \neq$ Nil And $(N, V) \in N_1$
Then $F \leftarrow car(L_{d(*)}(V)))$ (F is of the form $xA^{d(*)}y$ and $\Sigma(V)$ of the form K, F, H);
Generate a new node: W;
$\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) \leftarrow K, xA^*y, H$;
$\sigma(W) \leftarrow \sigma(V), 0$; $L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V)$;
$L_{d(*)}(W) \leftarrow cdr(L_{d(*)}(W))$; $L(W) \leftarrow Update(L(W), xA^*y)$;
$App(W) \leftarrow \emptyset$; $App(V) \leftarrow (d(*))$; $Ind(W) \leftarrow 1 + Ind(V)\ mod\ 20$;
If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
$L \leftarrow L \setminus \{V\}$
Endif

If $r_{Ind(V)} = (-*)$ And $L_{-*}(V) \neq$ Nil And $(N, V) \in N_1$
Then $F \leftarrow car(L_{-*}(V)))$ (F is of the form $x - A^*y$ and $\Sigma(V)$ of the form K, F, H);
Generate a new node: W;
$\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = K, x - Iy, H$;
$\sigma(W) \leftarrow \sigma(V), 0$; $L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V)$;
$L_{-*}(W) \leftarrow cdr(L_{-*}(W))$; $L(W) \leftarrow Update(L(W), x - Iy)$;
$App(W) \leftarrow \emptyset$; $App(V) \leftarrow (-*)$; $Ind(W) \leftarrow 1 + Ind(V)\ mod\ 20$;
If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
$L \leftarrow L \setminus \{V\}$
Endif

If $r_{Ind(V)} = (d(*))$ And $(N, V) \in N_2$
Then $F \leftarrow car(L_{d(*)}(V)))$ (F is of the form $xA^{d(*)}y$ and $\Sigma(V)$ of the form K, F, H);
Let α be the natural number such that $\sigma(V), \alpha = N$;
Generate a new node: W;
$\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = K, xA^{d(\alpha-1)}; 1y, H$;
$\sigma(W) \leftarrow \sigma(V), \alpha$; $L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V)$; $L_{d(*)}(W) \leftarrow cdr(L_{d(*)}(W))$;
$L(W) \leftarrow Update(L(W), XA^{d(\alpha-1)}; 1Y)$; $App(W) \leftarrow \emptyset$; $Ind(W) \leftarrow 1 + Ind(V)\ mod\ 20$;
If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
Endif

If $r_{Ind(V)} = (-*)$ And $(N, V) \in N_2$
Then $F \leftarrow car(L_{-*}(V)))$ (F is of the form $xA^{-*}y$ and $\Sigma(V)$ of the form K, F, H);
Let α be the natural number such that $\sigma(V), \alpha = N$;
Generate a new node: W;
$\mathcal{V} \leftarrow \mathcal{V} \cup \{W\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(V, W)\}$; $\Sigma(W) = K, x - A^\alpha y, H$;
$\sigma(W) \leftarrow \sigma(V), \alpha$; $L(W) \leftarrow L(V)$; $Var(W) \leftarrow Var(V)$;
$L_{-*}(W) \leftarrow cdr(L_{-*}(W))$; $L(W) \leftarrow Update(L(W), x\ - A^\alpha\ y)$;
$App(W) \leftarrow \emptyset$; $Ind(W) \leftarrow 1 + Ind(V)\ mod\ 20$;
If S' is not fundamental Then $L \leftarrow L \cup \{W\}$ EndIf;
Endif
If the rule $r_{Ind(V)}$ cannot be applied to $\Sigma(V)$
Then $Ind(V) \leftarrow 1 + Ind(V)\ mod\ 20$
Endif
Done
Return (T)

End

Let $\mathcal{T}$ be a tree obtained after a finite number of steps. Let n be a non-fundamental leaf of $\mathcal{T}$ on the branch b. The following facts can be easily shown:

(i) After at most $\phi^{-1}(\sigma(n))$ steps a rule shall be applied on n.

(ii) If the sequence S' can be inserted on the branch b, then after a finite number of steps, S' shall be inserted on b unless the branch b has been closed.

To prove (i) observe that $N \leq \phi^{-1}(\sigma(n))$. If $N = \phi^{-1}(\sigma(n))$ then (i) is proved. Otherwise after each step the set $\{i \mid \phi(i)$ is a non closed leaf and $i \geq \phi^{-1}(\sigma(n))\}$ strictly decreases until $\{i \mid \phi(i)$ is a non closed leaf and $i \geq \phi^{-1}(\sigma(n))\} = \{\phi^{-1}(\sigma(n))\}$. Indeed, when a rule is applied to a non-fundamental leaf n, then n is not anymore a leaf (see also the particular treatment of the infinitary rules). The proof of (ii) is by an easy verification knowing that the rules are applied in a fair way. This terminates the proof.
Q.E.D.

Function *Condition*1 (T, v, n)
input: T: tree, v:node, n:natural number
output: true or False %

Begin
If $App(v) \notin \{(d(*), (-*)\}$ Then Return (False) endIf;
If there is no $\alpha \in \omega$ such that $\phi(n) = \sigma(v), \alpha$ Then Return (False) endIf;
If there is no $\beta \in \omega$ and $v' \in \mathcal{V}$ such that $\sigma(v), \beta = \sigma(v')$ Then Return (False) endIf;
If $\{w \mid \phi(n) = \sigma(w), w \in \mathcal{V}\} \neq \emptyset$ Then Return (False) EndIf;
Return (True)
End

Figure 7: Procedure related to the application of infinitary rules

---