Fundamenta Informaticae 33 (1998) 1–22
IOS Press

Iteration-free PDL with Intersection: a Complete Axiomatization

Philippe Balbiani
Institut de recherche en informatique de Toulouse
118 route de Narbonne, 31062 Toulouse Cedex 4, France
balbiani@irit.fr

Dimiter Vakarelov
Department of Mathematical Logic with Laboratory for Applied Logic
Faculty of Mathematics and Computer Science, Sofia University
blvd James Bouchier 5, 1126 Sofia, Bulgaria
dvak@fmi.uni-sofia.bg

Abstract. This paper is devoted to the completeness issue of $\mathrm{PDL}_0^\cap$ – an iteration-free fragment of Propositional Dynamic Logic with intersection of programs. The trouble with $\mathrm{PDL}_0^\cap$ is that the operation of intersection is not modally definable. Using new techniques connected with rules for intersection and the notions of large and maximal programs, the paper demonstrates that the presented proof theory for $\mathrm{PDL}_0^\cap$ is complete for the standard Kripke semantics of $\mathrm{PDL}_0^\cap$.

Keywords: PDL with intersection, modal rules for intersection, maximal programs.

Introduction

Propositional Dynamic Logic – PDL, is one of the simplest applied modal logics designed for reasoning about the behaviour of programs (see [7], [24], [14]). PDL has various modifications and extensions with applications in different areas: PAL – Propositional Algorithmic Logic [19], DAL – Data Analysis Logic [6], BML – Boolean Modal Logic [9], RAL – Relative Accessibility Logic [20], logics for reasoning about knowledge [5], dynamic logic of information [23], deontic logic as a variant of dynamic logic [18], applications of PDL to linguistics [16], multi-dimensional dynamic logic [22], dynamic arrow logic [3] and so on.

A typical feature of PDL is that it is a multi-modal logic with an algebraic structure in the set of modalities: composition $\alpha;\beta$ ("sequential composition" of programs α and β), corresponding to the composition of the accessibility relations $R(\alpha)$ and $R(\beta)$; union $\alpha \vee \beta$ ("nondeterministic choice" or "disjunction" of programs α and β), corresponding to the union of $R(\alpha)$ and $R(\beta)$; star operation α^* ("nondeterministic iteration" of program α), corresponding to the reflexive and transitive closure of $R(\alpha)$; test $A?$ (an operation transforming the formula A into the program $A?$), corresponding to the partial identity relation in the subsets of the PDL-models in which the formula A is true.

One of the most interesting extensions of PDL is $\mathrm{PDL}^{\cap}$, i.e. PDL with intersection $\alpha \wedge \beta$ ("intersection" or "conjunction" of programs α and β), corresponding to the intersection of the accessibility relations $R(\alpha)$ and $R(\beta)$. The interest to introduce intersection lies in the fact that it formalizes some aspects of parallelism. Intersection appears also in many other applied modal logics quoted above.

Harel [13] proves that $\mathrm{PDL}^{\cap}$ with at least two deterministic programs is highly undecidable, while Danecki [4] proves that $\mathrm{PDL}^{\cap}$ with nondeterminstic programs is decidable. After Danecki's paper the problem of finding a good axiomatization of $\mathrm{PDL}^{\cap}$ remains open. Let us note that the difficulty of this problem is based on the fact that intersection of modalities is not modally definable in the language of PDL. A partial solution of this problem is found by Passy and Tinchev [21] in an extension of $\mathrm{PDL}^{\cap}$ with names – special propositional letters interpreted in the models in exactly one point (giving in this way a name of the corresponding point). In the named version of $\mathrm{PDL}^{\cap}$, intersection of modalities is definable but it is still an open problem if the named version of $\mathrm{PDL}^{\cap}$ is decidable. Balbiani and Fariñas del Cerro [2] present an axiomatization of a subsystem of $\mathrm{PDL}^{\cap}$ containing only composition and intersection, and prove its completeness by an adaptation of the subordination method introduced by Hughes and Cresswell [15]. Another completeness proof for this system is given by Balbiani [1] using a special modification of the "mosaic method" [17]. However it is not clear how to extend these completeness proofs to the full language of PDL with intersection, seeing that both proofs produce irreflexive models, while test and star are reflexive operations.

In this paper we attack the problem of axiomatization of iteration-free PDL with intersection – $\mathrm{PDL}_0^{\cap}$ – by a method using additional inference rules for intersection. Intersection rules for ordinary multi-modal languages have been introduced by the second author in [26]. In some ways they are similar to Gabbay's irreflexivity rule [8]. Marx and Venema [17] call such rules "non-sructural rules", while Goranko [11] names them "context dependent rules". The rule for intersection from [26] is based on the following idea (described in an unpublished manuscript [27]). Although intersection is not definable in ordinary quantifier-free multi-modal languages, it becomes definable in languages with propositional quantifiers. In a quantifier-free multi-modal language with intersection, we can imitate the rule for the universal quantifier only for the case of the definition of intersection. As a result we obtain the corresponding rule for intersection. The main effect of such rule is that it makes the canonical model (consisting of all maximal consistent sets closed under the rule) standard for the intersection. However the canonical model does not behave well with some existential conditions for the accessibility relations, because with such conditions one has to construct some special maximal consistent sets required by the condition, which in addition have to be closed under the rule of intersection. In general the construction of such maximal consistent sets is not always possible. In the case of PDL such a condition is connected with the semantics of composition: if $xR(\alpha;\beta)y$ then $(\exists z)(xR(\alpha)z$ and $zR(\beta)y)$. Thus the composition makes complications. Similar complications with composition arouse in some axiomatizations of PDL with an infinitary rule for the star operation (see [25], or in a similar situation [10]). Let us note that in the presence of an irreflexivity rule, as in the case of named extensions of PDL in [21], composition does not make complications due to some nice properties of Gabbay's irreflexivity rule, which are not possessed by the rule of intersection. In our case we overcome these complications by an idea suggested by the first author to use in the canonical model special constructs called "maximal programs". Like maximal consistent sets, which are special sets of formulas, maximal programs are special sets of programs, with a meaning to be explained later on in the paper.

The main achievement of the paper is not only the completeness theorem of a version of PDL with intersection but also a demonstration of a new technique connected with maximal programs and the rule of intersection. We hope to extend this technique to the full language of PDL with intersection, including the star operation and the converse of programs.

The structure of the paper is the following. In Section 1 we introduce the syntax and semantics of $\mathrm{PDL}_0^\cap$. Section 2 is based on [27] and here we discuss the definability of intersection in languages with quantifiers. For simplicity the presentation here is in an algebraic form in terms of set-theoretic dynamic algebras with intersection. Here we introduce the rule of intersection INT and prove its correctness. Section 3 is devoted to the axiomatic system for $\mathrm{PDL}_0^\cap$. In section 4 we introduce the notion of a theory and maximal theory (corresponding to the notion of a maximal consistent set) and we prove some basic facts for them, including the Lindenbaum Lemma and the Diamond Extension Lemma. Section 5 is devoted to the notion of a maximal program. In section 6 we introduce the canonical model for $\mathrm{PDL}_0^\cap$ and prove the completeness theorem. Section 7 is for some concluding remarks and open problems.

The results of this paper have been obtained during the visit of the second author in the Faculty of Computer Science at the University of Paris-Nord at Villetaneuse, which makes possible the collaboration between the two co-authors. Special thanks are due to the referees for some good suggestions improving the readability of the text and to Nathalie Chetcuti, Luis Fariñas del Cerro, Valentin Goranko, Dimiter Guelev, Andreas Herzig and Tinko Tinchev for some valuable comments and discussions.

1. Syntax and semantics

Syntax of $\mathrm{PDL}_0^\cap$ The language of $\mathrm{PDL}_0^\cap$ consists of a denumerable set Φ_0 of propositional variables and a denumerable set Π_0 of program variables and the following symbols:

- logical symbols: $\perp$ (logical falsehood).

• program symbols: ; (sequential composition), $\vee$ (disjunction), $\wedge$ (conjunction) and ? (test).

• parentheses (,), [,].

The set Φ of all formulas and the set Π of all programs are defined by the following parallel induction:

• All propositional variables and $\bot$ are formulas.

• If A is a formula and α is a program then $[\alpha]A$ is a formula.

• If α and β are programs and A is a formula then $(\alpha;\beta)$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$ and $A?$ are programs.

We adopt the standard rules for omission of the parentheses and define the implication as a special case of the box modality:

$A \Rightarrow B =_{\text{def}} [A?]B$.

Having the implication and the logical falsehood, we adopt the standard definitions for the remaining Boolean operations and for the diamond modality:

$\neg A =_{\text{def}} A \Rightarrow \bot$.

$\top =_{\text{def}} \neg\bot$.

$A \vee B =_{\text{def}} \neg A \Rightarrow B$.

$A \wedge B =_{\text{def}} \neg(A \Rightarrow \neg B)$.

$A \Leftrightarrow B =_{\text{def}} (A \Rightarrow B) \wedge (B \Rightarrow A)$.

$< \alpha > A =_{\text{def}} \neg[\alpha]\neg A$.

The notion of *subformula* is standard. The notion of *subprogram* is defined as follows:

• If α is a program variable or α is a test then the only subprogram of α is α.

• α is a subprogram of $\beta \circ \gamma$, $\circ \in \{;, \vee, \wedge\}$ if α coincides with $\beta \circ \gamma$ or α is a subprogram of β or α is a subprogram of γ.

Example: Let α and β be program variables. Then α is not a subprogram of $\gamma = [\alpha]A? \wedge \beta$, and the only subprograms of γ are γ, $[\alpha]A?$ and β.

Note that one and the same program may have different occurrences as a subprogram of another program. Program variables and tests occurring in a program α as subprograms are called atomic subprograms of α. Obviously α is built up from the set of its atomic subprograms by means of the program operations $\{;, \vee, \wedge\}$. Let $A(B)$ be an expression (program or formula) and B be a fixed expression which is a part of A. Then the result of the replacement of B in its place in A with another expression C will be denoted by $A(C)$.

Admissible forms

For the definition of the rule of intersection in the next section we will need programs of a special form, called admissible forms – AF. Let the language of $\text{PDL}_0^\cap$ be extended with a new propositional variable #. Each AF will have a natural number as a rank and the definition is by induction on the rank:

If $\alpha(\#?)$ is a program with unique occurrence of $\#?$ as a subprogram then $\alpha(\#?)$ is an AF of rank 1.

Let $\alpha(\#?)$ be an AF of rank 1, $\beta(\#?)$ be an AF of rank n and A be a formula not containing #. Then $\gamma(\#?) =_{\text{def}} \alpha(\neg[\beta(\#?)]A?)$ is an AF of rank n+1.

Note that in each AF α, $\#?$ has a unique occurrence, not in general as a subprogram of α. What is more, $\#?$ is a subprogram of α only if α is of rank 1.

Semantics of $\mathrm{PDL}_0^{\cap}$

To introduce the semantics of $\mathrm{PDL}_0^{\cap}$, we need the notion of a set-theoretical dynamic algebra, corresponding to the language of $\mathrm{PDL}_0^{\cap}$. Let $W \neq \emptyset$. Let $\mathcal{B}(W)$ denote the Boolean algebra of all subsets of W and $\mathcal{R}(W)$ denote the set of all binary relations in W. Then the pair $\mathcal{DA} =< \mathcal{B}(W), \mathcal{R}(W) >$ is called set-theoretical dynamic algebra over W if it is endowed with the following operations:

• All Boolean operations in the set $\mathcal{B}(W)$ of all subsets of W. For the sake of simplicity we use logical notations for Boolean operations. For instance the complement $W \setminus A$ will be denoted by $\neg A$, the union $A \cup B$ will be denoted by $A \vee B$ and so on.

• The following relational operations in the set $\mathcal{R}(W)$ of all binary relations in W $(S, T \subseteq W \times W)$:

$S;T =_{\mathrm{def}} \{(x, y)/(\exists z \in W)(xSz \& zTy)\}$ – composition of S and T.

$S \vee T =_{\mathrm{def}} S \cup T$ – union of S and T.

$S \wedge T =_{\mathrm{def}} S \cap T$ – intersection of S and T.

• We consider also two intersort operations ($A \subseteq W$ and $S \subseteq W \times W$):

$A? =_{\mathrm{def}} \{(x, x)/x \in A\}$ – test.

$[S]A =_{\mathrm{def}} \{x \in W/(\forall y \in W)(xSy \rightarrow y \in A)\}$ – box modality.

The diamond modality $< S > A$ is defined as usual:

$< S > A =_{\mathrm{def}} \neg[S]\neg A = \{x \in W/(\exists y \in W)(xSy \& y \in A)\}$.

We consider also infinitary unions and intersections in $\mathcal{B}(W)$ and $\mathcal{R}(W)$, which will be used in the next section.

PDL-models

Let $W \neq \emptyset$, R be a function from the set Π_0 of all program variables into the set $\mathcal{R}(W)$ of all binary relations in W and let V be a function from the set Φ_0 of all propositional variables into the set $\mathcal{B}(W)$ of all subsets of W. Then the triple $M = (W, R, V)$ is called a *PDL-model* and R and V are called valuations. We extend R and V to the set of all programs and to the set of all formulas by a parallel induction as follows:

• $V(\perp) =_{\mathrm{def}} \emptyset$.

• $V([\alpha]A) =_{\mathrm{def}} [R(\alpha)]V(A)$.

• $R(\alpha; \beta) =_{\mathrm{def}} R(\alpha); R(\beta)$.

• $R(\alpha \vee \beta) =_{\mathrm{def}} R(\alpha) \vee R(\beta)$.

• $R(\alpha \wedge \beta) =_{\mathrm{def}} R(\alpha) \wedge R(\beta)$.

• $R(A?) =_{\mathrm{def}} V(A)?$.

Having in mind this semantics and the definitions of the remaining Boolean operations and the diamond modality, we can see that their semantics have the intended meaning: $V(\neg A) = \neg V(A)$, $V(\top) = W$, $V(A \vee B) = V(A) \vee V(B)$, $V(A \wedge B) = V(A) \wedge V(B)$ and $V(< \alpha > A) = < R(\alpha) > V(A)$. A formula A is true in the model $M = (W, R, V)$ if $V(A) = W$. A formula A is called a PDL-tautology if A is true in all PDL-models. Let $x \in W$. The pair $< (W, R, V), x >$

is a model for a formula A if $x \in V(A)$. In this case we say also that A is true in x. The pair $<(W,R,V),x>$ is a model for a set Σ of formulas if $<(W,R,V),x)$ is a model for all formulas of Σ.

2. Modal definability of intersection and the rule of intersection

It is a well known fact that the semantics of intersection is not modally definable in the usual modal languages and also in the language of PDL. We will show that in the presence of propositional quantifiers it becomes definable. For simplicity of notations this will be done algebraically on the level of set-theoretical dynamic algebras. The formulas and programs of PDL$_0^\cap$ can be considered as expressions over dynamic algebras.

Proposition 2.1. Let $W \neq \emptyset$, $\mathcal{DA}(W)$ be the dynamic algebra ower W, S,T be binary relations in W and A be a subset of W. Then the following equality is true:

$<S \wedge T> A = \bigcap_{p \subseteq W}(<S>(A \wedge p) \vee <T>(A \wedge \neg p))$.

Proof. ($\subseteq$) Suppose $x \in <S \wedge T> A$, $p \subseteq W$ and proceed to show that $x \in (<S>(A \wedge p) \vee <T>(A \wedge \neg p))$. By the assumption there exists $y \in A$ such that xSy and xTy.

Case 1: $y \in p$. Then $y \in (A \wedge p)$, $x \in <S>(A \wedge p)$ and consequently $x \in (<S>(A \wedge p) \vee <T>(A \wedge \neg p))$.

Case 2: $y \in \neg p$. Then $y \in (A \wedge \neg p)$, $x \in <T>(A \wedge \neg p)$ and thus $x \in (<S>(A \wedge p) \vee <T>(A \wedge \neg p))$.

($\supseteq$) Suppose that $x \notin <S \wedge T> A$. We have to prove that there exists a $p \subseteq W$ such that $x \notin (<S>(A \wedge p) \vee <T>(A \wedge \neg p))$. Put: $p = \{y \in W / xTy\}$. Suppose, for the sake of contradiction that $x \in (<S>(A \wedge p) \vee <T>(A \wedge \neg p))$.

Case 1: $x \in <S>(A \wedge p)$. Then for some $y \in (A \wedge p)$ we have xSy. From here we obtain $y \in A$, $y \in p$ and, by the definition of the set p, xTy. Then we have $x(S \wedge T)y$ and $x \in <S \wedge T> A$ – a contradiction with the assumption.

Case 2: $x \in <T>(A \wedge \neg p)$. Then for some $y \in (A \wedge \neg p)$ we have xTy. From here we obtain $y \in A$, $y \in \neg p$ and, by the definition of the set p, $x\overline{T}y$ – a contradiction with xTy. ■

Corollary 2.2. The following PDL-formula with propositional quantifiers modally defines the semantics of intersection: $<\alpha \wedge \beta> A \Leftrightarrow (\forall p)(<\alpha>(A \wedge p) \vee <\beta>(A \wedge \neg p))$.

The implication from left to right can be expressed without propositional quantification: $<\alpha \wedge \beta> A \Rightarrow (<\alpha>(A \wedge p) \vee <\beta>(A \wedge \neg p))$. Assuming that p is not in A and substituting here first p with A and second p with $\neg A$ we obtain the following two formulas which we will take in the next section as axioms for the intersection:

- $<\alpha \wedge \beta> A \Rightarrow <\alpha> A$.
- $<\alpha \wedge \beta> A \Rightarrow <\beta> A$.

The implication from right to left cannot be expressed by a formula in our language but instead it will be imitated by a rule of inference. The simplest form of such a rule is the following one, the correctness of which follows immediately from proposition 2.1:

(INT_0) From $B \Rightarrow (< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))$ for all propositional variable p infer $B \Rightarrow < \alpha \wedge \beta > A$.

However, (INT_0) is not sufficient for the axiomatization of $PDL_0^{\cap}$. We will introduce a much more general rule which imitates the interaction of quantifiers with modal operators (like in the Barcan formula) and tests. For that purpose we will generalize proposition 2.1. First we will state some simple facts concerning infinitary intersections and unions in set-theoretical dynamic algebras.

Lemma 2.3. Let $\mathcal{DA}(W)$ be a dynamic algebra over W, I be an index set and let for any $i \in I$, A_i be a subset of W, S_i be a binary relation in W and α and β_i be programs considered as expressions ranging over $\mathcal{DA}(W)$. Then:

(i) $\neg \bigcap_{i \in I} A_i = \bigcup_{i \in I} \neg A_i$.

(ii) $(\bigcup_{i \in I} A_i)? = \bigcup_{i \in I} A_i?$.

(iii) $[\bigcup_{i \in I} S_i]A = \bigcap_{i \in I}[S_i]A$.

(iv) $\alpha \circ (\bigcup_{i \in I} \beta_i) = \bigcup_{i \in I}(\alpha \circ \beta_i)$, $\circ \in \{\vee, \wedge, ;\}$.

(v) $\alpha(\bigcup_{i \in I} \beta_i) = \bigcup_{i \in I} \alpha(\beta_i)$.

(vi) Let $\alpha(\#?)$ be an AF considered as an expression ranging over $\mathcal{DA}(W)$. Then $\alpha(\bigcup_{i \in I} A_i?) = \bigcup_{i \in I} \alpha(A_i?)$.

Proof. The proofs of (i)-(iv) follow directly from the definitions of the corresponding operations. The proof of (v) follows from (iv) by induction on the complexity of the program α. (vi) follows from (i)-(v) by induction on the rank of $\alpha(\#?)$, the basis of the induction is just (v). ■

Lemma 2.4. Let $W \neq \emptyset$, $\mathcal{DA}(W)$ be the dynamic algebra ower W, S, T be binary relations in W, A, B be subsets of W and $\gamma(\#)$ be an AF considered as an expression ranging over $\mathcal{DA}(W)$. Then the following equality is true:

$$[\gamma(\neg < S \wedge T > A?)]B = \bigcap_{p \subseteq W}[\gamma(\neg(< S > (A \wedge p) \vee < T > (A \wedge \neg p))?)]B.$$

Proof. By 2.1 we have: $< S \wedge T > A = \bigcap_{p \subseteq W}(< S > (A \wedge p) \vee < T > (A \wedge \neg p))$. Taking the complement we obtain: $\neg < S \wedge T > A = \bigcup_{p \subseteq W} \neg(< S > (A \wedge p) \vee < T > (A \wedge \neg p))$. Then by 2.3 (ii) we have: $(\neg < S \wedge T > A)? = \bigcup_{p \subseteq W} \neg(< S > (A \wedge p) \vee < T > (A \wedge \neg p))?$. Applying 2.3 (vi) and (iii) we get: $[\gamma(\neg < S \wedge T > A?)]B = \bigcap_{p \subseteq W}[\gamma(\neg(< S > (A \wedge p) \vee < T > (A \wedge \neg p))?)]B$ which proves the lemma. ■

The rule of intersection

From 2.4 we obtain the following more general rule of intersection, where $\gamma(\#?)$ is an AF.

(INT) From $[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B$ for all propositional variables p infer $[\gamma(\neg < \alpha \wedge \beta > A?)]B$.

Note that taking $\gamma(\#?) = \#?$ we obtain the rule (INT_0). Obviously, If C is a conclusion of (INT) then C can be represented in finitely many ways in the form $C = [\gamma(\neg < \alpha \wedge \beta > A?)]B$.

Example: Let $C = [\neg < \alpha_1 \wedge \beta_1 > A_1? \wedge \neg < \alpha_2 \wedge \beta_2 > A_2?]B$. Put $\gamma_1(\#?) = \#? \wedge \neg < \alpha_2 \wedge \beta_2 > A_2?$ and $\gamma_2(\#?) = \neg < \alpha_1 \wedge \beta_1 > A_1? \wedge \#?$. Then we have immediately $C = [\gamma_1(\neg < \alpha_1 \wedge \beta_1 > A_1?)]B = [\gamma_2(\neg < \alpha_2 \wedge \beta_2 > A_2?)]B$.

Lemma 2.5. Correctness of (INT).

The rule (INT) is correct with respect to the semantics of $\mathrm{PDL}_0^\cap$.

Proof. Immediately by 2.4. ■

3. Axiomatization

In this section we will propose an axiomatic system for $\mathrm{PDL}_0^\cap$.

Axiom schemes

(A0) All instances of classical tautologies.

(K) $[\alpha](A \Rightarrow B) \Rightarrow ([\alpha]A \Rightarrow [\alpha]B)$.

(;) $< \alpha; \beta > A \Leftrightarrow < \alpha >< \beta > A$.

($\vee$) $< \alpha \vee \beta > A \Leftrightarrow < \alpha > A \vee < \beta > A$.

($\wedge_1$) $< \alpha \wedge \beta > A \Rightarrow < \alpha > A$.

($\wedge_2$) $< \alpha \wedge \beta > A \Rightarrow < \beta > A$.

(Distr) $< \alpha \wedge (\beta \vee \gamma) > A \Leftrightarrow < (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) > A$.

(?) $< A? > B \Leftrightarrow (A \wedge B)$.

Rules of inference

(MP) Modus ponens: From A and $A \Rightarrow B$ infer B.

(N) Necessitation: From A infer $[\alpha]A$.

(INT) The rule of intersection: From $[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B$ for all propositional variables p infer $[\gamma(\neg < \alpha \wedge \beta > A?)]B$, where $\gamma(\#?)$ is an AF.

A formula A is a theorem of $\mathrm{PDL}_0^\cap$ if it belongs to the least set of formulas containing all axioms and closed under the rules.

Program inclusion and program equivalence.

We define the relations of *program inclusion* $\preceq$ and *program equivalence* $\equiv$ as follows:

$\alpha \preceq \beta$ iff $< \alpha > p \Rightarrow < \beta > p$ is a theorem of $\mathrm{PDL}_0^\cap$.

$\alpha \equiv \beta$ iff $< \alpha > p \Leftrightarrow < \beta > p$ is a theorem of $\mathrm{PDL}_0^\cap$.

In both cases p is a propositional variable not appearing in α, β.

Remarks

1. Let us note that the axioms ($\wedge_1$) and ($\wedge_2$) are equivalent, without the rule (INT), to the following formula: $< \alpha \wedge \beta > A \Rightarrow < \alpha > (A \wedge B) \vee < \beta > (A \wedge \neg B)$.

2. In the above formulation the rule (INT) is infinitary. It can be replaced by the following finitary one:

(INT_{fin}) From $[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B$ infer $[\gamma(\neg < \alpha \wedge \beta > A?)]B$, where $\gamma(\#?)$ is an AF and p is a propositional variable not occurring in γ, α, β, A, B.

We have the following lemma:

Lemma 3.1. The rules (INT) and (INT_{fin}) are equivalent in a sense that they are interchangable.

Proof. Obviously (INT_{fin}) implies (INT). For the converse, let us suppose that the premise $[\gamma(\neg(<\alpha>(A\wedge p)\vee<\beta>(A\wedge\neg p))?)]B$ of (INT_{fin}) is a theorem. Then substituting p with an arbitrary propositional variable q we again obtain a theorem. Since p is not in γ, α, β, A, B this theorem has the form $[\gamma(\neg(<\alpha>(A\wedge q)\vee<\beta>(A\wedge\neg q))?)]B$. Then by the rule (INT) we obtain the conclusion $[\gamma(\neg<\alpha\wedge\beta>A?)]B$. ■

We will use also the following special case of (INT_{fin}):

(INT_1) From $[\gamma](<\alpha>(A\wedge p)\vee<\beta>(A\wedge\neg p))$ infer $[\gamma]<\alpha\wedge\beta>A$, where p is a propositional variable neither occuring in A nor in the programs α, β, γ.

In order to demonstrate how the rule (INT) works we will give the proofs of some theorems of $\mathrm{PDL}_0^{\cap}$, which will be of later use.

Lemma 3.2. The following formulas are theorems of $\mathrm{PDL}_0^{\cap}$

(1) $<\alpha\wedge\beta>A\Rightarrow<\beta\wedge\alpha>A$.

(2) $<A?\wedge B?>C\Leftrightarrow<(A\wedge B)?>C$.

(3) $<A?\vee B?>C\Leftrightarrow<(A\vee B)?>C$.

(4) $<\alpha;(\beta\vee\gamma)>A\Leftrightarrow<(\alpha;\beta)\vee(\alpha;\gamma)>A$.

(5) $<\alpha(\beta\vee\gamma)>A\Leftrightarrow<\alpha(\beta)\vee\alpha(\gamma)>A$.

(6) $<\alpha(A?)>C\Leftrightarrow<\alpha((A\wedge B)?)>C\vee<\alpha((A\wedge\neg B)?)>C$.

Proof. (1) We mentioned that the axioms ($\wedge_1$) and ($\wedge_2$) together are equivalent without the rule (INT) to the following formula: $<\alpha\wedge\beta>A\Rightarrow<\alpha>(A\wedge B)\vee<\beta>(A\wedge\neg B)$. Then by the commutativity of the disjunction and taking B to be the negation of a propositional variable p not occurring in A, α, β we obtain: $<\alpha\wedge\beta>A\Rightarrow<\beta>(A\wedge p)\vee<\alpha>(A\wedge\neg p)$. Applying to this formula the rule (INT_0) we obtain the needed result: $<\alpha\wedge\beta>A\Rightarrow<\beta\wedge\alpha>A$.

(2) The implication $\Rightarrow$ follows without the rule of (INT) and we leave it to the reader. For the converse implication, let us note that the following formula is a theorem of the ordinary PDL: $<(A\wedge B)?>C\Rightarrow(<A?>(C\wedge p)\vee<B?>(C\wedge\neg p))$. Since p can be any propositional variable, then by an application of (INT_0), we obtain $<(A\wedge B)?>C\Rightarrow<A?\wedge B?>C$.

(3) and (4) are ordinary PDL-theorems.

(5) The proof is by induction on the complexity of α using (4), axiom (Distr) and some PDL-theorems concerning program disjunction.

(6) The proof is based on the fact that the program $A?$ in (6) is equivalent to the program $((A\wedge B)\vee(A\wedge\neg B))?$. This program by (3) is equivalent to the program $(A\wedge B)?\vee(A\wedge\neg B)?$ and the later can be substituted for $A?$. Then applying (5) we obtain the theorem. ■

Lemma 3.3. (i) The program equivalence is a congruence relation with respect to the program operations ($\equiv$ is stable with respect to replacement of equivalent programs).

(ii) If we consider $\equiv$ as equality then the set of all programs is a distributive lattice with respect to the program operations $\vee$ and $\wedge$. The ordering of the lattice coincides with the relation of program inclusion. This lattice has a smallest element $0=_{\mathrm{def}}\bot?$.

(iii) The operation of composition is associative and distributive with respect to the operation of disjunction, with unit $1=_{\mathrm{def}}\top?$ and zero $0=_{\mathrm{def}}\bot?$.

(iv) Each program is equivalent to a disjunction of programs not containing disjunction (*disjunctive normal form for programs*).

(v) Each formula is equivalent to a formula not containing the operation of program disjunction.

(vi) Let $\alpha(A?)$ be a program without program disjunction and containing a subprogram $A?$ such that $A? \equiv \bot?$. Then $\alpha \equiv \bot?$ and the formula $< \alpha > B \Leftrightarrow \bot$ is a theorem.

(vii) All program operations are monotonic with respect to program inclusion. If A, B are formulas then we have: $A \Rightarrow B$ is a theorem iff $A? \preceq B?$.

Proof. We left to the reader the proofs of (i)-(iii) and (vii) as an exercise. (iv) follows from the fact that composition ; and program conjunction $\wedge$ distribute with the operation of program disjunction $\vee$. (v) is a consequence of (iv) and the axiom ($\vee$) for program disjunction. (vi) follows from the fact that $\bot?$ is a zero for the operation of composition and the operation of conjunction. The last equivalence uses axiom (?) for the test and the fact that α can be replaced by $\bot?$. ■

We conclude this section by the soundness theorem for $\mathrm{PDL}_0^{\cap}$:

Proposition 3.4. If A is a theorem of $\mathrm{PDL}_0^{\cap}$ then A is true in all PDL-models.

Proof. Obviously the axioms of $\mathrm{PDL}_0^{\cap}$ are true in all PDL-models and the rules (MP) and (N) preserve validity. That the new rule (INT) preserves validity follows from 2.5. ■

4. Theories and quasi-canonical model

A set x of formulas is called a theory if it satisfies the following conditions:

(th 1) x contains the set of all theorems of $\mathrm{PDL}_0^{\cap}$.

(th 2) if $A \in x$ and $A \Rightarrow B \in x$ then $B \in x$ (x is closed under modus ponens).

(th 3) If $\gamma(\#?)$ is an AF and $[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B \in x$ for every propositional variable p then $[\gamma(\neg < \alpha \wedge \beta > A?)]B \in x$ (x is closed under the rule (INT)).

Obviously the smallest theory is the set TH of all theorems and the greatest theory is the set of all formulas. The later theory is called trivial theory. A theory x is called consistent if $\bot \notin x$, otherwise it is called inconsistent. It is a well known fact that a theory x is consistent iff it is not trivial and that x is inconsistent if it contains a formula A together with its negation $\neg A$. A theory x is called a maximal theory if it is consistent and for any formula A: $A \in x$ or $\neg A \in x$. A set of formulas Σ is called consistent if it is contained in a consistent theory. It can be shown that a single formula A is consistent (considered as a singleton $\{A\}$), iff it is not equivalent to $\bot$.

Remark. In the literature instead of maximal theory the notion of a maximal consistent set is used, where consistency is defined without using the notion of theory. It can be proved that each maximal theory is a maximal consistent set in the classical sense, and each maximal consistent set which is closed under the rule of (INT) is a maximal theory. We will use the following properties of maximal theories without explicit reference (x is a maximal theory):

Fact 4.1. (i) $\neg A \in x$ iff $A \notin x$,

(ii) $A \wedge B \in x$ iff $A \in x$ and $B \in x$,

(iii) $A \vee B \in x$ iff $A \in x$ or $B \in x$,

Let x be a set of formulas and δ be a program. Define $[\delta]x =_{\text{def}} \{A/[\delta]A \in x\}$. If A is a formula then define $x + A =_{\text{def}} \{B/A \Rightarrow B \in x\}$. Obviously $x + A = [A?]x$. In the next lemma we summarize some properties of theories.

Lemma 4.2. (i) If x is a theory then $[\delta]x$ is a theory too.

(ii) $x + A$ is the smallest theory containing A and x.

(iii) $x + A$ is inconsistent iff $\neg A \in x$.

(iv) Let x be a consistent theory, $\gamma(\#?)$ be an AF and $\neg[\gamma(\neg < \alpha \wedge \beta > A?)]B \in x$. Then there exists a propositional variable p such that $x + \neg[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B$ is consistent.

Proof. (i) Let A be a theorem. Then by the rule (N), $[\delta]A$ is a theorem and hence $[\delta]A \in x$, so $A \in [\delta]x$. Let $A \in [\delta]x$ and $A \Rightarrow B \in [\delta]x$. Then $[\delta](A \Rightarrow B) \in x$ and $[\delta]A \in x$. By the axiom (K), $[\delta](A \Rightarrow B) \Rightarrow ([\delta]A \Rightarrow [\delta]B) \in x$. Applying two times modus ponens we obtain that $[\delta]B \in x$, so $B \in [\delta]x$. Thus $[\delta]x$ is closed under the rule (MP). To show that $[\delta]x$ is closed under (INT), suppose that for any propositional variable p and for any AF $\gamma(\#?)$ we have: $[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B \in [\delta]x$. Then we obtain: $[\delta][\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B \in x$ and $[\delta; \gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B \in x$. Since x is closed under the rule (INT), we obtain: $[\delta; \gamma(\neg < \alpha \wedge \beta > A?)]B \in x$, $[\delta][\gamma(\neg < \alpha \wedge \beta > A?)]B \in x$ and hence $[\gamma(\neg < \alpha \wedge \beta > A?)]B \in [\delta]x$.

(ii) By (i) and the fact that $x + A = [A?]x$ we obtain that $x + A$ is a theory. The verification that it is the smallest theory containing A and x is by an easy application of the relevant definitions and is left to the reader.

(iii) We have the following sequence of equivalences: $x + A$ is inconsistent iff $\perp \in x + A$ iff $A \Rightarrow \perp \in x$ iff $\neg A \in x$.

(iv) Suppose that for any propositional variable p, $x + \neg[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B$ is inconsistent. Then for any propositional variable p, we have $[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B \in x$ and, by (INT), $[\gamma(\neg < \alpha \wedge \beta > A?)]B \in x$. But by assumption we have also $\neg[\gamma(\neg < \alpha \wedge \beta > A?)]B \in x$, which implies that x is inconsistent – a contradiction with the assumption. ■

Now we are ready for the main lemma in this section:

Lemma 4.3. Lindenbaum Lemma. Each consistent theory can be extended to a maximal theory.

Proof. Suppose x is a consistent theory and let $C_1, C_2, \ldots C_n, \ldots$ be an enumeration of all formulas. We define an increasing sequence of consistent theories $x_0, \ldots, x_n, \ldots$ by induction as follows. Let $x_0 =_{\text{def}} x$ and suppose that, for some integer n, x_n is defined. For x_{n+1} we consider two cases:

Case 1: $x_n + C_n$ is consistent. Then define $x_{n+1} =_{\text{def}} x_n + C_n$.

Case 2: $x_n + C_n$ is not consistent. Then $\neg C_n \in x$. In this case we consider two subcases:

Subcase 2.1: C_n is not in a form of a conclusion of the rule (INT). Then let $x_{n+1} =_{\text{def}} x_n$.

Subcase 2.2: C_n is in the form of a conclusion of (INT), namely C_n is in the following form $[\gamma(\neg < \alpha \wedge \beta > A?)]B$, where $\gamma(\#?)$ is an AF. Then there are finitely many such representations for C_n: $[\gamma_i(\neg < \alpha_i \wedge \beta_i > A_i?)]B$, $i = 1, \ldots, k$. We define inductively an increasing sequence of consistent theories x_n^i, $i = 0, \ldots, k$, as follows. Let $x_n^0 =_{\text{def}} x_n$. Suppose x_n^i is defined and consistent. Then it contains $\neg C_n = \neg[\gamma_i(\neg < \alpha_i \wedge \beta_i > A_i?)]B$ and by 4.2 (iv) there exists a propositional variable p_i such that $x_n^i + \neg[\gamma_i(\neg(< \alpha_i > (A_i \wedge p_i) \vee < \beta_i > (A_i \wedge \neg p_i))?)]B$ is consistent. We define x_n^{i+1} as follows: $x_n^{i+1} =_{\text{def}} x_n^i + \neg[\gamma_i(\neg(< \alpha_i > (A_i \wedge p_i) \vee < \beta_i > (A_i \wedge \neg p_i))?)]B$. Now we put $x_{n+1} =_{\text{def}} x_n^k$. Finally, we define $y =_{\text{def}} \bigcup_{i=0}^{\infty} x_i$. It is straightforward that y is a maximal theory which extends x. ■

Let L denote the logic $\text{PDL}_0^{\cap}$ and W_L be the set of all maximal theories of L. We define in W_L a function Q_L (denoted sometimes by Q) which assigns to each program α a binary relation $Q_L(\alpha)(= Q(\alpha))$ as follows:

For all $x, y \in W_L$, $xQ(\alpha)y$ iff $[\alpha]x \subseteq y$.

Let us note that we also have the following equivalence, which will be used later on: $xQ(\alpha)y$ iff $(\forall A \in y)(< \alpha > A \in x)$. This is a standard construction in the canonical models for various modal logics but we will call the pair (W_L, Q_L) quasi-canonical model for some reasons to be explained later on. The following lemma summarizes some properties of the relation Q_L.

Lemma 4.4. Let $x, y, z \in W_L$, α and β be programs and A be a formula. Then:

(i) $< \alpha > A \in x$ iff $(\exists y \in W_L)([\alpha]x \subseteq y \& A \in y)$.

(ii) $[\alpha \wedge \beta]x \subseteq y$ iff $[\alpha]x \subseteq y$ and $[\beta]x \subseteq y$.

(iii) $[\alpha \vee \beta]x \subseteq y$ iff $[\alpha]x \subseteq y$ or $[\beta]x \subseteq y$.

(iv) $[A?]x \subseteq y$ iff $x = y$ and $A \in x$.

(v) $[\alpha]x \subseteq z$ and $[\beta]z \subseteq y$ iff $(\forall C \in z)([\alpha; C?; \beta]x \subseteq y)$.

Proof. (i) The implication $(\leftarrow)$ is easy.

$(\rightarrow)$ Suppose $< \alpha > A \in x$. Then $[\alpha]\neg A \notin x$ and $\neg A \notin [\alpha]x$. Then by 4.2 (iii), $[\alpha]x + A$ is a consistent theory and by the Lindenbaum lemma there exists a maximal theory y such that $[\alpha]x + A \subseteq y$. From here we obtain $[\alpha]x \subseteq y$ and $A \in y$.

(ii) $(\rightarrow)$ Suppose $[\alpha \wedge \beta]x \subseteq y$ and for the sake of contradiction that $[\alpha]x \not\subseteq y$. Then there exists formula B such that $B \in [\alpha]x$ and $B \notin y$. From here we obtain that $B \notin [\alpha \wedge \beta]x$ and consequently $[\alpha \wedge \beta]B \notin x$. So $< \alpha \wedge \beta > \neg B \in x$ and by axiom $(\wedge 1)$ for program conjunction $< \alpha > \neg B \in x$. By the maximality of x, we get $[\alpha]B \notin x$ and $B \notin [\alpha]x$ – a contradiction. So $[\alpha \wedge \beta]x \subseteq y \rightarrow [\alpha]x \subseteq y$. In the same way one can prove the implication $[\alpha \wedge \beta]x \subseteq y \rightarrow [\beta]x \subseteq y$.

$(\leftarrow)$ Suppose $[\alpha]x \subseteq y$, $[\beta]x \subseteq y$ and for the sake of contradiction that $[\alpha \wedge \beta]x \not\subseteq y$. Then for some formula A we have that $A \in y$ and $< \alpha \wedge \beta > A \notin x$. By the rule (INT), there exists a propositional variable p such that $< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p) \notin x$. By the maximality of x, we obtain: (1) $< \alpha > (A \wedge p) \notin x$ and (2) $< \beta > (A \wedge \neg p) \notin x$. We consider two cases for p.

Case 1: $p \in y$. Since $A \in y$ then $A \wedge p \in y$, $\neg(A \wedge p) \notin y$ and by $[\alpha]x \subseteq y$, we obtain $[\alpha]\neg(A \wedge p) \notin x$ and $< \alpha > (A \wedge p) \in x$ – a contradiction with (1).

Case 2: $\neg p \in y$. Since $A \in y$, then $A \wedge \neg p \in y$, $\neg(A \wedge \neg p) \notin y$ and by $[\beta]x \subseteq y$, we obtain $[\beta]\neg(A \wedge \neg p) \notin x$ and $< \beta > (A \wedge \neg p) \in x$ – a contradiction with (2).

(iii) The proof of (iii) follows directly from the axiom ($\vee$) for the program disjunction without the rule (INT).

(iv) We have the following sequence of equivalences: $[A?]x \subseteq y$ iff $x + A \subseteq y$ iff $x + A = x$ and $x = y$ (because x is maximal and $x + A$ is consistent) iff $A \in x$ and $x = y$.

(v) ($\rightarrow$) Suppose $[\alpha]x \subseteq z$ and $[\beta]z \subseteq y$ and in order to obtain a contradiction that $(\forall C \in z)([\alpha; C?; \beta]x \subseteq y)$ is not the case. Then there exist formulas C and A such that $C \in z$, $[\alpha; C?; \beta]A \in x$ and $A \notin y$. From the later and $[\beta]z \subseteq y$, we get $[\beta]A \notin z$. From here and $C \in z$, we obtain $C \Rightarrow [\beta]A \notin z$. From this and $[\alpha]x \subseteq z$, we obtain $[\alpha](C \Rightarrow [\beta]A) \notin x$. This is equivalent to $[\alpha; C?; \beta]A \notin x$ – a contradiction.

($\leftarrow$) We have to prove the implications:

(a) $(\forall C \in z)([\alpha; C?; \beta]x \subseteq y) \rightarrow [\alpha]x \subseteq z$.

(b) $(\forall C \in z)([\alpha; C?; \beta]x \subseteq y) \rightarrow [\beta]z \subseteq y$.

Suppose that (a) is not true. Then we have (3) $(\forall C \in z)[\alpha; C?; \beta]x \subseteq y$ and (4) $[\alpha]x \not\subseteq z$. From (4), there exists a formula A such that (5) $\neg A \in z$ and (6) $A \in [\alpha]x$. So from (5) and (3), we obtain (7) $[\alpha; \neg A?; \beta]x \subseteq y$. Since $\bot \notin y$, from (7) we get (8) $[\alpha; \neg A?; \beta]\bot \notin x$. From (8), we conclude that (9) $\neg A \Rightarrow [\beta]\bot \notin [\alpha]x$. We observe that (6) implies (10) $\neg A \Rightarrow [\beta]\bot \in [\alpha]x$, which contradicts (9). This proves (a).

To prove (b) suppose that $(\forall C \in z)[\alpha; C?; \beta]x \subseteq y$ and that $[\beta]z \not\subseteq y$. From the later, we obtain that there exists a formula B such that $B \notin y$ and $[\beta]B \in z$ and from the former, that $[\alpha; [\beta]B?; \beta]B \notin x$. This is equivalent to $[\beta]B \Rightarrow [\beta]B \notin [\alpha]x$, which is impossible because $[\beta]B \Rightarrow [\beta]B$ is a theorem and consequently belongs to every theory. ■

Remark. If in 4.4 the condition (v) is in a stronger form, namely for any maximal theories x, y: $[\alpha; \beta]x \subseteq y$ iff $(\exists z \in W_L)([\alpha]x \subseteq z$ and $[\beta]z \subseteq y)$, then the quasi-canonical model would be sufficient to prove the completeness theorem for $\mathrm{PDL}_0^{\cap}$. However the above equivalence is not true and that is why we call the model quasi-canonical. We will overcome this defect of quasi-canonical models by introducing in the next section other objects – large and maximal programs.

We conclude this section with some technical statements about maximal theories.

Lemma 4.5. (i) Let x be a maximal theory, $< \delta(C?) > B \in x$, where $C?$ is a subprogram of δ. Then for every formula F we have either $< \delta((C \wedge F)?) > B \in x$ or $< \delta((C \wedge \neg F)?) > B \in x$.

(ii) Let x be a maximal theory, $< \delta((C \wedge \neg F)?) > B \in x$, where $(C \wedge \neg F)?$ is a subprogram of the program δ and F is in the form $[\gamma(\neg < \alpha \wedge \beta > A?)]B'$ for some admissible form $\gamma(\#?)$. Then there exists a propositional variable p such that $< \delta((C \wedge \neg F \wedge \neg[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B')?) > B \in x$.

Proof. (i) The proof follows from 3.2 (6) and the fact that x is a maximal theory.

(ii) Suppose that the statement is not true and proceed to obtain a contradiction. Then we have $< \delta((C \wedge \neg F)?) > B \in x$ and for every propositional variable p, $< \delta((C \wedge \neg F \wedge \neg[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B')?) > B \notin x$ or equivalently $[\delta((C \wedge \neg F \wedge \neg[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?)]B')?)]\neg B \in x$. Applying 3.2 (2), we obtain: $[\delta(C? \wedge \neg F? \wedge \neg[\gamma(\neg(< \alpha > (A \wedge p) \vee < \beta > (A \wedge \neg p))?]B'?)]\neg B \in x$ for every propositional variable p. Observe that $\delta(C? \wedge \neg F? \wedge \neg[\gamma(\#?)]B'?)$ is an admissible form. Then since x is (INT)-closed we obtain: $[\delta(C? \wedge \neg F? \wedge \neg[\gamma(\neg < \alpha \wedge \beta > A?)]B'?)]\neg B \in x$. Again by 3.2 (2), we obtain: $[\delta((C \wedge \neg F \wedge \neg[\gamma(\neg < \alpha \wedge \beta > A?)]B')?)]\neg B \in x$. Since $F = [\gamma(\neg < \alpha \wedge \beta > A?)]B'$, we obtain: $[\delta((C \wedge \neg F)?)]\neg B \in x$ and $< \delta((C \wedge \neg F)?) > B \notin x$ – a contradiction. ■

The following technical lemma will be important in the next section.

Lemma 4.6. Diamond Extension Lemma. Let $\delta = \delta(F_1?, \ldots, F_n?)$ be a program not containing the operation of program disjunction and $F_1?, \ldots, F_n?$ be the sequence of all subprograms of δ which are tests. Let x be a maximal theory such that $< \delta > B \in x$. Then there exist maximal theories $x_1, \ldots, x_n$ such that:

(i) For any $i = 1, \ldots, n$, $F_i \in x_i$.

(ii) For any $A_1 \in x_1, \ldots, A_n \in x_n$, $< \delta(A_1?, \ldots, A_n?) > B \in x$.

Proof. The proof is rather heavy and technical, so the reader can skip it in the first reading. We will construct the maximal theories $x_1, \ldots, x_n$ by a step-by-step construction similar to the construction in the proof of the Lindenbaum Lemma. Let $C_1, C_2, \ldots$ be an enumeration of all formulas. We define by induction n sequences of formulas F_i^j, where $i = 1, \ldots, n$ and $j = 0, 1, \ldots$, with the following properties:

(•1) All formulas F_i^j are consistent, namely not equivalent to $\bot$.

(•2) $< \delta(F_1^j?, \ldots, F_n^j?) > B \in x$.

(•3) If $j > k$ then $F_i^j \Rightarrow F_i^k$ is a theorem.

These sequences will be used to define the required maximal theories.

Basis on j**:** $j = 0$, $i = 1, \ldots, n$. We put: $F_i^0 = F_i$.

Hypothesis on j**,** $i = 1, \ldots, n$**.** Suppose F_i^j, $i = 1, \ldots, n$, are defined and satisfy the required conditions and proceed to define F_i^{j+1}, $i = 1, \ldots, n$.

Step $j + 1$. Now we start an internal induction on i.

Basis: for the given j **and** $i = 1$. We consider two cases:

Case 1: $< \delta((F_1^j \wedge C_j)?, \ldots, F_n?) > B \in x$. In this case define $F_1^{j+1} =_{\text{def}} F_1^j \wedge C_j$.

Case 2: $< \delta((F_1^j \wedge C_j)?, \ldots, F_n?) > B \notin x$. Then by 4.5 (i), we have $< \delta((F_1^j \wedge \neg C_j)?, \ldots, F_n?) > B \in x$.

Subcase 2.1: C_j is not in the form of a conclusion of the rule (INT). Then in this case define $F_1^{j+1} =_{\text{def}} F_1^j \wedge \neg C_j$.

Subcase 2.2: C_j is in the form of a conclusion of (INT). There are finitely many, say k, such representations of C_j: $[\gamma_1(\neg < \alpha_1 \wedge \beta_1 > A_1?)]B_1, \ldots, [\gamma_k(\neg < \alpha_k \wedge \beta_k > A_k?)]B_k$. We define in this case a sequence of formulas $F_1^{j,1}, \ldots, F_1^{j,k}$ by induction as follows. By 4.5 (ii), there exists a propositional variable p_1 such that $< \delta(F_1^j \wedge \neg C_j \wedge \neg[\gamma_1(\neg(< \alpha_1 > (A_1 \wedge p_1) \vee < \beta_1 > (A_1 \wedge \neg p_1))?)]B_1)?, \ldots, F_n^j) > B \in x$. Take such a propositional variable p_1 and define: $F_1^{j,1} = F_1^j \wedge \neg C_j \wedge \neg[\gamma_1(\neg(<\alpha_1>(A_1 \wedge p_1) \vee <\beta_1>(A_1 \wedge \neg p_1))?)]B_1$. Repeating again 4.5 (ii) and proceeding as above, we find a propositional variable p_2 such that $<\delta((F_1^{j,1} \wedge \neg[\gamma_2(\neg(<\alpha_2>(A_2 \wedge p_2) \vee <\beta_2>(A_2 \wedge \neg p_2))?)]B_2)?, \ldots, F_n^j)> B \in x$. For this propositional variable p_2, define: $F_1^{j,2} = F_1^{j,1} \wedge \neg[\gamma_2(\neg(<\alpha_2>(A_2 \wedge p_2) \vee <\beta_2>(A_2 \wedge \neg p_2))?)]B_2$. Proceed further in this way till k and finally define: $F_1^{j+1} = F_1^{j,k}$. The case $i = 1$ is finished.

Hypothesis on i for the given j. Suppose $i < n$ and let F_k^{j+1} be defined for $k = 1, \ldots, i$.

Step for $i+1$ and the given j. We proceed similarly as for $i = 1$ and consider two cases.

Case 1. $<\delta(F_1^{j+1}?, \ldots, F_i^{j+1}?, (F_{i+1}^j \wedge C_j)?, \ldots, F_n^j)> B \in x$. Define in this case $F_{i+1}^{j+1} = F_{i+1}^j \wedge C_j$.

Case 2. $<\delta(F_1^{j+1}?, \ldots, F_i^{j+1}?, (F_{i+1}^j \wedge C_j)?, \ldots, F_n^j)> B \notin x$. Then by 4.5 (i), we have $<\delta(F_1^{j+1}?, \ldots, F_i^{j+1}?, (F_{i+1}^j \wedge \neg C_j)?, \ldots, F_n^j)> B \in x$.

Subcase 2.1. C_j is not in the form of a conclusion of the rule INT. Then in this case define $F_{i+1}^{j+1} = F_{i+1}^j \wedge \neg C_j$.

Subcase 2.2. C_j is in a form of a conclusion of the rule INT and there are, say k, such representations of C_j. Then proceed similarly as in the subcase 2.2 for $i = 1$, defining inductively a sequence of formulas $F_{i+1}^{j,1}, \ldots, F_{i+1}^{j,k}$. Put $F_{i+1}^{j+1} = F_{i+1}^{j,k}$.

The induction is finished. Condition (•2) is fulfilled by the construction. Also by the construction one can see that the formula F_i^{j+1} contains as conjuncts the formulas F_i^k for $k = 0, \ldots, j$, so the condition (•3) is fulfilled. Let us note that from the condition (•2) and the assumption that the program δ does not contain program disjunction, one can conclude, by 3.3 (6), that each formula F_i^j is consistent. So condition (•1) is fulfilled. Let us remind that TH is the set of all theories. Then define:

$x_i^j =_{\text{def}} TH + F_i^j$, $i = 1, \ldots, n$, $j = 0, 1, \ldots$,

$x_i =_{\text{def}} \bigcup_{j=0}^{\infty} x_i^j$.

Each theory x_i^j is consistent, because the formulas F_i^j are consistent. The fact that the formula F_i^j contains as conjuncts the formulas F_i^k for $k = 0, \ldots, j-1$ implies that for fixed i the sequence $\{x_i^j / j = 0, 1, \ldots\}$ is increasing. Consequently x_i is a consistent theory. Further, by the construction, we obtain that for any formula C (equal to some formula C_n from the enumeration), either C or $\neg C$ is in x_i and that x_i is closed under (INT). So x_i is a maximal theory, which obviously contains F_i. Thus part (i) from the lemma is fulfilled.

To prove the condition (ii) suppose that $A_1 \in x_1, \ldots, A_n \in x_n$. Then for some $j^1, \ldots, j^n$ we have $A_1 \in x_1^{j^1}, \ldots, A_n \in x_n^{j^n}$. Let j be the greatest number in the set $\{j^1, \ldots, j^n\}$. Then $A_i \in x_i^j$, $i = 1, \ldots, n$. By the definition of x_i^j, we obtain that $F_i^j \Rightarrow A_i \in TH$, $i = 1, \ldots, n$. Then by 3.3 (vii), $F_i^j? \preceq A_i?$. Again by 3.3 (vii), we conclude that $\delta(F_1^j?, \ldots, F_n^j?) \preceq \delta(A_1?, \ldots, A_n?)$. From here and $<\delta(F_1^j?, \ldots, F_n^j?)> B \in x$, we obtain $<\delta(A_1?, \ldots, A_n?)> B \in x$. This ends the proof of part (ii) of the lemma. ■

5. Large programs

In this section we introduce the notions of large and maximal programs, the latter having a key role in the completeness theorem. In some sense large programs enable us to consider and operate with many usual programs simultaneously.

By a *large program* we mean any non-empty set Γ of programs. If Γ is a singleton $\{\alpha\}$ then it is identified with α. In this way each usual program can be considered as a large program. If x is a non-empty set of formulas then the large program $x? =_{\rm def} \{A?/A \in x\}$ is called a large test. If x is a maximal theory then $x?$ is called a maximal test. The standard program operations are extended in the set of large programs in the following way (Γ and Δ are large programs):

- $\Gamma \vee \Delta =_{\rm def} \{\gamma \vee \delta/\gamma \in \Gamma, \delta \in \Delta\}$.
- $\Gamma \wedge \Delta =_{\rm def} \{\gamma \wedge \delta/\gamma \in \Gamma, \delta \in \Delta\}$.
- $\Gamma;\Delta =_{\rm def} \{\gamma;\delta/\gamma \in \Gamma, \delta \in \Delta\}$.

It is obvious that in case Γ and Δ are singletons, then the above operations coincide with the standard program operations. Two large programs Γ and Δ are *equivalent*, in symbols $\Gamma \equiv \Delta$, if each element of Γ is equivalent to some element of Δ and vice versa. Sometimes we will treat equivalent large programs as equal. A large program Γ is called *maximal program* if it is generated by singletons and maximal tests, using the above operations. The following lemma describes the structure of maximal programs.

Lemma 5.1. The structure of maximal programs

(i) Let $\alpha = \alpha(A_1?, \ldots, A_n?)$ be a program, $A_1?, \ldots, A_n?$ be a sequence (possibly empty) of all subprograms of α which are tests and let $x_1?, \ldots, x_n?$ be a sequence of maximal tests. Then $\alpha(x_1?, \ldots, x_n?)$ is a maximal program and all maximal programs can be obtained in this way.

(ii) Let Γ be a maximal program, $\alpha \in \Gamma$ and $\alpha = \alpha(A_1?, \ldots, A_n?)$, where $A_1?, \ldots, A_n?$ is the sequence (possibly empty) of all subprograms of α which are tests. Then there exists a sequence of maximal tests $x_1?, \ldots, x_n?$ such that $A_1? \in x_1?, \ldots, A_n? \in x_n?$ and $\Gamma = \alpha(x_1?, \ldots, x_n?)$.

Proof. (i) follows directly by the definition of a maximal program.

(ii) By (i), Γ is in the form $\gamma(x_1?, \ldots, x_n?)$, where γ is a program. From $\alpha \in \Gamma$, we obtain that $\alpha = \gamma(A_1?, \ldots, A_n?)$ with $A_1? \in x_1?, \ldots, A_n? \in x_n?$, which ends the proof. ∎

In section 4, we introduced the notion of quasi-model (W_L, Q_L), where W_L is the set of all maximal theories and $Q = Q_L$ is a function which assigns to each program α a binary relation $Q(\alpha)$: $xQ(\alpha)y$ iff $[\alpha]x \subseteq y$. Now we extend the function Q to large programs as follows. Let Γ be a large program and x, y be maximal theories. Then define:

$xQ(\Gamma)y$ iff $(\forall\alpha \in \Gamma)(xQ(\alpha)y)$ iff $(\forall\alpha \in \Gamma)([\alpha]x \subseteq y)$.

Obviously, if Γ is a singleton large program then Q is the same as for usual programs. The following lemma generalizes some statements from 4.4.

Lemma 5.2. Let Γ and Δ be large programs and x, y, z be maximal theories. Then:

(i) $xQ(\Gamma \wedge \Delta)y$ iff $xQ(\Gamma)y$ and $xQ(\Delta)y$.

(ii) $xQ(\Gamma \vee \Delta)y$ iff $xQ(\Gamma)y$ or $xQ(\Delta)y$.

(iii) $xQ(z?)y$ iff $x = y = z$.

(iv) $xQ(\Gamma; z?; \Delta)y$ iff $xQ(\Gamma)z$ and $zQ(\Delta)y$.

Proof. (i) Applying 4.4 (ii), we obtain the following: $xQ(\Gamma \wedge \Delta)y$ iff $(\forall \alpha \wedge \beta \in \Gamma \wedge \Delta)([\alpha \wedge \beta]x \subseteq y)$ iff $(\forall \alpha \in \Gamma)(\forall \beta \in \Delta)([\alpha]x \subseteq y \& [\beta]x \subseteq y)$ iff $(\forall \alpha \in \Gamma)([\alpha]x \subseteq y)$ and $(\forall \beta \in \Delta)([\beta]x \subseteq y)$ iff $xQ(\Gamma)y$ and $xQ(\Delta)y$.

(ii) The proof is similar to that of (i) by an application of 4.4 (iii).

(iii) By 4.4 (iv), we obtain the following equivalences: $xQ(z?)y$ iff $(\forall A? \in z?)([A?]x \subseteq y)$ iff $(\forall A \in z)(x = y \& A \in x)$ iff $x = y$ and $(\forall A \in z)(A \in x)$ iff $x = y$ and $z \subseteq x$ iff $x = y$ and $z = x$ iff $x = y = z$.

(iv) $xQ(\Gamma; z?; \Delta)y$ iff $(\forall \alpha \in \Gamma)(\forall C? \in z?)(\forall \beta \in \Delta)([\alpha; C?; \beta]x \subseteq y)$ iff $(\forall \alpha \in \Gamma)(\forall \beta \in \Delta)([\alpha]x \subseteq z \& [\beta]z \subseteq y)$ iff $(\forall \alpha \in \Gamma)([\alpha]x \subseteq z)$ and $(\forall \beta \in \Delta)([\beta]z \subseteq y)$ iff $xQ(\Gamma)z$ and $zQ(\Delta)y$. We have used 4.4 (v). ■

Lemma 5.3. Let x, y be maximal theories and Γ be a large program. Then $xQ(\Gamma)y$ iff $(\forall \alpha \in \Gamma)(\forall B \in y)(< \alpha > B \in x)$.

Proof. Let us note that for maximal theories we have the following equivalence, which proves the lemma: $[\alpha]x \subseteq y$ iff $(\forall A \in y)(< \alpha > A \in x)$. ■

The following is one of the important propositions for maximal programs in this section. The proof uses the Diamond Extension Lemma from section 4.

Lemma 5.4. The Main Lemma for Maximal Programs. Let x be a maximal theory and $< \delta > B \in x$. Then there exist a maximal program Γ and a maximal theory y such that $\delta \in \Gamma$, $B \in y$ and $xQ(\Gamma)y$.

Proof. First we suppose that the lemma is proved for the case where δ does not contain program disjunction and show the general case. By lemma 3.3, δ is equivalent to a disjunction $\delta_1 \vee \ldots \vee \delta_k$ of programs not containing program disjunction. For simplicity, we assume $k = 2$ (the general case can be proved in the same way). From here, we obtain $< \delta_1 > B \in x$ or $< \delta_2 > B \in x$. We consider two cases.

Case 1: $< \delta_1 > B \in x$. Then by the lemma, there exist a maximal theory y and a maximal program Γ_1 such that $B \in y$, $\delta_1 \in \Gamma_1$ and $xQ(\Gamma_1)y$. Let Γ_2 be a maximal program such that $\delta_2 \in \Gamma_2$ and $\Gamma_1 \vee \Gamma_2$ be a maximal program (such a program can be constructed by lemma 5.1). Then $xQ(\Gamma_1)y$ implies the disjunction $(xQ(\Gamma_1)y$ or $xQ(\Gamma_2)y)$. By 5.2 (ii), the later is equivalent to $xQ(\Gamma_1 \vee \Gamma_2)y$. Obviously, $\delta_1 \vee \delta_2 \in (\Gamma_1 \vee \Gamma_2)$ and in this case the lemma is proved.

Case 2: $< \delta_2 > B \in x$. The proof is similar to the above one.

Now we start proving the lemma with the additional assumption that δ does not contain program disjunction. First suppose that δ do not contain tests as subprograms. Then δ itself is a maximal program and the proof follows from 4.4 (i). Now let δ contains tests and $F_1?, \ldots, F_n?$ be the sequence of all tests which are subprograms of δ. So we have $\delta = \delta(F_1?, \ldots, F_n?)$ and $< \delta(F_1?, \ldots, F_n?) > B \in x$. The later is equivalent to $< \delta(F_1?, \ldots, F_n?); B? > \top \in x$. By the Diamond Extension Lemma 4.6, there exist maximal theories $x_1, \ldots, x_n, y$ such that firstly (a) $F_1 \in x_1, \ldots, F_n \in x_n$ and $B \in y$ and secondly (b) if $A_1 \in x_1, \ldots, A_n \in x_n, B' \in y$ then $< \delta(A_1?, \ldots, A_n?); B'? > \top \in x$. Let $\Gamma = \delta(x_1?, \ldots, x_n?)$. Then by (a), we have $\delta = \delta(A_1?, \ldots, A_n?) \in \Gamma$ and $B \in y$. We will show that $xQ_L(\Gamma)y$, which will end the proof. In order to apply 5.3, suppose $B' \in y$, $\gamma \in \Gamma$ and proceed to show that $< \gamma > B' \in x$. From $\gamma \in \Gamma$, we obtain $\gamma = \delta(A'_1?, \ldots, A'_n?)$ for some $A'_1 \in x_1, \ldots, A'_n \in x_n$. Then by (b), we have $< \delta(A'_1?, \ldots, A'_n?; B'? > \top \in x$. From here, we obtain $< \delta(A'_1?, \ldots, A'_n?) > B' \in x$ and finally $< \gamma > B' \in x$, which ends the proof. ■

6. Canonical model for PDL$_0^\cap$ and the completeness theorem

For the definition of the canonical model for PDL$_0^\cap$ we introduce the following translation f of the set of programs into itself:

- $f(\pi) = \pi$ for a program variable π.
- $f(\alpha;\beta) = f(\alpha); \top?; f(\beta)$.
- $f(\alpha \circ \beta) = f(\alpha) \circ f(\beta)$, $\circ \in \{\vee, \wedge\}$.
- $f(A?) = A?$ for a formula A.

Let us note that one can prove by induction that for every program α, we have that $\alpha \equiv f(\alpha)$. The technical importance of the translation f is that it inserts tests between the components of the composition of programs which will be used later on in the proof of the characteristic property of composition in the canonical model. Now the canonical model $M_L = (W_L, R_L, V_L)$ for PDL$_0^\cap$ is defined as follows:

- W_L is the set of all maximal theories of PDL$_0^\cap$.
- R_L is a function from the set of all program variables into the binary relations in W_L defined as follows:

For all program variable π and for all $x, y \in W_L$, $xR_L(\pi)y$ iff $[\pi]x \subseteq y$ iff $xQ_L(\pi)y$.

- For all propositional variable p, $V_L(p) = \{x \in W_L / p \in x\}$.

This is obviously a standard model for PDL$_0^\cap$ and the valuations R_L and V_L are extended for arbitrary programs and arbitrary formulas inductively as it was defined for arbitrary models. It should be remarked that for all program α and for all $x, y \in W$, $xR_L(\alpha)y$ iff $xR_L(f(\alpha))y$. This can be proved by induction on the formation of α. The difference between the quasi-canonical model and the canonical model is that in quasi-canonical model the definition of the relations $Q_L(\alpha)$ is for an arbitrary program α while in the canonical model this relation is defined only for a program variable π and then extended inductively to arbitrary programs. In general these extensions are not the same as those in the quasi-canonical model. The following lemma plays an important role in the proof of the completeness theorem.

Lemma 6.1. Truth Lemma for the Canonical Model.

(i) For every formula A: $V_L(A) = \{x \in W_L / A \in x\}$.

(ii) For every program γ and for all $x, y \in W_L$: $xR_L(\gamma)y$ iff there exists a maximal program Γ such that $f(\gamma) \in \Gamma$ and $xQ_L(\Gamma)y$.

Proof. The proof of (i) and (ii) will be given simultaneously by a double induction on formation of formulas and programs.

Basis. For propositional variables the lemma follows from the definition of V_L and R_L and from the fact that program variables are itself maximal programs.

Induction Hypothesis (IH). Let us suppose that for a formula A and for programs γ, δ, the lemma is true (IH) and let us proceed to prove the lemma for formulas and programs built from A, γ and δ.

Proof of (i).

• Case for $\bot$. Since for any maximal theory x, $\bot \notin x$ we have $V_L(\bot) = \emptyset$.

• Case for $< \gamma > A$.

We have to show that for $x \in W_L$: $x \in V_L(< \gamma > A)$ iff $< \gamma > A \in x$.

($\rightarrow$) Suppose $x \in V_L(< \gamma > A)$. Then there exists $y \in V_L(A)$ such that $xR_L(\gamma)y$. By the (IH), $A \in y$ and there exists a maximal program Γ such that $f(\gamma) \in \Gamma$ and $xQ_L(\Gamma)y$. Then by the definition of $Q_L(\Gamma)$, we have: for all $\alpha \in \Gamma$, $[\alpha]x \subseteq y$, so for $\alpha = f(\gamma)$ we obtain $[f(\gamma)]x \subseteq y$. From here and $A \in y$, we get $< f(\gamma) > A \in x$ and since $f(\gamma)$ is equivalent to γ, we have that $< \gamma > A \in x$.

($\leftarrow$) Suppose $< \gamma > A \in x$. From here we get $< f(\gamma) > A \in x$. Then by the Main Lemma for Maximal Programs 5.4, there exist a maximal program Γ and a maximal theory y such that $f(\gamma) \in \Gamma$ and $xQ_L(\Gamma)y$ and $A \in y$. By (IH), we have $y \in V_L(A)$ and $xR_L(\gamma)y$. Hence we get $x \in V_L(< \gamma > A)$, which completes the proof for this case.

Proof of (ii).

• Case for $\gamma \vee \delta$. We have to prove: $xR_L(\gamma \vee \delta)y$ iff there exists a maximal program Σ such that $xQ_L(\Sigma)y$ and $f(\gamma \vee \delta) \in \Sigma$.

($\rightarrow$) Suppose for $x, y \in W_L$, $xR_L(\gamma \vee \delta)y$. Then we have $xR_L(\gamma)y$ or $xR_L(\delta)y$. By the (IH) for γ and δ, we have that for some maximal programs Γ and Δ: $f(\gamma) \in \Gamma$, $f(\delta) \in \Delta$ and ($xQ_L(\Gamma)y$ or $xQ_L(\Delta)y$). From here, we obtain $f(\gamma) \vee f(\delta) = f(\gamma \vee \delta) \in \Gamma \vee \Delta$. Then by 5.1, $(\Gamma \vee \Delta)$ is a maximal program and by 5.2 (ii), $xQ_L(\Gamma \vee \Delta)y$. So in this case, $\Sigma = \Gamma \vee \Delta$.

($\leftarrow$) Suppose for some maximal program Σ that $xQ_L(\Sigma)y$ and $f(\gamma \vee \delta) \in \Sigma$. Then by 5.1, Σ is in the form $\Gamma \vee \Delta$ and $f(\gamma) \in \Gamma$ and $f(\delta) \in \Delta$. Thus we have $xQ_L(\Gamma \vee \Delta)y$. Then by 5.2 (ii), we obtain the disjunction ($xQ_L(\Gamma)y$ or $xQ_L(\Delta)y$). By the (IH), we get ($xR_L(\gamma)y$ or $xR_L(\delta)y$), which is equivalent to $xR_L(\gamma \vee \delta)y$.

• Case for $\gamma \wedge \delta$.

The proof is similar to the proof of the previous case by using 5.1 and 5.2 (i).

• Case for $A?$. We have to show: $xR_L(A?)y$ iff for some maximal program Σ: $A? \in \Sigma$ and $xQ_L(\Sigma)y$.

($\rightarrow$) Suppose $xR_L(A?)y$. Then $x = y$ and $x \in V_L(A)$. By the (IH), we get: $x = y$ and $A \in x$. Then by 5.2 (iii), we obtain $xQ_L(x?)y$. Thus we have a maximal program $x?$ such that $A? \in x?$ and $xQ_L(x?)y$. In this case $\Sigma = x?$.

($\leftarrow$) Suppose for some maximal program Σ that $xQ_L(\Sigma)y$ and $A? \in \Sigma$. By 5.1 (ii), $\Sigma = z?$ for some maximal theory z and also $A? \in z?$. From here, we get $A \in z$ and $xQ_L(z?)y$. Then by

5.2 (iii), we obtain $x = y = z$, which implies $xR_L(A?)y$.

• Case for $\gamma;\delta$. We have to prove: $xR_L(\gamma;\delta)y$ iff there exists a maximal program Σ such that $f(\gamma;\delta) \in \Sigma$ and $xQ_L(\Sigma)y$.

($\rightarrow$) Suppose for $x, y \in W_L$ $xR_L(\gamma;\delta)y$. Then for some $z \in W_L$ we have $xR_L(\gamma)z$ and $zR_L(\delta)y$. By the (IH) for γ and δ, we have that for some maximal programs Γ and Δ: $f(\gamma) \in \Gamma$ and $f(\delta) \in \Delta$ and $(xQ_L(\Gamma)z$ and $zQ_L(\Delta)y)$. Then by 5.2 (iv), we have $xQ_L(\Gamma; z?; \Delta)y$. From $f(\gamma) \in \Gamma$, $\top? \in z?$ and $f(\delta) \in \Delta$ we get $f(\gamma); \top?; f(\delta) \in \Gamma; z?; \Delta$, hence $f(\gamma;\delta) \in \Gamma; z?; \Delta$. In this case, $\Sigma = \Gamma; z?; \Delta$.

($\leftarrow$) Suppose for some maximal program Σ that $xQ_L(\Sigma)y$ and that $f(\gamma;\delta) = f(\gamma); \top?; f(\delta) \in \Sigma$. Then by 5.1, Σ is in the form $\Gamma; z?; \Delta$ for some maximal programs Γ, Δ and $z?$ with $f(\gamma) \in \Gamma$ and $f(\delta) \in \Delta$. From here, we obtain $xQ_L(\Gamma; z?; \Delta)y$ and by 5.2 (iv), we have $xQ_L(\Gamma)z$ and $zQ_L(\Delta)y$. By the (IH), we get $xR_L(\gamma)z$ and $zR_L(\delta)y$. From here, we obtain $xR_L(\gamma;\delta)y$ which ends the proof.

The induction is finished and the lemma is proved. ■

Remark. The important part of 6.1 is (i), which is essential in the proof of the completeness theorem for $\mathrm{PDL}_0^{\cap}$. However we cannot prove (i) without proving (ii). From here one can see the technical importance of the notion of a maximal program. Note also that one can see the role of the translation f examining the proof of (ii) for the case of composition $\gamma;\delta$.

Proposition 6.2. Canonical Model Lemma. For every formula A, if A is true in the canonical model, then A is a theorem of $\mathrm{PDL}_0^{\cap}$.

Proof. Suppose A is true in the canonical model $M_L = (W_L, R_L, V_L)$ but for the sake of contradiction that A is not a theorem of $\mathrm{PDL}_0^{\cap}$. Then $A \notin TH$, where TH is the set of all theorems. By 4.2 (iii) $TH + \neg A$ is a consistent theory and by the Lindenbaum Lemma 4.3 it can be extended into a maximal theory, say x. From here we obtain that $A \notin x$ and by the Truth Lemma for Canonical Model 6.1 $x \notin V_L(A)$, so $V_L(A) \neq W_L$. So A is not true in the canonical model – a contradiction. ■

Now we are ready for the main theorem of the paper.

Theorem 6.3. Completeness Theorem for $\mathrm{PDL}_0^{\cap}$. The following conditions are equivalent for any formula A of $\mathrm{PDL}_0^{\cap}$:

(i) A is a theorem of $\mathrm{PDL}_0^{\cap}$.

(ii) A is true in all PDL-models.

Proof. $(i) \rightarrow (ii)$ is just the Soundness Theorem for $\mathrm{PDL}_0^{\cap}$.

$(ii) \rightarrow (i)$. Suppose A is true in all PDL-models. Then A is true in the canonical model. By the Canonical Model Lemma 6.2, A is a theorem of $\mathrm{PDL}_0^{\cap}$. ■

Theorem 6.4. Model Existency Theorem. If a set of formulas Σ is consistent, then it has a model.

Proof. Let Σ be a consistent set. Then by definition, there exists a consistent theory x_0 containing Σ. By the Lindenbaum Lemma, x_0 can be extended into a maximal theory x. Then by the Truth Lemma for Canonical Model, the pair $< (W_L, R_L, V_L), x >$ is a model of Σ. ■

7. Concluding remarks

In this paper we have given a complete axiomatization of star-free PDL with intersection using a special rule (INT) for intersection. In the standard examples of such completeness theorems, the main method of proving the completeness is the method of canonical models consisting of the set of all maximal consistent sets of the logic, equipped with standard definitions for the canonical accessibility relations like in the classical canonical constructions. In our case however, this method cannot work, because of the bad behaviour of composition. So we extend the canonical construction introducing a new machinery, connected with the new notions of *large and maximal programs*. We expect that this new canonical techniques can be applied to other logics, for instance in the axiomatization of the full version of PDL with intersection containing the star operation and possibly the converse operation. Another novelty in the paper is the proof that intersection is modally definable in a language with propositional quantifiers and that the rule (INT) in a sense simulates the quantifier rule for universal quantification in the context of the definition of intersection. This is a new look on the nature of some context dependent rules like (INT). Let us note that the notion of admissible form, used in the formulation of (INT) is too complicated. So a general question is whether (INT) can be formulated by a more simple structure of admissible forms, like for instance in (INT_1). Connected with this is the following general question: is it possible to eliminate the rule INT in general and to replace it with a finite or an infinite set of additional axiom schemes? We conclude with a remark that $PDL_0^{\cap}$ is decidable. This follows from Danecki's theorem about decidability of the full PDL with intersection [4].

References

[1] P. Balbiani. A new proof of completeness for a relative modal logic with composition and intersection. *Journal of Applied Non-Classical Logics,* to appear.

[2] P. Balbiani, L. Fariñas del Cerro. Complete axiomatization of a relative modal logic with composition and intersection. *Journal of Applied Non-Classical Logics,* 8:325–335, 1998.

[3] J. van Benthem. A note on dynamic arrow logics. J. van Eijck, A. Visser (Editors): *Logic and Information Flow.* 15–29, MIT Press, 1994.

[4] R. Danecki. Non-deterministic propositional dynamic logic is decidable. A. Skowron (Editor): *Computation Theory.* Lecture Notes in Computer Science 208, 34–53, Springer-Verlag, 1985.

[5] R. Fagin, J. Halpern, Y. Moses, M. Vardi. Reasoning About Knowledge. MIT Press, 1995.

[6] L. Fariñas del Cerro, E. Orlowska. DAL — a logic for data analysis. *Theoretical Computer Science,* 36:251–264, 1985.

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

[8] D. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions in tense frames. U. Monnich (Editor): *Aspects of Philosophical Logic.* 67–89, Reidel, 1981.

[9] G. Gargov, S. Passy, A note on Boolean modal logic. P. Petkov (Editor): *Mathematical Logic.* 299–309, Plenum Press, 1990.

[10] R. Goldblatt. Axiomatising the Logic of Computer Programming. Lecture Notes in Computer Science 130, Springer-Verlag, 1982.

[11] V. Goranko. Axiomatizations with context rules of inference in modal logic. *Studia Logica,* 61:179–197, 1998.

[12] V. Goranko, S. Passy. Using the universal modality: gains and questions. *Journal of Logic and Computation,* 2:203–233, 1992.

[13] D. Harel. Recurring dominoes: making the highly undecidable highly understandable. M. Karpinski (Editor): *Foundations of Computation Theory.* Lecture Notes in Computer Science 158, 177–194, Springer-Verlag, 1983.

[14] D. Harel. Dynamic logic. D. Gabbay, F. Guenthner (Editors): *Handbook of Philosophical Logic.* Volume 2, 497–604, Reidel, 1984

[15] G. Hughes, M. Cresswell. A Companion to Modal Logic. Methuen, 1984.

[16] M. Kracht. Syntactic codes and grammar refinement, *Journal of Logic, Language and Information,* 4:41–60, 1995.

[17] M. Marx, Y. Venema. Multi-Dimensional Modal Logic. Kluwer, 1997.

[18] J.-J. Meyer. A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic,* 29:109–136, 1988.

[19] G. Mirkowska. PAL — propositional algorithmic logic. *Fundamenta Informaticæ,* 4:675–760, 1981.

[20] E. Orlowska. Kripke models with relative accessibility and their applications to inference from incomplete information. *Mathematical Problems in Computation theory.* Banach Center Publications volume 21, 329–339, Polish Scientific Publishers, 1988.

[21] S. Passy, T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation,* 93:263–332, 1991.

[22] A. Petkov. Propositional dynamic logic in two and more dimensions. D. Skordev (Editor): *Mathematical Logic and its Applications.* 323–329, Plenum Press, 1987.

[23] M. de Rijke. A system of dynamic modal logic. *Journal of Philosophical Logic,* 27:109–142, 1998.

[24] K. Segerberg. A completeness theorem in the modal logic of programs. T. Traczyk (Editor): *Universal Algebra and Applications.* Banach Center Publications volume 9, 31–36, Polish Scientific Publishers, 1982.

[25] T. Tinchev, D. Vakarelov. Propositional dynamic logics with counters and stacks. A. Skowron (Editor): *Computation Theory.* Lecture Notes in Computer Science 208, 364–374, Springer-Verlag, 1985.

[26] D. Vakarelov. Modal rules for intersection. Abstract of the 10-th International Congress of Logic, Methodology and Philosophy of Science, Florence, Italy, 1995, page 183.

[27] D. Vakarelov. Modal rules for intersection. Unpublished manuscript.