Proof theoretic methodology for Propositional Dynamic Logic

Daniel Leivant
Department of Computer Science
Cornell University
Ithaca, NY 14853, USA

Abstract. We relate by syntactic techniques finitary and infinitary axiomatizations for the iterator-construct $*$ of Propositional Dynamic Logic PDL. This is applied to derive the Interpolation Theorem for PDL, and to provide a new proof of the semantic completeness of Segerberg's axiomatic system for PDL.

Contrary to semantic techniques used to date, our proof of completeness is relatively insensitive to changes in the language and axioms used, provided some minimum syntactic closure properties hold. For instance, the presence of the test-operator adds no difficulty, and the proof also establishes the Interpolation Theorem and the closure under iteration of a constructive variant of PDL.

Introduction.

The synthesis of modal logic and verification logic, in the guises of Dynamic Logic and Algorithmic Logic, has considerably enhanced our understanding of the process of reasoning about programs. These two logics, introduced by Pratt [Pr1] and Salwicki [Sa] respectively, proceed beyond the Floyd/Hoare tradition in that they permit unrestricted specification of extensional (i.e. "before-after") properties of programs, by introducing a modal operator $[\alpha]X$ to express "the statement X is true after each execution of the (non-deterministic) program α." For example, $\neg[\alpha]\underline{false}$ expresses the termination of some execution of α. Moreover, the propositional fragment PDL of Dynamic Logic, invented by Fischer and Ladner [FL], was proved to enjoy such pleasant properties as decidability and axiomatizability.

Axioms for PDL were invented by K. Segerberg [Se], and several proofs were proposed for their semantic completeness [Be,KP,Ga,Mi,Ni,Pa1,Pa2,Pr2,Pr3,Se]. It seems that all correct proofs among these are essentially stylistic variants of the one discovered independently by D.M. Gabbay [Ga] and R. Parikh [Pa1]. This includes the proof in [Ni], where the use of a proof theoretic machinary is a redundant complication. Remarkably, all these presentations establish completeness only for test-free formulas of PDL (see 1.1 below), claiming extendibility to the entire language. It is indeed quite clear that the formula $[?X]Y \Leftrightarrow (X \rightarrow Y)$ completely axiomatizes the test construct $?$. It is not that obvious however that Segerberg's Induction Axiom still axiomatizes the iterator construct $*$ when tests are also present. In fact, the extension of the Gabbay-Parikh proof to full PDL is quite tedious.

We present a proof theoretic analysis of PDL which provides a totally new proof of the semantic completeness of Segerberg's axioms. First, we observe that PDL is easily axiomatized by a calculus D that uses an infinitary inference rule OMEGA: from $Y \to [\alpha^n]X$ for all natural numbers n, conclude $Y \to [\alpha^*]X$. This kind of completeness result has been known for long (in particular for Algorithmic Logic). However, our proof, based on Smullian's Tableaux Method, seems to add insight.

We then show that Segerberg's axioms are complete for D, by induction on proofs in D. The only non-trivial step is for the infinitary inference rule OMEGA. Our main observation is that if in a proof of D a very long branch can mention only a given number of formulas, then that branch must contain a repetitive pattern. Specifically, if P_n is a proof of $Y \to [\alpha^n]X$, and n is sufficiently large, then there is a repetitive pattern in P_n, from which one concludes that $Y \to [\alpha^k][(\alpha^m)^*]X$ is deducible from Segerberg's axioms for some m,k where $m < v$ for a certain v that depends only on Y. Applying this argument repeatedly for several values for n, it easily follows that $Y \to [\alpha^*]X$ is deducible.

Using a similar analysis of proofs in D we prove the Interpolation Theorem for PDL: if $X \to Y$ is a theorem, then there is a formula K, using only atomic propositions and programs common to X and Y, such that $X \to K$ and $K \to Y$ are both theorems.

We believe that this technique elucidates the relation between finitary and infinitary axiomatizations of programming logics, clarifying in particular the role of Segerberg's Induction Axiom. A technical advantage of our method is its adaptability. The completeness of Segerberg's Induction for uses of OMEGA depends only on simple and common syntactic relations between the premises and conclusions of inference rules. Thus, Interpolation and appropriate completeness results can be proved for both subtheories and supertheories of PDL.

A subtheory of some interest is Constructive PDL, i.e. PDL based on Constructive (Intuitionistic) Logic (see section 3). For technical reasons, the system D we use actually generates Constructive PDL. On the side of extensions, we hope that our technique, and more generally the syntactic methodology on which it is based, will show their usefulness in the study of full Dynamic Logic and of Process Logics.

1. Propositional Dynamic Logic.

1.1 Syntax.

There are letters a_i and p_i $(i=0,1,\ldots)$ for atomic programs and propositions respectively, for which we use $a,b,\ldots$ and $p,q,\ldots$ as syntactic variables. Compound formulas are built up inductively from the atoms: if α,β are programs and X,Y are formulas then $\alpha;\beta$, $\alpha\vee\beta$, α^* and $?X$ are programs, and $\neg X$, $X\wedge Y$, $X\to Y$ and $[\alpha]X$ are formulas. We also write $\alpha\beta$ for $\alpha;\beta$. $\langle\alpha\rangle$ abbreviates $\neg[\alpha]\neg$, α^k $(k\geq 1)$ abbreviates $\alpha(\alpha\ldots(\alpha\alpha)..)$ with k terms; similarly $[\alpha]^kX$ abbreviates $[\alpha]\ldots[\alpha]X$. $[\alpha^0]X$ and $[\alpha]^0X$ are identified with X.

1.2 Semantics.

A (Kripke-) model of PDL consists of a set W of <u>states</u>, a set $V_p \subset W$ for each p, and a binary relation $\to^a$ on W for each a. We use $\to^a$ in infix notation. If $s \in V_p$ then we write $s \models p$ and say that p is <u>true at</u> s. This interpretation is extended to compound programs and formulas:

$\to^{\alpha\beta}$ is $\to^{\alpha} \circ \to^{\beta}$ (composition)

$\to^{\alpha\vee\beta}$ is $\to^{\alpha} \cup \to^{\beta}$ (union)

$\to^{\alpha^*}$ is $\cup_i (\to^{\alpha})^i$ (iteration)

$\to^{?X}$ is $\{\langle s,s\rangle : s \models X\}$

$s \models \neg X$ iff not $s \models X$

$s \models X\to Y$ iff $s \models Y$ or not $s \models X$

$s \models [\alpha]X$ iff $t \models X$ whenever $s \to^{\alpha} t$

1.3 Segerberg's Axioms.

K. Segerberg [Se] proposed an axiomatization of PDL of which the following system S is a slight variant.

<u>Axiom schemes</u>.

A1. All propositional tautologies.

A2. $[\alpha]$<u>true</u>

A3. $[\alpha](X\wedge Y) \Leftrightarrow [\alpha]X \wedge [\alpha]Y$

A4. $[\alpha\beta]X \Leftrightarrow [\alpha][\beta]X$

A5. $[\alpha\vee\beta] \Leftrightarrow [\alpha]X \wedge [\beta]Y$

A6. $[\alpha^*]X \Leftrightarrow X \wedge [\alpha][\alpha^*]X$

A7. $[?X]Y \Leftrightarrow (X \to Y)$

Inference rules.

R1. If $\vdash X$ and $\vdash X \to Y$ then $\vdash Y$ (Detachment)

R2. If $\vdash X \to Y$ then $\vdash [\alpha]X \to [\alpha]Y$ (Generalization)

R3. If $\vdash X \to [\alpha]X$ then $\vdash X \to [\alpha^*]X$ (Induction)

1.3.1. PROPOSITION. S is sound for the Kripke semantics for PDL. I.e. every theorem of S is true in every state of every model.

PROOF. Straightforward by induction on the length of proofs.□

1.3.2. REMARKS.

1. In R2 X and $[\alpha]X$ may be dropped, by A2.
2. By R2 and A3 we get $[\alpha](X \to Y) \to . [\alpha]X \to [\alpha]Y$.
3. By R3, R2 and A3 applied to $X \equiv Y \wedge [\alpha^*](Y \to [\alpha]Y)$ we get the induction axiom: $[\alpha^*](Y \to [\alpha]Y) \to [\alpha^*]Y$

1.4 Eliminability of the choice construct $\vee$.

The non-deterministic choice construct $\vee$ is important on several accounts. The constructs ;, $\vee$ and * of PDL are analogous to the constructs of regular expressions. In addition, $\vee$ is important in allowing the simulation of deterministic branching constructs, e.g. if X then α else β is simulated by $(?X)\alpha \vee (?\neg X)\beta$. However, $\vee$ is eliminable in PDL:

1.4.1. PROPOSITION. For each formula X of PDL there is a formula X' without $\vee$ such that $X \Leftrightarrow X'$ is a theorem of S.□

2. Complete semi-calculi for PDL.

2.1. An infinitary sequential calculus for PDL.

Our analysis of PDL is based on the method of sequential calculi, invented by Gerhard Gentzen (see e.g. [Ta]). In what follows a sequent is a pair $f \vdash g$, where f,g are finite sets of PDL formulas. f is the antecedent and g the succedent of $f \vdash g$. Intuitively, $f \vdash g$ expresses that the conjunction of all formulas in f implies one of the formulas in g. We write f,X for $f \cup \{X\}$, and $[\alpha]f$ for $\{ [\alpha]X : X \in f \}$. We use $\to$ and $\neg$ as the only propositional connectives.

We define a calculus of sequents D. The rules of inference of D are as follows.

$$\neg R: \quad \frac{f, X \vdash g}{f \vdash g, \neg X} \qquad\qquad \neg L: \quad \frac{f \vdash g, X}{f, \neg X \vdash g}$$

$$\rightarrow R: \quad \frac{f, X \vdash g, Y}{f \vdash g, X \rightarrow Y} \qquad\qquad \rightarrow L: \quad \frac{f \vdash g, X \qquad f, Y \vdash g}{f, X \rightarrow Y \vdash g}$$

$$;R: \quad \frac{f \vdash g, [\alpha][\beta]X}{f \vdash g, [\alpha\beta]X} \qquad\qquad ;L: \quad \frac{f, [\alpha][\beta]X \vdash g}{f, [\alpha\beta]X \vdash g}$$

$$\vee R: \quad \frac{f \vdash g, [\alpha]X \qquad f \vdash g, [\beta]X}{f \vdash g, [\alpha\vee\beta]X} \qquad\qquad \vee L: \quad \frac{f, [\alpha]X, [\beta]X \vdash g}{f, [\alpha\vee\beta]X \vdash g}$$

$$*L: \quad \frac{f, X, [\alpha][\alpha^*]X \vdash g}{f, [\alpha^*]X \vdash g} \qquad\qquad *R: \quad \frac{\{f \vdash g, [\alpha]^n X\}_{n=0,1,\ldots}}{f \vdash g, [\alpha^*]X}$$

$$?R: \quad \frac{f, X \vdash g, Y}{f \vdash g, [?X]Y} \qquad\qquad ?L: \quad \frac{f \vdash g, X \qquad f, Y \vdash g}{f, [?X]Y \vdash g}$$

$$\text{GEN:} \quad \frac{f \vdash X}{[\alpha]f \vdash [\alpha]X}$$

$$\text{WEAK:} \quad \frac{f \vdash g}{f' \vdash g'} \qquad \text{where } f' \supset f \text{ and } g' \supset g.$$

For each rule the sequent(s) above the line is (are) the _premise(s)_, and the one below is the _conclusion_. The displayed formulas are _active_. A sequent $f \vdash g$ is _initial_ if $f \cap g \neq \emptyset$. A _proof_ D is a tree P of sequents, where all branches are finite, the leaves are initial, and the internal nodes relate according to the inference rules. The sequent at the root is _proved_ by the proof, and X is _proved_ if $\vdash X$ is.

As we shall readily see, it is fairly easy to prove that D is semantically complete. An infinitary rule similar to (*R) has been proposed as part of a complete axiomatization of Algorithmic Logic. However, we consider D merely as a tool. In our opinion, the semantical completeness of D does not, by itself, enhance our understanding of PDL. By comparison, a complete axiomatization of First Order Arithmetic using a similar infinitary rule has been known for long, but it does not contribute directly to our understanding of the axiomatization of arithmetic.

2.2 An auxiliary extension of PDL.

In PDL atomic programs are also assumed to be non-deterministic. It will be convenient to explicitly spell out executions of atomic programs in an extension PEL of PDL. The syntax of PEL contains, in addition to the atoms of PDL, letters c_i $(i=1,2\ldots)$, for which we use c as a syntactic variable. An <u>execution</u> is an expression $a(c)$; we use e and m as syntactic variables for executions and finite sequences of executions, respectively. If e is $a(c)$ then we write $\bar{e}$ for a, and we let $mf = \{mX \mid X \in f\}$. Compound formulas and programs are defined as for PDL, except that when X is a formula then so is eX.

A (<u>Kripke</u>) <u>model</u> for PEL is defined as for PDL, with the additional clauses:
(1) $\to^{a(c)}$ is a partial function (i.e. a univalent relation) on the set W of states.
(2) $\to^{a}$ is $\cup_c \to^{a(c)}$.
Thus, $[a]X$ is understood as an abbreviation for $\forall c\ a(c)X$. Clearly, every model of PEL is also a model PDL, and every model of PDL is uniquely extendable to a model of PEL.

We define an infinitary (semi-) calculus E for PEL. The inference rules are those of D as displayed above, except GEN and WEAK, but with the syntactic variable m in front of all displayed formulas. For instance,

$$\to R:\quad \frac{f, mX \vdash g, mY}{f \vdash g, m(X\to Y)}$$

In addition, there are rules spelling out the meaning of executions:

$$[\,]R:\quad \frac{f \vdash g, ma(c)X}{f \vdash g, maX};\qquad [\,]L:\quad \frac{f, ma(c)X \vdash g}{f, maX \vdash g}$$

(c should not occur
in the lower sequent)

Let T be a tree with all branches finite. By _tree-induction_ (_on_ T) we mean the following principle. Suppose a property A is true of all leaves of T, and suppose that A is true of a node N of T whenever A is true of all children of N in T. Then A is true of the root of T.

Let $\wedge f$ and $\vee f$ denote the conjunction and disjunction of all formulas in f, respectively. Let $\neg f = \{ \neg X \mid X \in f \}$.

2.2.1. PROPOSITION. (Soundness of E). If $f \vdash g$ is proved in E then $\wedge f \to \vee g$ is true in every state of every model.
PROOF. By tree-induction on the proof of $f \vdash g$. □

2.2.2. PROPOSITION. (the Inversion Principle) Suppose $f \vdash g$ is a premise of $f' \vdash g'$ by an inference of E. Then $\wedge f \ \& \ \wedge\neg g \to \wedge f' \ \& \ \wedge\neg g'$.□

2.3. Completeness of D for E.

A formula of PEL is _simple_ if it is of the form mX where X is a PDL formula. A set f of simple formulas is _stratified_ if all executions in f can be listed without repetition, $e_1, \ldots, e_k$, so that no occurrence of e_i in f is in the scope of e_j if $i<j$. We say that that order _stratifies_ f. A sequent $f \vdash g$ of E is _stratified_ if $f \cup g$ is stratified. For each stratified sequent $f \vdash g$ of PEL we define a formula $\{f \vdash g\}$ of PDL, by induction on the number of distinct executions in $f \vdash g$. If $f \vdash g$ is in PDL then $\{f \vdash g\}$ is $\wedge f \to \vee g$. Fix some order O of executions that stratifies $f \cup g$. Let e be the O-first executions in $f \vdash g$; so $f \vdash g$ is $ef_1, f_2 \vdash eg_1, g_2$, where e does not occur in f_1, f_2, g_1, g_2. Define $\{f \vdash g\} = \{f_2 \vdash g_2, [\bar{e}]\{f_1 \vdash g_1\}\}$.

Suppose $f = f_0 \cup e_1 f_1 \cup \cdots \cup e_k f_k$, $g = g_0 \cup e_1 g_1 \cup \cdots \cup e_k g_k$, where each e_i occurs only in f_i, g_i. Then

$\{f \vdash g\} = \{ f_0 \vdash g_0, [\bar{e}_1]\{f_1 \vdash g_1\}, \ldots, [\bar{e}_k]\{f_k \vdash g_k\}\}$.

I.e. the definition of $\{f \vdash g\}$ is independent of the choice of stratifying order. In particular, we have:

2.3.1. LEMMA. If e does not occur in f_1, g_1, then

$\{f_1, ef_2 \vdash g_1, eg_2\} = \{f_1 \vdash g_1, [\bar{e}]\{f_2 \vdash g_2\}\}$.□

2.3.2. LEMMA. (i) Suppose e does not occur in f, g. Then

$\{f \vdash g, meX\} = \{f \vdash g, m\bar{e}X\}$.

(ii) $\{f, mX \vdash g, mY\} = \{f \vdash g, m(X \to Y)\}$.

PROOF. (i) By induction on the number of executions in f, g, m that 0-precede e. The basis is 2.3.1.

(ii) By induction on the number of executions occurring in m or 0-preceding an execution in m.□

2.3.3. LEMMA. If $f \vdash g$ is a stratified sequent proved in E, then $\{f \vdash g\}$ is a theorem of D.

PROOF. By induction on the proof of $f \vdash g$ in E. Note that a premise sequent to a stratified sequent is stratified.

Basis. $f \vdash g$ is $mX \vdash mX$ where X is a PDL formula. $\{mX \vdash mX\}$ is seen to be a theorem of D by a trivial induction on the length of m.

Induction Step. By inspection on the inference rules of E. The cases for $\to$R and []R are lemma 2.3.2. Other cases are tedious but straightforward. (As should be expected, the rules of Genralization and Weakening are essential).□

2.4 Semantic completeness of E and D.

We show that E is semantically complete. By 2.3.4 this implies the semantic completeness of D. Our method is an adaptation of Smullian's Tableaux Method [S]. For a PEL formula X, a failed attempt to generate a proof of X in E yields a model and state at which X is false. The states of the model are all execution sequences $e_1 \cdots e_k$; such a sequence may be understood as the state reached by successively executing $e_1, \ldots, e_k$.

Fix a canonical enumeration of all PEL formulas. This can be extended to an enumeration of all sequents, and hence to an enumeration $(I_k)_k$ of all inferences of E. If I_k has the sequent σ as consequence, we say that k is a σ-index.

A formula is basic if it is of the form mp. A sequent is basic if all formulas therein are basic.

The canonical resolution $CR(f \vdash g)$ of a sequent $f \vdash g$ is the tree T of sequents and inference rules built up inductively from the root as follows.

(i) The sequent at the root of T is $f \vdash g$.

Let $N = (\sigma, ??)$ be a node of T.

(ii) If σ is initial then N is a leaf of T.

(iii) If σ is basic, then N is a leaf, and if not initial it is said to be critical;

(iv) Otherwise, let k be the first σ-index different from all σ-indices used below N in T. Set $??=I_k$, and for the children $N_i = (\sigma_i, ??)$ of N in T set σ_i to be the i'th premise of I_k.

Suppose that $CR(f \vdash g)$ is not a proof of E. The canonical counter-example $CCE(f \vdash g)$ is a model defined as follows.

(1) The states are the sequences of executions (including the empty sequent Λ).

(2) For an atomic program a, $m \to^a m'$ just in case $m' = em$ for some $e = a(c)$.

There are two cases for the definition of satisfaction of atomic formulas.

Case 1. There is an infinite branch $B = (f_i \vdash g_i)_{i=1}^{\infty}$ in $CR(f \vdash g)$. Let $m \models p$ just in case $mp \in \cup_i f_i$.

Case 2. $CR(f \vdash g)$ is finite. Then there is a critical leaf $N = f' \vdash g'$. Let $m \models p$ just in case $mp \in f'$.

LEMMA 2.4.1. (i) Suppose $CR(f \vdash g)$ has an infinite branch B as above. If $X \in \cup_i f_i$ then X is true at Λ in $CCE(f \vdash g)$. If $X \in \cup_i g_i$ then X is false at Λ.
(ii) Suppose $N = f' \vdash g'$ is a critical node in $CR(f \vdash g)$. If $f'' \vdash g''$ is a sequent under N, then all formulas in f'' are true in $CCE(f \vdash g)$ at Λ, and all formulas in g'' are false at Λ.

PROOF. (i) By induction on X.
(ii) By induction on the number of inferences from N to $f'' \vdash g''$, using 2.2.2.□

THEOREM 2.4.2 E and D are semantically complete.

PROOF. Suppose X is not a theorem of E. Then $CR(\vdash X)$ is not a proof. Then, by 2.4.1 X is false in the model $CCE(\vdash X)$ at Λ.□

2.5 A finitary variant of D.

Fischer and Ladner [FL] have shown that if for a PDL formula X there is amodel and state s at which X is true, then X is true at a state s in a finite model of size $\leq 2\#|X|$, where $\#$ denotes exponentiation and $|X|$ is the size of X. We show that this implies that if $[\alpha]^n X$ is valid for all $n \leq k = 2 \# |[\alpha^*]X|$, then $[\alpha^*]X$ is valid.

Suppose that $[\alpha]^n X$ is valid for all $n \leq k$. If $[\alpha^*]X$ is not valid, then it is false at some state s of a model with at most k states. Assume that, in that model,

$$s = s_1 \to^{\alpha} s_2 \to^{\alpha} \dots \to^{\alpha} s_m = t.$$

If $m \leq k$ then $s \models [\alpha]^m X$ by assumption, so $t \models X$. If $m > k$ then, since there are only $\leq k$ states in the model, there must be a subsequence

$$s = s_{i_1} \to^{\alpha} \cdots \to^{\alpha} s_{i_n} = t,$$

without repetitions, where $n \leq k$. Since $s \models [\alpha]^n$ by assumption, $t \models X$ follows as before. Thus $s \models [\alpha^*]X$.

We can similarly see that, given f, g, and X, the inference rule

$$\frac{\{f \vdash g, [\alpha]^n X\}_{n=1}^{k}}{f \vdash g, [\alpha^*]X}$$

$k = 2 \# (|f| + |g| + |X|)$, is valid. Thus, the infinitary rule (*R) can be modified to a finitary rule, but where the number of premises depends on the instance.

3. Constructive PDL.

3.1. Constructive PDL as an auxiliary system.

Constructive (or Intuitionistic) Logic differs from usual (or "classical") logic in its insistence on a constructive interpretation of the logical operators, rather than a truth-values semantic. E.g. $A \vee B$ is constructively true if either A or B can be asserted. Thus, $p \vee \neg p$ is not a valid schema; but $\neg\neg(p \vee \neg p)$ is valid, because assuming $\neg(p \vee \neg p)$ leads to a contradiction.

Constructive Logic stemmed from the work of L.E.J. Brouwer in the Philosophy of Mathematics in the early years of the century, and has enjoyed a regain of interest since World War II, for some part because as the logic underlining constructive reasoning it is related to Recursion Theory. A concern for constructive reasoning has also motivated the incorporation of CL into programming logics by R. Constable [CO].

A constructive variant CPDL of PDL has not been studied to date, and we do not propose to carry out such a study here. However, the technical aspect of our analysis is considerably simplified if we treat constructive rather than usual PDL. (The suitability of constructive logics for syntactic analysis has been well-known in Proof Theory.) Moreover, the results we obtain for CPDL quickly translate into similar results for PDL. Thus, we kill two birds with one stone: simplify the proofs, and obtain results simultaneously for both usual and constructive PDL.

In Constructive Logic none of the propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, is definable in terms of the others. A complete system for constructive PDL should therefore use all connective, and should similarly differentiate between $\langle\alpha\rangle X$ and $[\alpha]X$. We use however only $\rightarrow$, $\neg$ and $[\,]$, since CPDL is not our main concern here. At the same time our proofs can be easily extended to full CPDL.

3.2 Formal systems for Constructive PDL.

A constructive variant CS of Segerberg's axiom system S is obtained simply by using only the constructively valid propositional schemas. E.g. $p \vee \neg p$ is excluded.

Given a PDL formula X, Let X^o result from X by inserting $\neg\neg$ in front of atomic subformulas and in front of each logical operator. For example, $(X\rightarrow Y)^o$ is $\neg\neg(X^o\rightarrow Y^o)$, and $([\alpha]X)^o$ is $\neg\neg[\alpha](X^o)$.

By standard proof theoretic methods it is easy to prove:

3.2.1. PROPOSITION. A formula X is a theorem of S iff X^o is a theorem of CS.□

(Actually, X^o can be taken here as the result of inserting $\neg\neg$ only in front of atomic subformulas.)

Constructive variants CD and CE of D and E can be obtained by restricting the kind of sequents $f \vdash g$ used to ones with $|g| \leq 1$. Again, routine proof theoretic methods can be used to show:

3.2.2. PROPOSITION. A PDL formula X is a theorem of D iff X^0 is a theorem of CD. (A similar statement holds for PEL).□

4. Semantic completeness of Segerberg's axioms.

4.1 Counting in S.

4.1.1. LEMMA. The following schemas are provable in S.

(i) $[(\alpha^w)^*][\alpha]X \rightarrow [\alpha][(\alpha^w)^*]X$.

(ii) $\wedge_{a<w}[\alpha^a][(\alpha^w)^*]X \rightarrow [\alpha^*]X$.

(iii) If $s1,\ldots,sa$ are distinct modulo w then

$$\wedge_{i<k}[\alpha]^i X \ \& \ \wedge_{a<w}[\alpha^{sa}][(\alpha^w)^*]X \rightarrow [\alpha^*]X,$$

where $k = \max(s1,\ldots,sa)$.□

4.2 Proofs from non-initial sequents.

A proof in CD of $f \vdash X$ from $F = \{f_i \vdash X_i\}_i$ is a proof-figure of CD where leaves are initial or in F. The positive closure PC(f) of a set f of PDL formulas is the smallest set $g \supset f$ such that

$(X \rightarrow Y) \in g \implies Y \in g;$

$[\alpha]X \in g \implies X \in g;$

$[\alpha\beta]X \in g \implies [\alpha][\beta]X \in g;$

$[\alpha \vee \beta]X \in g \implies [\alpha]X,\ [\beta]X \in g;$

$[\alpha^*]X \in g \implies [\alpha\alpha^*]X \in g;$

$[?X]Y \in g \implies Y \in g.$

4.2.1. LEMMA. Suppose P is a proof in CD of $f \vdash [\alpha]q$ from $\{f_i \vdash q\}_i$, where q does not occur in f. If $f' \vdash [\alpha]^r q$ is a sequent in P (under a non-initial leaf) then $PC(f') \subset PC(f)$.

PROOF. By tree-induction on P.□

4.2.2. LEMMA. Assume P is a proof in CD of $f \vdash [\alpha]^r X$ from $\{f_i \vdash X\}_i$, where $X \notin PC(f)$. Then P is $P'[X/q]$ for some proof P' of $f \vdash [\alpha]q$ from $\{ f_i \vdash q \}_i$, where q is new.

PROOF. By tree-induction on P.□

4.3 Completeness and closure under iteration of Segerberg's axioms.

We write $f \vdash_{CS} X$ if $\wedge f \rightarrow X$ is a theorem of CS. For a proof P, $P[X/q]$ will denote the result of substituting X for the letter q in P.

4.3.1. LEMMA. (i) If P is a proof in CD deriving $f \vdash X$, then $f \vdash_{CS} X$.
(ii) If P is a proof in CD of $f \vdash [\alpha]q$ from $\{ f_i \vdash q \}_i$, and $f_i \vdash_{CS} X$ for $i \in I$, then $f \vdash_{CS} [\alpha]X$.

PROOF. By tree-induction on P, simultaneously for (i) and (ii). The basis is trivial. The induction step is by cases on the last inference of P. The only non-trivial case is (*R). For both (i) and (ii) it suffices to prove that if P is a proof of the form

$$\frac{\begin{matrix} P_n \\ \{ f \vdash [\alpha]^n X \}_n \end{matrix}}{f \vdash [\alpha^*]X}$$

then $f \vdash_{CS} [\alpha^*]X$.

Let $u = |PC(f)|$, $v = 2^u$, and let d be such that $[\alpha]^d X \notin PC(f)$. Let $h > v+d$. By 4.2.2. and 4.2.1 P is of the form

$$\begin{matrix} Q_i \\ \{ f_i \vdash [\alpha]^d X \}_i \\ R \\ f \vdash [\alpha]^h X \end{matrix}$$

where R is $R'[[\alpha]^d X/q]$ for some R' deriving $f \vdash [\alpha]^{h-d} q$ from $\{f_i \vdash q\}_i$.

Also R' is of the form

$$\begin{matrix} \{ f_j \vdash q \}_{j\in I0} \\ R_j^1 \\ \{ f_j \vdash [\alpha]q \}_{j\in I1} \\ R_j^2 \\ \{ f_j \vdash [\alpha]^2 q \}_{j\in I2} \\ \cdots\cdots \end{matrix}$$

By 4.2.1 and our choice of v there are some m,n $(m < n < v)$ such that $\{f_j\}_{j\in Im} = \{f_j\}_{j\in In}$. Thus we extract from R' proofs T_j $(j \in In)$ of $f_j \vdash [\alpha]^r[\alpha]^m q$ from $\{f_j \vdash [\alpha]^m q\}_{j\in In}$, where $r = n-m$.

Since q does not occur in $PC(f_j)$ we get by 4.2.1 that T_j $(j \in In)$ is $T_j'[[\alpha]^m q/p]$ for some T_j' deriving $f_j \vdash [\alpha]^r p$ from $\{f_j \vdash p\}_{j\in In}$.

Let $Y = \vee_{j\in In} \wedge f_j$. Then $f_j \vdash_{CS} Y$ trivially, so by induction assumption $f_j \vdash_{CS} [\alpha]^r Y$ for each $j \in In$. From this it follows, by the Induction Rule of S, that

(1) $f_{jn} \vdash_{CS} [(\alpha^r)^*]Y$.

Also, by induction assumption, $f_j \vdash_{CS} [\alpha]^m X$ for each $j \in Im = In$, and so $Y \vdash_{CS} [\alpha]^m X$ wherefrom $[(\alpha^r)^*]Y \vdash_{CS} [(\alpha^r)^*\alpha^m]X$. Combined with (1), $f_j \vdash_{CS} [(\alpha^r)^*\alpha^m]X$. By induction assumption (applied to the bottom portion of P_h) it follows that $f \vdash_{CS} [\alpha^s(\alpha^r)^*\alpha^m]X$, where $s = h-n$. So $f \vdash_{CS} [\alpha^{s+m}(\alpha^w)^*]X$ by 4.1.1(i), where w is the least common multiplier of $2,\ldots,v$.

Now let $h' \not\equiv h \pmod w$. The same argument yields s',m' where $s'+m' \not\equiv s+m$

(mod w), for which $f \vdash_{CS} [\alpha^{s'+m'}(\alpha^w)^*]X$. Repeating the argument w times yields by 4.1.1(iii) $f \vdash_{CS} [\alpha^*]X$.□

As an immediate consequence of 4.3.1 we have:

4.3.2. THEOREM. CS is closed under iteration: if $f \vdash_{CS} [\alpha]^n X$ for all (sufficiently large) n, then $f \vdash_{CS} [\alpha^*]X$.□

4.3.3. THEOREM. Segerberg's axioms system S is semantically complete (and so also closed under iteration).

PROOF. X true ==> X is provable in D (by 2.2.4)
==> X^o is provable in CD (by 3.2.2)
==> X^o is provable in CS (by 4.4.1)
==> X is provable in S (by 3.2.1).□

5. Interpolation.

5.1. Maehara's method.

Let f be a set of PDL formulas. Denote by $L(f)$ the set of atomic propositions and programs in f. We wish to prove that PDL satisfies the interpolation property: If $X \to Y$ is a theorem of PDL, then there is a formula K, said to be an interpolant for X,Y, such that $L(K) \subset L(X) \cap L(Y)$, with $X \to K$ and $K \to Y$ both true.

We adapt a method of Maehara (see [Ta]) for proving interpolation using sequential calculi. Given a sequent $f \vdash X$ and a partition $f^-;f^+$ of f, K is an interpolant for $f^-;f^+ \vdash X$ if $L(K) \subset L(f^-) \cap L(f^+,X)$, with $\wedge f^- \to K$ and $\wedge(f^+,K) \to X$ both true. It is fairly easy to construct an interpolant for any partitioned $f^-;f^+ \vdash X$ if $f \vdash X$ is derived in CD by a proof P that does not use the infinitary rule (*R). This is done by induction on P. E.g., suppose that the last inference of P is $(\to L)$,

$$(*) \qquad \frac{g \vdash X \qquad g,Y \vdash Z}{g,X\to Y \vdash Z}$$

Suppose that the given partition of f is $f^-,X\to Y;f^+$. Then by induction assumption there are formulas K_1,K_2 such that the sequents $f^+ \vdash K_1$, $f^-,K_1 \vdash X$, $f^-,Y \vdash K_2$ and $f^+,K_2 \vdash Z$ are all true. Let $K \equiv K_1 \to K_2 \equiv [?K_1]K_2$. Then $f^-,X\to Y \vdash K$ and $f^+,K \vdash Z$ are both true.

Let P be a proof in CD of $f \vdash X$. Let f^-, f^+ be a partition π of f. Let $g \vdash Y$ be a sequent of height d in P. The partition $g^-;g^+$ of g induced by π is defined by induction on d. For $d=0$, $g=f$, $g^-=f^-$, $g^+=f^+$. Let $g \vdash Y$ be the premise of $h \vdash Z$ by $(*R)$. Then $g=h$, $g^-=h^-$, $g^+=h^+$. Otherwise, g^- and g^+ are defined by the appropriate case in Maehara's method. For instance, for $(\to L)$ as in $(*)$ above, if $g^-, X\to Y; g^+$ is the given partition for the derived sequent, then $g^+;g^-$ is the partition for the left premise, and $g^-, Y; g^+$ the one for the right premise. Similarly, for $\dfrac{f \vdash X}{[\alpha]f \vdash [\alpha]X}$, if $[\alpha]f^-;[\alpha]f^+$ is the partition obtained for the derived sequent, then $f^-;f^+$ is the partition for the premise.

5.2 The limit of linear transformations on a set of PDL formulas.

Suppose $Y_1,\ldots,Y_k$ are PDL formulas, and let

$(*)$ $\quad Y_i' \equiv \wedge_j\, \beta_{ij} Y_j \qquad i = 1,\ldots,k.$

where βY stands for $[\beta]Y$. We wish to show that for each i the infinite conjunction $Y_i \wedge Y_i' \wedge (Y_i')' \wedge Y_i^{(3)} \wedge \ldots$ is expressible in PDL. Formally, $(*)$ suggests a linear transformation. Writing $\mathbf{Y}$ for the vector $\begin{pmatrix} Y_1 \\ \cdot \\ Y_k \end{pmatrix}$, and (β) for the matrix of programs $\begin{pmatrix} \beta_{11} & \cdots & \beta_{1k} \\ \cdots & \cdots & \cdots \\ \beta_{k1} & \cdots & \beta_{kk} \end{pmatrix}$, we see that $\mathbf{Y}' = (\beta)\mathbf{Y}$, where we use formal matrix multiplication, and let conjunction stand for addition of formulas. Define the product $(\beta)(\mathcal{Y})$ of matrices of programs as usual, with the program constructs $\vee$ and $;$ standing for addition and multiplication respectively. It is easy to see then that $(\beta)((\mathcal{Y})\mathbf{Y})$ is equivalent to $((\beta)(\mathcal{Y}))\mathbf{Y}$.

The following two lemmas deal with finding, given a matrix of regular expressions β, a matrix $\mathcal{Y}$ equivalent to

$(\beta)^* = I + (\beta) + (\beta)^* + \ldots$

where I is the $k\times k$ matrix with the empty string Λ on the diagonal and the empty set elsewhere. I am indebted to D. Kozen for suggesting that a solution might be obtained via equation (1) below.

5.2.1. LEMMA. Let (β) be a $k\times k$ matrix of regular expressions over a given alphabet. There exists a matrix $(\mathcal{Y})$ of regular expressions such that

$$(1) \quad I + (\beta)(\mathcal{Y}) = (\mathcal{Y})$$

PROOF. Spelled out, for the first column say, (1) reads:

$(1,1) \quad \Lambda + \beta_{11}\mathcal{Y}_{11} + \beta_{12}\mathcal{Y}_{21} + \cdots + \beta_{1k}\mathcal{Y}_{k1} = \mathcal{Y}_{11}$

$(1,2) \quad \beta_{21}\mathcal{Y}_{11} + \beta_{22}\mathcal{Y}_{21} + \cdots + \beta_{2k}\mathcal{Y}_{k1} = \mathcal{Y}_{21}$

.....

(1,k) $\beta_{k1}\gamma_{11} + \beta_{k2}\gamma_{21} + \cdots + \beta_{kk}\gamma_{k1} = \gamma_{k1}$.

We show that these equations have a solution satisfying, in addition,

(2) $\gamma_{j1} = \lambda_j \gamma_{11}$ for some expression λ_j, $j= 2,\ldots,k$.

(2) implies that (1,j) (for $j= 2,\ldots,k$) is equivalent to

$$(\beta_{j1} + \beta_{j2}\lambda_2 + \cdots + \beta_{jk}\lambda_k)\gamma_{11} = \lambda_j \gamma_{11}.$$

For this it suffices to have

(3) $\lambda_j = \beta_{j1} + \beta_{j2}\lambda_2 + \cdots + \beta_{jk}\lambda_k \qquad j= 2,\ldots,k$.

The obvious solution for λ_2 is

(4) $\lambda_2 = \beta_{22}^*(\beta_{21} + \beta_{23}\lambda_3 + \cdots + \beta_{2k}\lambda_k)$.

Substituting this in the equation for λ_3

(3) $\lambda_3 = \beta'_{32} + \beta'_{33}\lambda_3 + \cdots + \beta'_{3k}\lambda_k$

where the β'_{3i} 's are expressed in terms of the β_{2i} 's and the β_{3i} 's. In k-1 steps we get an equation

$$\lambda_k = \mu + \vartheta\lambda_k,$$

(where μ, ϑ are expressions in the β_{ij} 's) from which the solution, $\lambda_k = \vartheta^*\mu$ may be substituted back to obtain solutions for all λ_i's.

It remains to find a solution to

$$\Lambda + \delta\gamma_{11} = \gamma_{11},$$

where

$$\delta = \beta_{11} + \beta_{12}\lambda_2 + \cdots + \beta_{1k}\lambda_k.$$

Let $\gamma_{11} = \delta^*$.□

5.2.2. LEMMA. Suppose (β) is such that $(\beta)^n \neq (\beta)^{n+1}$ for all n. Then there is at most one (and hence exactly one) solution to equation (1).□

5.3 Interpolation for PDL.

5.3.1. LEMMA. Let $f^-;f^+$ be a partition π of f. (i) Suppose P is a proof in CD of $f \vdash X$. Then there is an interpolant K for $f^-;f^+ \vdash X$.

(ii) Suppose P is a proof of $f \vdash [\alpha]q$ from $\{f_i \vdash q\}_{i<k}$, where q does not occur in f. Let $f_i^-;f_i^+$ be the partitions of f_i (i<k) induced by π. If K_i is an interpolant for $f_i^-;f_i^+ \vdash X$ (i<k), then there is an interpolant K of the form $\wedge_i[\beta_i]K_i$ for $f^-;f^+ \vdash [\alpha]X$.

PROOF. By tree-induction on P, simultaneously for (i) and (ii). The basis is trivial. The induction step is by cases on the last inference rule of P. For all cases other than (*R) the proof proceeds as outlined in 5.1 above. We prove (i) for the case (*R) (the proof of (ii) is similar).

Suppose P_n is a proof of $f \vdash [\alpha]^n X$ (n= 0,1,...). As in the proof of 4.3.1 we select a sufficiently large h, for which P_h has the form

$$\begin{array}{c} Q_i \\ \{f_i^-;f_i^+ \vdash [\alpha]^m X\}_{i\in I} \\ R_j \\ \{f_j^-;f_j^+ \vdash [\alpha]^{m+r} X\}_{j\in I} \\ U \\ f^-;f^+ \vdash [\alpha]^h X \end{array}$$

where the partitions are the ones induced, and where $R_j = R_j'[[\alpha]^r X/q]$, $U = U'[[\alpha]^s X/q]$, where $s = h-m-r$.

By induction assumption there are interpolants K_i for $f_i^-;f_i^+ \vdash [\alpha]^m X$, ($i\in I$), and there is a matrix (β) such that if M_i is an interpolant for $f_i^-;f_i^+ \vdash Y$ (for $i \in I$) then $M_i' = ((\beta)\mathbf{M})_i$ ($i \in I$) are interpolants for $f_i^-;f_i^+ \vdash [\alpha]^r Y$. Thus $f_i^- \vdash ((\beta)^n \mathbf{K})_i$ for all n, and $f_i^+,((\beta)^n\mathbf{K})_i \vdash [\alpha]^m[\alpha]^{r*n}X$ for all n.
If there is an n such that $(\beta)^n = (\beta)^{n+1}$, and so $(\beta)^n = (\beta)^{n+k}$ for all k, then let $(\gamma) = I + (\beta) + (\beta)^2 + \dots + (\beta)^n$. Otherwise let (γ) be the solution of (1) in 5.2.1. Then in any case $f_i^- \vdash ((\gamma)\mathbf{K})_i$, and $f_i^+,((\gamma)\mathbf{K})_i \vdash [\alpha]^m[(\alpha^r)^*]X$.

By induction assumption applied to U it then follows that there is an interpolant H_1 for $f^-;f^+ \vdash [\alpha]^{m+s}[(\alpha^w)^*]X$, where w is a sufficiently large number depending only on $f^-;f^+$. Similarly, interpolants H_i (i= 2,...,w) are obtained for sequents as above with w sums m+s pairwise distinct modulo w. It follows, as in the proof of 4.3.1, that $\wedge_i H_i$ is an interpolant for $f^-;f^+ \vdash [\alpha^*]X$.□

5.3.2. THEOREM. (Intrpolation Theorem for PDL and CPDL).
(i) PDL satisfies the Intrpolation Property: if $X \to Y$ is a theorem of PDL, then there is a formula K with $L(K) \subset L(X)\cap L(Y)$, such that $X \to K$ and $K \to Y$ are theorems of PDL.
(ii) The same holds for Constructive PDL.

PROOF. Suppose $X \to Y$ is a theorem of PDL. Then $X^o \to Y^o$ is a theorem of D, by 3.2.2. By 5.3.1 there is then an interpolant for X^o,Y^o, which by 3.2.1 is also an interpolant for X,Y.

To prove (ii) we have to observe that 5.3.1 may be refined: the interpolation statements $f^- \vdash K$, $f^+,K \vdash X$ can be proved in CS. This depends, in turn, on formalizing the observations of 5.2 in CS.□

References.

[Be] F. Berman, A completeness technique for D-axiomatizable semantics; Proc. Eleventh ACM Symp. on Theory of Computing (1979) 160-166.

[CO] R.L. Constable and M.J. O'Donnell, A Programming Logic; Winthrop, Cambridge (1978).

[FL] M. Fischer and R. Ladner, Propositional Dynamic Logic of regular programs; Jour.Comp.Syst.Sc. 18 (1979) 194-211.

[Ga] D.M. Gabbay, Axiomatizations of Logics of Programs; manuscript, (1977).

[KP] D. Kozen and R. Parikh, An elementary proof of the completeness of PDL; to appear in Theor. Comp. Sc..

[Mi] G. Mirkowska, Model existence theorems in algorithmic logic with non-deterninistic programs; unpublished report, University of Warsaw.

[Ni] H. Nishimura, Sequential methods in Propositional Dynamic Logic; Acta Informatica 12 (1979) 377-400.

[Pa1] R. Parikh, A completeness result for PDL; Symp. on Math. Foundations of Computer Science, Springer LNCS (1978), and to appear in Theoretical Comp Sc..

[Pa2] R. Parikh, Propositional logics of programs; Proc. Seventh ACM Symp. on Principles of Programming languages (1980) 186-192.

[Pr1] V.R. Pratt, Semantical considerations in Floyd/Hoare Logic; Proc. 17th IEEE Symp. Foundations of Comp. Sc.(1976) 109-121.

[Pr2] V.R. Pratt, A practical decision method for Propositional Dynamic Logic; Proc. Tenth ACm Symp. on Theory of Computing (1978), 326-337.

[Pr3] V.R. Pratt, Models of program logics;Proc. Twentieth IEEE Symp. on Foundations of Computer Science (1979), 115-122.

[Sa] A. Salwicki, Formalized algorithmic languages; Bull. Acad. Pol. Sc. Ser. Sc. Math. Astr. Phys. 18 (1970) 227-232.

[Se] K. Segerberg, A completeness theorem in the modal logic of programs; Notices Amer. Math. Soc. 24 (1977) A522.

[Sm] R.M. Smullian, First Order Logic; Springer, Berlin (1968).

[T] G. Takeuti, Proof Theory; North-Holland, Amsterdam (1975).