Advanced Logic 2014–15

Exercises Set 5

Let $\textsc{Var} = \{p, q, \ldots\}$ be a set of propositional variables, and $\mathrm{A} = \{a, b, \ldots\}$ a set of *atomic programs*. The sets $\textsc{Prog}$ and $\textsc{Form}$ of *PDL-programs* and *PDL-formulas over* A *and* $\textsc{Var}$ are defined by mutual induction:

$$\alpha ::= a \mid \alpha \,;\, \alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \varphi? \qquad (a \in \mathrm{A})$$
$$\varphi ::= p \mid \bot \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\alpha\rangle\varphi \qquad (p \in \textsc{Var})$$

We let ? and $*$ bind stronger than ; which in turn binds stronger than $\cup$. Moreover, we sometimes write $\alpha\beta$ instead of $\alpha \,;\, \beta$.
Some standard program constructs that can be defined in this syntax are:

$$\textsf{if } \varphi \textsf{ then } \alpha \textsf{ else } \beta = (\varphi? \,;\, \alpha) \cup (\neg\varphi? \,;\, \beta)$$
$$\textsf{while } \varphi \textsf{ do } \alpha = (\varphi? \,;\, \alpha)^* \,;\, \neg\varphi?$$

The semantics for PDL-formulas is the one given for multi-modal formulas, now over the index set $\textsc{Prog}$ (so for every $\alpha \in \textsc{Prog}$ the modality $\langle\alpha\rangle$ is interpreted by the relation R_α), plus some additional conditions for interpreting program constructors:
A *PDL-frame* $\mathcal{F} = (W, \{R_\alpha \mid \alpha \in \textsc{Prog}\})$ is a $\textsc{Prog}$-frame satisfying:

$$R_{\alpha\beta} = R_\alpha \circ R_\beta$$
$$R_{\alpha\cup\beta} = R_\alpha \cup R_\beta$$
$$R_{\alpha^*} = (R_\alpha)^*$$

Here $R \circ S$ is the composition of relations R and S, $R \cup S$ is their union, and R^* is the reflexive transitive closure of R, defined by:

$$R \circ S = \{\, (x, z) \mid \exists y\, (\, Rxy \wedge Syz\,) \,\} \qquad \mathrm{Id} = \{\, (x, y) \mid x = y \,\}$$
$$R \cup S = \{\, (x, y) \mid Rxy \vee Sxy \,\} \qquad R^0 = \mathrm{Id}$$
$$R^* = \bigcup_{n \geq 0} R^n \qquad R^{n+1} = R^n \circ R$$

A *PDL-model* $\mathcal{M} = (\mathcal{F}, V)$ consists of a PDL-frame $\mathcal{F} = (W, \{R_\alpha \mid \alpha \in \textsc{Prog}\})$ and a valuation $V : \textsc{Var} \to \mathcal{P}(W)$, and satisfies:

$$R_{\varphi?} = \{(w, w) \mid \mathcal{M}, w \vDash \varphi\}$$

1. Let $\mathcal{M} = (W, \{R_\alpha \mid \alpha \in \textsc{Prog}\}, V)$ be a PDL-model. Determine the transition relations corresponding to the following programs:

   (a) $\mathsf{while}\ p\ \mathsf{do}\ \alpha$

   (b) $\mathsf{if}\ p\ \mathsf{then}\ \alpha\ \mathsf{else}\ \beta$

2. (a) Show that $R^1 = R$.

   (b) Show that if xR^*y, then, for some $n \geq 0$, there are $x_0, x_1, \ldots, x_n$ such that $x_0 = x$, $x_n = y$ and $x_i R x_{i+1}$ for all $0 \leq i < n$;

   (c) Show that R^* is the smallest reflexive and transitive relation which contains R, i.e.:

      (i) R^* is reflexive and transitive;

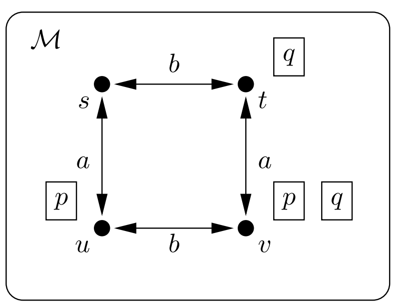
      (ii) if R' is a reflexive, transitive relation and $R \subseteq R'$, then $R^* \subseteq R'$.

   (d) What is the reflexive-transitive closure of $\{(n, n+1) \mid n \in \mathbb{N}\}$?

Once the relations for interpreting atomic programs are fixed, we also know the relations corresponding to the composed programs. Put differently, an A-model (with A the set of atomic programs), induces a PDL-model (for programs over A), as follows: Let $\mathcal{M} = (W, \{R_a\}_{a\in A}, V)$ be an A-model. The *PDL-extension $\widehat{\mathcal{M}}$ of $\mathcal{M}$* is the model $\widehat{\mathcal{M}} = (W, \{\widehat{R}_\alpha \mid \alpha \in \textsc{Prog}\}, V)$, with $\widehat{R}_\alpha$ defined inductively on the structure of α:

$$\widehat{R}_a = R_a \qquad \widehat{R}_{\alpha\cup\beta} = \widehat{R}_\alpha \cup \widehat{R}_\beta \qquad \widehat{R}_{\alpha^*} = (\widehat{R}_\alpha)^*$$
$$\widehat{R}_{\alpha\beta} = \widehat{R}_\alpha \circ \widehat{R}_\beta \qquad \widehat{R}_{\varphi?} = \{(x,x) \mid \mathcal{M}, x \vDash \varphi\}$$

3. Let $\textsc{Var} = \{p, q\}$ and $A = \{a, b\}$, and consider the following A-model $\mathcal{M}$:



   (a) Prove that $\widehat{\mathcal{M}} \vDash p \leftrightarrow [(ab^*a)^*]p$.

   (b) Prove that $\widehat{\mathcal{M}} \vDash q \leftrightarrow [(ba^*b)^*]q$.

   (c) Let α be the program[1] defined by:

$$\alpha = (aa \cup bb \cup (ab \cup ba)(aa \cup bb)^*(ab \cup ba))^*$$

   Prove that $\widehat{\mathcal{M}} \vDash \varphi \leftrightarrow [\alpha]\varphi$ for every PDL-formula φ.

[1] Viewed as a regular expression, α generates all words over the alphabet $\{a, b\}$ with an even number of a's, and an even number of b's.

4. Let α and β be PDL-programs. Which of the following two formulas is valid in PDL, which is not?

   (a) $[(\alpha \cup \beta)^*]p \to [\alpha^*]p \wedge [\beta^*]p$

   (b) $[\alpha^*]p \wedge [\beta^*]p \to [(\alpha \cup \beta)^*]p$

   Give a counterexample for the invalid one, and prove validity of the other.

5. Let $\alpha, \beta, \gamma \in \textsc{Prog}$ be PDL-programs and φ a PDL-formula. Show that the following formulas are validities PDL:

   (a) $[\alpha(\beta \cup \gamma)]\varphi \leftrightarrow [\alpha\beta \cup \alpha\gamma]\varphi$

   (b) $[(\alpha \cup \beta)\gamma]\varphi \leftrightarrow [\alpha\gamma \cup \beta\gamma]\varphi$

   (c) $[\alpha^*]\varphi \leftrightarrow \varphi \wedge [\alpha][\alpha^*]\varphi$