Dynamic Logic

Patrick Stünkel

February 8, 2022

1 Propositional Logic (PL)

Propositional logic is an important constituent in the definition of Dynamic logic. Thus, I recall the respective content here shortly. Propositional logic is built around propositions. We denote Π_0 as the set of (atomic) proposition symbols $\{P, Q, R, \ldots\}$/ The set of all propositional formulae $\Phi^{\mathbf{PL}}$ is defined via the following grammar:

$$\Phi^{\mathbf{PL}} ::= \top \mid \bot \mid p \mid \neg\Phi^{\mathbf{PL}} \mid \Phi^{\mathbf{PL}} \wedge \Phi^{\mathbf{PL}} \mid \Phi^{\mathbf{PL}} \vee \Phi^{\mathbf{PL}} \mid \Phi^{\mathbf{PL}} \Rightarrow \Phi^{\mathbf{PL}} \mid \Phi^{\mathbf{PL}} \Leftrightarrow \Phi^{\mathbf{PL}} \tag{1}$$

A *model* m of propositional logic is a valuation function $m : \Pi_2 \to \mathbf{2}$, or equivalently a subset of Π_0 (i.e. the propositions assumed to be true). Models can be enumerated in the form of *truth tables*. The relation between models and basic sentences in propositional logic are summarised in the following truth table.

P	S	$\top$	$\bot$	P	$\neg P$	$P \wedge S$	$P \vee S$	$P \Rightarrow S$	$P \Leftrightarrow S$
false	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*false*	*true*	*false*	*true*	*true*	*false*
true	*false*	*true*	*false*	*true*	*false*	*false*	*true*	*false*	*false*
true	*true*	*true*	*false*	*true*	*false*	*true*	*true*	*true*	*true*

Table 1: Relationship between propositional models and sentences

The tautologies of propositional logic are listed below.

$$\top \tag{Verum}$$
$$\phi \vee \neg\phi \tag{Excluded middle}$$
$$\phi \wedge \psi \Leftrightarrow \psi \wedge \phi \tag{Commutativity of $\wedge$}$$
$$\phi \vee \psi \Leftrightarrow \psi \vee \phi \tag{Commutativity of $\vee$}$$
$$\phi \wedge (\psi \wedge \chi) \Leftrightarrow (\phi \wedge \psi) \wedge \chi \tag{Associativity of $\wedge$}$$
$$\phi \vee (\psi \vee \chi) \Leftrightarrow (\phi \vee \psi) \vee \chi \tag{Associativity of $\vee$}$$
$$\neg(\neg\phi) \Leftrightarrow \phi \tag{Double negation elimiation}$$
$$(\phi \Rightarrow \psi) \Leftrightarrow (\neg\psi \Rightarrow \neg\phi) \tag{Contraposition}$$
$$(\phi \Rightarrow \psi) \Leftrightarrow (\neg\phi \vee \psi) \tag{Implication elimination}$$
$$(\phi \Leftrightarrow \psi) \Leftrightarrow ((\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)) \tag{Biconditional elimination}$$
$$\neg(\phi \wedge \psi) \Leftrightarrow (\neg\phi \vee \neg\psi) \tag{De Morgan}$$
$$\neg(\phi \vee \psi) \Leftrightarrow (\neg\phi \wedge \neg\psi) \tag{De Morgan}$$
$$(\phi \wedge (\psi \vee \chi)) \Leftrightarrow ((\phi \wedge \psi) \vee (\phi \wedge \chi)) \tag{Distributivity of $\wedge$ over $\vee$}$$
$$(\phi \vee (\psi \wedge \chi)) \Leftrightarrow ((\phi \vee \psi) \wedge (\phi \vee \chi)) \tag{Distributivity of $\vee$ over $\wedge$}$$

The inference rules in propositional logic are as follows.

$$\frac{\phi, \phi \Rightarrow \psi}{\psi} \tag{Modus Ponens}$$
$$\frac{\phi \wedge \psi}{\phi} \tag{And Elimination}$$

2 Propositional Dynamic Logic (PDL)

Propositional Dynamic Logic is simpler variant of dynamic logic, which adds the notions of *dynamics* and *regular actions* to propositional logic. Let A_0 describe a set of atomic action labels, e.g. the possible interactions with a vending machine (*insert coin*, *push button*, ...). Atomic actions can be composed via the regular combinators *sequence* (;), *choice* ($\cup$), *iteration* (*), and *test* (?). This induces the set of all regular actions A, defined via the following grammar:

$$A ::= A_0 \mid A^* \mid A\,;\,A \mid A \cup A \mid \Phi^{\mathbf{PL}}? \tag{2}$$

The sentences of PDL $\Phi^{\mathbf{PDL}}$ are defined via the following grammar

$$\Phi^{\mathbf{PDL}} ::=$$
$$\top \mid \bot \mid \Phi_0 \tag{Atomic Sentences}$$
$$\langle A \rangle \Phi^{\mathbf{PDL}} \tag{Possibility}$$
$$[A]\Phi^{\mathbf{PDL}} \tag{Necessity}$$
$$\neg\Phi^{\mathbf{PDL}} \mid \Phi^{\mathbf{PDL}} \wedge \Phi^{\mathbf{PDL}} \mid \Phi^{\mathbf{PDL}} \vee \Phi^{\mathbf{PDL}} \mid$$
$$\Phi^{\mathbf{PDL}} \Rightarrow \Phi^{\mathbf{PDL}} \mid \Phi^{\mathbf{PDL}} \Leftrightarrow \Phi^{\mathbf{PDL}} \tag{Connectives}$$

3 Labelled Kripke Structures & Regular Actions

Labelled Kripke Structures are the models of Propositional Dynamic Logic. A labelled Kripke structure K is defined as follows:

$$K = (S^K, T^K \subseteq S \times A_0 \times S, L^K : S \to 2^{\Phi_0})$$

where S is a set of states, T is the transition relation, and L is the labelling function that assigns those propositional symbols to a state $s \in S^K$ that are satisfied in the particular state.

The semantics of the regular actions A is defined as follows. Each atomic action label $a \in A_0$ is interpreted as a relation R_a:

$$R_a := \{(s, s') \mid (s, a, s') \in T^K\} \tag{3}$$

The composite operations sequence, choice, and iteration are defined recursively: Sequence is interpreted by relation composition.

$$R_{a;b} := R_b \circ R_a = \{(s, s') \mid \exists s'' \in S^K.(s, s'') \in R_a \wedge (s'', s') \in R_b\} \tag{4}$$

Choice is interpreted as union.

$$R_{a\cup b} := R_a \cup R_b = \{(s, s') \mid (s, s') \in R_a \vee (s, s') \in R_b\} \tag{5}$$

Iteration is interpreted as reflexive transitive closure.

$$\begin{aligned} R_a^0 &:= \{(s, s) \mid s \in S\} \\ R_a^1 &:= R_a \\ R_a^{n+1} &:= R_a^n \circ R_a^{n-1} && (n \geq 2) \\ R_{a^*} &:= \bigcup_i R_a^i \end{aligned}$$

The test action is a subset of the diagonal relation.

$$R_{\varphi?} := \{(s, s) \mid L^K(s) \models \varphi\} \tag{6}$$

The satisfaction of a PDL formula φ in a given model m is defined as follows. Let K be the Kripke structure of m, then one can first verify the satisfaction of φ in a given state $s \in S^K$ of K.

$$m, s \models \varphi \text{ iff. } \begin{cases} P \in L^K(s) & \text{iff. } \varphi = P, P \in \Phi_0 \\ \exists (s, a, s') \in T^K.m, s' \models \psi & \text{iff. } \varphi = \langle a \rangle \psi, a \in A, \psi \in \Phi^{\mathbf{PDL}} \\ \forall (s, a, s') \in T^K.m, s' \models \psi & \text{iff. } \varphi = [a]\psi, a \in A, \psi \in \Phi^{\mathbf{PDL}} \\ m, s \not\models \psi & \text{iff. } \varphi = \neg\psi, \psi \in \Phi^{\mathbf{PDL}} \\ m, s \models \psi \text{ and } m, s \models \chi & \varphi = \psi \wedge \chi, \psi, \chi \in \Phi^{\mathbf{PDL}} \\ m, s \models \psi \text{ or } m, s \models \chi & \varphi = \psi \vee \chi, \psi, \chi \in \Phi^{\mathbf{PDL}} \\ m, s \models \psi \text{ implies } m, s \models \chi & \varphi = \psi \Rightarrow \chi, \psi, \chi \in \Phi^{\mathbf{PDL}} \\ m, s \models \psi \text{ iff. } m, s \models \chi & \varphi = \psi \Leftrightarrow \chi, \psi, \chi \in \Phi^{\mathbf{PDL}} \end{cases}$$

4 PDL theorem proving

PDL has a inference system that is sound and complete. Its axioms are given by the propositional tautologies, enumerated above, together with the following list:

$$
\begin{aligned}
[\alpha]\phi &\Leftrightarrow \neg\langle\alpha\rangle(\neg\phi) && \text{(Duality)}\\
[\alpha](\phi \Rightarrow \psi) &\Rightarrow ([\alpha]\phi \Rightarrow [\alpha]\psi) && \text{(“K axiom”: } [\alpha] \text{ distributes over } \Rightarrow)\\
[\alpha](\phi \wedge \psi) &\Leftrightarrow [\alpha]\phi \wedge [\alpha]\psi && \text{(Modal And distribution)}\\
[\alpha \cup \beta]\phi &\Leftrightarrow [\alpha]\phi \wedge [\beta]\phi && \text{(Choice)}\\
[\alpha; \beta]\phi &\Leftrightarrow [\alpha][\beta]\phi && \text{(Sequuence)}\\
[\psi?]\phi &\Leftrightarrow (\psi \Rightarrow \phi) && \text{(Test)}\\
(\phi \wedge [\alpha][\alpha^*]\phi) &\Leftrightarrow [\alpha^*]\phi && \text{(Mix)}\\
(\phi \wedge [\alpha^*](\phi \Rightarrow [\alpha]\phi)) &\Rightarrow [\alpha^*]\phi && \text{(Induction)}
\end{aligned}
$$

(The interested reader is invited to verify those axioms on an arbitrary Kripke structure)

The two main inference rules in PDL are as follows:

$$
\frac{\phi, \phi \Rightarrow \psi}{\psi} \qquad \text{(Modus Ponens)}
$$

$$
\frac{\phi}{[\alpha]\phi} \qquad \text{(Modal Generalisation)}
$$

5 Dynamic Logic

Dynamic Logic extends the definitions of PDL by replacing propositional logic by first-order logic[1]. The idea is that first-order theory axioms are used to specify the semantics of common base types in programming languages such as `int`, `string`, or `float`. Free variables represent arbitrary program variables and the Kripke structure shows the various states of a program during its executions, i.e. current variable values. Hence, the set of all regular actions A differs compared to PDL in the sense that, atomic actions A_0 in this case are given by *variable assignments*

$$v := t$$

, where v is a free variable and t is a term and that the test operation $\varpi?$ allows φ to be a FOL sentence. The sentences of dynamic logic correspond to those of PDL with the only difference that atomic sentences are given by predicate terms.

[1]For an overview on first-order logic, its sentences, models, and inference system, I refer to common textbooks in the area.