Formale Systeme II: Theorie

Dynamic Logic:
Propositional Dynamic Logic

SS 2016

Prof. Dr. Bernhard Beckert · Dr. Mattias Ulbrich
Slides partially by Prof. Dr. Peter H. Schmitt
Goals

Overview – a family of logics

Modal Logics

→ Propositional Dynamic Logic

→ Dynamic Logic

Hybrid DL  Java DL

Modal Logics: → Formal Systems I (recap here)
Java DL: Logic used in KeY
→ lecture “Formal Systems II – Applications”
Goals

**Dynamic Logic** as . . .

- abstract reasoning framework for descriptions of actions
- means to formalise and reason about semantics of programs
- vehicle for examining/proving theoretical results on program reasoning
  - what is decidable, what is not?
  - relative completeness
- concept of program verification on a while language
- logic for a verification engine for a realworld programming language
Literature

- *Formale Systeme II*
  Vorlesungsskript
  Peter H. Schmitt
  → Website

- *Dynamic Logic*
  Series: Foundations of Computing
  David Harel, Dexter Kozen and Jerzy Tiuryn
  MIT Press
  → Department Library
Motivating Example
The Towers of Hanoi
The Instructions

1. Move alternatingly the smallest disk and another one.
2. If moving the smallest disk put it on the stack it did not come from in its previous move.
3. If not moving the smallest disk do the only legal move,

More formally:
sequence of actions

\[ moveS ; moveO ; moveS ; moveO ; \ldots \]

more concisely:

\[ (moveS ; moveO)^* \]

improved:

\[ moveS ; testForStop ; (moveO ; moveS ; testForStop)^* \]
Properties

**Atomic statement:** \( S_1 \) true iff smallest piece on first stack

**Moving away**

(1) \( S_1 \rightarrow \langle \text{moveS} \rangle \neg S_1 \)
... after moving the smallest, it is no longer on the first stack

**Moving other**

(2) \( S_1 \rightarrow \langle \text{moveO} \rangle S_1 \)
... after moving something else, it is still on the first stack

**Conclusions from (1) and (2)**

\( S_1 \rightarrow \langle \text{moveO} ; \text{moveS} \rangle \neg S_1 \)
\( S_1 \rightarrow \langle (\text{moveO})^* ; \text{moveS} \rangle \neg S_1 \)

↑ THAT’S DYNAMIC LOGIC ↑
Dynamic Logic
Dynamic Logic

- Allows reasoning about properties of composite actions.
- Actions are explicitly part of the language.
- Extends modal logic
- We look at two instances:
  - Propositional Dynamic Logic
  - First Order Dynamic Logic
Recap: Modal Logic

Syntax/semantics of dynamic logic build on top of modal logic.

**Syntax:**
- Signature $\Sigma$: set of propositional variables
- $Fml_{\Sigma}^{ML}$ smallest set with:
  - $\Sigma \subseteq Fml_{\Sigma}^{mod}$
  - $true, false \in Fml_{\sigma}^{mod}$
  - $A, B \in Fml_{\Sigma}^{mod} \implies A \land B, A \lor B, A \rightarrow B, \neg A \in Fml_{\Sigma}^{mod}$
  - $A \in Fml_{\Sigma}^{mod} \implies \Box A, \Diamond A \in Fml_{\Sigma}^{mod}$

- pronounced “Box” and “Diamond”
Recap: Modal Logic – Semantics

Modal Logics

Logics of *necessity* and *possibility*.

Meaning of Modalities:

**Modal**
- □A  It is necessary that . . .
- ◊A  It is possible that . . .

**Deontic** (from Greek for duty)
- □A  It is obligatory that . . .
- ◊A  It is permitted that . . .

**Epistemic** (logic of knowledge)
- □A  I know that . . .
- ◊A  I consider it possible that . . .
Recap: Modal Logic – Semantics

Unified Semantics

In late 1950s Saul Kripke defined unified semantics for all “meanings” of modal operators: “worlds” and “accessibility” between them.

Kripke Frame \((S, R)\):
- Set \(S\) of worlds (or states)
- Relation \(R \subseteq S \times S\), the accessibility relation

Kripke Structure \((S, R, I)\):
- Given a signature \(\Sigma\)
- Kripke Frame \((S, R)\)
- Interpretation \(I : S \to 2^\Sigma\)
Recap: Modal Logic – Semantics

For a signature $\Sigma$ and Kripke structure $\langle S, R, I \rangle$

$I, s \models \varphi \iff$ Formula $\varphi$ holds in state $s \in S$

$I \models \varphi \iff$ Formula $\varphi$ holds in all states $s \in S$

$I, s \models p \iff p \in I(s) \quad \text{for } p \in \Sigma$

$I \models$ is homomorphic for $\land, \lor, \rightarrow, \neg$, i.e.:

$I, s \models \varphi \land \psi \iff I, s \models \varphi$ and $I, s \models \psi$

$I, s \models \varphi \lor \psi \iff I, s \models \varphi$ or $I, s \models \psi$

$I, s \models \varphi \rightarrow \psi \iff I, s \models \varphi$ implies $I, s \models \psi$

$I, s \models \neg \varphi \iff$ not $I, s \models \varphi$

$I, s \models \square \varphi \iff I, s' \models \varphi$ for all $s' \in S$ with $(s, s') \in R$

$I, s \models \diamond \varphi \iff I, s' \models \varphi$ for some $s' \in S$ with $(s, s') \in R$

Example: Chalkboard
More than one modality

Multi-modal logic

Have different Box operators with different accessibility relations:

\( \Box \alpha, \Box \beta, \Box \gamma, \ldots \)

(\( \rightarrow \) basic actions ins “Towers of Hanoi”)

Propositional Dynamic Logic (PDL):

- Signature \( \Sigma \) of propositional variables
- Set \( A = \{ \alpha, \beta, \ldots \} \) of atomic actions/programs
- We write \([\alpha] \) instead of \( \Box _{\alpha} \)
## PDL – Regular Programs

### Compose Programs

Atomic programs can be into composed into larger programs

For a given signature $\Sigma$ and atomic programs $A$, the set of programs $\Pi_{\Sigma,A}$ is the smallest set such that

1. $A \subseteq \Pi_{\Sigma,A}$
2. $p, q \in \Pi_{\Sigma,A} \implies (p ; q) \in \Pi_{\Sigma,A}$
3. $p, q \in \Pi_{\Sigma,A} \implies (p \cup q) \in \Pi_{\Sigma,A}$
4. $p \in \Pi_{\Sigma,A} \implies p^* \in \Pi_{\Sigma,A}$
5. $F \in \text{Fml}^{PDL}_{\Sigma,A} \implies ?F \in \Pi_{\Sigma,A}$

Regular Programs =

Regular Expressions over atomic programs and tests
PDL – Formulae

For a given signature $\Sigma$ and atomic programs $A$, the set of formulae $\text{Fml}_{\Sigma,A}^{PDL}$ is the smallest set such that

1. $\text{true}, \text{false} \in \text{Fml}_{\Sigma,A}^{PDL}$

2. $\Sigma \subseteq \text{Fml}_{\Sigma,A}^{PDL}$

3. $A, B \in \text{Fml}_{\Sigma,A}^{PDL} \implies A \land B, A \lor B, A \rightarrow B, \neg A \in \text{Fml}_{\Sigma,A}^{PDL}$

4. $P \in \Pi_{\Sigma,A}, A \in \text{Fml}_{\Sigma,A}^{PDL} \implies [P]A, \langle P \rangle A \in \text{Fml}_{\Sigma,A}^{PDL}$

Programs and Formulae are mutually dependent definitions and must be seen simultaneously.
PDL Formulas – Examples

→ Towers of Hanoi

\[ A = \{ \text{move}S, \text{move}O \}, \quad \Sigma = \{ S1 \} \]
\[ S1 \rightarrow \langle (\text{move}O)^* \ ; \ \text{move}S \rangle \neg S1 \]

multi-level and nested modalities

\[ A = \{ \alpha, \beta \}, \quad \Sigma = \{ P, Q \} \]
\[
[\alpha \cup (\neg P; \beta)^*]Q \\
[\alpha]P \rightarrow [\alpha^*]P \\
[\alpha]\langle \beta \rangle (P \rightarrow [\alpha^*]Q) \\
[\alpha ; \neg \langle \beta \rangle P; \beta]Q
\]
Given a signature $\Sigma$ and atomic programs $A$

### (multi-modal propositional) Kripke frame $(S, \rho)$

- set of states $S$
- function $\rho : A \rightarrow S \times S$ accessibility relations for atomic programs

### Kripke structure $(S, \rho, I)$

- Kripke frame $(S, \rho)$
- interpretation $I : S \rightarrow 2^\Sigma$

$\Rightarrow$ same as for modal logic
Extension of $\rho$

from $\rho : A \rightarrow S^2$ to $\rho : \Pi_{\Sigma,A} \rightarrow S^2$

- $\rho(\alpha)$ base case for $\alpha \in A$
- $\rho(\pi_1 \cup \pi_2) = \rho(\pi_1) \cup \rho(\pi_2)$
- $\rho(\pi_1 ; \pi_2) = \rho(\pi_1) ; \rho(\pi_2)$
  $$= \{(s, s') \mid \text{ex. } t \text{ with } (s, t) \in \rho(\pi_1) \text{ and } (t, s') \in \rho(\pi_2)\}$$
- $\rho(\pi^*) = \text{rtcl}(\rho(\pi)) = \bigcup_{n=0}^{\infty} \rho(\pi)^n$ refl. transitive closure
  $$= \{(s_0, s_n) \mid \text{ex. } n \text{ with } (s_i, s_{i+1}) \in \rho(\pi) \text{ for } 0 \leq i < n\}$$
- $\rho(\forall A) = \{(s, s) \mid l, s \models A\}$
PDL – Semantics

For a signature $\Sigma$, basic programs $A$ and Kripke structure $(S, \rho, I)$

$l, s \models p \iff p \in I(s)$ for $p \in \Sigma$

$\models$ is homomorphic for $\land, \lor, \to, \neg$.

$l, s \models [\pi] \varphi \iff l, s' \models \varphi$ for all $s' \in S$ with $(s, s') \in \rho(\pi)$

$l, s \models \langle \pi \rangle \varphi \iff l, s' \models \varphi$ for some $s' \in S$ with $(s, s') \in \rho(\pi)$