Formalizing and Proving Theorems in Coq — Lecture 5

Tobias Kappé

Today's lecture

You will learn about...

- advanced topics involving recursion and inductives (continued)
- the process for your project during the coming two weeks

Propositional Dynamic Logic — Syntax

Let P and A be sets of *propositions* and *actions* respectively.

The set of *programs* Π is generated by the following grammar:

$$\alpha, \beta ::= a \ (a \in A) \mid \alpha + \beta \mid \alpha ; \beta \mid \alpha^*$$

The set of *formulas* Φ is generated by the following grammar:

$$\phi, \psi ::= p, \overline{p} \ (p \in P) \mid \phi \vee \psi \mid \phi \wedge \psi \mid \langle \alpha \rangle \phi \mid [\alpha] \phi$$

Propositional Dynamic Logic — Semantics

A *frame* is a triple (W, σ, π) where

- W is a set of *worlds*;
- $\sigma : A \to \mathcal{P}(W \times W)$ assigns a relation to each action;
- $\pi : P \to \mathcal{P}(W)$ assigns a set of worlds to each proposition.

Propositional Dynamic Logic — Semantics

We can extend $\sigma : A \to \mathcal{P}(W \times W)$ to act on Π:

$$\sigma(\alpha + \beta) = \sigma(\alpha) \cup \sigma(\beta) \qquad \sigma(\alpha; \beta) = \sigma(\alpha) \circ \sigma(\beta) \qquad \sigma(\alpha^*) = \sigma(\alpha)^*$$

We can also extend $\pi : P \to \mathcal{P}(W)$ to act on Φ:

$$\pi(\overline{p}) = W \setminus \pi(p) \qquad \pi(\phi \vee \psi) = \pi(\phi) \cup \pi(\psi) \qquad \pi(\phi \wedge \psi) = \pi(\phi) \cap \pi(\psi)$$

$$\pi(\langle\alpha\rangle\phi) = \{w \in W : \exists w' \in W.\ w' \in \pi(\phi) \wedge w\ \sigma(\alpha)\ w'\}$$

$$\pi([\alpha]\phi) = \{w \in W : \forall w' \in W.\ w\ \sigma(\alpha)\ w' \implies w' \in \pi(\phi)\}$$

Propositional Dynamic Logic — demonstration

We are going to

1. encode our definition of a frame
2. encode the syntax of programs
3. assign a semantics to programs
4. encode the syntax of formulas
5. assign a semantics to formulas
6. prove some elementary properties of PDL

Demo

Propositional Dynamic Logic II — Syntax

Let P and A be sets of *propositions* and *actions* respectively.

The set of *programs* Π is generated by the following grammar:

$$\alpha, \beta ::= a \ (a \in A) \mid \alpha + \beta \mid \alpha ; \beta \mid \alpha^*$$

The set of *formulas* Φ is generated by the following grammar:

$$\phi, \psi ::= p \ (p \in P) \mid \phi \vee \psi \mid \neg \phi \mid \langle \alpha \rangle \phi$$

NB: no more negative propositions, conjunctions, or boxes!

Propositional Dynamic Logic II — Semantics

We can extend $\sigma : A \to \mathcal{P}(W \times W)$ to act on Π:

$$\sigma(\alpha + \beta) = \sigma(\alpha) \cup \sigma(\beta) \qquad \sigma(\alpha ; \beta) = \sigma(\alpha) \circ \sigma(\beta) \qquad \sigma(\alpha^*) = \sigma(\alpha)^*$$

We can also extend $\pi : P \to \mathcal{P}(W)$ to act on Φ:

$$\pi(\overline{p}) = W \setminus \pi(p) \qquad \pi(\phi \vee \psi) = \pi(\phi) \cup \pi(\psi) \qquad \pi(\neg\phi) = W \setminus \pi(\phi)$$

$$\pi(\langle\alpha\rangle\phi) = \{w \in W : \exists w' \in W.\ w' \in \pi(\phi) \wedge w\ \sigma(\alpha)\ w'\}$$

Propositional Dynamic Logic II — demonstration

We are going to

1. reuse our definition of a frame
2. resue the syntax/semantics of programs
3. encode the syntax of formulas
4. assign a semantics to formulas
5. prove some elementary properties of PDL

Demo

Propositional Dynamic Logic III — Syntax

Let P and A be sets of *propositions* and *actions* respectively.

The set of *programs* Π is generated by the following grammar:

$$\alpha, \beta ::= a \ (a \in A) \mid \phi? \ (\phi \in \Phi) \mid \alpha + \beta \mid \alpha ; \beta \mid \alpha^*$$

The set of *formulas* Φ is generated by the following grammar:

$$\phi, \psi ::= p \ (p \in P) \mid \phi \vee \psi \mid \neg \phi \mid \langle \alpha \rangle \phi$$

NB: mutual recursion!

Propositional Dynamic Logic III — Semantics

We can extend $\sigma : A \to \mathcal{P}(W \times W)$ to act on Π:

$$\sigma(\alpha + \beta) = \sigma(\alpha) \cup \sigma(\beta) \qquad \sigma(\phi?) = \{(w, w) : w \in \pi(\phi)\}$$

$$\sigma(\alpha; \beta) = \sigma(\alpha) \circ \sigma(\beta) \qquad \sigma(\alpha^*) = \sigma(\alpha)^*$$

We can also extend $\pi : P \to \mathcal{P}(W)$ to act on Φ:

$$\pi(\overline{p}) = W \setminus \pi(p) \qquad \pi(\phi \vee \psi) = \pi(\phi) \cup \pi(\psi) \qquad \pi(\neg\phi) = W \setminus \pi(\phi)$$

$$\pi(\langle\alpha\rangle\phi) = \{w \in W : \exists w' \in W.\ w' \in \pi(\phi) \wedge w\ \sigma(\alpha)\ w'\}$$

Propositional Dynamic Logic III — demonstration

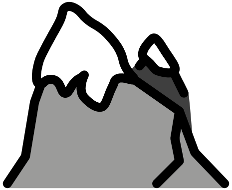
We are going to

1. reuse our definition of a frame
2. encode the syntax of programs/formulas
3. encode the semantics of programs/formulas
4. prove some elementary properties of PDL

Demo

Project phase: the plan

- Settle on one of your two proposed projects.
- Send me three goals:
  1. a minimal goal
  2. a realistic goal
  3. a stretch goal
- Start encoding basic definitions and proof objectives.
- Work top-down or bottom up to prove your goals.
- *Keep asking questions on Piazza*.
- Friday of next week: mid-project check-in.



Next lecture

- learn about the Curry-Howard isomorphism
- synthesize programs, run proofs

Homework:

- Complete the proofs in `lecture-5*.v`.