

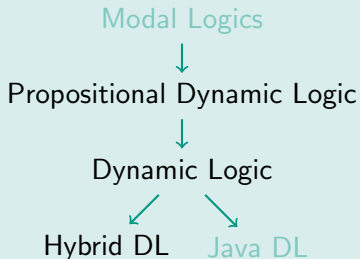
# Formale Systeme II: Theorie

## Dynamic Logic: Propositional Dynamic Logic

SS 2018

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

## Overview – a family of logics



**Modal Logics:** → Formal Systems I (recap here)

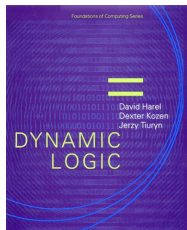
**Java DL:** Logic used in KeY

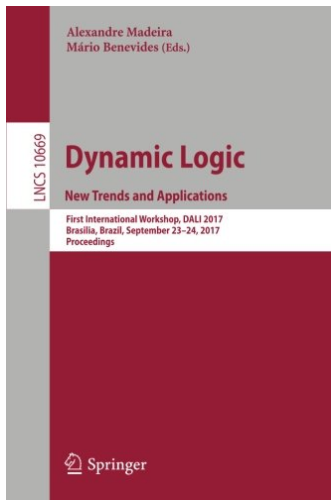
→ lecture “Formal Systems II – Applications”

## 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 verification engines for realworld programming languages

- *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



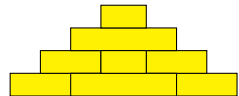
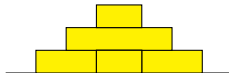
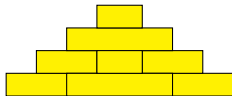


## From the table of contents

- A Dynamic Logic for Learning Theory (Baltag et al.)
- Axiomatization and Computability of a Variant of Iteration-Free PDL with Fork (Balbiani et al.)
- Dynamic Preference Logic as a Logic of Belief Change (Souza et al.)
- Dynamic Logic: A Personal Perspective (**Vaughan Pratt**)
- . . .

# Motivating Example

# Introductory Example



The Towers of Hanoi

# The Instructions

- 1 Move alternately 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

$$\text{move}S ; \text{move}O ; \text{move}S ; \text{move}O ; \dots$$

more concisely:

$$(\text{move}S ; \text{move}O)^*$$

improved:

$$\text{move}S ; \text{testForStop} ; (\text{move}O ; \text{move}S ; \text{testForStop})^*$$



**Atomic statement:**  $S1$  true iff smallest piece on first stack

## Moving away

$$(1) \quad S1 \rightarrow \langle \text{moveS} \rangle \neg S1$$

... after moving the smallest, it is no longer on the first stack

## Moving other

$$(2) \quad S1 \rightarrow \langle \text{moveO} \rangle S1$$

... after moving something else, it is still on the first stack

## Conclusions from (1) and (2)

$$S1 \rightarrow \langle \text{moveO} ; \text{moveS} \rangle \neg S1$$

$$S1 \rightarrow \langle (\text{moveO})^* ; \text{moveS} \rangle \neg S1$$

↑ THAT'S 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

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

## Syntax:

- Signature  $\Sigma$ : set of propositional variables
- $Fml_{\Sigma}^{ML}$  smallest set with:
  - $\Sigma \subseteq Fml_{\Sigma}^{mod}$
  - $true, false \in Fml_{\Sigma}^{mod}$
  - $A, B \in Fml_{\Sigma}^{mod} \implies A \wedge B, A \vee B, A \rightarrow B, \neg A \in Fml_{\Sigma}^{mod}$
  - $A \in Fml_{\Sigma}^{mod} \implies \Box A, \Diamond A \in Fml_{\Sigma}^{mod}$
- pronounced “Box” and “Diamond”

## Modal Logics

Logics of *necessity* and *possibility*.

### Meaning of Modalities:

#### Modal

- $\Box A$  It is necessary that ...
- $\Diamond A$  It is possible that ...

#### Deontic (from Greek for duty)

- $\Box A$  It is obligatory that ...
- $\Diamond A$  It is permitted that ...

#### Epistemic (logic of knowledge)

- $\Box A$  I know that ...
- $\Diamond A$  I consider it possible that ...

## 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  $I : S \rightarrow 2^\Sigma$

# Recap: Modal Logic – Semantics

For a signature  $\Sigma$  and Kripke structure  $(S, R, I)$

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

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

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

$\models$  is homomorphic for  $\wedge, \vee, \rightarrow, \neg$ , i.e.:

$$I, s \models \varphi \wedge \psi \iff I, s \models \varphi \text{ and } I, s \models \psi$$

$$I, s \models \varphi \vee \psi \iff I, s \models \varphi \text{ or } I, s \models \psi$$

$$I, s \models \varphi \rightarrow \psi \iff I, s \models \varphi \text{ implies } I, s \models \psi$$

$$I, s \models \neg \varphi \iff \text{not } I, s \models \varphi$$

$$I, s \models \Box \varphi \iff I, s' \models \varphi \text{ for all } s' \in S \text{ with } (s, s') \in R$$

$$I, s \models \Diamond \varphi \iff I, s' \models \varphi \text{ for some } s' \in S \text{ with } (s, s') \in R$$

Example: Chalkboard

## Multi-modal logic

Have different Box operators with different accessibility relations:

$$\Box_{\alpha}, \Box_{\beta}, \Box_{\gamma}, \dots$$

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

## Propositional Dynamic Logic (PDL):

- Signature  $\Sigma$  of propositional variables
- Set  $A = \{\alpha, \beta, \dots\}$  of atomic actions/programs
- We write  $[\alpha]$  instead of  $\Box_{\alpha}$



## Compose Programs

Atomic programs can be 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  $A \subseteq \Pi_{\Sigma,A}$  atomic programs
- ②  $p, q \in \Pi_{\Sigma,A} \implies (p ; q) \in \Pi_{\Sigma,A}$  sequential composition
- ③  $p, q \in \Pi_{\Sigma,A} \implies (p \cup q) \in \Pi_{\Sigma,A}$  nondeterministic choice
- ④  $p \in \Pi_{\Sigma,A} \implies p^* \in \Pi_{\Sigma,A}$  indeterminate iteration
- ⑤  $F \in Fml_{\Sigma,A}^{PDL} \implies ?F \in \Pi_{\Sigma,A}$  tests

Regular Programs =

Regular Expressions over atomic programs and tests

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

- 1  $true, false \in Fml_{\Sigma,A}^{PDL}$
- 2  $\Sigma \subseteq Fml_{\Sigma,A}^{PDL}$
- 3  $A, B \in Fml_{\Sigma,A}^{PDL} \implies A \wedge B, A \vee B, A \rightarrow B, \neg A \in Fml_{\Sigma,A}^{PDL}$
- 4  $P \in \Pi_{Sigma,A}, A \in Fml_{\Sigma,A}^{PDL} \implies [P]A, \langle P \rangle A \in Fml_{\Sigma,A}^{PDL}$

Programs and Formulae are mutually dependent definitions and must be seen simultaneously.

→ Towers of Hanoi

$$A = \{\text{moveS}, \text{moveO}\}, \quad \Sigma = \{S1\}$$
$$S1 \rightarrow \langle (\text{moveO})^* ; \text{moveS} \rangle \neg S1$$

multi-level and nested modalities

$$A = \{\alpha, \beta\}, \quad \Sigma = \{P, Q\}$$

$$[\alpha \cup (?P ; \beta)^*]Q$$
$$[\alpha]P \rightarrow [\alpha^*]P$$
$$[\alpha]\langle \beta \rangle (P \rightarrow [\alpha^*]Q)$$
$$[\alpha ; ?\langle \beta \rangle P ; \beta]Q$$

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  $I : S \rightarrow 2^\Sigma$
- ⇒ 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*  
 $= \{(s_0, s_n) \mid \text{ex. } n \text{ with } (s_i, s_{i+1}) \in \rho(\pi) \text{ for } 0 \leq i < n\}$

$\rho(?F) = \{(s, s) \mid I, s \models F\}$

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

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

$\models$  is homomorphic for  $\wedge, \vee, \rightarrow, \neg$ .

$$I, s \models [\pi]\varphi \iff I, s' \models \varphi \text{ for all } s' \in S \text{ with } (s, s') \in \rho(\pi)$$

$$I, s \models \langle \pi \rangle \varphi \iff I, s' \models \varphi \text{ for some } s' \in S \text{ with } (s, s') \in \rho(\pi)$$

## Dual operators

$$[\pi]\varphi \leftrightarrow \neg\langle\pi\rangle\neg\varphi$$

- $[\pi_1 ; \pi_2]\varphi \leftrightarrow [\pi_1][\pi_2]\varphi$
- $[\pi_1 \cup \pi_2]\varphi \leftrightarrow [\pi_1]\varphi \wedge [\pi_2]\varphi$
- $[?\psi]\varphi \leftrightarrow \psi \rightarrow \varphi$
- $[\pi^*]\varphi \leftrightarrow \varphi \wedge [\pi ; \pi^*]\varphi$
  
- $\langle\pi_1 ; \pi_2\rangle\varphi \leftrightarrow \langle\pi_1\rangle\langle\pi_2\rangle\varphi$
- $\langle\pi_1 \cup \pi_2\rangle\varphi \leftrightarrow \langle\pi_1\rangle\varphi \vee \langle\pi_2\rangle\varphi$
- $\langle?\psi\rangle\varphi \leftrightarrow \psi \wedge \varphi$
- $\langle\pi^*\rangle\varphi \leftrightarrow \varphi \vee \langle\pi ; \pi^*\rangle\varphi$
  
- all tautologies for modal logic **K**

## Axioms

All propositional tautologies

$[\pi](\varphi \rightarrow \psi)$	$\rightarrow$	$([\pi]\varphi \rightarrow [\pi]\psi)$	(ML1 = K)
$[\pi](\varphi \wedge \psi)$	$\leftrightarrow$	$[\pi]\varphi \wedge [\pi]\psi$	(ML2)
$[\pi_1; \pi_2]\varphi$	$\leftrightarrow$	$[\pi_1][\pi_2]\varphi$	(PDL1)
$[\pi_1 \cup \pi_2]\varphi$	$\leftrightarrow$	$[\pi_1]\varphi \wedge [\pi_2]\varphi$	(PDL2)
$[?\varphi]\psi$	$\leftrightarrow$	$\varphi \rightarrow \psi$	(PDL3)
$[\pi^*]\varphi$	$\leftrightarrow$	$\varphi \wedge [\pi][\pi^*]\varphi$	(PDL4)
$\varphi \wedge [\pi^*](\varphi \rightarrow [\pi]\varphi)$	$\rightarrow$	$[\pi^*]\varphi$	(IND)

## Rules

$$\frac{\varphi, \varphi \rightarrow \psi}{\psi} \quad (\text{MP})$$

$$\frac{\varphi}{[\pi]\varphi} \quad (\text{GEN})$$



## Theorem

The presented calculus is sound and complete.

## Proof

See e.g., pp. 559-560

in David Harel's article *Dynamic Logic*  
in the *Handbook of Philosophical Logic, Volume II*,  
published by D.Reidel in 1984.

or

D. Harel, D. Kozen and J. Tiuryn

*Dynamic Logic*

in *Handbook of Philosophical Logic, 2<sup>nd</sup> edition*, volume 4  
by Kluwer Academic Publisher, 2001.

## Syntactic Sugar

- PDL syntax has elementary program operators
- Enrich it by defining new operators (“macros”)

**skip** :=  $?true$

**fail** :=  $?false$

**if**  $\varphi$  **then**  $\alpha$  **else**  $\beta$  :=  $(?\varphi ; \alpha) \cup (? \neg \varphi ; \beta)$

**while**  $\varphi$  **do**  $\alpha$  :=  $(?\varphi ; \alpha)^* ; ? \neg \varphi$

# More PDL Tautologies

$$[\mathbf{skip}]\varphi \leftrightarrow \varphi$$

$$\langle \mathbf{skip} \rangle \varphi \leftrightarrow \varphi$$

$$[\mathbf{fail}]\varphi \leftrightarrow \mathit{true}$$

$$\langle \mathbf{fail} \rangle \varphi \leftrightarrow \mathit{false}$$

$$[\mathbf{if} \ \varphi \ \mathbf{then} \ \alpha \ \mathbf{else} \ \beta]\psi \leftrightarrow (\varphi \rightarrow [\alpha]\psi) \wedge (\neg\varphi \rightarrow [\beta]\psi)$$

$$\langle \mathbf{if} \ \varphi \ \mathbf{then} \ \alpha \ \mathbf{else} \ \beta \rangle \psi \leftrightarrow (\varphi \rightarrow \langle \alpha \rangle \psi) \wedge (\neg\varphi \rightarrow \langle \beta \rangle \psi)$$

# Decidability

Is PDL decidable?



Is there an algorithm that terminates on every input and computes whether a PDL-formula  $\phi \in Fml_{\Sigma, A}^{PDL}$  is **satisfiable**.



Is there an algorithm that terminates on every input and computes whether a PDL-formula  $\phi \in Fml_{\Sigma, A}^{PDL}$  is **valid**.

**Answer:**

**YES**, PDL is decidable!

## General Idea:

$\varphi \in Fml^{PDL}$  has a model  $\iff \varphi$  has a model of bounded size.

For every Kripke structure, a bounded Kripke structure can be defined which is indistinguishable for  $\varphi$ .

## Preliminary lemma: Decidability for modal logic

The proof idea is the same, yet simpler.

## Reduced syntax

Only connectors  $\rightarrow$ , *false*,  $\Box$  are allowed  $\Rightarrow$  simplifies proofs.

## Operator

$$FL^{mod} : Fml^{mod} \rightarrow 2^{Fml^{mod}}$$

assigns to  $\varphi$  the set of subformulas of  $\varphi$ .

$$FL^{mod}(\varphi \rightarrow \psi) = \{\varphi \rightarrow \psi\} \cup FL^{mod}(\varphi) \cup FL^{mod}(\psi)$$

$$FL^{mod}(false) = \{false\}$$

$$FL^{mod}(p) = \{p\} \quad p \in \Sigma$$

$$FL^{mod}(\Box\varphi) = \{\Box\varphi\} \cup FL^{mod}(\varphi)$$

## Observation

$$|FL^{mod}(\varphi)| \leq |\varphi|$$

## Filtration

For a Kripke structure  $S, R, I$  define a bounded structure  $\tilde{S}, \tilde{R}, \tilde{I}$  with

$$S, R, I, s \models \varphi \iff \tilde{S}, \tilde{R}, \tilde{I}, \tilde{s} \models \varphi$$

## Central Idea

States are **undistinguishable** for  $\varphi$  if they are equal on  $FL^{mod}(\varphi)$ .

$$s \equiv t \iff (I, s \models \psi \iff I, t \models \psi \text{ for all } \psi \in FL^{mod}(\varphi))$$

$$\tilde{s} := \{s' \mid s' \equiv s\} \quad \dots \text{ equivalence classes}$$

$$\tilde{S} := \{\tilde{s} \mid s \in S\}$$

$$\tilde{R} := \{(\tilde{s}, \tilde{s}') \mid (s, s') \in R\}$$

$$\tilde{I}(p) := \{\tilde{s} \mid s \in I(p)\}$$



$$\tilde{s} := \{s' \mid s' \equiv s\}$$

$$\tilde{S} := \{\tilde{s} \mid s \in S\}$$

$$\tilde{R} := \{(\tilde{s}, \tilde{t}) \mid (s, t) \in R\}$$

$$\tilde{I}(p) := \{\tilde{s} \mid s \in I(p)\}$$

## Lemma

$$|\tilde{S}| \leq 2^{|FL^{mod}(\varphi)|} \leq 2^{|\varphi|}$$

## Lemma (proved by structural induction)

$$S, R, I, s \models \varphi \iff \tilde{S}, \tilde{R}, \tilde{I}, \tilde{s} \models \varphi$$

## Theorem

Modal Logic (K) can be decided by inspecting a bounded number of models.

## Operator

$$FL : Fml^{PDL} \rightarrow 2^{Fml^{PDL}}$$

$FL(\varphi)$  smallest set satisfying

- 1  $\varphi \in FL(\varphi)$
- 2  $(\psi_1 \rightarrow \psi_2) \in FL(\varphi) \Rightarrow \psi_1 \in FL(\varphi) \text{ and } \psi_2 \in FL(\varphi)$
- 3  $[\pi]\psi \in FL(\varphi) \Rightarrow \psi \in FL(\varphi)$
- 4  $[\pi_1; \pi_2]\psi \in FL(\varphi) \Rightarrow [\pi_1][\pi_2]\psi \in FL(\varphi)$
- 5  $[\pi_1 \cup \pi_2]\psi \in FL(\varphi) \Rightarrow [\pi_1]\psi \in FL(\varphi) \text{ and } [\pi_2]\psi \in FL(\varphi)$
- 6  $[\pi]^*\psi \in FL(\varphi) \Rightarrow [\pi][\pi^*]\psi \in FL(\varphi)$
- 7  $[\psi_1?]\psi_2 \in FL(\varphi) \Rightarrow \psi_1 \in FL(\varphi)$

Lemma (not obvious)

$$|FL(\varphi)| \leq |\varphi|$$

Same construction as for modal logic

extended:  $\tilde{\rho}(a) := \{(\tilde{s}, \tilde{t}) \mid (s, t) \in \rho(a)\}$  for all  $a \in A$

## Lemma

$$S, R, I, s \models \varphi \iff \tilde{S}, \tilde{R}, \tilde{I}, \tilde{s} \models \varphi$$

Prove by structural induction:

- If  $\psi \in FL(\varphi)$  then  $s \models \psi$  iff  $\tilde{s} \models \psi$
- $(s, t) \in \rho(\pi)$  implies  $(\tilde{s}, \tilde{t}) \in \tilde{\rho}(\pi)$  for all  $\pi \in \Pi_{\Sigma, A}$
- If  $(\tilde{s}, \tilde{t}) \in \tilde{\rho}(\pi)$  and  $s \models [\pi]\psi$ , then  $t \models \psi$  for  $[\pi]\psi \in FL(\varphi)$

full proof:  $\rightsquigarrow$  lecture notes or [Harel et al., Lemma 6.4]

## Naive approach used for proof

- $FL(\varphi) \in O(|\varphi|)$
  - $|\tilde{S}| \leq 2^{FL(\varphi)} \in O(2^{|\varphi|})$  many states in filtration
  - $|\text{models}| \leq (2^\Sigma)^{|\tilde{S}|} \in O(2^{2^{|\varphi|}})$
- ⇒ double exponential complexity

One can do better:

## Complexity of Deciding PDL

The decision problem for PDL is in EXPTIME:  
can be decided by a deterministic algorithm in  $O(2^{p(n)})$  for some polynomial  $p$ .

↪ [Harel et al. Ch. 8]

# Deduction Theorem and Compactness

$$M \subseteq \text{Fml}^{PDL}, \quad \varphi \in \text{Fml}^{PDL}$$

## Global Consequence

$$M \models^G \varphi : \iff$$

for all Kripke structures  $(S, \rho, I)$ :

$$I, s \models M \text{ for all } s \in S \quad \text{implies} \quad I, s \models \varphi \text{ for all } s \in S$$

## Local Consequence

$$M \models^L \varphi : \iff$$

for all Kripke structures  $(S, \rho, I)$ :

$$\text{for all } s \in S: \quad I, s \models M \text{ implies } I, s \models \varphi$$

**Local consequence is stronger:**  $M \models^L \varphi \implies M \models^G \varphi$   
 $\not\Leftarrow$

*Recall:* In propositional logic:

$$M \cup \{\varphi\} \models \psi \iff M \models \varphi \rightarrow \psi$$

**Not valid for PDL:**

$$p \models^G [\alpha]p \text{ but } \not\models^G p \rightarrow [\alpha]p$$

**Problem:**

Decidability has been shown only for  $\models \varphi$ .

## Questions

- 1 Is  $\psi \models^G \varphi$  decidable for PDL?
- 2 Is  $M \models^G \varphi$  decidable for PDL?

## Lemma

$$\psi \models^G \varphi \iff \models ([(\beta_1 \cup \dots \cup \beta_k)^*] \psi) \rightarrow \varphi$$

with  $B := \{\beta_1, \dots, \beta_k\}$  the atomic programs occurring in  $\psi, \varphi$ .

$\Leftarrow$  simple  $\rightsquigarrow$  Exercise

- $\Rightarrow$
- 1 Kripke structure  $(S, \rho, I)$ ,  $s \in S$ .
  - 2  $S^-(s) := \{s' \mid s' \text{ reachable from } s \text{ via } B.\} \subseteq S$
  - 3  $S^-(s) \models \psi \iff S^-(s), s \models [B^*] \psi$
  - 4  $S^-(s) \models \psi$  entails  $S^-(s) \models \varphi$  by assumption
  - 5  $S^-(s), s \models \varphi \iff S, s \models \varphi$

## Decidable:

The consequence problem  $\psi \models^G \varphi$  is decidable for PDL.



## Recall: Compactness Theorem

$$M \models^G \varphi \iff \text{exists finite } E \subseteq M \text{ with } E \models^G \varphi$$

### Holds for:

Propositional Logic, First Order Logic, **not** for higher order logic

## Counterexample for PDL

$$M := \{p \rightarrow \underbrace{[\alpha; \dots; \alpha]q}_{n \text{ times}} \mid n \in \mathbb{N}\}, \quad \varphi := p \rightarrow [\alpha^*]q$$

- $M \models^G \varphi$  ? *yes*
- $E \subset M, E \models^G \varphi$  ? *no*

## PDL is not compact

because it has transitive closure “built in”.

## Quote:

*[T]he problem of whether an arbitrary PDL formula  $p$  is deducible from a single fixed axiom scheme is of extremely high degree of undecidability, namely  $\Pi_1^1$ -complete.*

Meyer, Streett, Mirkowska:

*The Deducibility Problem in Propositional Dynamic Logic, 1981*

# Variants and Conclusion

Idea: Add actions reverting action effects

Add further program constructor  $\cdot^{-1}$ :

$$\pi \in \Pi \implies \pi^{-1} \in \Pi$$

$$\text{with } \rho(\pi^{-1}) = \rho(\pi)^{-1}$$

**Axiom schemes:** for all  $\varphi \in \text{Fml}^{PDL}$ ,  $\pi \in \Pi$

- $\varphi \rightarrow [\pi]\langle\pi^{-1}\rangle\varphi$
- $\varphi \rightarrow [\pi^{-1}]\langle\pi\rangle\varphi$

**Complete**

Adding the axioms to the known PDL calculus gives a correct and complete calculus for PDL with Converse.

# Variante: Context-free Programs

Idea: Go beyond regular programs

Instead of regular programs, allow context-free grammar

**For example:**

Produced context-free grammar  $X ::= \alpha X \gamma \mid \beta$

with  $L(X) = \{\alpha^n \beta \gamma^n \mid n \in \mathbb{N}\}$

Undecidability result

Validity is undecidable if instead of regular programs, context-free programs are allowed.

Expressiveness

Without fixed semantics of  $\mathbb{N}$ , recursion is strictly more expressive than looping.

A propositional Kripke structure  $\mathcal{K} = (S, \rho, I)$  is determined by:

$S$  the set of states  
 $\rho : A \rightarrow S \times S$  the accessibility relations for atomic programs  $a$   
 $I : S \rightarrow 2^\Sigma$  evaluation of propositional atoms in states

**Choose now:**  $S \subseteq 2^\Sigma$  the set of states

We call this the **state vector semantics**.

- Strictly larger set of tautologies.
- Obviously decidable.
- Evaluation of propositional variables fixes the state (and the accessibility of successor states)

Let

- $A = \{a_1, \dots, a_k\}$
- $\pi_{all}$  stands for the program  $(a_1 \cup \dots \cup a_k)^*$ .
- $U \subseteq \Sigma$  be a subset of the set of propositional atoms.
- $state_U$  abbreviate  $\bigwedge_{p \in U} p \wedge \bigwedge_{p \notin U} \neg p$ .
- $F$  an arbitrary PDL formula.

Then

$$\langle \pi_{all} \rangle (state_U \wedge F) \rightarrow [\pi_{all}] (state_U \rightarrow F)$$

is true in all state vector Kripke structures.

Let  $H$  be the set of all formulas

$$\langle \pi_{all} \rangle (state_U \wedge F) \rightarrow [\pi_{all}] (state_U \rightarrow F)$$

with the notation from the previous slide.

Then:

- 1  $\{F\} \cup H$  is satisfiable iff  $F$  is state vector satisfiable.
- 2  $H \models F$  iff  $\models_{sv} F$ .



- extension of modal logic
- abstract notion of actions / atomic logic statements
- regular programs, with non-deterministic choice and Kleene-iteration
- correct and complete calculus for tautologies
- satisfiability is decidable (in EXPTIME)
- logic is not compact
- deducibility is utterly undecidable
- deduction theorem can be rescued

**Detection of dynamic execution errors in  
IBM system automation's rule-based expert system**

# **An Application of PDL**

[SinzEtAl02]

Carsten Sinz, Thomas Lumpp, Jürgen Schneider, and Wolfgang Küchlin:

**Detection of dynamic execution errors in IBM System Automation's rule-based expert system.**

*Information and Software Technology*, 44(14):857–873, November 2002.

# Context



## IBM zSeries

- **z** = zero downtime
- high availability: 99.999%
- $< 5.3min/yr$  downtime

## System Automation

- full automation of a data center
- starting, stopping, migration of applications
- recovery from system failures
- ...
- **complex, rule-based configuration**

## Example

Flight booking center: 100s of users, many parallel apps

# Example Rule

```
correlation set/status/compound/satisfactory :  
when      status/compound NOT E {Satisfactory}  
          AND status/startable E {Yes}  
          AND ( ( status/observed E {Available, WasAvailable}  
                AND status/desired E {Available}  
                AND status/automation E {Idle, Internal}  
                AND correlation/external/stop/failed E {false}  
              )  
            OR  
            ( status/observed E {SoftDown, StandBy}  
              AND status/desired E {Unavailable}  
              AND status/automation E {Idle, Internal}  
            )  
          )  
then SetVariable status/compound = Satisfactory  
      RecordVariableHistory status/compound
```

Fig. 4. Example of a correlation rule.

(taken from [SinzEtAl02])

**when** *cond* **then** *var* = *d*

- **AND, OR, NOT** allowed in conditions
- *var* **E** { *d*<sub>1</sub>, . . . , *d*<sub>2</sub> } – “element of”
- the **then** part can be executed if **cond** is true

- One boolean atom per var/value-pair
- $P_{var,d} = true \iff var = d$
- Encode that  $var$  has exactly one value (of  $d_1, \dots, d_k$ )
- $$\left( \bigvee_{i=1..k} P_{var,d_i} \right) \wedge \left( \bigwedge_{\substack{i,j=1..k \\ i < j}} \neg(P_{var,d_i} \wedge P_{var,d_j}) \right)$$
- Atomic Actions:  $var = d \rightsquigarrow \alpha_{var,d}$
- Axiom  $[\alpha_{var,d}]P_{var,d}$



## Semantics of a rule as program:

*?when ; then*

## Semantics of all rules as program:

$$R := ((?when_1 ; then_1) \cup \dots \cup (?when_r ; then_r))^*$$

## Uniqueness of final state:

under assumption of a precondition  $PRE$

$$PRE \rightarrow (\langle R \rangle p \leftrightarrow [R] p)$$

## Confluence:

$$PRE \rightarrow (\langle R \rangle [R] p \rightarrow [R] \langle R \rangle p)$$

## Absence of Oscillation:

modelled using an extension of PDL with non-termination operator

## Verification Technique

- state vector semantics
- translation of PDL to boolean SAT
- solving using SAT solver (Davies-Putnam)

### Experiment:

- $\sim 40$  rules
- resulted in  $\sim 1500$  boolean variables
- SAT solving  $< 1$  sec
- !! **violations found – before deployment**