

# Propositional Dynamic Logic

Mathematical Logic for Computer Science

Alessio Bandiera

1985878

# Dynamic Logics

**Dynamic Logics** are modal logics for representing the states and the events of dynamic systems. The first DL system was developed in 1976 by Vaughan Pratt, an early pioneer of computer science

His original DL was a *first-order* modal logic, and **Propositional Dynamic Logic** (PDL) is the propositional counterpart of it

Its only two syntactic categories are **propositions** and **programs**, and *possibility* and *necessity* are expressed through modal operators that also indicate the programs they are referring to

- $\langle \pi \rangle \phi$  is read "there is an execution of  $\pi$  that ends in a state in which  $\phi$  is true"
- $[\pi] \psi$  is read "all executions of the program  $\pi$  end in states in which  $\psi$  is true"

## Formulas

Given  $\Phi_0$  the set of *atomic formulas*, for any  $\phi, \psi \in \Phi_0$

- $\phi \in \text{Form}(\Phi_0)$
- $\neg\phi \in \text{Form}(\Phi_0)$
- $\phi \vee \psi \in \text{Form}(\Phi_0)$
- $[\alpha]\phi \in \text{Form}(\Phi_0)$

where  $\alpha \in \text{Prog}(\Pi_0)$

## Programs

Given  $\Pi_0$  the set of *atomic programs*, for any  $\alpha, \beta \in \Pi_0$

- $\alpha \in \text{Prog}(\Pi_0)$
- $(\alpha; \beta) \in \text{Prog}(\Pi_0)$
- $(\alpha \cup \beta) \in \text{Prog}(\Pi_0)$
- $\alpha^* \in \text{Prog}(\Pi_0)$
- $\phi? \in \text{Prog}(\Pi_0)$

where  $\phi \in \text{Form}(\Phi_0)$

## Relations

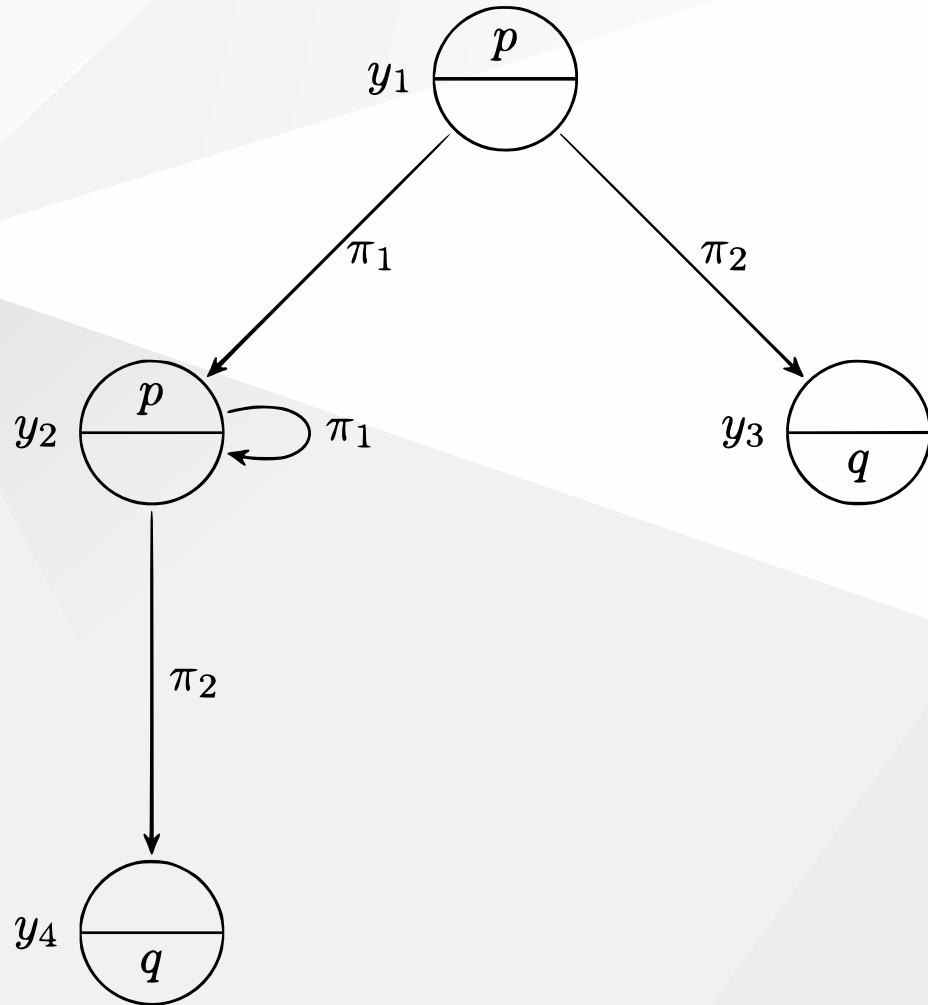
$$(x, y) \in R(\pi) \iff x \xrightarrow{\pi} y$$

- $(x, y) \in R(\alpha; \beta) \iff \exists z \in W \quad (x, z) \in R(\alpha) \wedge (z, y) \in R(\beta)$
- $(x, y) \in R(\alpha \cup \beta) \iff (x, y) \in R(\alpha) \vee (x, y) \in R(\beta)$
- $(x, y) \in R(\alpha^*) \iff \exists z_0, \dots, z_n \in W \quad \begin{cases} z_0 = x \\ z_n = y \\ (z_{k-1}, z_k) \in R(\alpha) \end{cases}$
- $(x, y) \in R(\phi?) \iff x = y \wedge y \in V(\phi)$

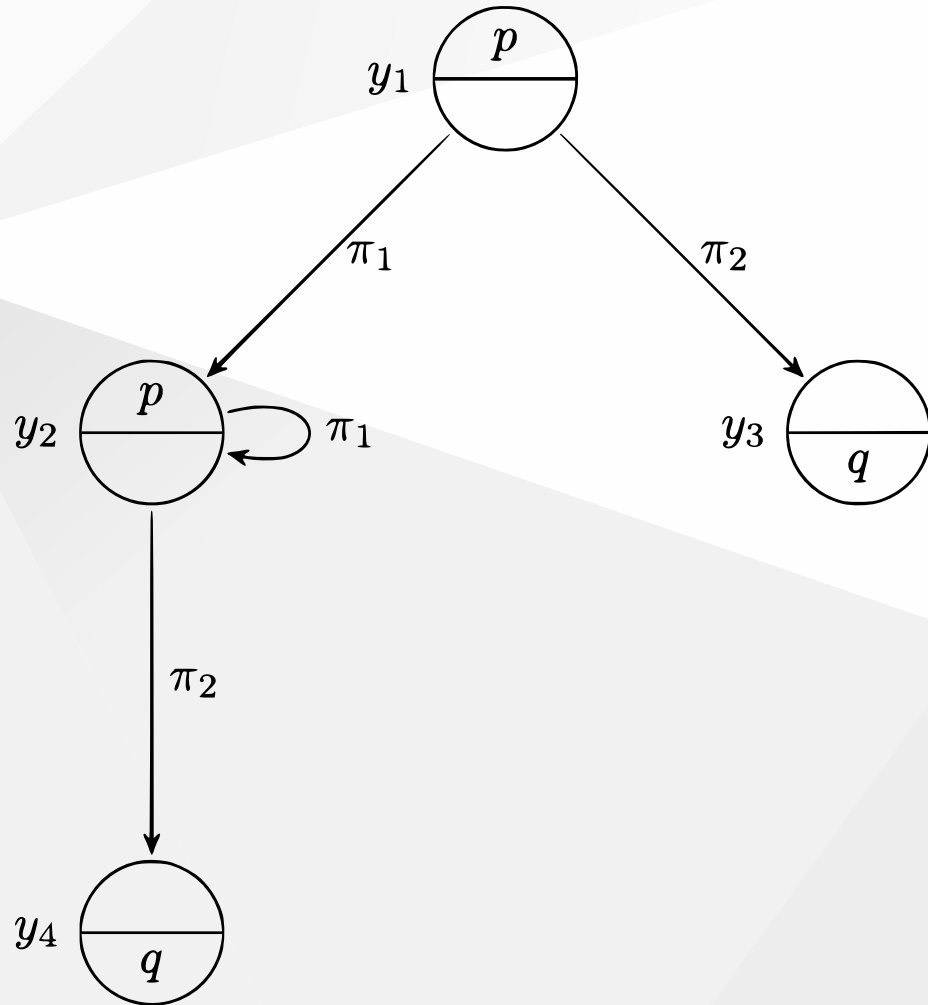
## Valuations

$$x \in V(p) \iff p \text{ is true at } x$$

- $V(\perp) = \emptyset$
- $V(\top) = W$
- $V(\neg\phi) = W - V(\phi)$
- $V(\phi \vee \psi) = V(\phi) \cup V(\psi)$
- $V([\alpha]\phi) = \{x \mid \forall y \in W \quad (x, y) \in R(\alpha) \implies y \in V(\phi)\}$



- $W = \{y_1, y_2, y_3, y_4\}$
- $R(\pi_1) = \{(y_1, y_2), (y_2, y_2)\}$
- $R(\pi_2) = \{(y_1, y_3), (y_2, y_4)\}$
- $V(p) = \{y_1, y_2\}$
- $V(q) = \{y_3, y_4\}$



- $\mathfrak{M}, y_1 \models \langle \pi_1^*; \pi_2 \rangle q$
- $\mathfrak{M}, y_2 \models [\pi_1^*]p$
- $\mathfrak{M}, y_1 \models [\pi_1 \cup \pi_2](p \vee q)$
- $\mathfrak{M}, y_3 \models [\pi_1 \cup \pi_2]\perp$



## Validity

We write  $\mathfrak{M}, w \models \phi$  if and only if  $w \in V(\phi)$

$\phi$  is *valid* in  $\mathfrak{M}$ , written as  $\mathfrak{M} \models \phi$ , if and only if

$$\mathfrak{M} \models \phi \iff \forall w \in W \quad \mathfrak{M}, w \models \phi$$

$\phi$  is **valid**, written as  $\models \phi$ , if and only if

$$\models \phi \iff \forall \mathfrak{M} \quad \mathfrak{M} \models \phi$$

## Goal

The goal is to define a **decidability predicate**  $\vdash$  such that  $\vdash$ -deductions are both *sound* and *complete* in terms of validity, i.e. for any  $\phi$  it holds that

$$\vdash \phi \iff \models \phi$$

## K and N axioms

$$(K) \quad [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi)$$

$$(N) \quad \frac{\phi}{[\pi]\phi}$$

A modal logic is **normal** if it obeys (K) and (N)

## PDL axioms

PDL is the *least normal* modal logic containing every instance of

- (A1)  $[\alpha; \beta]\phi \leftrightarrow [\alpha][\beta]\phi$
- (A2)  $[\alpha \cup \beta]\phi \leftrightarrow [\alpha]\phi \wedge [\beta]\phi$
- (A3)  $[\alpha^*]\phi \leftrightarrow \phi \wedge [\alpha][\alpha^*]\phi$
- (A4)  $[\phi?]\psi \leftrightarrow (\phi \rightarrow \psi)$

and closed under the *loop invariance* rule of inference

$$(I) \quad \frac{\phi \rightarrow [\alpha]\phi}{\phi \rightarrow [\alpha^*]\phi}$$

## $\vdash$ -deducibility

A formula  $\phi$  is  $\vdash$ -deducible from  $\Sigma \subseteq \text{Form}(\Phi_0)$  if there exists a sequence  $\phi_0, \dots, \phi_n$  such that  $\phi_n = \phi$ , and for all  $i \in [n]$

- $\phi_i$  is an instance of an axiom schema
- $\phi_i$  is an instance of a formula of  $\Sigma$
- $\phi_i$  comes from earlier formulas of the sequence by inference

Are  $\vdash$ -deductions sound and complete?

## Segerberg's axioms

In 1977 Segerberg proposed to replace

$$(I) \quad \frac{\phi \rightarrow [\alpha]\phi}{\phi \rightarrow [\alpha^*]\phi}$$

with the following fifth axiom

$$(A5) \quad \phi \wedge [\alpha^*](\phi \rightarrow [\alpha]\phi) \rightarrow [\alpha^*]\phi$$

in order to prove that such axiomatization was sound and complete

## Segerberg's axioms

Indeed, it is easy to prove that (I) can be replaced with (A5)

- |   |   |
|---|---|
| 1. $\vdash \phi \rightarrow [\alpha]\phi$   | (premise)                                 |
| 2. $\vdash [\alpha^*](\phi \rightarrow [\alpha]\phi)$   | (from 1 using (N) with $\pi = \alpha^*$ ) |
| 3. $\vdash \phi \wedge [\alpha^*](\phi \rightarrow [\alpha]\phi) \rightarrow [\alpha^*]\phi$        | (A5)                                      |
| 4. $\vdash [\alpha^*](\phi \rightarrow [\alpha]\phi) \rightarrow (\phi \rightarrow [\alpha^*]\phi)$ | (from 3 through prop. reasoning)          |
| 5. $\vdash \phi \rightarrow [\alpha^*]\phi$   | (from 2 and 4 using <i>Modus Ponens</i> ) |

## Soundness

To prove that  $\vdash$  is sound w.r.t.  $\models$ , i.e. that

$$\vdash \phi \implies \models \phi$$

a proof by induction on the length of  $\phi$ 's deduction in  $\vdash$  suffices.

So, what about completeness? It requires to prove that

$$\models \phi \implies \vdash \phi$$



## Parikh's proof

Seegerberg's work was the first attempt to prove the completeness of  $\vdash$ , however in 1978 he found a flaw in his argument

Then in the same year Parikh published what is now considered the first proof of the completeness of  $\vdash$

## PDL satisfiability

$\phi$  is *satisfiable* in  $\mathfrak{M}$  if there is a world  $w \in W$  such that  $\mathfrak{M}, w \models \phi$

$\phi$  is **satisfiable** if there is a model  $\mathfrak{M}$  such that  $\phi$  is satisfiable in  $\mathfrak{M}$

$\text{PDL-SAT} := \{ \langle \phi \rangle \mid \phi \text{ is a satisfiable PDL formula} \}$

## Unsatisfiable formulas

$\phi$  is *unsatisfiable* if and only if  $\neg\phi$  is *valid*

Therefore, we can use the recursive definition of valid PDL formulas and build a procedure  $P$  that enumerates all the  $\vdash$ -deducible formulas

Hence, given enough time if  $\neg\phi$  is  $\vdash$ -deducible  $P$  will eventually find it and determine that  $\phi$  is *unsatisfiable*

This proves that  $\text{PDL-SAT} \in \text{coREC}$

## Satisfiable formulas

However, if  $\phi$  is satisfiable  $P$  never terminates.

Nonetheless, we can leverage the **finite model property** of PDL

$$\forall \phi \in \text{Form}(\Phi_0) \quad \langle \phi \rangle \in \text{PDL-SAT} \implies \exists \mathfrak{M}_{fin} \text{ finite} \quad \phi \text{ satisfiable in } \mathfrak{M}_{fin}$$

Therefore, there is a procedure  $P'$  that enumerates all the finite models  $\mathfrak{M}_{fin}$  and checks for each model if  $\phi$  is satisfiable in  $\mathfrak{M}_{fin}$

Thus,  $P$  and  $P'$  can be run in parallel to decide PDL-SAT. However, this is very inefficient, can we do any better?

## Small model property

Kozen and Parikh proved that PDL has also the **small model property**

$$\forall \phi \in \text{Form}(\Phi_0) \quad \langle \phi \rangle \in \text{PDL-SAT} \implies \exists \mathfrak{M}_{fin} \text{ finite} \quad \begin{cases} |\mathfrak{M}_{fin}| < \exp(|\phi|) \\ \phi \text{ satisfiable in } \mathfrak{M}_{fin} \end{cases}$$

This property implies that we can stop  $P'$  as soon as all the "small" models have been exhausted, to conclude that  $\phi$  is not satisfiable

This concludes that  $\text{PDL-SAT} \in \text{NEXP}$ . In 1980 Pratt was able to prove that  $\text{PDL-SAT} \in \text{EXP-complete}$

## Variations of PDL

Over the years multiple versions of PDL have been studied

We will discuss the following

- Test-free PDL
- CPDL

## Expressive power

The "?" operator seems *different* with respect to the other programs, can we remove this operator from PDL?

Let  $\text{PDL}_0$  be the test-free version of PDL. In 1981 Berman and Paterson proved that this PDL formula

$$\langle (P?; A)^*; \neg P?; A; P? \rangle \top$$

has no  $\text{PDL}_0$  equivalent formula

## Ultimate periodicity

The idea of their counterexample is based on the following result in the theory of context-free languages:

A unary language  $L = \{1^n \mid n \in \mathbb{N}\}$  is *regular* if and only if the set  $\{n \in \mathbb{N} \mid 1^n \in L\}$  is *ultimately periodic*

A set  $S \subseteq \mathbb{N}$  is **ultimately periodic** if there are integers  $X \in \mathbb{N}$  and  $Y > 0$  such that

$$\forall k \geq X \quad k \in S \iff k + Y \in S$$



## Ultimate periodicity

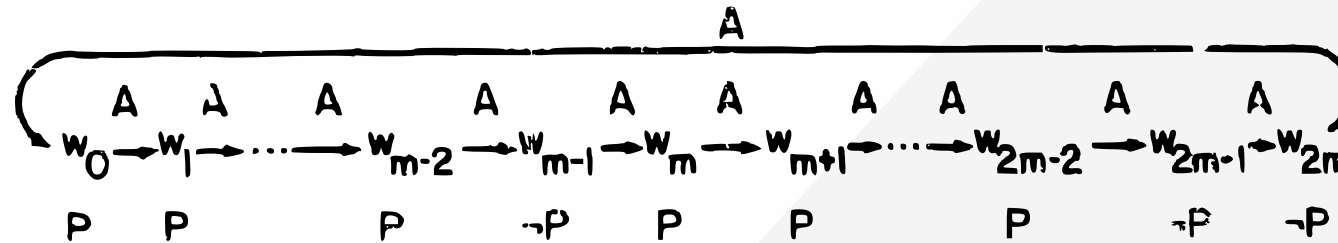
By removing tests from PDL formulas, programs are restricted to regular expressions

Hence, Berman and Paterson built a family of models  $\mathfrak{A}_m$  for  $m \geq 2$  in which the only program present is  $A$

Therefore, by ultimate periodicity each program over  $\mathfrak{A}_m$  can be rewritten as a regex

$$A^h(A^n)^*$$

## The counterexample

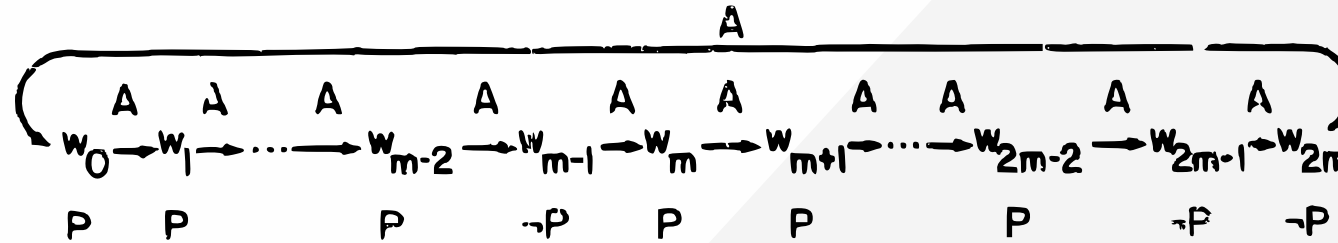


Each  $\mathfrak{A}_m$  consists of  $2m + 1$  worlds, where  $2m + 1$  is *prime*

This forces  $(A^n)^*$  to generate all the possible residues modulo  $2m + 1$ , i.e. each world will be able to reach any other world

Hence, test-free PDL formulas *cannot distinguish* the worlds in which we are performing the evaluation

## The counterexample



However, tests *can* distinguish the worlds by building programs which **depend on the truthness of propositions**

$$\langle (P?; A)^*; \neg P?; A; P? \rangle \top$$

In fact, this formula is satisfied at  $w_0$  but not satisfied at  $w_m$

## The converse operator

CPDL is a variant which adds the **converse** operator to PDL programs

$$(x, y) \in R(\alpha^{-1}) \iff (y, x) \in R(\alpha)$$

To get a sound and complete system, two additional axioms are needed

$$(A6) \quad \phi \rightarrow [\alpha] \langle \alpha^{-1} \rangle \phi$$

$$(A7) \quad \phi \rightarrow [\alpha^{-1}] \langle \alpha \rangle \phi$$

As for PDL, CPDL has the **small model property** too, and  $\text{CPDL-SAT} \in \text{EXP-complete}$  as well

## Expressive power

What about the expressive power? Consider these two models

$$\mathfrak{M} = (M, R, V) \qquad \mathfrak{M}' = (W', R', V')$$

$\mathfrak{M}$	$\mathfrak{M}'$
$W = \{x, y\}$	$W' = \{y'\}$
$R(\pi) = \{(x, y)\}$	$R'(\pi) = \emptyset$
$V(x) = V(y) = \emptyset$	$V'(y') = \emptyset$

## Expressive power

$\mathfrak{M}$	$\mathfrak{M}'$
$W = \{x, y\}$	$W' = \{y'\}$
$R(\pi) = \{(x, y)\}$	$R'(\pi) = \emptyset$
$V(x) = V(y) = \emptyset$	$V'(y') = \emptyset$

From the perspective of PDL  $y$  and  $y'$  are *indistinguishable*, in fact

$$\mathfrak{M}, y \models \phi \iff \mathfrak{M}', y' \models \phi$$

## Expressive power

$\mathfrak{M}$	$\mathfrak{M}'$
$W = \{x, y\}$	$W' = \{y'\}$
$R(\pi) = \{(x, y)\}$	$R'(\pi) = \emptyset$
$V(x) = V(y) = \emptyset$	$V'(y') = \emptyset$

However CPDL *can* distinguish  $y$  and  $y'$  because

$$\mathfrak{M}, y \models \langle \pi^{-1} \rangle \top \quad \mathfrak{M}', y' \not\models \langle \pi^{-1} \rangle \top$$

meaning that CPDL has **more expressive power** than PDL

Thanks for your attention

Mathematical Logic for Computer Science  
Alessio Bandiera  
1985878