# Day 5: The Epistemology of Nondeterminism

NASSLLI 2022

Adam Bjorndahl

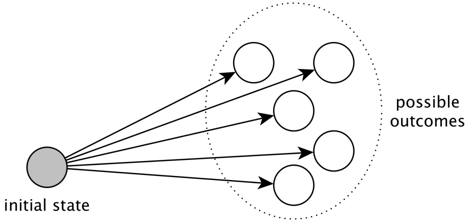
Carnegie Mellon University

# Propositional dynamic logic

Propositional dynamic logic (PDL) is a framework for reasoning about *nondeterministic program execution*.

Models are relational structures interpreted as state transition diagrams:

- Each program $\pi$ is associated with a binary relation $R_\pi$ on the state space.
- $xR_\pi y$ means that the state $y$ is one possible result of executing $\pi$ in $x$.



# Propositional dynamic logic

The language of PDL includes a unary modality $\langle \pi \rangle$ for each program $\pi$, where

$$x \models \langle \pi \rangle \varphi \quad \Leftrightarrow \quad \exists y (x R_\pi y \text{ and } y \models \varphi).$$

Thus, $\langle \pi \rangle \varphi$ is true just in case some possible execution of $\pi$ results in $\varphi$.

# Propositional dynamic logic

The language of PDL includes a unary modality $\langle\pi\rangle$ for each program $\pi$, where

$$x \models \langle\pi\rangle\varphi \quad\Leftrightarrow\quad \exists y(xR_\pi y \text{ and } y \models \varphi).$$

Thus, $\langle\pi\rangle\varphi$ is true just in case some possible execution of $\pi$ results in $\varphi$.

What is the sense of *possibility* at play here? We explore an epistemic account.

# Nondeterminism as uncertainty?

Can we understand nondeterminism as a kind of uncertainty?

# Nondeterminism as uncertainty?

Can we understand nondeterminism as a kind of uncertainty?

1. Reinterpret program execution as fundamentally deterministic.
   - Replace each relation $R_\pi \subseteq X^2$ with a function $f_\pi : X \to X$.
   - Write $\bigcirc_\pi \varphi$ for "after $\pi$, $\varphi$":

$$x \models \bigcirc_\pi \varphi \quad \Leftrightarrow \quad f_\pi(x) \models \varphi.$$

# Nondeterminism as uncertainty?

Can we understand nondeterminism as a kind of uncertainty?

1. Reinterpret program execution as fundamentally deterministic.
   - Replace each relation $R_\pi \subseteq X^2$ with a function $f_\pi : X \to X$.
   - Write $\bigcirc_\pi \varphi$ for "after $\pi$, $\varphi$":

$$x \models \bigcirc_\pi \varphi \quad \Leftrightarrow \quad f_\pi(x) \models \varphi.$$

2. Enrich the logical setting with a standard knowledge modality $K$, with dual $\hat{K}$, and define

$$\langle \pi \rangle \varphi \ \equiv \ \hat{K} \bigcirc_\pi \varphi.$$

# Nondeterminism as uncertainty?

$$\langle \pi \rangle \varphi \;\equiv\; \hat{K} \bigcirc_\pi \varphi.$$

On this view, the nondeterministic outcomes of $\pi$ are simply those compatible with the agent's current state of knowledge.

# Nondeterminism as uncertainty?

$$\langle \pi \rangle \varphi \;\equiv\; \hat{K} \bigcirc_{\pi} \varphi.$$

On this view, the nondeterministic outcomes of $\pi$ are simply those compatible with the agent's current state of knowledge.

- But this seems to miss the essence of nondeterminism.

# Nondeterminism as uncertainty?

$$\langle \pi \rangle \varphi \; \equiv \; \hat{K} \bigcirc_{\pi} \varphi.$$

On this view, the nondeterministic outcomes of $\pi$ are simply those compatible with the agent's current state of knowledge.

- But this seems to miss the essence of nondeterminism.
- For a very uninformed agent, we must interpret $\pi$ as having many possible nondeterministic outcomes.

# Nondeterminism as uncertainty?

$$\langle \pi \rangle \varphi \;\equiv\; \hat{K} \bigcirc_{\pi} \varphi.$$

On this view, the nondeterministic outcomes of $\pi$ are simply those compatible with the agent's current state of knowledge.

- But this seems to miss the essence of nondeterminism.
- For a very uninformed agent, we must interpret $\pi$ as having many possible nondeterministic outcomes.
- There is a clear intuitive distinction between those outcomes of $\pi$ that are possible as far as some (possibly quite ignorant) agent knows, and those outcomes that would remain possible *even with good information*.

# Nondeterminism as uncertainty?

Suppose you run a random number generator. This seems a canonical example of a nondeterministic process.

- Not only do you not know what number will be generated, but you are *unable in principle* to determine this in advance.

By contrast, suppose you run a program that prints the next entry in a given database.

- We do not want to call this program nondeterministic, even if you happen to currently be ignorant about the contents of the database.

# Nondeterminism as uncertainty?

Suppose you run a random number generator. This seems a canonical example of a nondeterministic process.

- Not only do you not know what number will be generated, but you are *unable in principle* to determine this in advance.

By contrast, suppose you run a program that prints the next entry in a given database.

- We do not want to call this program nondeterministic, even if you happen to currently be ignorant about the contents of the database.

This is a distinction we want to respect.

- The relevant epistemic notion is not what any given agent currently happens to know, but what they *could come to know*.

# Topological semantics

We need models rich enough to encode this notion of “potential knowledge” or “knowability”. Of course, we use topology.

# Topological semantics

We need models rich enough to encode this notion of "potential knowledge" or "knowability". Of course, we use topology.

Consider the modal language generated by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box\varphi,$$

where $p \in \textsc{prop}$ and $\Box\varphi$ stands for "$\varphi$ is knowable".

Formulas of this language are interpreted in **topological models** $M = (X, \mathcal{T}, v)$, where:

- $(X, \mathcal{T})$ is a topological space, and
- $v : \textsc{prop} \to 2^X$ is a valuation.

$$\begin{array}{lcl} x \models p & \Leftrightarrow & x \in v(p) \\ x \models \neg\varphi & \Leftrightarrow & x \not\models \varphi \\ x \models \varphi \wedge \psi & \Leftrightarrow & x \models \varphi \text{ and } x \models \psi \\ x \models \Box\varphi & \Leftrightarrow & x \in int([\![\varphi]\!]). \end{array}$$

# Topological semantics

Recall that we write $\Diamond$ for $\neg\Box\neg$, the dual of $\Box$, and correspondingly interpret it as the dual of the interior operator:

$$\begin{aligned} x \models \Diamond\varphi \quad &\Leftrightarrow \quad x \in cl(\llbracket\varphi\rrbracket) \\ &\Leftrightarrow \quad \forall U \in \mathcal{T}(x \in U \text{ implies } U \cap \llbracket\varphi\rrbracket \neq \emptyset). \end{aligned}$$

# Topological semantics

Recall that we write $\Diamond$ for $\neg\Box\neg$, the dual of $\Box$, and correspondingly interpret it as the dual of the interior operator:

$$\begin{aligned} x \models \Diamond\varphi &\Leftrightarrow x \in cl([\![\varphi]\!]) \\ &\Leftrightarrow \forall U \in \mathcal{T}(x \in U \text{ implies } U \cap [\![\varphi]\!] \neq \emptyset). \end{aligned}$$

We then read $\Diamond\varphi$ as "$\varphi$ is unfalsifiable": no measurement one could take at state $x$ would rule out the possibility of $\varphi$.

# Dynamic topological logic

Topological models equipped with a function $f : X \to X$ have been studied in depth.[1] These are called **dynamic topological models**.

[1] Kremer, P. and Mints, G. *Dynamic Topological Logic*.

# Dynamic topological logic

Topological models equipped with a function $f : X \to X$ have been studied in depth.[1] These are called **dynamic topological models**.

Of course, we can also equip a topological model with a family of functions $\{f_\pi\}_{\pi \in \Pi}$, and expand the basic language with the dynamic modalities defined previously:

$$x \models \bigcirc_\pi \varphi \quad \Leftrightarrow \quad f_\pi(x) \models \varphi.$$

[1] Kremer, P. and Mints, G. *Dynamic Topological Logic*.

# Nondeterminism as unfalsifiability

This setting is rich enough to realize our earlier intuition.

Identify the nondeterministic outcomes of a program $\pi$ with those that cannot be ruled out by *any* feasible measurement:

$$\langle \pi \rangle \varphi \;\equiv\; \Diamond \bigcirc_\pi \varphi.$$

# Nondeterminism as unfalsifiability

This setting is rich enough to realize our earlier intuition.

Identify the nondeterministic outcomes of a program $\pi$ with those that cannot be ruled out by *any* feasible measurement:

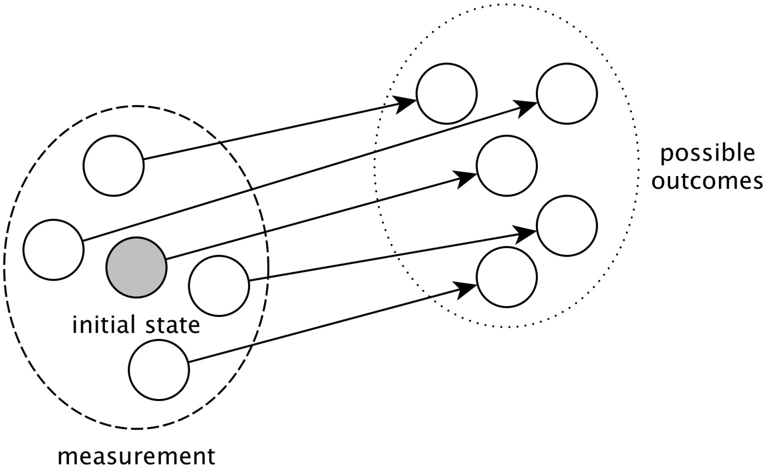
$$\langle \pi \rangle \varphi \;\equiv\; \Diamond \bigcirc_{\pi} \varphi.$$

Dually,

$$[\pi] \varphi \;\equiv\; \Box \bigcirc_{\pi} \varphi.$$

That is, $\varphi$ is a guaranteed outcome of $\pi$ just in case there is some measurement the agent can take *before running* $\pi$ that guarantees $\varphi$ will be true *after running* $\pi$.

# Nondeterminism as unfalsifiability



# Embedding PDL in dynamic topological logic

We will return to refine this understanding of nondeterminism, but first we should check that our topological reinterpretation of PDL is true to the original.

The most basic version of (serial) PDL (without any operations on programs) is axiomatized by

| | |
|---|---|
| CPL | propositional tautologies |
| $\mathsf{K}_\pi$ | $[\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi)$ |
| $\mathsf{D}_\pi$ | $[\pi]\varphi \to \langle\pi\rangle\varphi$ |
| MP | from $\varphi$ and $\varphi \to \psi$ deduce $\psi$ |
| $\mathsf{Nec}_\pi$ | from $\varphi$ deduce $[\pi]\varphi$. |

Call this system $\mathsf{SPDL}_0$.

# Embedding PDL in dynamic topological logic

Of course, we can interpret the language of PDL directly in dynamic topological models via our characterization of $\langle \pi \rangle$:

$$\begin{aligned} x \models \langle \pi \rangle \varphi \quad &\Leftrightarrow \quad \Big( x \models \Diamond \bigcirc_\pi \varphi \Big) \\ &\Leftrightarrow \quad x \in cl(f_\pi^{-1}(\llbracket \varphi \rrbracket)). \end{aligned}$$

# Embedding PDL in dynamic topological logic

Of course, we can interpret the language of PDL directly in dynamic topological models via our characterization of $\langle\pi\rangle$:

$$\begin{aligned} x \models \langle\pi\rangle\varphi \quad &\Leftrightarrow \quad \Big(x \models \Diamond\bigcirc_\pi\varphi\Big) \\ &\Leftrightarrow \quad x \in cl(f_\pi^{-1}(\llbracket\varphi\rrbracket)). \end{aligned}$$

Under these semantics, we obtain the following result.

## Proposition

$\mathsf{SPDL}_0$ *is a sound and complete axiomatization of the language of PDL with respect to the class of all dynamic topological models.*

# Embedding PDL in dynamic topological logic

In fact, given any (serial) PDL model $M = (X, (R_\pi)_{\pi \in \Pi}, v)$, we can construct a dynamic topological model

$$\tilde{M} = (\tilde{X}, \mathfrak{T}, (f_\pi)_{\pi \in \Pi}, \tilde{v})$$

where for every $x \in X$ there exists $\alpha \in \tilde{X}$ such that $x$ and $\alpha$ agree on all formulas of PDL.

# Embedding PDL in dynamic topological logic

In fact, given any (serial) PDL model $M = (X, (R_\pi)_{\pi \in \Pi}, v)$, we can construct a dynamic topological model

$$\tilde{M} = (\tilde{X}, \mathfrak{T}, (f_\pi)_{\pi \in \Pi}, \tilde{v})$$

where for every $x \in X$ there exists $\alpha \in \tilde{X}$ such that $x$ and $\alpha$ agree on all formulas of PDL.

- Points $\alpha \in \tilde{X}$ are *networks* of $R_\pi$-paths through $X$:
  - $\alpha : \Pi^* \to X$ where $(\forall \vec{\pi} \in \Pi^*)(\forall \pi \in \Pi)(\alpha(\vec{\pi}) R_\pi \alpha(\vec{\pi}, \pi))$.
- The topology is generated by open sets that group together networks that start at the same point:
  - Basic opens are $U_x = \{\alpha \in \tilde{X} \ : \ \alpha(\emptyset) = x\}$.
- Each $f_\pi$ increments networks by prefixing the program $\pi$:
  - $f_\pi(\alpha)(\vec{\pi}) = \alpha(\pi, \vec{\pi})$.
- Primitive propositions get their values from the network starting point:
  - $\tilde{v}(p) = \{\alpha \in \tilde{X} \ : \ \alpha(\emptyset) \in v(p)\}$.

# Continuity

*Continuity* is a fundamental notion in topology.

# Continuity

*Continuity* is a fundamental notion in topology.

- A function $f$ is *continuous* if the preimage of every open set is open: i.e., $f^{-1}(U)$ is open whenever $U$ is open.
- Intuitively, a function is continuous if very small changes to the input produce very small changes in the output.

# Continuity

*Continuity* is a fundamental notion in topology.

- A function $f$ is *continuous* if the preimage of every open set is open: i.e., $f^{-1}(U)$ is open whenever $U$ is open.
- Intuitively, a function is continuous if very small changes to the input produce very small changes in the output.

What does continuity correspond to in the present framework?

# Continuity

Continuity of $f_\pi$ can be characterized in the language of dynamic topological logic via the scheme

$$\bigcirc_\pi \Box \varphi \to \Box \bigcirc_\pi \varphi.$$

# Continuity

Continuity of $f_\pi$ can be characterized in the language of dynamic topological logic via the scheme

$$\bigcirc_\pi \Box \varphi \to \Box \bigcirc_\pi \varphi.$$

"If, **after** executing $\pi$, $\varphi$ is (not only true, but also) measurably true, then it is possible to take a measurement **right now** that guarantees that $\varphi$ **will** be true after executing $\pi$."

# Continuity

Continuity of $f_\pi$ can be characterized in the language of dynamic topological logic via the scheme

$$\bigcirc_\pi \Box \varphi \rightarrow \Box \bigcirc_\pi \varphi.$$

"If, **after** executing $\pi$, $\varphi$ is (not only true, but also) measurably true, then it is possible to take a measurement **right now** that guarantees that $\varphi$ **will** be true after executing $\pi$."

As an axiom scheme, this says: whatever one could learn about the state of the system after the program execution one could also learn in advance of the program execution.

# Continuity

Continuity of $f_\pi$ can be characterized in the language of dynamic topological logic via the scheme

$$\bigcirc_\pi \Box\varphi \to \Box\bigcirc_\pi \varphi.$$

"If, **after** executing $\pi$, $\varphi$ is (not only true, but also) measurably true, then it is possible to take a measurement **right now** that guarantees that $\varphi$ **will** be true after executing $\pi$."

As an axiom scheme, this says: whatever one could learn about the state of the system after the program execution one could also learn in advance of the program execution.

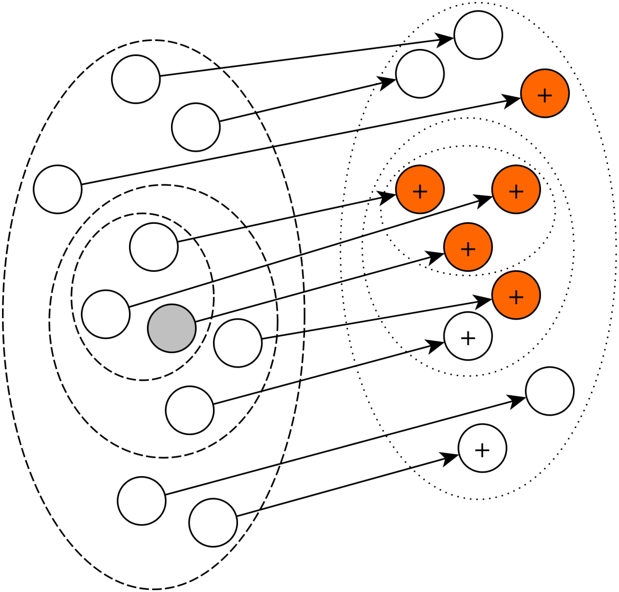
This is determinism! Continuity is determinism.

# Continuity is determinism

This refines our earlier intuitions about (non)determinism.

- It may be that no measurement at $x$ rules out *all* the other states.
  - This is the case whenever $\{x\}$ is not open.
- But this *does not* imply nondeterminism!
- It may still be possible to learn (in advance) everything *that can be known* about the effects of executing $\pi$.
  - This is the case iff $f$ is continuous at $x$.

# Continuity is determinism



# Incorporating knowledge

This interpretation of nondeterminism is epistemic in a certain sense, but involves only “knowability”, not knowledge itself.

What if we want to reason about knowledge as well?

# Topological subset models

*Topological subset models* are well-suited to the simultaneous representation of both knowledge and knowability.

- They underlie previous work on public announcements in topological spaces.
  - The precondition for an announcement of $\varphi$ is taken to be the *knowability* of $\varphi$ (i.e., $\Box\varphi$).
- They have been used to study the interplay between knowledge, knowability, and belief.
  - Stalnaker's principle of "strong belief" (or "full belief"), $B\varphi \to BK\varphi$, is weakened to $B\varphi \to B\Box\varphi$.

# Topological subset models

A **topological subset model** is a topological space $(X, \mathcal{T})$ together with a valuation $v : \text{PROP} \to 2^X$ where truth is defined with respect to *pairs* of the form $(x, U)$ where $x \in U \in \mathcal{T}$.

- $x$ represents the actual world, and $U$ represents the agent's current evidence.

# Topological subset models

A **topological subset model** is a topological space $(X, \mathcal{T})$ together with a valuation $v : \text{PROP} \to 2^X$ where truth is defined with respect to *pairs* of the form $(x, U)$ where $x \in U \in \mathcal{T}$.

- $x$ represents the actual world, and $U$ represents the agent's current evidence.

$$\begin{array}{lcl}
(x, U) \models p & \Leftrightarrow & x \in v(p) \\
(x, U) \models \neg\varphi & \Leftrightarrow & (x, U) \not\models \varphi \\
(x, U) \models \varphi \wedge \psi & \Leftrightarrow & (x, U) \models \varphi \text{ and } (x, U) \models \psi \\
(x, U) \models K\varphi & \Leftrightarrow & U \subseteq [\![\varphi]\!]^U \\
(x, U) \models \Box\varphi & \Leftrightarrow & x \in int([\![\varphi]\!]^U),
\end{array}$$

where $[\![\varphi]\!]^U = \{x \in U \ : \ (x, U) \models \varphi\}$.

# Dynamic topological subset models

We can define **dynamic topological subset models** simply by incorporating functions $f_\pi$ as before.

# Dynamic topological subset models

We can define **dynamic topological subset models** simply by incorporating functions $f_\pi$ as before.

But we need subset-style semantics for the dynamic modalities:

$$(x, U) \models \bigcirc_\pi \varphi \quad \Leftrightarrow \quad (f_\pi(x), ??) \models \varphi.$$

# Dynamic topological subset models

We can define **dynamic topological subset models** simply by incorporating functions $f_\pi$ as before.

But we need subset-style semantics for the dynamic modalities:

$$(x, U) \models \bigcirc_\pi \varphi \quad \Leftrightarrow \quad (f_\pi(x), ??) \models \varphi.$$

Perhaps the most natural definition sets the updated information state to be $f_\pi(U)$.

- This only works if $f_\pi(U)$ is open.

$$(x, U) \models \bigcirc_\pi \varphi \quad \Leftrightarrow \quad (f_\pi(x), f_\pi(U)) \models \varphi.$$

# Dynamic topological subset models

This logic is axiomatized by combining the S5 system for $K$ and the S4 system for $\Box$ with the following:

$$
\begin{array}{rl}
\text{KI} & K\varphi \to \Box\varphi \\
\neg\text{-C}_\pi & \bigcirc_\pi \neg\varphi \leftrightarrow \neg\bigcirc_\pi \varphi \\
\wedge\text{-C}_\pi & \bigcirc_\pi(\varphi \wedge \psi) \leftrightarrow (\bigcirc_\pi \varphi \wedge \bigcirc_\pi \psi) \\
K\text{-C}_\pi & \bigcirc_\pi K\varphi \leftrightarrow K\bigcirc_\pi \varphi \\
\text{O}_\pi & \Box\bigcirc_\pi \varphi \to \bigcirc_\pi \Box\varphi \\
\text{Nec}_\pi & \text{from } \varphi \text{ deduce } \bigcirc_\pi \varphi.
\end{array}
$$

# Dynamic topological subset models

When the functions $f_\pi$ are allowed to be partial, the extra axioms have to be adjusted:

KI $K\varphi \to \Box\varphi$

$\neg$-PC$_\pi$ $\bigcirc_\pi \neg\varphi \leftrightarrow (\neg\bigcirc_\pi\varphi \wedge \bigcirc_\pi\top)$

$\wedge$-C$_\pi$ $\bigcirc_\pi(\varphi \wedge \psi) \leftrightarrow (\bigcirc_\pi\varphi \wedge \bigcirc_\pi\psi)$

$K$-PC$_\pi$ $\bigcirc_\pi\top \to (\bigcirc_\pi K\varphi \leftrightarrow K(\bigcirc_\pi\top \to \bigcirc_\pi\varphi))$

O$_\pi$ $(\Box\neg\bigcirc_\pi\varphi \wedge \bigcirc_\pi\top) \to \bigcirc_\pi\Box\neg\varphi$

Mon$_\pi$ from $\varphi \to \psi$ deduce $\bigcirc_\pi\varphi \to \bigcirc_\pi\psi$.

# Learning?

In a sense, this setting cannot capture *learning*.

- True, an agent's state of knowledge changes in accordance with program execution...
- ...but every "live" possibility $y \in U$ is preserved as the corresponding state $f_\pi(y)$ in the updated information set $f_\pi(U)$.
- Possibilities are never eliminated, only updated.

# Learning?

In a sense, this setting cannot capture *learning*.

- True, an agent's state of knowledge changes in accordance with program execution...
- ...but every "live" possibility $y \in U$ is preserved as the corresponding state $f_\pi(y)$ in the updated information set $f_\pi(U)$.
- Possibilities are never eliminated, only updated.

Perhaps we should incorporate some tools from *dynamic epistemic logic* (e.g., announcements) to represent acts of information update that truly eliminate states?

# Learning?

In a sense, this setting cannot capture *learning*.

- True, an agent's state of knowledge changes in accordance with program execution...
- ...but every "live" possibility $y \in U$ is preserved as the corresponding state $f_\pi(y)$ in the updated information set $f_\pi(U)$.
- Possibilities are never eliminated, only updated.

Perhaps we should incorporate some tools from *dynamic epistemic logic* (e.g., announcements) to represent acts of information update that truly eliminate states?

In fact, PDL already provides some tools we can adapt for this purpose.

# Test programs

The standard language of PDL is sometimes extended to include "test programs", written $\varphi?$, where $\varphi$ is a formula in the language.

The corresponding relation is defined by

$$xR_{\varphi?}y \text{ iff } x = y \text{ and } x \in [\![\varphi]\!].$$

So the test $\varphi?$ does nothing if $\varphi$ is true and crashes otherwise.

# Test programs

The standard language of PDL is sometimes extended to include "test programs", written $\varphi?$, where $\varphi$ is a formula in the language.

The corresponding relation is defined by

$$xR_{\varphi?}y \text{ iff } x = y \text{ and } x \in [\![\varphi]\!].$$

So the test $\varphi?$ does nothing if $\varphi$ is true and crashes otherwise.

This process is already deterministic; for primitive propositions $p$, we can import $R_{p?}$ as a *partial* function:

$$f_{p?}(x) = \begin{cases} x & \text{if } x \in [\![p]\!] \\ \text{undefined} & \text{otherwise.} \end{cases}$$

# Test programs

Problem: the function $f_{p?}$ is open iff $[\![p]\!]$ is open.

- What if $p$ is true but not *measurably true* at some state?

# Test programs

Problem: the function $f_{p?}$ is open iff $[\![p]\!]$ is open.

- What if $p$ is true but not *measurably true* at some state?
- Intuition: at such states the program should crash since it fails to determine that $p$ is true.

# Test programs

Problem: the function $f_{p?}$ is open iff $[\![p]\!]$ is open.

- What if $p$ is true but not *measurably true* at some state?
- Intuition: at such states the program should crash since it fails to determine that $p$ is true.

Take two:

$$f_{p?}(x) = \begin{cases} x & \text{if } x \in int([\![p]\!]) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

## Test programs

Problem: the function $f_{p?}$ is open iff $[\![p]\!]$ is open.

- What if $p$ is true but not *measurably true* at some state?
- Intuition: at such states the program should crash since it fails to determine that $p$ is true.

Take two:

$$f_{p?}(x) = \begin{cases} x & \text{if } x \in int([\![p]\!]) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

So we have:

$$\begin{aligned} (x, U) \models \bigcirc_{p?}\varphi &\Leftrightarrow (f_{p?}(x), f_{p?}(U)) \models \varphi \\ &\Leftrightarrow (x, U \cap int([\![p]\!])) \models \varphi \text{ (and this is defined).} \end{aligned}$$

# Test programs

Problem: the function $f_{p?}$ is open iff $[\![p]\!]$ is open.

- What if $p$ is true but not *measurably true* at some state?
- Intuition: at such states the program should crash since it fails to determine that $p$ is true.

Take two:

$$f_{p?}(x) = \begin{cases} x & \text{if } x \in int([\![p]\!]) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

So we have:

$$\begin{aligned} (x, U) \models \bigcirc_{p?}\varphi \quad &\Leftrightarrow \quad (f_{p?}(x), f_{p?}(U)) \models \varphi \\ &\Leftrightarrow \quad (x, U \cap int([\![p]\!])) \models \varphi \text{ (and this is defined).} \end{aligned}$$

This coincides exactly with the topological definition of a public announcement of $p$.

# Ongoing work

There are many further questions...

# Ongoing work

There are many further questions...

- Can we make sense of public announcements of epistemic formulas in this setting?

# Ongoing work

There are many further questions...

- Can we make sense of public announcements of epistemic formulas in this setting?
- Can we import nondeterministic operations on programs, like union or Kleene star?

# Ongoing work

There are many further questions...

- Can we make sense of public announcements of epistemic formulas in this setting?
- Can we import nondeterministic operations on programs, like union or Kleene star?
  - Or is it better to focus on deterministic analogues? E.g.:
    - "If $\varphi$, do $\pi_1$, else do $\pi_2$."
    - "Do $\pi$ until $\varphi$."

# Ongoing work

There are many further questions...

- Can we make sense of public announcements of epistemic formulas in this setting?
- Can we import nondeterministic operations on programs, like union or Kleene star?
  - Or is it better to focus on deterministic analogues? E.g.:
    - "If $\varphi$, do $\pi_1$, else do $\pi_2$."
    - "Do $\pi$ until $\varphi$."
- Can we augment this framework to provide a formal epistemic theory of *probabilisitic nondeterminism*, i.e., chance?

# Ongoing work

There are many further questions...

- Can we make sense of public announcements of epistemic formulas in this setting?
- Can we import nondeterministic operations on programs, like union or Kleene star?
  - Or is it better to focus on deterministic analogues? E.g.:
    - "If $\varphi$, do $\pi_1$, else do $\pi_2$."
    - "Do $\pi$ until $\varphi$."
- Can we augment this framework to provide a formal epistemic theory of *probabilisitic nondeterminism*, i.e., chance?
  - Represent chance as neither an external feature of the world, nor an internal, subjective feature of an agent...

# Ongoing work

There are many further questions...

- Can we make sense of public announcements of epistemic formulas in this setting?
- Can we import nondeterministic operations on programs, like union or Kleene star?
  - Or is it better to focus on deterministic analogues? E.g.:
    - "If $\varphi$, do $\pi_1$, else do $\pi_2$."
    - "Do $\pi$ until $\varphi$."
- Can we augment this framework to provide a formal epistemic theory of *probabilisitic nondeterminism*, i.e., chance?
  - Represent chance as neither an external feature of the world, nor an internal, subjective feature of an agent...
  - ...but rather as a relationship between the world and the agent's ability to gather information.

Thank you!

# Operations on programs

PDL is often enhanced by equipping $\Pi$ with certain operations:

- sequencing: $\pi_1 ; \pi_2$ executes $\pi_1$ followed immediately by $\pi_2$;
- nondeterministic union: $\pi_1 \cup \pi_2$ nondeterministically executes either $\pi_1$ or $\pi_2$;
- iteration: $\pi^*$ repeatedly executes $\pi$ some nondeterministic finite number of times.

# Operations on programs

PDL is often enhanced by equipping $\Pi$ with certain operations:

- sequencing: $\pi_1; \pi_2$ executes $\pi_1$ followed immediately by $\pi_2$;
- nondeterministic union: $\pi_1 \cup \pi_2$ nondeterministically executes either $\pi_1$ or $\pi_2$;
- iteration: $\pi^*$ repeatedly executes $\pi$ some nondeterministic finite number of times.

The latter two may be difficult to interpret in a setting where program execution is fundamentally deterministic.

However, sequencing appears straightforward. Define

$$f_{\pi_1;\pi_2} = f_{\pi_2} \circ f_{\pi_1}.$$

# Topological dynamics with sequencing

(Serial) PDL with sequencing is axiomatized by

$$\mathsf{SPDL}_0 + \langle \pi_1 ; \pi_2 \rangle \varphi \leftrightarrow \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi.$$

# Topological dynamics with sequencing

(Serial) PDL with sequencing is axiomatized by

$$\mathsf{SPDL}_0 + \langle \pi_1 ; \pi_2 \rangle \varphi \leftrightarrow \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi.$$

This scheme is *not* valid in arbitary dynamic topological models:

$$[\![\langle \pi_1 ; \pi_2 \rangle \varphi]\!] = cl(f^{-1}_{\pi_1 ; \pi_2}([\![\varphi]\!])) = cl(f^{-1}_{\pi_1}(f^{-1}_{\pi_2}([\![\varphi]\!]))),$$

whereas

$$[\![\langle \pi_1 \rangle \langle \pi_2 \rangle \varphi]\!] = cl(f^{-1}_{\pi_1}(cl(f^{-1}_{\pi_2}([\![\varphi]\!])))).$$

# Topological dynamics with sequencing

(Serial) PDL with sequencing is axiomatized by

$$\mathsf{SPDL}_0 + \langle \pi_1 ; \pi_2 \rangle \varphi \leftrightarrow \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi.$$

This scheme is *not* valid in arbitary dynamic topological models:

$$[\![ \langle \pi_1 ; \pi_2 \rangle \varphi ]\!] = cl(f^{-1}_{\pi_1;\pi_2}([\![ \varphi ]\!])) = cl(f^{-1}_{\pi_1}(f^{-1}_{\pi_2}([\![ \varphi ]\!]))),$$

whereas

$$[\![ \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi ]\!] = cl(f^{-1}_{\pi_1}(cl(f^{-1}_{\pi_2}([\![ \varphi ]\!])))).$$

The extra closure operator means we have

$$[\![ \langle \pi_1 ; \pi_2 \rangle \varphi ]\!] \subseteq [\![ \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi ]\!]$$

but not, in general, equality.

# Topological dynamics with sequencing

A function $f$ is called *open* if it maps open sets to open sets: i.e., if $f(U)$ is open whenever $U$ is open.

It is not hard to see that if $f_{\pi_1}$ is open, then

$$[\![\langle \pi_1 ; \pi_2 \rangle \varphi]\!] = [\![\langle \pi_1 \rangle \langle \pi_2 \rangle \varphi]\!].$$

So the sequencing axiom is valid on the class of dynamic topological models where each $f_\pi$ is open.

# Topological dynamics with sequencing

A function $f$ is called *open* if it maps open sets to open sets: i.e., if $f(U)$ is open whenever $U$ is open.

It is not hard to see that if $f_{\pi_1}$ is open, then

$$[\![\langle \pi_1; \pi_2 \rangle \varphi]\!] = [\![\langle \pi_1 \rangle \langle \pi_2 \rangle \varphi]\!].$$

So the sequencing axiom is valid on the class of dynamic topological models where each $f_\pi$ is open.

- Roughly speaking, $f_\pi$ being open has a "perfect recall" type flavour: what is knowable in advance of executing $\pi$ is also knowable after executing $\pi$.