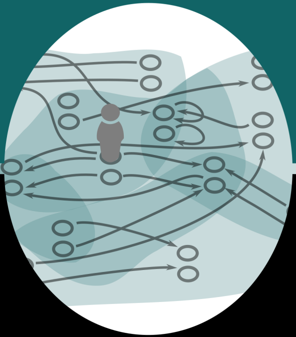
Semantics of Nondeterministic Construction



Master's Thesis Defense

Jacob Neumann

14 August 2020

What is this about?

- **Philosophy**
  - Give mathematical models of a dynamic-epistemic situation
  - Formalize epistemic intuitions about coin flipping
- **Mathematics**
  - Specify framework for studying extensions to dynamic topological logic
  - Replicate aspects of relational PDL
  - Study various interesting mathematical objects

Outline

- **Section 0: The Epistemic Interpretation**
  How dynamic topological logic models agents in situations
- **Section 1: The Logic of Coin Flipping**
  How to extend these models to interpret coin flipping
- **Section 2: The Logic of Program Construction**
  How to study (and generalize) the coin-flipping augmentation

Section 0

The Epistemic Interpretation

What we're studying:

Agents who can

- Perform actions (or execute programs)
- Know (some) true statements about their situation (including statements about the results of their actions).

We wish to understand the *logic* of such agents

Dynamic Topological Models

A Σ-DTM $\mathfrak{M}$ is a kind of mathematical object.It includes:

- A set, denoted $|\mathfrak{M}|$, of *states*
- The set Σ, its *program set*

DTMs serve as mathematical models of an agent's possibilities in a situation.

Evaluating $\mathcal{L}_{\Box\bigcirc}(\Sigma)$

$\mathcal{L}_{\Box\bigcirc}(\Sigma)$ is given by:

$$\varphi, \psi ::= p \mid \varphi \wedge \psi \mid \neg\varphi \mid \bigcirc_\sigma \varphi \mid \Box\varphi \qquad (p \in \Phi,\ \sigma \in \Sigma)$$

The judgment

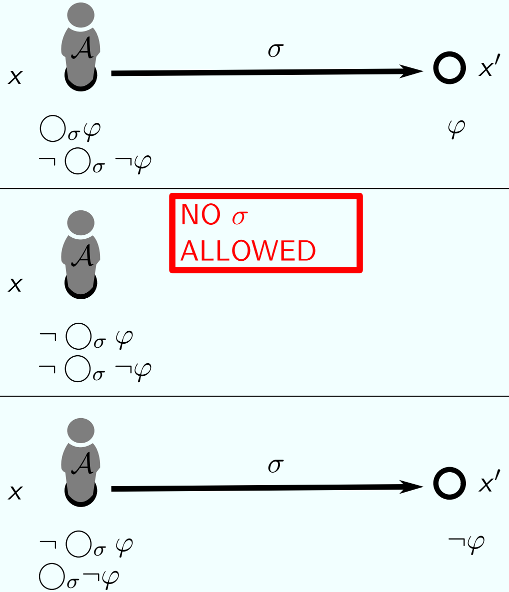
$$(\mathfrak{M}, x) \models \varphi$$

is pronounced "$(\mathfrak{M}, x)$ validates φ" or "x is a φ-world". The conditions for determining whether or not $(\mathfrak{M}, x) \models \varphi$ are defined by recursion on the structure of φ.

Epistemic Interpretation: Action

We think of our agent as being "at" a world x of a Σ-DTM $\mathfrak{M}$. The formulas validated by $(\mathfrak{M}, x)$ express the properties of her interaction with that situation.

- $\bigcirc_\sigma \varphi$ – **"after σ, φ"**
  In the present state of the situation, the agent is able to perform the action denoted by σ, and doing so will result in a state where φ is true.


x
σ
x'
$\bigcirc_\sigma \varphi$
$\neg \bigcirc_\sigma \neg\varphi$
φ
NO σ ALLOWED
x
$\neg \bigcirc_\sigma \varphi$
$\neg \bigcirc_\sigma \neg\varphi$
x
σ
x'
$\neg \bigcirc_\sigma \varphi$
$\bigcirc_\sigma \neg\varphi$
$\neg\varphi$

Epistemic Interpretation: Knowledge

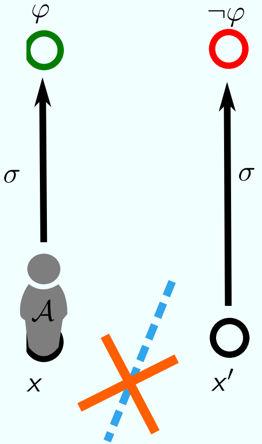
We think of our agent as being "at" a world x of a Σ-DTM $\mathfrak{M}$. The formulas validated by $(\mathfrak{M}, x)$ express features of her experience of that situation.

- $\Box\varphi$ – **"φ is knowably true"**
  In the present state of the situation, φ is true and, moreover, there is some observation the agent could make which would allow her to *know* that φ
- $\Diamond\varphi$ – **"the agent cannot rule out the possibility of φ"**
  In the present state of the situation, there is no observation the agent could make which would allow her to know that φ is not the case

In the text of the thesis, I explain in (much) greater detail why this reading is warranted.

	ADTL Axioms and Inferences
(CPL)	enough tautologies of CPL
(MP)	from $\varphi \to \psi$ and φ, infer ψ
(K)	$\Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi$
(T)	$\Box\varphi \to \varphi$
(4)	$\Box\varphi \to \Box\Box\varphi$
(Nec)	from φ, infer $\Box\varphi$
($\neg$-PC)	$\bigcirc_\sigma \neg\varphi \leftrightarrow (\neg \bigcirc_\sigma \varphi \wedge \bigcirc_\sigma \top)$
($\wedge$-C)	$\bigcirc_\sigma(\varphi \wedge \psi) \leftrightarrow \bigcirc_\sigma\varphi \wedge \bigcirc_\sigma\psi$
(Mon)	from $\varphi \to \psi$, infer $\bigcirc_\sigma\varphi \to \bigcirc_\sigma\psi$

Epistemic Opacity



$\Diamond \bigcirc_\sigma \varphi \wedge \Diamond \bigcirc_\sigma \neg\varphi$

- It is the case that "after σ, φ", but, as far as the agent can know, it could be the case that "after σ, $\neg\varphi$".

Section 1

The Logic of Coin Flipping

Principles of Coin Flipping

We study agents who use a coin flip to decide between two possible actions

σ_0 or σ_1 : "**flip a coin. If heads, do σ_0. If tails, do σ_1**"

Obvious facts about how coin flipping should work:

- σ_0 or σ_1 should ultimately be either σ_0 or σ_1 (Honesty)
- The agent should not be able to determine which one it will be, prior to flipping the coin (Epistemic Opacity)

OR-Augmentation

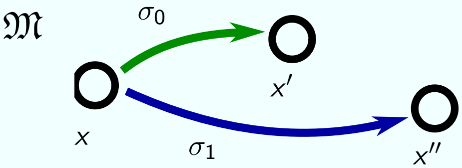
We want to "augment" DTMs to interpret this.

$$\Pi^{\mathsf{or}} = \Pi \cup \{\pi_0 \;\mathsf{or}\; \pi_1 \;:\; \pi_0, \pi_1 \in \Pi\}.$$

We specify a *transformation* sending a Π-DTM $\mathfrak{M}$ to a Π^{or}-DTM $\mathfrak{M}^{\mathsf{OR}}$.

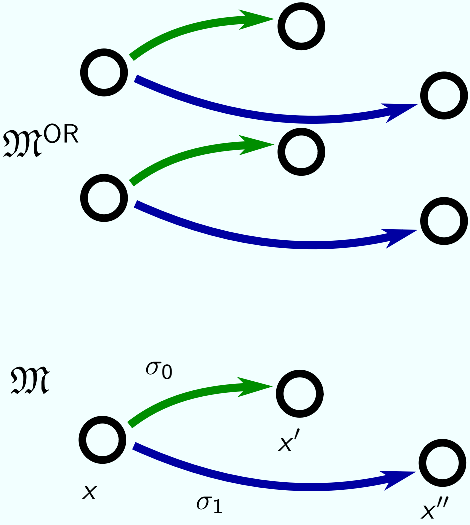
- $\mathfrak{M}$ and $\mathfrak{M}^{\mathsf{OR}}$ should interpret Π programs the same way
- or-programs should match our intuitions about coin flipping

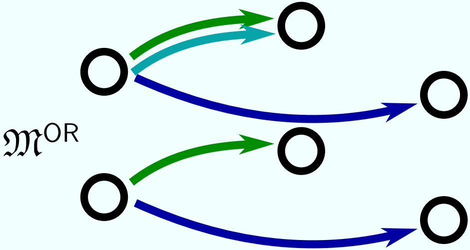
Note: this does not allow for "nesting", e.g. $\pi_0 \;\mathsf{or}\; (\pi_1 \;\mathsf{or}\; \pi_2)$

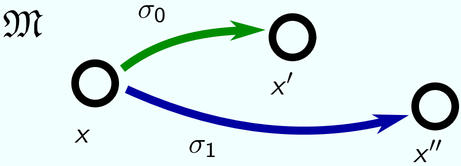


$\mathfrak{M}$
σ_0
x'
x
σ_1
x''

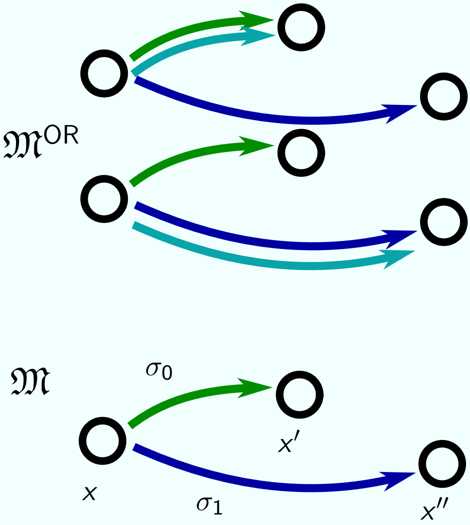
.


$\mathfrak{M}^{OR}$
$\mathfrak{M}$
σ_0
x'
x
σ_1
x''

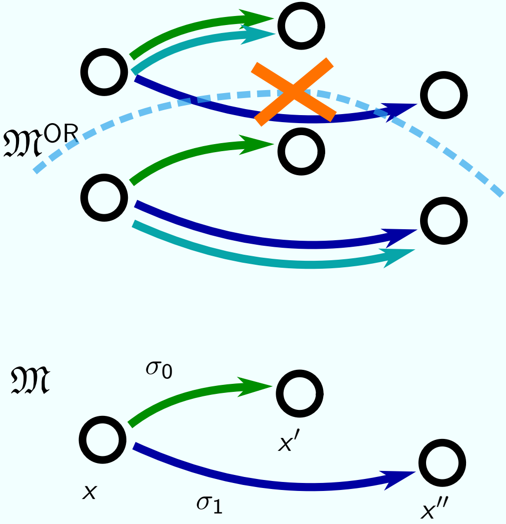




σ_0 or σ_1



σ_0 or σ_1



σ_0 or σ_1

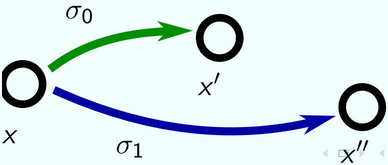
Results

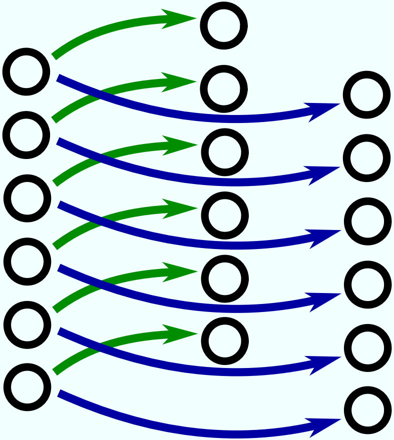
Essential features of OR-augmented models:

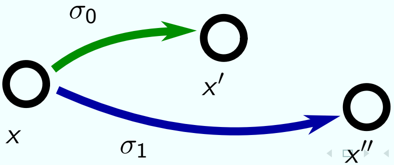
- Primitive equivalence: Π-actions (those not requiring the coin) behave the same, regardless of whether the agent has a coin or not
- Honesty: σ_0 or σ_1 is either σ_0 or σ_1.
- Epistemic Opacity: The agent cannot rule out the possibility that the coin will come up heads, or that it will come up tails.
- Regularity: The outcome of the flip does not depend on *what the flip is being used to decide between*.
- Persistence: The agent may only flip the coin once – repeated flips will give the same result.

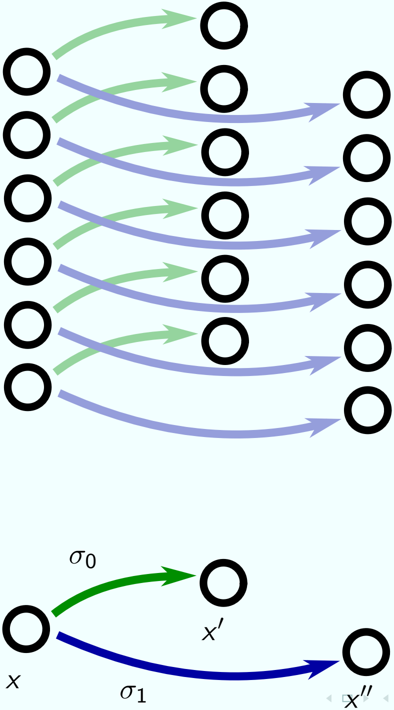
Section 2

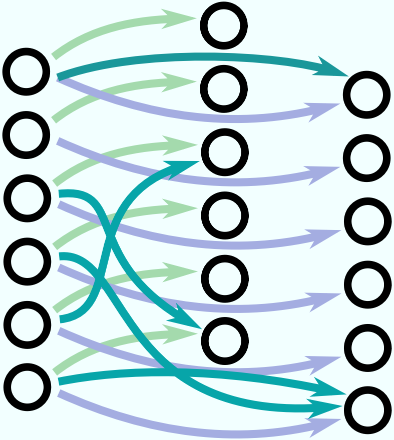
The Logic of Program Construction

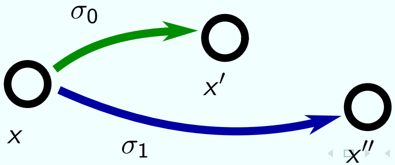

σ_0
x'
x
σ_1
x''




σ_0
x'
x
σ_1
x''


σ_0
x'
x
σ_1
x''




σ_0
x'
x
σ_1
x''

Π^c

If c is some n-ary function symbol, Π^c is the least set containing Π closed under c: i.e. those σ given by

$$\sigma ::= \pi \mid c(\sigma_1, \ldots, \sigma_n)$$

where π ranges over Π.

Unless noted (as we did with or), there is no restriction on syntactic "nesting".

Program Construction

Defn. 2.1 A **program constructor** C interpreting c consists of a rule assigning to each Π-DTM $\mathfrak{M}$ a Π^c-DTM $\mathfrak{M}^C$, which is structured as pictured above.

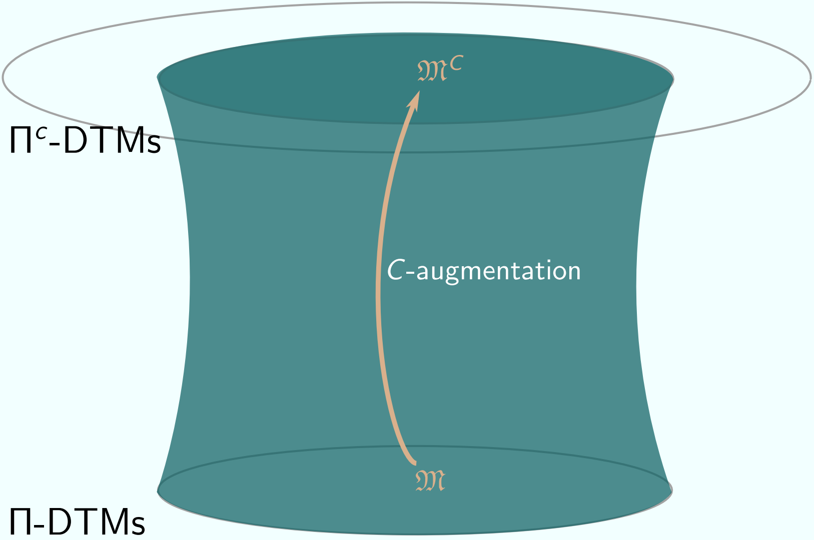
Some examples:

- OR
- Uω and U∞
- OR1, OR2, OR3, . . .
- SKIP
- SEQ
- STARω, STAR∞, . . .
- Test programs, loops, conditionals, more elaborate constructions?

Things I didn't get to talk about (today)

- More epistemological development
- Proof theory
- Propositional dynamic logic
- More details about various program constructors
- Bisimulations
- Refined Frame Theory and Characterization
- Details/proofs of the major results:
  - Prop. 1.5 New proof of PDL_0 soundness & completeness
  - Thms. 3.6 & 3.7 $PDL_0 + (U)$ soundness & completeness w.r.t. $U\omega$-augmented (or $U\infty$-augmented) models
  - Thm. 4.3 Undefinability of the class of OR-augmented DTMs
  - Thm. 7.1 Characterization of OR

Thank you!



$\mathfrak{M}^C$
Π^c-DTMs
C-augmentation
$\mathfrak{M}$
Π-DTMs