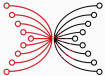
# 3rd-year PhD Report

Orestis Melkonian

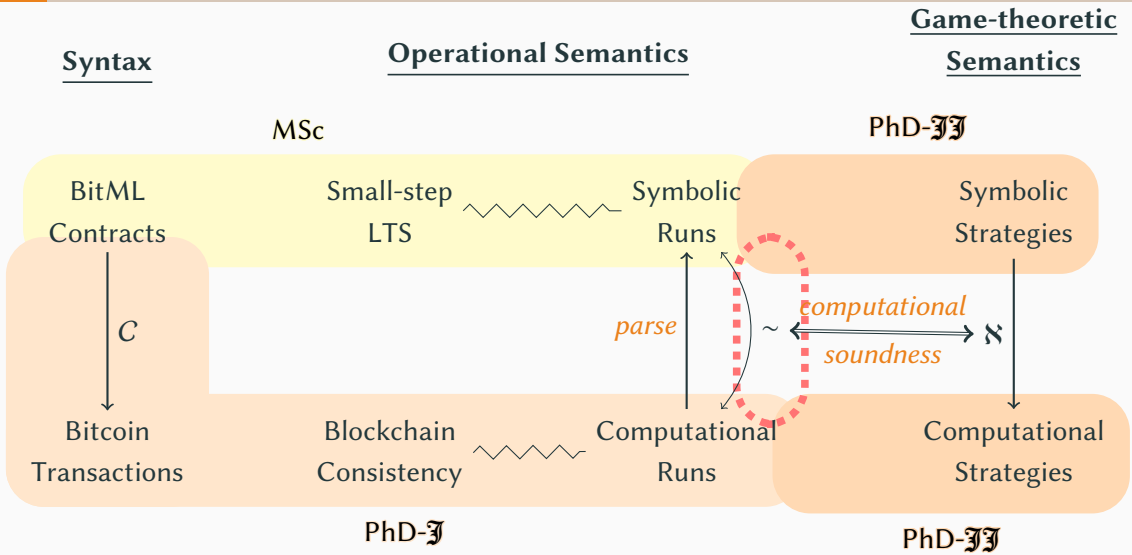
September 19, 2022

THE UNIVERSITY *of* EDINBURGH

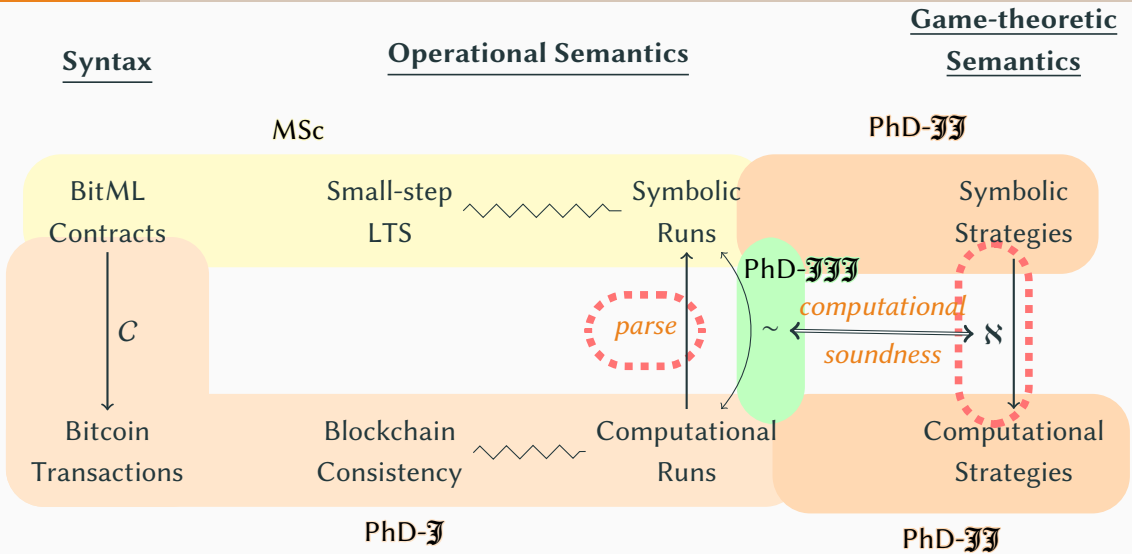


INPUT | OUTPUT

# BitML: Years I & II



# BitML: Year III



# Challenges for coherence

- Proofs that construct mappings required meta-properties on lists
- Tracing a contract's lifetimer required temporal hyper-properties
- Scaling up → hit Agda's type-checking performance limits

# The Abstraction Problem

- Constructive proofs are too involved ⟶ slow/infinite type-checking
- Now even stuck on a concrete example!

3 possible solutions:

(0). hire a super-computer... (nah)

1. fix Agda itself (+ community service)
2. tweak placement of abstract (slow edit/check loop)
3. revert to *spartan type theory* (i.e. no typeclasses, modules, etc..)

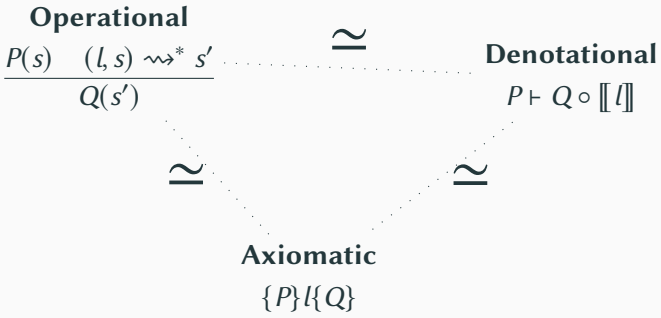
Given a computational run $R^c$ and symbolic strategies $\sigma^s$:

$$\frac{R^c \text{ conforms to } \Sigma^c \qquad \text{where } \Sigma^c := \aleph(\Sigma^s)}{\exists R^s.\ R^s \sim R^c \qquad R^s \text{ conforms to } \Sigma^s}$$

**BACKUP PLAN**

Formalize only the first half of *computational soundness* (i.e. parsing).

# Separation Logic for UTxO: Year 𝔍𝔍




**SL: [FRAME] rule**

$$\frac{l \# R \quad \{P\}\, l\, \{Q\}}{\{P * R\}\, l\, \{Q * R\}}$$

**CSL: [PARALLEL] rule**

$$\frac{l_1 \parallel l_2 = l \quad l_1 \# P_2 \quad l_2 \# P_1 \quad \{P_1\}\, l_1\, \{Q_1\} \quad \{P_2\}\, l_2\, \{Q_2\}}{\{P_1 * P_2\}\, l\, \{Q_1 * Q_2\}}$$

# Separation Logic for UTxO: Year ℑℑℑ

- Previous model does not translate easily to UTxO
- Main issue: compositionality, due to side-conditions $l \# P$
- A step back: regain compositionality by separating on values instead of participants
- No side-conditions needed anymore!

**SL: [FRAME] rule**

$$\frac{\cancel{l \# R} \quad \{P\}\, l\, \{Q\}}{\{P * R\}\, l\, \{Q * R\}}$$

**CSL: [PARALLEL] rule**

$$\frac{l_1 \parallel l_2 = l \quad \cancel{l_1 \# P_2} \quad \cancel{l_2 \# P_1} \quad \{P_1\}\, l_1\, \{Q_1\} \quad \{P_2\}\, l_2\, \{Q_2\}}{\{P_1 * P_2\}\, l\, \{Q_1 * Q_2\}}$$

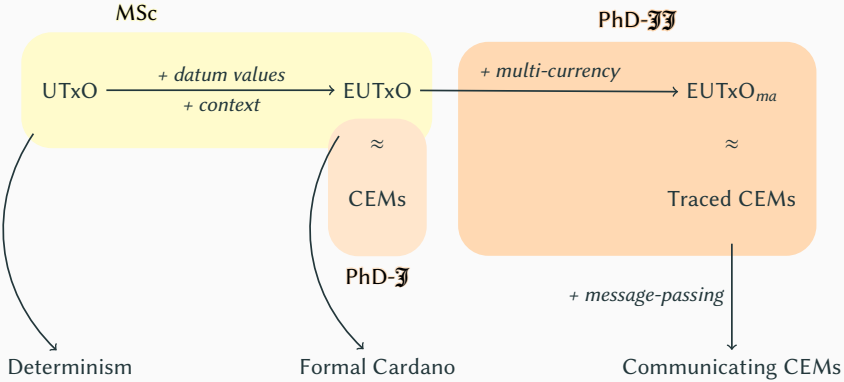
# Separation Logic for UTxO: Year IV

- Still some work to accommodate UTxO, due to names/addresses/hashes
  - Currently formulating *Abstract UTxO* (AUTxO)
  - Main idea: reference unspent outputs by *value*, so as to utilize its monoidal structure
  - Have the ledger model + semantics, but need to modify the underlying separation logic

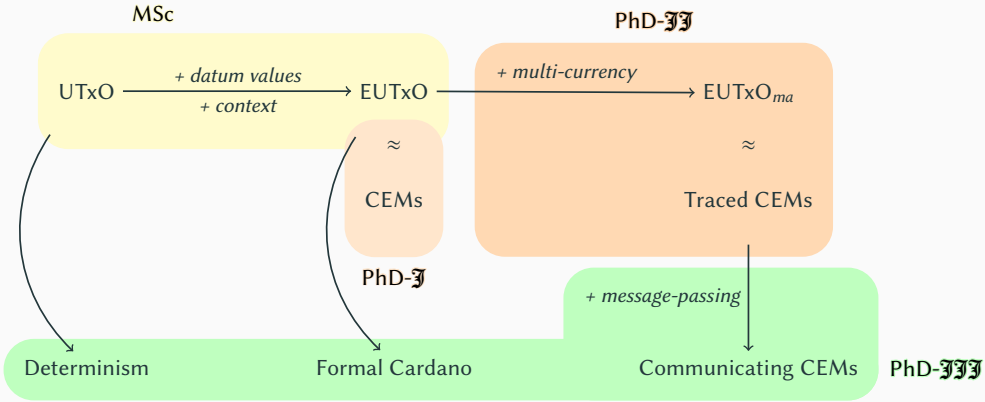
**BACKUP PLAN**

Write a functional pearl for the non-UTxO case only.

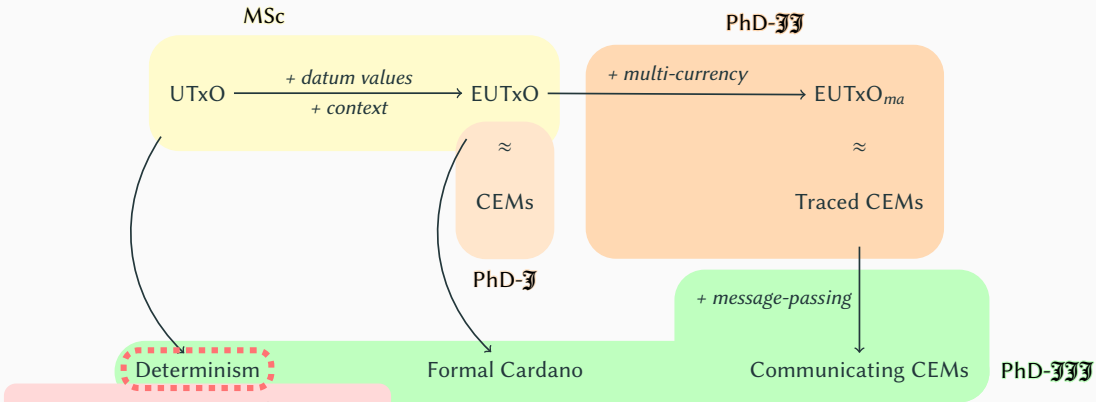
# UTxO: Year I & II



# UTxO: Year III



# UTxO: Year III

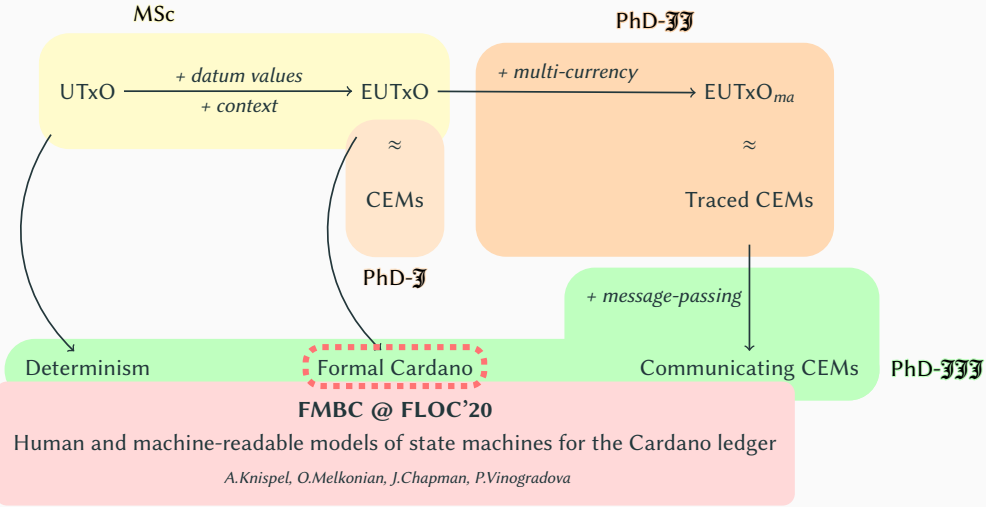



**FMBC @ FLOC'20**

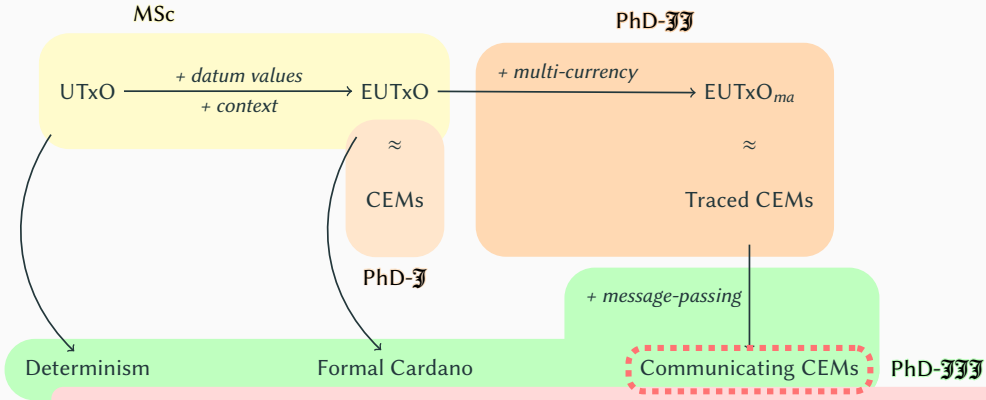
Determinism of ledger updates

*P.Vinogradova, A.Knispel, J.Chapman, O.Melkonian*

# UTxO: Year 𝔍𝔍𝔍



# UTxO: Year 𝔍𝔍𝔍




**FMBC @ FLOC'20**

Designing EUTxO smart contracts as communicating state machines: the case of simulating accounts

*P.Vinogradova, M.Chakravarty, J.Chapman, T.Ferariu, M.P.Jones, J.Krijnen*

**CPP @ POPL'22**

Reasonable Agda is Correct Haskell: Writing Verified Haskell using AGDA2HS

**ICFP'22**

Reasonable Agda is Correct Haskell: Writing Verified Haskell using AGDA2HS

**Haskell Symposium @ ICFP'22**

Reasonable Agda is Correct Haskell: Writing Verified Haskell using AGDA2HS

*J.Cockx, O.Melkonian, L.Escot, J.Chapman, U.Norell*

- In collaboration with Jamie Gabbay (Herriot-Watt), after reviewing IEUTxO
- More of an educational process, but also quite connected to all of my projects
- Current formalized features:
  - atoms
  - swapping + permutations
  - abstraction + concretion
  - support + freshness + И quantifier
  - ULC case study: syntax + α-equivalence + reduction rules

# Timeplan

- **[2022 - mid 2023]** Closure
  - ideally two more papers on BitML and separation logic at prestigious venues
  - if time runs out → fallback to backup plans
- **[mid 2023 - late 2023]** Thesis write-up
  - IMHO enough material to fill up a dissertation

# Discussion