Game & Strategies in Type Theory

Peio Borthelle, USMB, Chambéry, France
w/ Tom Hirschowitz & Guilhem Jaber & Yannick Zakowski

TYPES 2023 – Valencia

Motivations

Goal

Study higher-order programming languages with effects such as non-termination starting from their operational semantics.

Goal

Study higher-order programming languages with effects such as non-termination starting from their operational semantics.

Contextual Equivalence

"a and b are observationally indistinguishable"

$$a \simeq_{\mathrm{ctx}} b \quad := \quad \forall E,\, E[a] \simeq_{\mathrm{op}} E[b]$$

Program Equivalence

Goal

Study higher-order programming languages with effects such as non-termination starting from their operational semantics.

Contextual Equivalence

"a and b are observationally indistinguishable"

$$a \simeq_{\mathrm{ctx}} b \quad := \quad \forall E,\, E[a] \simeq_{\mathrm{op}} E[b]$$

💀🤬

We want an easier-to-check $\simeq_M$ such that

$$a \simeq_M b \quad \Longrightarrow \quad a \simeq_{\mathrm{ctx}} b$$

Operational Game Semantics

Trace Semantics

- Sequences of observations of an execution.
- "perform an effect", "call a free variable", "return a value"

An example: OGS

- In the spirit of process calculi: keep the labels first-order.
- Computations[1] are hidden as fresh variables.
- We don't observe full values but only patterns.

[1]functions, thunks … CBPV negative types

Our Contribution

- A generic account of Operational Game Semantics.
- Implemented and proved correct in Coq.

Formalization

Given an evaluator "$\mathrm{term} \to \mathcal{D}(\mathrm{nf})$":

- Construct the OGS LTS ;
- Show it correct for contextual equivalence.

Crux of the proof

$$[\![E[a]]\!] \approx_M [\![E]\!] \parallel [\![a]\!]$$

assuming the evaluator verifies:

$$\begin{tikzcd}
E[a] \arrow[r, mapsto, "*"] \arrow[d, mapsto, "*", "\text{“active”}"'] & v \\
E[w] \arrow[r, mapsto, "\text{“sync”}"'] & b \arrow[u, mapsto, "*"', "\text{“passive”}"]
\end{tikzcd}$$

Technical Choices

Intensional representations

- LTS more intensional than prefix-closed set of traces
- coalgebraic LTS compute more than relational LTS

$\Rightarrow$ guarded coinduction in Type using negative records

$\Rightarrow$ coinduction-up-to in Prop using Coq-Coinduction[2]

Rigid structures

- dependent variant of interaction trees[3]
- well-typed & well-scoped variables

$\Rightarrow$ dependent programming in Coq using Coq-Equations[4]

---

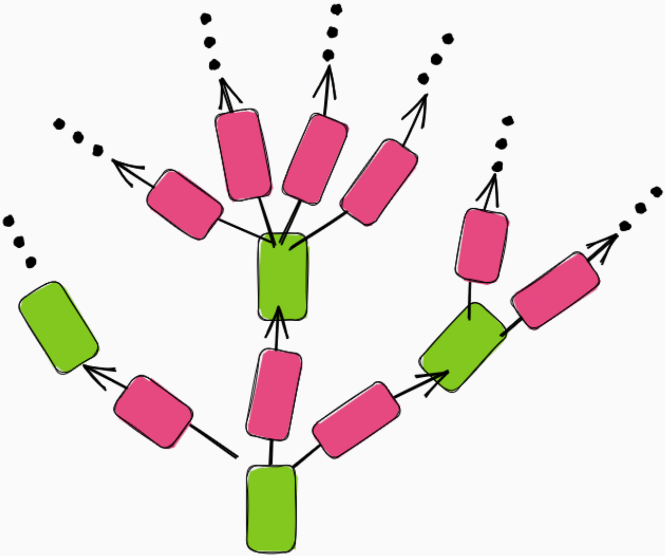
[2] by Damien Pous

[3] original by Li-Yao Xia *et al.*

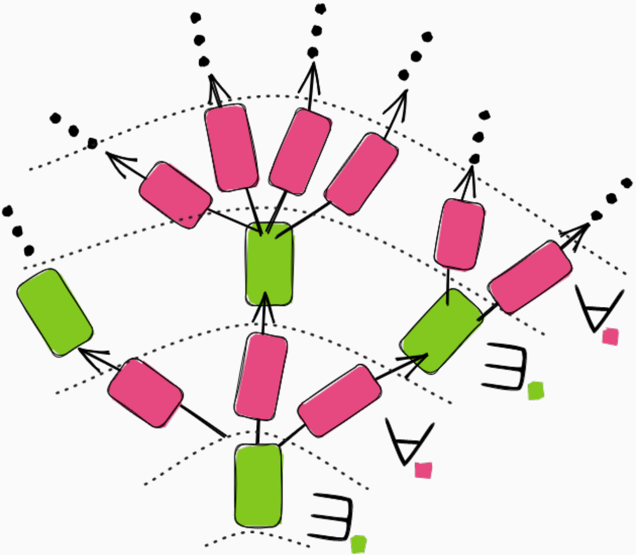
[4] by Matthieu Sozeau

What are game rules?

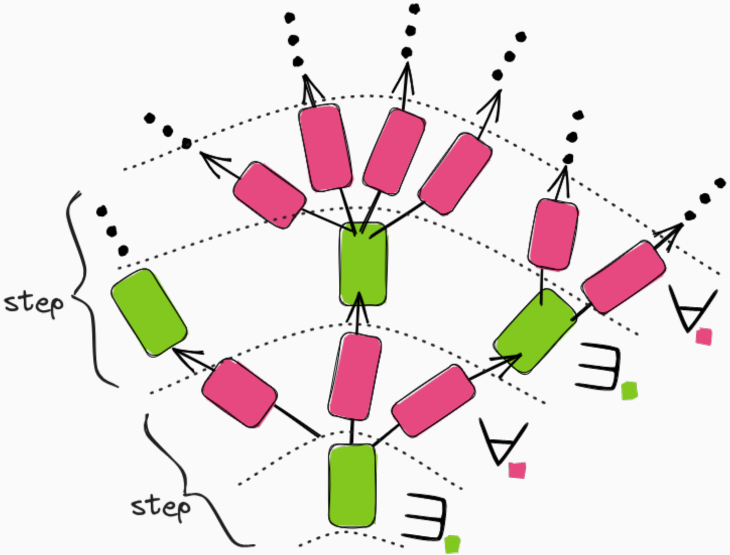
Two Sets



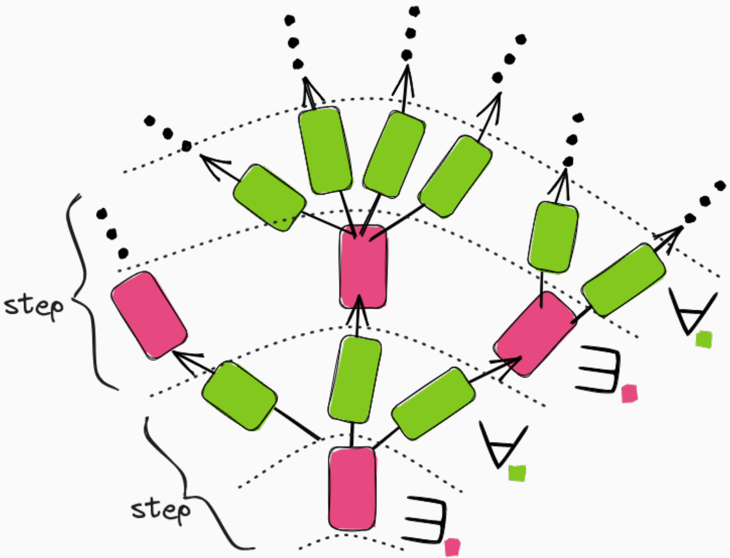
Two Sets



Two Sets



Two Sets



Simple Strategies, Formally

$$Step_P, Step_O : \mathbf{SET} \to \mathbf{SET}$$

$$Step_P\ X \coloneqq M_P \times (M_O \to X) \qquad \text{player step}$$

$$Step_O\ X \coloneqq M_O \times (M_P \to X) \qquad \text{opponent step}$$

Building Blocks

$$Act_M, Pas_M : \mathbf{SET} \to \mathbf{SET}$$

$$Act_M\ X \coloneqq M \times X \qquad \text{active half-step}$$

$$Pas_M\ X \coloneqq M \to X \qquad \text{passive half-step}$$

$$sync_M : Act_M\ X \times Pas_M\ Y \to X \times Y \qquad \text{interaction law}$$

Simple Strategies, Formally

$$Step_P, Step_O : \mathbf{SET} \to \mathbf{SET}$$

$$Step_P\ X \coloneqq M_P \times (M_O \to X) \qquad \text{player step}$$

$$Step_O\ X \coloneqq M_O \times (M_P \to X) \qquad \text{opponent step}$$

Building Blocks

$$Act_M, Pas_M : \mathbf{SET} \to \mathbf{SET}$$

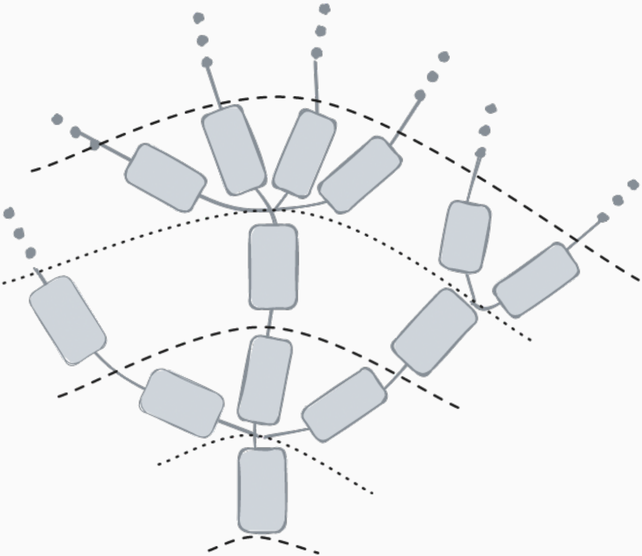
$$Act_M\ X \coloneqq M \times X \qquad \text{active half-step}$$

$$Pas_M\ X \coloneqq M \to X \qquad \text{passive half-step}$$

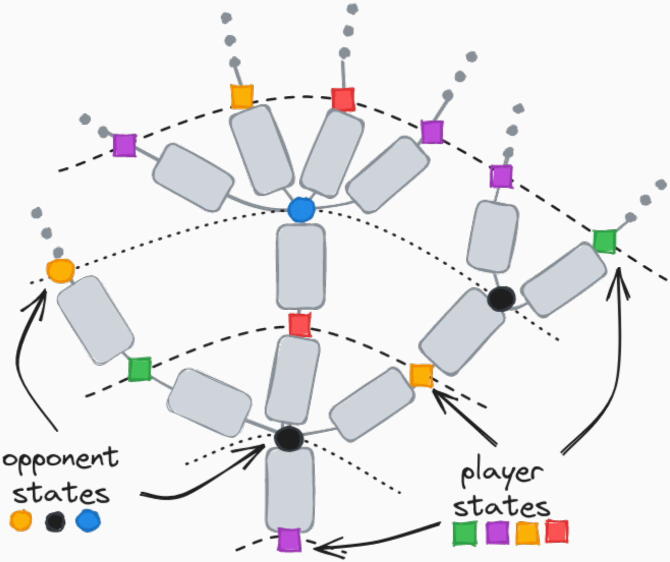
$$sync_M : Act_M\ X \times Pas_M\ Y \to X \times Y \qquad \text{interaction law}$$

Pretty nice, but not very expressive.

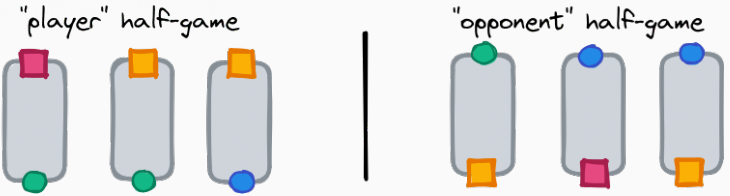
More *Precise* Strategies



More *Precise* Strategies



Two *Bi-Indexed Sets*[5]



$$HalfGame\ (I\ J : \mathbf{SET}) : \mathbf{SET} := \begin{cases} M : I \to \mathbf{SET} \\ f_i : S\ i \to J \end{cases}$$

$$Act, Pas : (J \to \mathbf{SET}) \to (I \to \mathbf{SET})$$

$$Act\ X\ i := (s : M\ i) \times X\ (f_i\ s) \qquad \text{active half-step}$$

$$Pas\ X\ i := (s : M\ i) \to X\ (f_i\ s) \qquad \text{passive half-step}$$

$$sync : \Sigma_i (Act\ X\ i \times Pas\ Y\ i) \to \Sigma_j (X\ j \times Y\ j) \qquad \text{interaction law}$$

[5] See also Paul Levy & Sam Staton: Transition systems over games

From pairs of "half-games" to indexed containers

$$Game\,(I\,J : \mathbf{SET}) : \mathbf{SET} := \begin{cases} P : HalfGame\ I\ J \\ O : HalfGame\ J\ I \end{cases}$$

- $Act_P \circ Pas_O : PolyEndo(I \to \mathbf{SET})$, player point of view
- $Act_O \circ Pas_P : PolyEndo(J \to \mathbf{SET})$, opponent point of view

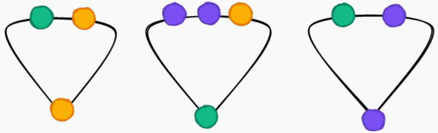
Moral: containers loose information

- In containers, implicitely $J = (i : I) \times M_P$.
- Let's cut containers in half!

Tree constructions

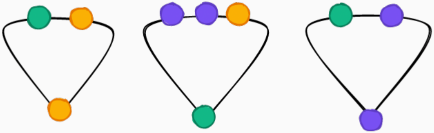
Classical container fixpoints

Given the “tiles”, how are we allowed to combine them?



Classical container fixpoints

Given the “tiles”, how are we allowed to combine them?



μ: inductive trees

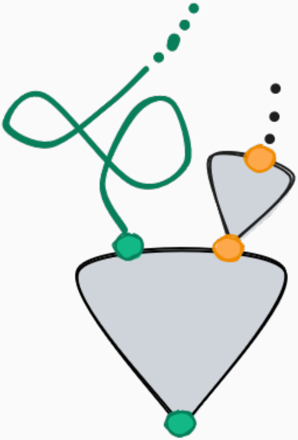
- “any combination of finite depth”
- “any algebra of the step-functor”

ν: coinductive trees

- “any combination”
- “any coalgebra of the step-functor”

Fixpoints, continued

But we also need these!



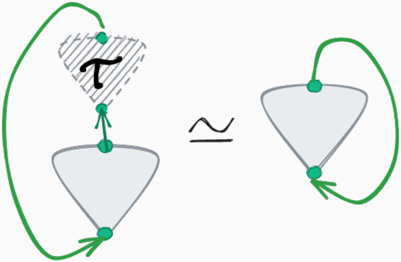
Interaction Trees! (no fancy greek letter for this fixpoint yet)

- “any combination with arbitrary loops”
- “any iterative algebra of the step-functor”
- “any coalgebra of ‘Step + Id’ modulo weak bisimilarity”

[6]See extensive work by Stefan Milius

Indexed Interaction Trees

$$ITree\,(X : I \to SET) := \nu A.(i \mapsto \underbrace{X\,i}_{\text{return}} + \overbrace{A\,i}^{\text{silent step}} + \underbrace{[\![G]\!]\;A\,i}_{\text{play a move}})$$



Weak bisimilarity skips over any finite number of τ nodes

$\Rightarrow$ Mixed inductive-coinductive,

but not a problem for Coq-Coinduction!

Conclusion

Contributions

- Formalized generic soundness theorem for operational game semantics.
- New datastructure: indexed interaction trees.

Ongoing & future work

- Clarify the categorical structure of "games" and "half-games".
- Fully formalized game semantics.

Conclusion

Contributions

- Formalized generic soundness theorem for operational game semantics.
- New datastructure: indexed interaction trees.

Ongoing & future work

- Clarify the categorical structure of “games” and “half-games”.
- Fully formalized game semantics.

Thanks for listening!

Composition of dual strategies