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.
**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”

$$a \sim_{ctx} b \;:=\; \forall E, \; E[a] \sim_{op} E[b]$$
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”

\[ a \simeq_{\text{ctx}} b := \forall E, E[a] \simeq_{\text{op}} E[b] \]

We want an easier-to-check \(\simeq_{M}\) such that

\[ a \simeq_{M} b \implies a \simeq_{\text{ctx}} b \]
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
• A generic account of Operational Game Semantics.
• Implemented and proved correct in Coq.
Our Contribution

Formalization

Given an **evaluator** “term $\rightarrow D(nf)$”:
- 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:

![Diagram](image)
Technical Choices

Intensional representations

• LTS more intensional than prefix-closed set of traces
• coalgebraic LTS compute more than relational LTS
⇒ guarded coinduction in TYPE using negative records
⇒ coinduction-up-to in PROP using COQ-COINDUCTION²

Rigid structures

• dependent variant of interaction trees³
• well-typed & well-scoped variables
⇒ dependent programming in CoQ using COQ-EQUATIONS⁴

²by Damien Pous
³original by Li-Yao Xia et al.
⁴by Matthieu Sozeau
What are game rules?
Two Sets
Two Sets
Two Sets
Two Sets
Simple Strategies, Formally

\[ \text{Step}_P, \text{Step}_O : \text{SET} \to \text{SET} \]

\[ \text{Step}_P X := M_P \times (M_O \to X) \] \hspace{1cm} \text{player step}

\[ \text{Step}_O X := M_O \times (M_P \to X) \] \hspace{1cm} \text{opponent step}

Building Blocks

\[ \text{Act}_M, \text{Pas}_M : \text{SET} \to \text{SET} \]

\[ \text{Act}_M X := M \times X \] \hspace{1cm} \text{active half-step}

\[ \text{Pas}_M X := M \to X \] \hspace{1cm} \text{passive half-step}

\[ \text{sync}_M : \text{Act}_M X \times \text{Pas}_M Y \to X \times Y \] \hspace{1cm} \text{interaction law}
Simple Strategies, Formally

\[ \text{Step}_P, \text{Step}_O : \text{SET} \rightarrow \text{SET} \]

\[ \text{Step}_P X := \mathcal{M}_P \times (\mathcal{M}_O \rightarrow X) \quad \text{player step} \]

\[ \text{Step}_O X := \mathcal{M}_O \times (\mathcal{M}_P \rightarrow X) \quad \text{opponent step} \]

Building Blocks

\[ \text{Act}_M, \text{Pas}_M : \text{SET} \rightarrow \text{SET} \]

\[ \text{Act}_M X := \mathcal{M} \times X \quad \text{active half-step} \]

\[ \text{Pas}_M X := \mathcal{M} \rightarrow X \quad \text{passive half-step} \]

\[ \text{sync}_M : \text{Act}_M X \times \text{Pas}_M Y \rightarrow X \times Y \quad \text{interaction law} \]

Pretty nice, but not very expressive.
More Precise Strategies
More Precise Strategies
Two Bi-Indexed Sets

HalfGame \( (I, J : \text{SET}) : \text{SET} := \)

\[
\begin{align*}
M : I & \rightarrow \text{SET} \\
f_i : S_i & \rightarrow J
\end{align*}
\]

Act, Pas : \((J \rightarrow \text{SET}) \rightarrow (I \rightarrow \text{SET})\)

Act \(X_i := (s : M_i) \times X (f_i s)\)  

active half-step

Pas \(X_i := (s : M_i) \rightarrow X (f_i s)\)  

passive half-step

sync : \(\Sigma_i(\text{Act } X_i \times \text{Pas } Y_i) \rightarrow \Sigma_j(X_j \times Y_j)\)  

interaction law

---

5See also Paul Levy & Sam Staton: Transition systems over games
From pairs of “half-games” to indexed containers

\[
\text{Game } (I J : \text{SET}) : \text{SET} := \begin{cases} 
P : \text{HalfGame } I J \\
O : \text{HalfGame } J I 
\end{cases}
\]

- \(\text{Act}_P \circ \text{Pas}_O : \text{PolyEndo}(I \rightarrow \text{SET})\), player point of view
- \(\text{Act}_O \circ \text{Pas}_P : \text{PolyEndo}(J \rightarrow \text{SET})\), opponent point of view

**Moral:** containers loose information

- In containers, implicitly \(J = (i : I) \times M_P\).
- Let’s cut containers in half!
Tree constructions
Given the “tiles”, how are we allowed to combine them?
Given the “tiles”, how are we allowed to combine them?

\( \mu \): inductive trees
- “any combination of finite depth”
- “any algebra of the step-functor”

\( \nu \): coinductive trees
- “any combination”
- “any coalgebra of the step-functor”
Fixpoints, continued

But we also need these!
Completely Iterative Monads

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”

\[\text{See extensive work by Stefan Milius}\]
Indexed Interaction Trees

\[ ITree (X : I \to SET) := \nu A. (i \mapsto X_i + A_i + \mathbb{[[G]]} A_i) \]

Weak bisimilarity skips over any finite number of \( \tau \) nodes
\[ \Rightarrow \text{Mixed inductive-coinductive,} \]
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.

Thanks for listening!
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

composition + hiding