Positional Injectivity for Innocent Strategies

Lison Blondeau-Patissier & Pierre Clairambault
I2M & LIS, AMU, Marseille

Journées du GT Scalp 2021
Introduction

Programs

Game Semantics

Dynamic: Represent programs with temporal information. Strategies

Relational Semantics

Static: Represent states. Positions

Theorem: Positional Injectivity (for Hyland Ong games)
Pos is injective for total finite innocent strategies.
Introduction

Game Semantics

Dynamic: Represent programs with temporal information.
Strategies

Relational Semantics

Static: Represent states.
Positions

Theorem: Positional Injectivity (for Hyland Ong games)

Pos is injective for total finite innocent strategies.
Outline

1 Introduction to Game Semantics
   - Arenas
   - Plays
   - Innocent Strategies

2 Positional Injectivity

3 Proof Method
Introduction to Game Semantics (Hyland-Ong games)

Opponent (Context) \rightarrow Player (Process)

Arenas (Types): The game with its rules
Plays (Executions): A game between two players
Strategies (Programs): Guideline for Player
Introduction to Game Semantics (Hyland-Ong games)

Arenas \((Types)\): The game with its rules
Plays \((Executions)\): A game between two players
Strategies \((Programs)\): Guideline for Player
An arena $A$ is a tuple $\langle |A|, \leq_A, \text{pol}_A \rangle$ such that:

- $|A|$ is a set of events;
- $\text{pol}_A : |A| \rightarrow \{-, +\}$ is a labelling function;
- $\leq_A$ defines a negative and alternating tree.

\[
\begin{array}{c}
\text{q}^- \\
\text{tt}^+ \quad \text{ff}^+
\end{array}
\]

Arena bool
An **arena** $A$ is a tuple $\langle |A|, \leq_A, \text{pol}_A \rangle$ such that:

- $|A|$ is a set of **events**;
- $\text{pol}_A : |A| \rightarrow \{-, +\}$ is a labelling function;
- $\leq_A$ defines a **negative** and **alternating** tree.

```
Arena bool 
Arena bool_1 \Rightarrow bool_2
```
An **arena** $A$ is a tuple $\langle |A|, \leq_A, \text{pol}_A \rangle$ such that:

- $|A|$ is a set of *events*;
- $\text{pol}_A : |A| \to \{-, +\}$ is a labelling function;
- $\leq_A$ defines a *negative* and *alternating* tree.
An arena $A$ is a tuple $\langle |A|, \leq_A, \text{pol}_A \rangle$ such that:

- $|A|$ is a set of events;
- $\text{pol}_A : |A| \to \{-, +\}$ is a labelling function;
- $\leq_A$ defines a negative and alternating tree.
A play on arena $A$ is a pointing string $s = s_1 \ldots s_n \in |A|^*$ such that:

- The pointers respect $\rightarrow_A$;
- alternating: $\forall 1 \leq i < n$, $\pol_A(s_i) \neq \pol_A(s_{i+1})$;
- legal: $\forall 1 \leq i \leq n$, either $s_i = \min(A)$ or $s_i$ has a pointer.

Typical play for $\lambda f^o \rightarrow o. \lambda x^o. x$

Typical play for $\lambda f^o \rightarrow o. \lambda x^o. f x$

Typical play for $\lambda f^o \rightarrow o. \lambda x^o. f f x$
A strategy $\sigma : A$ is a non-empty set $\sigma \subseteq \text{Plays}^+(A)$ satisfying:

- **prefix-closed:** $\forall s \in \sigma, \forall t \sqsubseteq^+ s, t \in \sigma$,
- **deterministic:** $\forall s \in \sigma, \text{sab}, \text{sab}' \in \sigma \Rightarrow \text{sab} = \text{sab}'$.

$$((o \Rightarrow o) \Rightarrow o) \Rightarrow o$$

$$K_x : \lambda f^{(o \Rightarrow o) \Rightarrow o}. f (\lambda x^o. f (\lambda y^o. x))$$
A **strategy** $\sigma : A$ is a non-empty set $\sigma \subseteq \text{Plays}^+(A)$ satisfying:

- **prefix-closed:** $\forall s \in \sigma, \forall t \sqsubseteq^+ s, t \in \sigma$,
- **deterministic:** $\forall s \in \sigma, \text{sab}, \text{sab}' \in \sigma \implies \text{sab} = \text{sab}'$.

\[
((o \Rightarrow o) \Rightarrow o) \Rightarrow o \quad \text{or} \quad K_x : \lambda f^{(0 \Rightarrow o) \Rightarrow o}. f (\lambda x^o. f (\lambda y^o. x))
\]

\[
((o \Rightarrow o) \Rightarrow o) \Rightarrow o \quad \text{or} \quad \lambda f^{(0 \Rightarrow o) \Rightarrow o}. f (\lambda x^o. f (\lambda y^o. x))
\]

\[
((o \Rightarrow o) \Rightarrow o) \Rightarrow o \quad \text{or} \quad \lambda f^{(0 \Rightarrow o) \Rightarrow o}. f (\lambda x^o. f (\lambda y^o. x))
\]
A **P-view** is a play where Opponent moves point to the previous Player move.

A strategy is **innocent** if Player reacts the same way to every duplication of Opponent moves.

\[ ((o \Rightarrow o) \Rightarrow o) \Rightarrow o \]
A **P-view** is a play where Opponent moves point to the previous Player move.

\[( (o \Rightarrow o) \Rightarrow o ) \Rightarrow o \]
A **P-view** is a play where Opponent moves point to the previous Player move.

A strategy is **innocent** if Player reacts the same way to every duplication of Opponent moves.
Outline

1 Introduction to Game Semantics

2 Positional Injectivity
   - Positions
   - Positionality
   - Positional Injectivity

3 Proof Method
The **position** of a play $s$, noted $(s)$, is the desequentialization of $s$ (it is $s$ without its temporal order).

$$((o \Rightarrow o) \Rightarrow o) \Rightarrow o$$

A play $s$ of $K_x$

$(s)$
A strategy $\sigma : A$ is **positional** if for all $sab, t \in \sigma$, $ta' \in \text{Plays}(A)$,

$$(sa) = (ta') \Rightarrow \exists ta'b \in \sigma, (sab) = (ta'b).$$

Two maximal P-views for $\lambda f^\circ \rightarrow^o \rightarrow^o. \lambda x^o. \lambda y^o. f (fx)(fy)$
A strategy $\sigma : A$ is **positional** if for all $sab, t \in \sigma$, $ta' \in \text{Plays}(A)$,

$$(sa) = (ta') \Rightarrow \exists ta'b \in \sigma, (sab) = (ta'b).$$

Two maximal P-views for $\lambda f^{o \rightarrow o \rightarrow o}. \lambda x^o. \lambda y^o. f \ (f \ x \ x) \ (f \ y \ y)$
A strategy $\sigma : A$ is **positional** if for all $sab, t \in \sigma$, $ta' \in \text{Plays}(A)$,

$$(sa) = (ta') \Rightarrow \exists ta'b \in \sigma, (sab) = (ta'b).$$

Two maximal P-views for $\lambda f^\circ\circ\to\circ. \lambda x^\circ. \lambda y^\circ. f (f x x) (f y y)$
The **positions** of a strategy $\sigma : A$ are $\{|\sigma|\} = \{s \mid s \in \sigma\}$.

**Question (Positional Injectivity)**

For any $\sigma, \tau : A$, do we have $\{|\sigma|\} = \{|\tau|\} \Rightarrow \sigma = \tau$?
The **positions** of a strategy $\sigma : A$ are $\{\|\sigma\|\} = \{\{s\} : s \in \sigma\}$.

**Question (Positional Injectivity)**

For any $\sigma, \tau : A$, do we have $\{\|\sigma\|\} = \{\|\tau\|\} \Rightarrow \sigma = \tau$ ?

$$((o \Rightarrow o) \Rightarrow o) \Rightarrow o$$

$$K_x : \lambda f^{(o \rightarrow o) \rightarrow o} \cdot f (\lambda x^o \cdot f (\lambda y^o \cdot x))$$

$$((o \Rightarrow o) \Rightarrow o) \Rightarrow o$$

$$K_y : \lambda f^{(o \rightarrow o) \rightarrow o} \cdot f (\lambda x^o \cdot f (\lambda y^o \cdot y))$$
The **positions** of a strategy \( \sigma : A \) are \( |\sigma| = \{ (s) \mid s \in \sigma \} \).

**Question (Positional Injectivity)**

For any \( \sigma, \tau : A \), do we have \( |\sigma| = |\tau| \Rightarrow \sigma = \tau ? \)

\[
((o \Rightarrow o) \Rightarrow o) \Rightarrow o
\]

\[
K_x : \lambda f^{(o \rightarrow o) \rightarrow o}. f (\lambda x^{o}. f (\lambda y^{o}. x))
\]

\[
K_y : \lambda f^{(o \rightarrow o) \rightarrow o}. f (\lambda x^{o}. f (\lambda y^{o}. y))
\]
A strategy $\sigma : A$ is **total** iff for all $s \in \sigma$,

$$sa \in \text{Plays}(A) \implies \exists b \text{ such that } sab \in \sigma.$$  

A strategy $\sigma : A$ is **finite** iff the set of P-views of $\sigma$ is finite.
A strategy $\sigma : A$ is **total** iff for all $s \in \sigma$,

$$sa \in \text{Plays}(A) \implies \exists b \text{ such that } sab \in \sigma.$$ 

A strategy $\sigma : A$ is **finite** iff the set of P-views of $\sigma$ is finite.

**Theorem (Positional Injectivity)**

For any $\sigma, \tau : A$ innocent total finite, $\sigma = \tau$ iff $\{\sigma\} = \{\tau\}$. 

Lison Blondeau-Patissier & Pierre Clairambault
Outline

1. Introduction to Game Semantics
2. Positional Injectivity
3. Proof Method
   - Augmentations
   - Characteristic Augmentations
   - Positional Injectivity
Augmentations = Plays without Opponent’s Scheduling
Augmentations

Augmentations = Plays without Opponent’s Scheduling
= Positions augmented with causal order
Augmentations

Augmentations = Plays without Opponent’s Scheduling

= Positions augmented with causal order

\[(o \Rightarrow o) \Rightarrow o\]
Augmentations

Augmentations = Plays without Opponent’s Scheduling
= Positions augmented with causal order

\(((o \Rightarrow o) \Rightarrow o) \Rightarrow o\)
Augmentations

Augmentations = Plays without Opponent’s Scheduling
= Positions augmented with causal order

$$((o \Rightarrow o) \Rightarrow o) \Rightarrow o$$

Are causal games positionally injective?
Using Duplications of Opponent Moves
Using Duplications of Opponent Moves
Using Duplications of Opponent Moves

Fork = \{ \ominus \text{’s with the same parent and arena image} \}
Using Duplications of Opponent Moves

**Fork** = \{ \ominus's with the same parent and arena image \}

**Clone class** = \{ “similar” \oplus’s \}

\[
\begin{align*}
\#F_1 &= 1 & \#F_2 &= 1 & \#F_3 &= 2 \\
\#C_1 &= 1 & \#C_2 &= 1 & \#C_3 &= 2
\end{align*}
\]
Using Duplications of Opponent Moves

Fork = \{ \ominus \text{’s with the same parent and arena image} \}

Clone class = \{ \text{“similar” } \oplus \text{’s} \}

\[
\begin{align*}
\#F_1 &= 1 & \#F_2 &= 1 & \#F_3 &= 2 \\
\#C_1 &= 1 & \#C_2 &= 1 & \#C_3 &= 2
\end{align*}
\]
A **characteristic augmentation** is an augmentation extracted from the maximal P-views of a strategy, such that each fork has for cardinality a unique power of 2.
Theorem: Positional Injectivity (for causal games)

For any $\sigma$, $\tau$ finite causal strategies,

$$\langle \sigma \rangle = \langle \tau \rangle \iff \sigma = \tau.$$
Positional Injectivity

**Theorem: Positional Injectivity (for causal games)**

For any $\sigma$, $\tau$ finite causal strategies,

$$\left\langle |\sigma| \right\rangle = \left\langle |\tau| \right\rangle \iff \sigma = \tau.$$  

**Theorem: Positional Injectivity (for Hyland Ong games)**

For any $\sigma$, $\tau$ total finite innocent strategies,

$$\left\langle |\sigma| \right\rangle = \left\langle |\tau| \right\rangle \iff \sigma = \tau.$$
Conclusion

Theorem: Positional Injectivity

Total finite innocent strategies are positionally injective.
Theorem: Positional Injectivity

Total finite innocent strategies are positionally injective.

Thank you!
\[ T_1 = f \ T_2 \ R \quad T_2 = f \ L \ T_1 \quad L = f \ L \perp \quad R = f \perp \ R \]

\[ \lambda f^{o \to o \to o} \cdot T_1 \]

\[ \lambda f^{o \to o \to o} \cdot T_2 \]