Justified sequences in string diagrams
A comparison between two approaches to concurrent game semantics

Clovis Eberhart and Tom Hirschowitz

LAMA, CNRS, Université Savoie Mont Blanc

CALCO 2017, Ljubljana, June 14–16, 2017
Game semantics for concurrent languages

Game semantics (Hyland-Ong ’00, Nickau ’94):
- Types $\rightarrow$ games
- Programs $\rightarrow$ strategies

Game semantics for concurrent languages via sheaves:
- Hirschowitz et al.: CCS, $\pi$-calculus
- Tsukada and Ong: non-deterministic $\lambda$-calculus
Sheaf-theoretic approaches to game semantics

Both approaches

Innocent strategies = sheaves for a Grothendieck topology induced by the embedding of views into plays

Different notions of plays:

- Hirschowitz et al.: string diagrams
- Tsukada and Ong: justified sequences

This work

- Design a string diagrammatic model of HON games
- Show a strong relationship between plays in both models
- Deduce equivalence of both notions of innocent strategies
The level of plays

justified sequences: \[ \forall_{A, B} \xRightarrow{i_{\text{HON}}} \mathbb{P}_{A, B} \]

string diagrams: \[ \mathcal{E}^V(A \vdash B) \xRightarrow{i} \mathcal{E}(A \vdash B) \]
The level of plays

justified sequences:

string diagrams:
The level of plays

justified sequences:

proof trees:

string diagrams:
The level of strategies

The square

\[
\begin{array}{c}
\mathbb{V}_{A,B} \\
F^V
\end{array}
\xymatrix{
\mathbb{V}_{A,B} & \ar[l]_{i_{HON}} \mathbb{P}_{A,B} \\
\mathbb{E}^V(A \vdash B) & \ar[u]_{i} \mathbb{E}(A \vdash B) \\
}
\]

is exact (Guitart, 1980), so

\[
\begin{array}{c}
\Delta_{F^V} \\
\Delta_F
\end{array}
\xymatrix{
\mathbb{V}_{A,B} & \ar[r]_{\Pi_{i_{HON}}} \mathbb{P}_{A,B} \\
\mathbb{E}^V(A \vdash B) & \ar[r]_{\Pi_i} \mathbb{E}(A \vdash B) \\
}
\]

commutes up to isomorphism.
Overview

1. The level of plays

2. The level of strategies
The level of plays

1. The level of plays

2. The level of strategies
Arenas

HON game semantics is based on arenas. Arena = forest of moves
Example: boolean arena $\mathbb{B}$:

```
  q
 / \   \
 t   f
```

Residual of an arena $A \cdot m$: forest below $m$
Plays in HON games

- **Play on** \((A, B)\) = justified sequence of moves in \(A\) or \(B\)
- **Example on** \((B, B)\):

```
q_r
q_l
  
  t_l
f_l
f_r
  
  t_r
```

- **View on** \((A, B)\) = particular kind of play (inductive definition)
Plays as string diagrams

- **Position** \(\approx\) set of players,
  - *positive* \(\circ\) (labelled by a *positive* sequent of arenas \((\Gamma \vdash)\)) or
  - *negative* \(\bullet\) (labelled by a *negative* sequent of arenas \((\Gamma \vdash A)\))
Plays as string diagrams

- **Position** \( \simeq \) set of players,
  - positive \( \circ \) (labelled by a positive sequent of arenas \( (\Gamma \vdash \cdot) \)) or
  - negative \( \bullet \) (labelled by a negative sequent of arenas \( (\Gamma \vdash A) \))

- **Two kinds of moves:**
  \[ \Gamma, A, \Delta \vdash A \cdot m \]
  \[ \Gamma, A \cdot m \vdash \]

  \[ \Gamma, A, \Delta \vdash \text{ and } \Gamma \vdash A \] , in context, e.g.:
Plays as string diagrams

- **Position** ≈ set of players,
  - **positive** (labelled by a positive sequent of arenas (\(\Gamma \vdash\)) or
  - **negative** (labelled by a negative sequent of arenas (\(\Gamma \vdash A\)))

- **Two kinds of moves:**
  \[
  \Gamma, A, \Delta \vdash A \cdot m \\
  \Gamma, A \cdot m \vdash \\
  \Gamma, A, \Delta \vdash \text{ and } \Gamma \vdash A
  \]

- **Play** = vertical pasting of moves

![String diagram of plays as string diagrams]
The big picture

justified sequence

string diagram
The big picture

1. Justified sequence
2. P-view tree
3. Proof tree
4. String diagram
5. Sequentialised proof tree
$P$-view trees

Justified sequence $\rightarrow$ tree whose branches are views

Example:
\textit{P-view trees}

Justified sequence $\rightarrow$ tree whose branches are views

Example:
**P-view trees**

Justified sequence $\rightarrow$ tree whose branches are views

Example:
Introduction

The level of plays

The level of strategies

Conclusion

Proof trees

Partial trees for:

\[ \ldots \quad \Gamma, A \cdot m(i) \vdash \ldots \quad (\forall i \in n) \]
\[ \Gamma \vdash A \]

\[ \Gamma, A, \Delta \vdash A \cdot m \]
\[ \Gamma, A, \Delta \vdash \]

Example:

\[ \mathbb{B}_l, \{t, f\}_r, \emptyset \vdash \emptyset \]
\[ f_r \]

\[ \mathbb{B}_l, \{t, f\}_r, \emptyset \vdash \{t, f\}_l \]
\[ t_r, t_l, f_l \]

\[ \mathbb{B}_l, \{t, f\}_r \vdash \{t, f\}_l \]
\[ q_l \]

\[ \mathbb{B}_l \vdash \mathbb{B}_r \]
\[ q_r \]
$P$-view trees versus proof trees

Differences:

- Presence of pointers
- Labelling
From proof trees to string diagrams (part 1)

\[
\begin{align*}
\frac{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash \emptyset}{t_r} & \quad \frac{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash \emptyset}{t_l, f_l} \\
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{q_l} & \quad \frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{q_r}
\end{align*}
\]

arbitrarily branching tree
\[\downarrow\]
binary tree
\[\downarrow\]
sequentialisation
\[\downarrow\]
n-ary node
\[\downarrow\]
comb
From proof trees to string diagrams (part 1)

\[
\frac{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash \emptyset}{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash} f_r
\frac{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash \emptyset}{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash} t_r
\]

\[
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} q_l
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} f_l
\]

\[
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} q_l
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} t_l
\]

\[
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} q_l
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} t_l
\]

arbitrarily branching tree
\[\downarrow\]
binary tree
sequentialisation
\[\downarrow\]
n-ary node
comb

\[
\frac{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash \emptyset}{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash} f_r
\frac{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash \emptyset}{\mathcal{B}_l, \{t, f\}_r, \emptyset \vdash} t_r
\]

\[
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} q_l
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} f_l
\]

\[
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} q_l
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} t_l
\]

\[
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} q_l
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} f_l
\]

\[
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} q_l
\frac{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l}{\mathcal{B}_l, \{t, f\}_r \vdash \{t, f\}_l} t_l
\]
From proof trees to string diagrams (part 2)
From proof trees to string diagrams (part 2)
From proof trees to string diagrams (part 2)
From proof trees to string diagrams (part 2)
From proof trees to string diagrams (part 2)
The level of strategies

1. The level of plays

2. The level of strategies
Deterministic strategies

Two notions of strategies

- **Behaviours** = prefix-closed set of views = \([\mathcal{V}^{op}, 2]\)
- **Innocent strategies** = prefix-closed set of plays + innocence = some functors \([\mathcal{P}^{op}, 2]\)

Problem

Milner’s coffee machines

accept the same traces: \(\varepsilon, a, ab,\) and \(ac\).
Non-deterministic strategies

### Solution

Accept trace or not $\rightarrow$ set of possible states after accepting trace

- **Behaviours** $= [\mathbb{V}^{\text{op}}, \text{Set}] = \hat{\mathbb{V}}$
- **Innocent strategies** $= \text{some functors } [\mathbb{P}^{\text{op}}, \text{Set}]$
  
  $= \text{some presheaves in } \hat{\mathbb{P}}$
  
  essential image of $\prod \! i$

\[
\begin{array}{c}
\begin{tikzpicture}
\node (a) at (0,0) {$a$};
\node (b) at (-1,-1) {$b$};
\node (c) at (1,-1) {$c$};
\node (x) at (0,-2) {$x$};
\node (x') at (0,-3) {$x'$};
\draw [->] (a) -- (b); \draw [->] (a) -- (c);
\end{tikzpicture}
\end{array}
\]

\[
\begin{array}{c}
\begin{tikzpicture}
\node (a) at (0,0) {$a$};
\node (x) at (0,-1) {$x$};
\node (x') at (0,-2) {$x'$};
\node (b) at (-1,-2) {$b\downarrow$};
\node (c) at (1,-2) {$c\downarrow$};
\draw [->] (a) -- (x); \draw [->] (x) -- (x'); \draw [->] (a) -- (b); \draw [->] (a) -- (c);
\end{tikzpicture}
\end{array}
\]

\[
S(a) = \{x\}
\]

\[
S(a) = \{x, x'\}
\]
Categories of innocent strategies

The square

\[
\begin{array}{ccc}
\Delta_{FV}^\uparrow & \Pi_i^{HON} & \Delta_F \\
\mathbb{E}^V(A \vdash B) & \longrightarrow & \mathbb{E}(A \vdash B)
\end{array}
\]

commutes up to isomorphism:

- Behaviours are equivalent
- Innocent strategies are equivalent
- compatible with innocentisation
- (Non-innocent strategies are not)
Conclusion

Done: link between two models of game semantics:

- At the level of plays:
  - Full embedding of justified sequences into string diagrams
  - Equivalence of categories of views

- At the level of strategies:
  - Equivalent categories of behaviours and innocent strategies
  - Compatible with innocentisation

To do: composition of strategies in our setting.