Game Semantics

Dan R. Ghica
University of Birmingham

July 20, 2018

1 Composition = Synch + Replication + Hiding

- Associative
- Identity?

$K_A : A_0 \to A_1$
Define via a next-move function $\hat{K_A}$
$q_1(a\langle b\rangle) \mapsto q_0(b\langle c\rangle)$
$p \cdot m_{1j}(a\langle b\rangle) \cdot m_j(c\langle d\rangle) \cdot p' \cdot n_j^0(d\langle e\rangle) \mapsto n_{1j}^0(b\langle f\rangle)$

1.1 Lemma

$K_A : A_1 \to A_2$
$K_A \downarrow A_1 = K_A \downarrow A_2$

1.2 Lemma

K_A is a strategy.

1.3 Theorem

$K_A; T \stackrel{?}{=} T; K_B \stackrel{?}{=} T$ - Not true!

1.4 Counterexample

$$\texttt{Unit} \xrightarrow{T_{\texttt{BAD}}} \texttt{Unit} \xrightarrow{K_{\texttt{Unit}}} \texttt{Unit}$$

1.5 How to fix identity?

(A) Discipline plays with extra conditions
(Rule out $T_{\texttt{BAD}}$, etc.)

(B) Add closure conditions on strategies

Have to be careful - these choices are language specific.
Concurrency $\rightarrow$ asynchronous games.
Need extra conditions because "static concurrency" $C_1 \parallel C_2$

1.5.1 Fork/Join

If a thread finishes then all its sub-threads must have finished, $(C_1 \parallel C_2); C_3$
Translation to game language: If a question is answered then all its justified questions must have been answered.
Typical Strategy: $T_{\parallel} : \texttt{Command}_1 \rightarrow \texttt{Command}_2 \rightarrow \texttt{Command}$

1.6 Definition: Strict Scoping

$P \cdot m(a\langle - \rangle) \cdot P' \in P$ and $m \in A$ then $a \notin P'$

1.7 Definition: Strict Nesting

$P_1 \cdot m_1(a\langle b \rangle) \cdot P_2 \cdot m_2(b\langle c \rangle) \cdot P_3 \cdot n_1(b\langle d \rangle) \in P$ and $n_1 \in A$ then $\exists n_2 \in A$ s.t. $n_2(c\langle - \rangle) \in P_3$

1.8 Theorem

$A \xrightarrow{T} B \xrightarrow{\tau} C$ have $S.N, S.S$ plays
then $T; \tau$ has $S.N, S.S$ plays.

1.9 When can two moves synch in a play?

1.9.1 Definition: Asynchronous Strategy

$T : A$ of SN/SS plays
$P \cdot m_0(a_0\langle b_0 \rangle) \cdot m_1(a_1\langle b_1 \rangle) \in T$
and $m_0 \in P_A$ or $m_1 \in O_A$ and $P \cdot P_1 \cdot P_0 \in P_A$
then $P \cdot P_1 \cdot P_0 \in T$

2 Concurrent Idealized Algol

PCF +
Local State (newvar, asg, der) +
concurrency (par) +
binary semantics (newsem, grab, release)
$\Gamma = x_0 : \theta_0, \ldots, x_R : \theta_R$
$\Gamma \vdash M : \Theta$

new x **in** M $\equiv$ newvar(λx.M)
sem x **in** M $\equiv$ newsem(λx.M)

2.1 Challenges

```
λc.new x = 0 in C; !x ≡ λc.C; 0
λf.new x = 0 in f(x := !x + 1)(!x)
          ≡ λf.new y = 0 in f(x := !x − 1)(−!x)
```

3 A New Definition of λ-Calculus

Cartesian Closed Category (CCC)

- Product: $A \times B$
- Unit: $I \times A \cong A \times I \cong A$
- Projections: $\Pi_i : A_1 \times A_2 \to A_i \quad (i = 1, 2)$
- Pairing: $\langle T; \tau \rangle : A_1 \times A_2 \to B$ if $T : A_1 \to B$ and $\tau : A_2 \to B$
  $\langle T; \tau \rangle = T \cup \tau$
- Exponential: $A \to B$
- Transpose: $\frac{T : A \times B \to C}{\lambda T : A \to B \to C}$
- Evaluation Strategy: $ev : (A \to B) \times A \to B$

3.1 Concurrent Idealized Algol (General Recipe)

$[\![\texttt{Nat}]\!] = \texttt{Nat}$
$[\![\texttt{Var}]\!] = \texttt{Var}$
$[\![\theta \to \theta']\!] = [\![\theta]\!] \to [\![\theta']\!]$
$[\![\Gamma]\!] = [\![\theta_0]\!] \times \cdots \times [\![\theta_i]\!]$
$\Gamma, x_\kappa : \theta_\kappa, \Gamma' \vdash x_\kappa : \theta_\kappa = \Pi_\kappa = \texttt{strat}(\kappa_{[\![\theta_\kappa]\!]})$
$[\![\Gamma \vdash \lambda x.M : \theta \to \theta']\!] = \lambda[\![\Gamma, x : \theta \vdash M : \theta']\!]$
$[\![\Gamma \vdash MN : \theta]\!] = \langle [\![M : \theta' \to \theta, [\![N : \theta']\!]]\!] \rangle; ev = ([\![M]\!] \cup [\![N]\!]), \texttt{strat}(\kappa_\theta \cup \kappa_{\theta'})$
$[\![\vdash M_0 : \theta_0]\!] = \Gamma_{M_0}$