classy

acondolu

March 2024

1 Calculus

Definition 1 (Terms).

$$t, u ::= \blacksquare \mid x \mid \langle t, u \rangle \mid [t, u] \mid \{t, u\}$$

Definition 2 (Types).

$$A, B ::= p \mid \bar{p} \mid A \wedge B \mid A \vee B$$

Definition 3 (Dual type $A^\perp$). .

$$\begin{array}{lll} p^\perp & \coloneqq & \bar{p} \\ \bar{p}^\perp & \coloneqq & p \\ (A \vee B)^\perp & \coloneqq & A^\perp \wedge B^\perp \\ (A \wedge B)^\perp & \coloneqq & A^\perp \vee B^\perp \end{array}$$

Definition 4 (Typing rules). .

$$\frac{}{x : A, x : A^\perp}$$

$$\frac{\Gamma}{\Gamma, \blacksquare : A}\ \textit{weakening}$$

$$\frac{\Gamma, t : A, u : A}{\Gamma, \{t, u\} : A}\ \textit{contraction}$$

$$\frac{\Gamma, t : A, u : B}{\Gamma, [t, u] : A \vee B}$$

$$\frac{\Gamma, t : A \qquad \Delta, u : B}{\Gamma, \Delta, \langle t, u \rangle : A \wedge B}$$

$$\frac{\Gamma, t : A \qquad \Delta, u : A^\perp}{\Gamma, \Delta}\ \textit{cut}$$

Definition 5 (Programs)**.** *A* cut expression *has the form $t * u$ where t and u are terms. The cut operator $*$ is commutative.*

Programs are denoted by Π and are unordered sequences of cut expressions.

$$\Pi ::= \epsilon \mid \Pi, t * u$$

Definition 6 (Reduction rules)**.** *.*

$$\begin{array}{llll} \textit{Destruct} & \Pi, \langle t_1, t_2 \rangle * [u_1, u_2] & \rightarrow_d & \Pi, t_1 * u_1, t_2 * u_2 \\ \textit{Send} & \Pi, x * t & \rightarrow_s & \Pi \{x \Leftarrow t\} \\ \textit{Copy} & \Pi, \{t_1, t_2\} * u & \rightarrow_c & \Pi, t_1 * u, t_2 * u^\alpha \end{array}$$

The Copy rule makes a copy of the term u by duplicating the free variables that occur in it. We denote by u^α the term obtained from u by replacing each variable x with a new fresh variable x'. Note: if u contains free variables, the notion of copy is undefined at the moment (TODO).

Example 1 (Divergent program)**.**

$$\omega \coloneqq [\{\langle x, y\rangle, x\}, y]$$

$$\begin{array}{rl} \Omega \coloneqq & \omega * \langle \omega', \mathbb{R} \rangle \\ \rightarrow_d & \{\langle x, y\rangle, x\} * \omega', y * \mathbb{R} \\ = & \{\langle x, y\rangle, x\} * [\{\langle x', y'\rangle, x'\}, y'], y * \mathbb{R} \\ \rightarrow_c & \langle x, y\rangle * [\{\langle x', y'\rangle, x'\}, y'], x * \omega'', y * \mathbb{R} \\ \rightarrow_d & x * \{\langle x', y'\rangle, x'\}, y * y', x * \omega'', y * \mathbb{R} \\ \rightarrow_s & x * \{\langle x', y'\rangle, x'\}, x * \omega'', y' * \mathbb{R} \\ \rightarrow_s & \{\langle x', y'\rangle, x'\} * \omega'', y' * \mathbb{R} \end{array}$$

Definition 7 (α-equivalence)**.** *We define $\equiv_\alpha$, the equivalence relation over classy-terms or classy-programs up to renaming of variables. (TODO)*

2 Confluence

Critical pairs may arise because of terms like:

- $x * y$: not actually critical pairs, they are α-equivalent.
- $\{t_1, t_2\} * x$: invalid at the moment, because x has a free variable (TODO).
- $\{t_1, t_2\} * \{u_1, u_2\}$: locally confluent.

3 Embedding λ-terms

An embedding of λ-terms into classy-programs can be defined through the usual type embedding that translates $A \to B$ into $A \vee B^{\perp}$.

We denote by σ a sequence of assignment of variables to terms of the form $(x \mapsto t)$.

We denote by σ_x the restriction of an assignment σ to the variable x:

$$\sigma_x := \{t \mid (x \mapsto t) \in \sigma\}.$$

The translation $\tau(t)$ of a λ-term into a program relies on an auxiliary function $\tau_{\text{aux}}(t)$, which takes in input a λ-term and returns a thriple $(\Pi; \sigma; t')$ composed of a program, a substitution, and a term.

The auxiliary function $\tau_{\text{aux}}(t)$ is defined as follows:

$$\begin{array}{lcl}
\tau_{\text{aux}}(\lambda x.t) & := & (\Pi; \sigma \setminus \sigma_x; \texttt{[}\{\sigma_x\}, t'\texttt{]}) \\
& & \text{where } (\Pi; \sigma; t') := \tau_{\text{aux}}(t) \\
\tau_{\text{aux}}(x t_1 \ldots t_n) & := & (\Pi_1 \ldots \Pi_n; \sigma_1 \ldots \sigma_n (x \mapsto \texttt{<}t'_1, \ldots \texttt{<}t'_n, z\texttt{>>}); z) \\
& & \text{where } (\Pi_i; \sigma_i; t'_i) := \tau_{\text{aux}}(t_i) \text{ for } i = 1 \ldots n \\
& & \text{and } z \text{ is a fresh variable} \\
\tau_{\text{aux}}(t_0 t_1 \ldots t_n) & := & (\Pi_0 \Pi_1 \ldots \Pi_n, t'_0 * \texttt{<}t'_1, \ldots \texttt{<}t'_n, z\texttt{>>}; \sigma_0 \ldots \sigma_n; z) \\
& & \text{where } (\Pi_i; \sigma_i; t'_i) := \tau_{\text{aux}}(t_i) \text{ for } i = 0 \ldots n \\
& & \text{and } z \text{ is a fresh variable} \\
& & \text{and } t_0 \text{ is a lambda abstraction.}
\end{array}$$

The translation $\tau(t)$ is then defined as follows:

$$\tau(t) := \Pi \text{ where } (\Pi; \sigma; t') := \tau_{\text{aux}}(t).$$

4 Correctness

Definition 8 (Π-path)**.** *Let Π be a program. A Π-path is given by:*

- $n > 0$
- *a sequence of variables $x_0, \ldots, x_n$ in Π*
- *terms $t_0 \circledast u_0, \ldots, t_n \circledast u_n$ in Π (where $t \circledast u$ is either $\texttt{<}t, u\texttt{>}$ or $t * u$)*
- *such that $x_i \in t_i$ and $x_{i+1} \in u_i$ for $i = 0 \ldots n-1$.*

Definition 9. *A program Π is valid if*

- *All subterms are unique, except variables: each variable occurs exactly twice.*
- *There are no cyclic Π-paths (cyclic meaning a Π-path $x_0, \ldots, x_n$ such that $x_0 = x_n$).*

5 Simulation

Theorem 1. *For every λ-terms t and u, if $t \to_\beta u$, then $\tau(t) \to \tau(u)$.*

Proof. Let C be an evaluation context such that $t = C\langle(\lambda x.t')u'\rangle$ and $u = C\langle t'\{x \Leftarrow u'\}\rangle$. $\square$

6 Stub correctness proofs

$$\frac{}{\parallel x, x \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma \quad \mathrm{wf}}{\Pi \parallel \Gamma, \blacksquare \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma, t, u \quad \mathrm{wf}}{\Pi \parallel \Gamma, \{t, u\} \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma, t, u \quad \mathrm{wf}}{\Pi \parallel \Gamma, [t, u] \quad \mathrm{wf}}$$

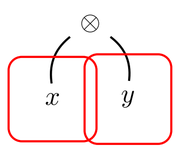
$$\frac{\Pi \parallel \Gamma, t \quad \mathrm{wf} \qquad \Pi' \parallel \Delta, u \quad \mathrm{wf}}{\Pi, \Pi' \parallel \Gamma, \Delta, \texttt{<}t, u\texttt{>} \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma, t \quad \mathrm{wf} \qquad \Pi' \parallel \Delta, u \quad \mathrm{wf}}{\Pi, \Pi', t * u \parallel \Gamma, \Delta \quad \mathrm{wf}}$$

- $x * a$
- $x * b$
- $< a, b >$

$$\begin{tikzcd} & Z \arrow[dl] \arrow[dr] & \\ X \arrow[rr] & & Y \end{tikzcd}$$

$$T := [\{\texttt{<} -x, -y\texttt{>}, +x\}, +y]$$



$$\frac{}{\parallel x, x \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma \quad \mathrm{wf}}{\Pi \parallel \Gamma, \blacksquare \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma, t, u \quad \mathrm{wf}}{\Pi, \alpha = \{t, u\} \parallel \Gamma, \alpha \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma, t, u \quad \mathrm{wf}}{\Pi, \alpha = [t, u] \parallel \Gamma, \alpha \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma, t \quad \mathrm{wf} \qquad \Pi' \parallel \Delta, u \quad \mathrm{wf}}{\alpha = \mathsf{<}\Pi t, \Pi' u\mathsf{>} \parallel \Gamma, \Delta, \alpha \quad \mathrm{wf}}$$

$$\frac{\Pi \parallel \Gamma, t \quad \mathrm{wf} \qquad \Pi' \parallel \Delta, u \quad \mathrm{wf}}{\alpha = \Pi t * \Pi' u \parallel \alpha \quad \mathrm{wf}}$$