classy

acondolu

March 2024

1 Calculus

Definition 1 (Terms)**.**

$$t, u ::= \blacksquare \mid x \mid \mathtt{<}t\mathtt{,}u\mathtt{>} \mid \mathtt{[}t\mathtt{,}u\mathtt{]} \mid \mathtt{\{}t, u\mathtt{\}}$$

Definition 2 (Types)**.**

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

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

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

Definition 4 (Typing rules)**.** .

$$\frac{}{x\colon A, x\colon A^\perp}\ \textit{axiom}$$

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

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

$$\frac{\Gamma, t\colon A, u\colon B}{\Gamma, \mathtt{[}t\mathtt{,}u\mathtt{]}\colon A \vee B}\ \textit{disjunction}$$

$$\frac{\Gamma, t\colon A \qquad \Delta, u\colon B}{\Gamma, \Delta, \mathtt{<}t\mathtt{,}u\mathtt{>}\colon A \wedge B}\ \textit{conjunction}$$

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

Note: the rules Axiom, Cut, Conjunction, Disjunction correspond to the sequent calculus rules of multiplicative linear logic (MLL).

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 (possibly empty) unordered sequences of cut expressions:

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

Definition 6 (Reduction rules)**.** *Reduction rules can apply to any cut expression in a program, and are defined as follows:*

$$\begin{array}{llll}
\text{Destruct} & \Pi, \mathtt{<} t_1 \mathtt{,} t_2 \mathtt{>} * \mathtt{[} u_1 \mathtt{,} u_2 \mathtt{]} & \to_d & \Pi, t_1 * u_1, t_2 * u_2 \\
\text{Send} & \Pi, x * t & \to_s & \Pi \{x \Leftarrow t\} \\
\text{Copy} & \Pi, \mathtt{\{} t_1, t_2 \mathtt{\}} * u & \to_c & \Pi, t_1 * u, t_2 * u^{\alpha} \qquad (*)
\end{array}$$

$(*)$ *The* Copy *rule makes a copy of the* closed *term u by duplicating all variables that occur in it. We denote by u^{α} a term obtained from u by replacing each variable x with a new globally fresh variable x'. u and u^{α} are α-equivalent according to upcoming definition of α-equivalence. Note: if u contains free variables, the notion of copy is undefined at the moment (TODO).*

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

Example 1 (Divergent program)**.**

$$\omega \coloneqq \mathtt{[\{<} x \mathtt{,} y \mathtt{>}, x \mathtt{\}} \mathtt{,} y \mathtt{]}$$

$$\begin{array}{rl}
\Omega \coloneqq & \omega * \mathtt{<} \omega' \mathtt{,} \mathbb{R} \mathtt{>} \\
\to_d & \mathtt{\{<} x \mathtt{,} y \mathtt{>}, x \mathtt{\}} * \omega', y * \mathbb{R} \\
= & \mathtt{\{<} x \mathtt{,} y \mathtt{>}, x \mathtt{\}} * \mathtt{[\{<} x' \mathtt{,} y' \mathtt{>}, x' \mathtt{\}} \mathtt{,} y' \mathtt{]}, y * \mathbb{R} \\
\to_c & \mathtt{<} x \mathtt{,} y \mathtt{>} * \mathtt{[\{<} x' \mathtt{,} y' \mathtt{>}, x' \mathtt{\}} \mathtt{,} y' \mathtt{]}, x * \omega'', y * \mathbb{R} \\
\to_d & x * \mathtt{\{<} x' \mathtt{,} y' \mathtt{>}, x' \mathtt{\}}, y * y', x * \omega'', y * \mathbb{R} \\
\to_s & x * \mathtt{\{<} x' \mathtt{,} y' \mathtt{>}, x' \mathtt{\}}, x * \omega'', y' * \mathbb{R} \\
\to_s & \mathtt{\{<} x' \mathtt{,} y' \mathtt{>}, x' \mathtt{\}} * \omega'', y' * \mathbb{R}
\end{array}$$

2 Confluence

Upon a first inspection, critical pairs may seem to arise because of terms of the form $x * y$, $\mathtt{\{} t_1, t_2 \mathtt{\}} * x$, or $\mathtt{\{} t_1, t_2 \mathtt{\}} * \mathtt{\{} u_1, u_2 \mathtt{\}}$.

- $\Pi \{y \Leftarrow x\} \leftarrow_s \Pi, x * y \to_s \Pi \{x \Leftarrow y\}$. This is actually not a critical pair, because $\Pi \{y \Leftarrow x\} \equiv_{\alpha} \Pi \{x \Leftarrow y\}$ in valid programs.
- $\Pi, \mathtt{\{} t_1, t_2 \mathtt{\}} * x$: cannot apply $\to_c$, because x has a free variable (TODO).
- $\mathtt{\{} t_1, t_2 \mathtt{\}} * \mathtt{\{} u_1, u_2 \mathtt{\}}$: it is clearly 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:

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

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:

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

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

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

4 Correctness

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

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

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 $x_0, \dots, x_n$ 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

$$\frac{}{\| \; x, x \quad \mathrm{wf}}$$

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

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

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

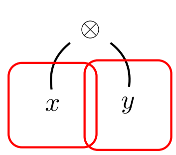
$$\frac{\Pi \| \Gamma, t \quad \mathrm{wf} \qquad \Pi' \| \Delta, u \quad \mathrm{wf}}{\Pi, \Pi' \| \Gamma, \Delta, \texttt{<}t\texttt{,}u\texttt{>} \quad \mathrm{wf}}$$

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

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

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

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



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

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

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

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

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

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