Examples and Proofs for the paper
Functions as processes: termination and the $\tilde{\lambda}\mu\tilde{\mu}$-calculus

Matteo Cimini
School of Computer Science
Reykjavik University
Reykjavik, Iceland
matteo@ru.is

Claudio Sacerdoti Coen
Department of Computer Science
University of Bologna
Bologna, Italy
sacerdot@cs.unibo.it

Davide Sangiorgi
Department of Computer Science
University of Bologna
Bologna, Italy
and INRIA, France
davide.sangiorgi@cs.unibo.it

1 Example of a term encoded by $[\cdot]$

In this section we provide an example of encoded term and we show its reduction steps. Let us consider the following command

$$c = \mu\beta. <\lambda x.\lambda y.\mu y. < x | y \cdot \gamma > | \lambda x. x \cdot \beta > | M \cdot \alpha >$$

The command $c$ is the encoding of the $\lambda$-term $\lambda x.\lambda y. (x y) \lambda x. x M'$ given by the call-by-name compilation of $[\cdot]$. We consider $M$ as the encoding of $M'$. The example is complex enough to show clearly how the encoding simulates $\tilde{\lambda}\mu\tilde{\mu}$. In what follows we write $env(x, P)$, with $x$ a variable and $P$ a process, the $\pi$-term $\langle x, P \rangle$. For sake of readability we omit the initial restrictions ($\nu$-operations) over the channels: every channel is considered private with the exception of $\mu, \tilde{\mu}$ and $\lambda$.

$$[c] = Rec \ y. \tilde{\mu}(\beta). \langle \lambda x.\lambda y.\mu y. < x | y \cdot \gamma > | \lambda x. x \cdot \beta > \rangle + \tilde{\mu}(x). !x. Y \ | \ Rec \ y. \tilde{\lambda}(\delta). ([M] \ | !\delta. ([\alpha])] + \mu(\beta)). !\beta. Y \rightarrow^\kappa \ (by \ \mu)$$

Rec $y. \lambda(\delta). \langle \tilde{\mu}x.\langle \lambda x.\lambda y.\mu y. < x | y \cdot \gamma > | \delta > \rangle + \tilde{\mu}(z). !z. Y \ | \ Rec \ y. \tilde{\lambda}(\delta). ([\lambda x. x] \ | !\delta. ([\beta])] + \mu(\beta)). !\beta. Y \ | \ env(y, [M]) \rightarrow^\kappa \ (by \ \lambda)$

Rec $y. \lambda(\delta'). \langle \tilde{\mu}y. < \gamma > | \delta' > \rangle + \tilde{\mu}(z). !z. Y \ | \ Rec \ y. \tilde{\delta} + \mu(\beta)). !\beta. Y \ | \ env(y, [M]) \rightarrow^\kappa \ (by \ \delta)$

Rec $y. \lambda(\delta'). \langle \tilde{\mu}y. < \gamma > | \delta' > \rangle + \tilde{\mu}(z). Y \ | \ Rec \ y. \tilde{\beta} + \mu(\beta)). !\beta. Y \ | \ env(y, [M]) \rightarrow^\kappa \ (by \ \delta')$

Rec $y. \lambda(\delta'). \langle \tilde{\mu}y. < \gamma > | \delta' > \rangle + \tilde{\mu}(z). !z. Y \ | \ Rec \ y. \tilde{\lambda}(\delta). ([M] \ | !\delta. ([\alpha])] + \mu(\beta)). !\beta. Y \ | \ env(y, [M]) \rightarrow^\kappa \ (by \ \lambda)$

Rec $y. \tilde{\mu}(y). \langle \mu y. < x | y \cdot \gamma > | \delta' > \rangle + \mu(\beta)). !\beta. Y \ | \ env(y, [M]) \rightarrow^\kappa \ (by \ \mu)$

Rec $y. \tilde{\mu}(y). \langle \mu y. < x | y \cdot \gamma > | \delta' > \rangle + \mu(\beta)). !\beta. Y \ | \ env(y, [M]) \rightarrow^\kappa \ (by \ \mu)$

Rec $y. \tilde{\mu}(y). \langle \mu y. < x | y \cdot \gamma > | \delta' > \rangle + \mu(\beta)). !\beta. Y \ | \ env(y, [M]) \rightarrow^\kappa \ (by \ \mu)$

Rec $y. \tilde{\lambda}(\delta). ([M] \ | !\delta. ([\alpha])] + \mu(\beta)). !\beta. Y \ | \ env(y, [M]) \rightarrow^\kappa \ (by \ \lambda)$
Rec \( Y \). \( \lambda(\delta''). \downarrow \mu. < z \mid \delta'' > \downarrow \mu(z). \downarrow Y \) | Rec \( Y \). \( \lambda(\delta). (\uparrow y) \mid ! \delta. (\uparrow y) \) + \( \mu(\beta). ! \beta. \downarrow Y \) | env(\( \beta. [M \cdot \alpha] \)) | env(\( \delta. [\beta] \)) | env(x, [\( \lambda x. x \)] y) | env(\( \delta'. [\alpha] \)) | env(y, [M]) | env(\( \delta''. [\gamma] \)) \rightarrow^\pi \) by \( \lambda \)
Rec \( Y \). \( \overline{\delta} + \overline{\mu}(z). \downarrow Y \) | Rec \( Y \). \( \overline{\mu}(z). < z \mid \delta'' > \downarrow \mu(\beta). ! \beta. \downarrow Y \) | env(\( \beta. [M \cdot \alpha] \)) | env(\( \delta. [\beta] \)) | env(x, [\( \lambda x. x \)] y) | env(\( \delta'. [\alpha] \)) | env(y, [M]) | env(\( \delta''. [\gamma] \)) \rightarrow^\pi \) by \( \mu \)
Rec \( Y \). \( \overline{\delta} + \overline{\mu}(z). \downarrow Y \) | Rec \( Y \). \( \overline{\delta''} + \overline{\mu}(\beta). ! \beta. \downarrow Y \) | env(\( \beta. [M \cdot \alpha] \)) | env(\( \delta. [\beta] \)) | env(x, [\( \lambda x. x \)] y) | env(\( \delta'. [\alpha] \)) | env(y, [M]) | env(\( \delta''. [\gamma] \)) | env(z, [y]) \rightarrow^\pi \) by \( \gamma \)
Rec \( Y \). \( \overline{\delta} + \overline{\mu}(\beta). ! \beta. \downarrow Y \) | env(\( \beta. [M \cdot \alpha] \)) | env(\( \delta. [\beta] \)) | env(x, [\( \lambda x. x \)] y) | env(\( \delta'. [\alpha] \)) | env(y, [M]) | env(\( \delta''. [\gamma] \)) | env(z, [y]) \rightarrow^\pi \) by \( \delta'' \)
Rec \( Y \). \( \overline{\gamma} \) | Rec \( Y \). \( \overline{\delta''} + \overline{\mu}(\beta). ! \beta. \downarrow Y \) | env(\( \beta. [M \cdot \alpha] \)) | env(\( \delta. [\beta] \)) | env(x, [\( \lambda x. x \)] y) | env(\( \delta'. [\alpha] \)) | env(y, [M]) | env(\( \delta''. [\gamma] \)) | env(z, [y]) \rightarrow^\pi \) by \( \gamma \)
Rec \( Y \). \( \overline{\gamma} \) | Rec \( Y \). \( \overline{\gamma} \) | env(\( \beta. [M \cdot \alpha] \)) | env(\( \delta. [\beta] \)) | env(x, [\( \lambda x. x \)] y) | env(\( \delta'. [\alpha] \)) | env(y, [M]) | env(\( \delta''. [\gamma] \)) | env(z, [y]) \rightarrow^\pi \) by \( \gamma \)
Rec \( Y \). \( \overline{\gamma} \) | Rec \( Y \). \( \overline{\gamma} \) | env(\( \beta. [M \cdot \alpha] \)) | env(\( \delta. [\beta] \)) | env(x, [\( \lambda x. x \)] y) | env(\( \delta'. [\alpha] \)) | env(y, [M]) | env(\( \delta''. [\gamma] \)) | env(z, [y]) \rightarrow^\pi \) by \( \gamma \)

Note that the final process is equivalent to \( \llbracket M \rrbracket \mid [\alpha] \) (by \( \equiv^\pi \)) since none of the other processes is accessible. The process \( \llbracket c \rrbracket \) reduces to \( \llbracket \lambda M \mid \alpha > \rrbracket \), as expected since \( c \) reduces to \( \lambda M \mid \alpha > \). In the step \( \pi \), and in the analogous steps, we could have chosen to perform a communication with channel \( \mu \) and capture directly the variable \( \delta' \). The proof in Section 2 ensures that the result would not change.

2 Correctness of \( \llbracket . \rrbracket \)

Our proof follows the same line of the proofs in 2 given by Milner. In what follows \( P \rightarrow^\pi \) means that there is no \( P' \) such that \( P \rightarrow^\pi P' \). We write \( \rightarrow^\pi \) for the transitive closure of \( \rightarrow^\pi \), \( \rightarrow^\pi^* \) for the transitive and reflexive closure of \( \rightarrow^\pi \) and \( P \downarrow^\pi P' \) means \( P \rightarrow^\pi^* P' \) and \( P' \rightarrow^\pi \). We use the notation \( c \rightarrow \) and \( c \downarrow c' \) for commands in the analogous way.

**Definition 2.1** (Relation R). Let \( R \) be a relation containing pairs \( (c, P) \), with \( c \) a command and \( P \) a process. Let \( \nu \) be a sequence of \( \lambda \mu \bar{\mu}\)-terms. Let \( \bar{x} \) be a sequence of term variables (of \( \lambda \mu \bar{\mu} \)). Let \( \bar{e} \) be a sequence of context variables. Let \( \bar{a} \) be a sequence of context variables.
Let \( R \) contain all pairs such that:

\[
c \equiv c' [\bar{\nu}/\bar{x}, \bar{e}/\bar{a}] \\
\]

\[
P \equiv^\pi \llbracket c' \rrbracket | \text{env} (\bar{x}, \bar{v}) | \text{env}(\bar{a}, \bar{e})
\]

Assume variables are progressively numbered, i.e. \( \bar{x} = x_1, x_2, x_3, \ldots x_k \), the length of \( \bar{x} \) and \( \bar{v} \) sequences is \( k \), the length of \( \bar{a} \) and \( \bar{e} \) sequences is \( m \). We write \( \text{env}(x, v) \) for the process \( !x. \llbracket v \rrbracket \) and we write \( \text{env}(\bar{x}, \bar{v}) \) for \( \text{env}(x_1, v_1) | \text{env}(x_2, v_2) | \ldots | \text{env}(x_k, v_k) \).

Moreover assume \( \text{FV}(c') \subset \bar{x} \cup \bar{a} \) and what follows:
Theorem 2.2 (correctness of the encoding). Let \((c, P) \in R:\)

- (a) if \(c \rightarrow \text{ then } P \downarrow^\pi P'\) with \( (c, P') \in R \).
- (b) \(c \rightarrow c' \text{ then } P \rightarrow^\pi P'\) with \( (c', P') \in R \).

Proof. Let \((c, P) \in R\). So \(c = c'[\bar{v}/\bar{x}, \bar{e}/\bar{a}]\). Let \(y\) and \(\beta\) be variables not appearing in \(\bar{x}\) nor in \(\bar{a}\), i.e. free in \(c\). The proof proceeds by case analysis on \(c'\) and cope both (a) and (b) since either \(c \rightarrow\) or \(c \rightarrow d\) for some \(d\). For sake of readability we omit the initial sequence of restrictions in writing \(\pi\)-processes.

We firstly focus on normal forms.

- \(c' = <y | \beta >\) (and \(c = <y | \beta > [\bar{v}/\bar{x}, \bar{e}/\bar{a}]\))
  \[ c \rightarrow \]
  \[ \llbracket c \rrbracket = \text{Rec } Y. \bar{y} + \mu (\alpha). !z. Y | \text{Rec } Y. \bar{\beta} + \mu (\alpha). !\alpha. Y | \text{env}(\bar{x}, \bar{v}) | \text{env}(\bar{a}, \bar{e}) \]
  \[ \llbracket c \rrbracket \rightarrow^\pi \text{ and } (c, \llbracket c \rrbracket) \in R \text{ by definition of } R. \]

- \(c' = <y \cdot v_2 | e >\) (and \(c = <y \cdot v_2 | e > [\bar{v}/\bar{x}, \bar{e}/\bar{a}]\))
  \[ c \rightarrow \]
  \[ \llbracket c \rrbracket = \text{Rec } Y. \bar{y} + \mu (\alpha). !z. Y | \text{Rec } Y. \bar{\lambda}(\delta). (\llbracket v_2 \rrbracket | !\delta. \llbracket e \rrbracket) + \mu (\alpha). !\alpha. Y | \text{env}(\bar{x}, \bar{v}) | \text{env}(\bar{a}, \bar{e}) \]
  \[ \llbracket c \rrbracket \rightarrow^\pi \text{ and } (c, \llbracket c \rrbracket) \in R \text{ by definition of } R. \]

- \(c' = <\lambda x. v | \beta >\) (and \(c = <\lambda x. v | \beta > [\bar{v}/\bar{x}, \bar{e}/\bar{a}]\))
  \[ c \rightarrow \]
  \[ \llbracket c \rrbracket = \text{Rec } Y. \bar{\lambda}(\delta). (\llbracket x \rrbracket \cdot v_1 | \delta >) + \mu (\alpha). !z. Y | \text{Rec } Y. \bar{\beta} + \mu (\alpha). !\alpha. Y | \text{env}(\bar{x}, \bar{v}) | \text{env}(\bar{a}, \bar{e}) \]
  \[ \llbracket c \rrbracket \rightarrow^\pi \text{ and } (c, \llbracket c \rrbracket) \in R \text{ by definition of } R. \]

Next we focus on the cases for which a reduction takes place.

- \(c' = <\lambda x. v_1 \cdot v_2 | e >\) (and \(c = <\lambda x. v_1 \cdot v_2 | e > [\bar{v}/\bar{x}, \bar{e}/\bar{a}]\))
  \[ c \rightarrow c'' = <v_2 \cdot v_2 \cdot v_1 | e > [\bar{v}/\bar{x}, \bar{e}/\bar{a}] \]
  \[ P = \text{Rec } Y. \bar{\lambda}(\delta). (\llbracket \mu \chi. v_1 | \delta >) + \mu (\alpha). !z. Y | \text{Rec } Y. \bar{\lambda}(\delta). (\llbracket v_2 \rrbracket | !\delta. \llbracket e \rrbracket) + \mu (\beta). !\beta. Y | \text{env}(\bar{x}, \bar{v}) | \text{env}(\bar{a}, \bar{e}) \]
  \[ P \rightarrow^\pi P' = \llbracket v_2 \rrbracket | \llbracket \mu \chi. v_1 | \delta > \rrbracket | \llbracket \mu \chi. v_1 | \delta > \rrbracket | \llbracket \mu \chi. v_1 | \delta > \rrbracket \]
  \[ (c'', P') \in R. \]
  Indeed \(c'' = <v_2 \cdot v_1 | \delta > [e/\delta][\bar{v}/\bar{x}, \bar{e}/\bar{a}]. \)

- \(c' = <\mu \beta. c | e >\) (and \(c = <\mu \beta. c | e > [\bar{v}/\bar{x}, \bar{e}/\bar{a}]\))
  (*) Assume \(e\) is not a \(\mu\)-abstraction nor a variable appearing in \(\bar{a}\), we treat those cases apart.
  \[ c \rightarrow c' = c [e/\beta][\bar{v}/\bar{x}, \bar{e}/\bar{a}] \]
  \[ P = \text{Rec } Y. \bar{\mu}(\beta). (\llbracket c \rrbracket + \mu (\alpha). !z. Y | \text{Rec } Y. Q + \mu (\beta). !\beta. Y | \text{env}(\bar{x}, \bar{v}) | \text{env}(\bar{a}, \bar{e}) \]
  \[ Q \text{ is the first operand of the main sum in } \llbracket e \rrbracket. \text{ Given (*)}, \text{ no matter what the form } e \text{ it does not perform an output along the channel } \bar{\mu} \text{ nor it can activate any process. The only possible reduction} \]
is the following:

\[ P \rightarrow^x P' = [e] | env(\beta, [e]) | env(\tilde{x}, \tilde{v}) | env(\tilde{a}, \tilde{v}) \]

\((c'', P') \in R\) by definition of \(R\).

• \(c' = \langle v | \mu x. c >\) (and \(c = \langle v | \mu x. c > [\tilde{v}/\tilde{x}, \tilde{e}/\tilde{a}]\))

(*) Assume \(v\) is not a \(\mu\)-abstraction nor a variable in \(\tilde{x}\), we treat those cases apart. The proof is analogous to the previous one.

• \(c' = \langle \mu \beta. c_1 | \mu x. c_2 >\) (and \(c = \langle \mu \beta. c_1 | \mu x. c_2 > [\tilde{v}/\tilde{x}, \tilde{e}/\tilde{a}]\))

\[ P = Rec\ Y. (\mu\beta). [e] + \mu(\tilde{x}). !z. Y | Rec\ Y. (\mu\beta). !z. Y | env(\tilde{x}, \tilde{v}) | env(\tilde{a}, \tilde{v}) \]

Two possible reductions: by cases

(1)

\[ c \rightarrow c'' = c_1[\mu x. c_2/\beta] [\tilde{v}/\tilde{x}, \tilde{e}/\tilde{a}] \]

\[ P \rightarrow^x P' = [c_1] | env(\mu x. c_2, \beta) | env(\tilde{x}, \tilde{v}) | env(\tilde{a}, \tilde{v}) \]

\((c'', P') \in R\) by definition of \(R\).

(2)

\[ c \rightarrow c'' = c_2[\mu \beta. c_1/x] [\tilde{v}/\tilde{x}, \tilde{e}/\tilde{a}] \]

\[ P \rightarrow^x P' = [c_2] | env(\mu \beta. c_1, x) | env(\tilde{x}, \tilde{v}) | env(\tilde{a}, \tilde{v}) \]

\((c'', P') \in R\) by definition of \(R\).

Next we focus on the cases where a bound variable is involved and an access to an environment entry may occur.

• \(c' = \langle x_i | e >\) (and \(c = \langle x_i | e > [\tilde{v}/\tilde{x}, \tilde{e}/\tilde{a}]\))

(*) Assume \(e\) is not a \(\mu\)-abstraction nor a variable in \(\tilde{a}\), we treat those cases apart.

Proof by complete induction on \(k - i\)

\[ c = \langle x_i | e > [\tilde{v}/\tilde{x}, \tilde{e}/\tilde{a}] \]

\[ P = Rec\ Y. \tilde{x} + \mu(\tilde{x}). !z. Y | Rec\ Y. Q + \mu(\tilde{x}). !z. Y | env(\tilde{x}, \tilde{v}) | env(\tilde{a}, \tilde{v}) \]

(recall that \(env(x_i, v_i) = ![x_i] [v_i]\))

\(Q\) is the first operand of the main sum in \([e]\). Given (*), no matter what the form \(e\) is it does not perform an output along the channel \(\mu\) nor it can activate any process. The only possible reduction is the following:

\[ P \rightarrow^x P' = \langle v_i | e > | env(\tilde{x}, \tilde{v}) | env(\tilde{a}, \tilde{v}) \]

If \(v_i\) is a variable appearing in \(\tilde{x}\), say \(x_j\), then \(j > i\) and \(k - j < k - i\), thus we apply the induction hypothesis. If \(v_i\) is not a variable appearing in \(\tilde{x}\) then its form matches another proved case of the proof.

• \(c' = \langle v | \alpha_j >\) (and \(c = \langle v | \alpha_j > [\tilde{v}/\tilde{x}, \tilde{e}/\tilde{a}]\))

Assume \(v\) is not a \(\mu\)-abstraction nor a variable in \(\tilde{x}\), we treat those cases apart. The proof is analogous to the previous one (the induction is on \(m - i\)).

• \(c' = \langle x_i | \alpha_j >\) (and \(c = \langle x_i | \alpha_j > [\tilde{v}/\tilde{x}, \tilde{e}/\tilde{a}]\))

The proof is a straightforward combination of the schema of the previous two cases.
• $c' = \langle \mu \beta. \text{c}_1 \mid a_i \rangle$ (and $c = \langle \mu \beta. \text{c}_1 \mid a_i \rangle \equiv [\bar{v}/\bar{x}, \bar{e}/\bar{a}]$)

$P = \text{Rec Y}. \bar{\mu}(\text{c}_1) + \mu(x).!x.Y \mid \text{Rec Y}. \bar{\alpha}i + \mu(\beta)\cdot!\beta.Y \mid \text{env}(\bar{x}, \bar{v}) \mid \text{env}(\bar{a}, \bar{e})$

Two possible reductions: by cases

1. $c \rightarrow c'' = c_1[a_i/\beta][\bar{v}/\bar{x}, \bar{e}/\bar{a}]$

$P \rightarrow^\pi P' = [c_1] \mid \text{env}(a_i, \beta) \mid \text{env}(\bar{x}, \bar{v}) \mid \text{env}(\bar{a}, \bar{e})$

$(c'', P') \in R$ by definition of $R$.

2. $P \rightarrow^\pi P' = [\mu \beta. \text{c}_1] \mid [\text{c}] \mid \text{env}(\bar{x}, \bar{v}) \mid \text{env}(\bar{a}, \bar{e})$

Proof by complete induction on $m - i$

If $e_i$ is a variable appearing in $\bar{a}$, say $\alpha_j$, then $j > i$ and $m - j < m - i$, thus we apply the induction hypothesis. If $e_i$ is not a variable appearing in $\bar{a}$ then its form matches another proved case of the proof.

• $c' = \langle x_i \mid \alpha y. \text{c} \rangle$ (and $c = \langle x_i \mid \alpha y. \text{c} \rangle \equiv [\bar{v}/\bar{x}, \bar{e}/\bar{a}]$)

The proof is analogous to the previous one (the induction is on $k - i$).

$\square$

The encoding intimately mimicks $\lambda \mu \bar{\alpha}$. Indeed as a consequence of Theorem 2.2 either one of the following conditions holds:

• $c \uparrow$ and $[c] \uparrow$

• $c \downarrow c_1$ and $[c] \downarrow P$ with $(c_1, P) \in R$

3 \textbf{ $[\cdot]$ is an operational full abstraction (correspondence $\pi$-M1) }

Let $\rho$ be an environment that binds the variables $\bar{y}$. Assume that $x$ and $\beta$ are not appearing in $\bar{y}$. In what follows $s \rightarrow^{M1}$ means that there is no $s'$ such that $s \rightarrow^{M1} s'$.

\textbf{Proof.} $\Rightarrow$

The proof proceeds by case analysis on the states of $M1$.

Normal forms:

• $\langle x \mid \beta \rangle \Downarrow^\rho \rightarrow^{M1}$

$P = [\langle x \mid \beta \rangle \Downarrow^\rho] = v\bar{y}.(\text{Rec Y}. \bar{x} + \bar{\mu}(z).!z.Y \mid \text{Rec Y}. \bar{\alpha}i + \mu(\alpha).!\alpha.Y \mid [\rho])$

$P \rightarrow^\pi$

• $\langle x \mid v_2 \cdot e \rangle \Downarrow^\rho \rightarrow^{M1}$

$P = [\langle x \mid v_2 \cdot e \rangle \Downarrow^\rho] = v\bar{y}.(\text{Rec Y}. \bar{x} + \bar{\mu}(z).!z.Y \mid \text{Rec Y}. \bar{\alpha}(\delta).([v_2] \mid !\delta. [e] + \mu(\beta)\cdot!\beta.Y \mid [\rho]))$

$P \rightarrow^\pi$

• $\langle \lambda x. v \mid \beta \rangle \Downarrow^\rho \rightarrow^{M1}$

$P = [\langle \lambda x. v \mid \beta \rangle \Downarrow^\rho] = v\bar{y}.(\text{Rec Y}. \lambda(\delta).\bar{\mu}x. <v_1 \mid \delta \rangle + \bar{\mu}(z).!z.Y \mid \text{Rec Y}. \bar{\alpha}i + \mu(\alpha).!\alpha.Y \mid [\rho])$

$P \rightarrow^\pi$

Reduction cases:
\[ <\mu\beta.c | e > in \rho \rightarrow_{M1} c[\beta' / \beta] in \rho + [\beta' = e] \]

\[ P = [\[ <\mu\beta.c | e > in \rho \rightarrow_{\beta} \mu Y.(Rec Y. \, \overline{\mu}(\beta).\, c + \overline{\mu} (x).! x. Y | Rec Y. Q + \mu(\beta).! \beta. Y | \rho) \] (a simple \alpha-conversion is sufficient)\]

\[ \bullet < v \mid \overline{\mu}.c > in \rho \rightarrow_{M1} c[x'/x] in \rho + [x' = v] \]

The proof is analogous to the previous one.

\[ \bullet < x | e > in \rho + [x = v] \rightarrow_{M1} < v | e > in \rho + [x = v] \]

\[ P = [\[ < x | e > in \rho + [x = v] \] = v x . v y . (Rec Y. x + \mu(x).! z . Y | e | ! x . (v | \rho)) \]

\[ P \rightarrow^* P' \equiv^* v x_1 \cdot v y_1. (e | v | e | \rho) \] and \[ P' \equiv^* \[ < v | e > in \rho + [x = v] \] \]

\[ \bullet < v | \beta > in \rho + [\beta = e] \rightarrow_{M1} < v | e > in \rho + [\beta = e] \]

The proof is analogous to the previous one.

\[ \bullet < x . v_1 | (v_1 . e) > in \rho \rightarrow_{M1} < v_2 | \overline{\mu}.< v_1 | \delta >> in \rho + [\delta = e] \]

\[ P = [\[ < x . v_1 | (v_1 . e) > in \rho \] \] = v y_1 . (Rec Y. \lambda(\delta). \[ \overline{\mu}.< v_1 | \delta >> + \overline{\mu} (z) .! z . Y | Rec Y. \, \lambda(\delta). (v_1 | ! \delta . (e | \rho)) \]

\[ P \rightarrow^* P' \equiv^* v \delta . v y_1 . (v_2 | \rho | e | e | \rho) \] and \[ P' \equiv^* \[ < v_2 | \overline{\mu}.< v_1 | \delta >> in \rho + [\delta = e] \] \]

\[ \square \]

**Proof.** \[ \Leftarrow \]

Since we reason by cases, the proof for \( \Rightarrow \) takes into account the same cases for \( \Leftarrow \), thus the deterministic cases are already proved (it is easy to check that the encodings are deterministic too). The same consideration holds for normal forms. What we need to check is whether the remaining transitions lead to what we expect in \( M1 \) for every possible \( \pi \) step.

The non deterministic cases are \[ [\[ < \mu\beta.c | \alpha > in \rho + [\alpha = e] \], [\[ < x | \mu y . c > in \rho + [x = v] \] ] \] and \[ [\[ < \mu\beta.c_1 | \mu \beta . c_2 > in \rho \] ] \]

\[ [s = < \mu\beta.c | \alpha > in \rho + [\alpha = e] \]

\[ P = [\[ s \] \] = v a . v y . (Rec Y. \, \overline{\mu}(\beta).\, c + \overline{\mu} (x).! x. Y | Rec Y. \, \overline{\alpha} + \mu(\beta).! \beta. Y | ! \alpha . (e | \rho) ] \]

Two reduction cases:

1. \[ P \rightarrow^* P' \equiv^* v \beta' . v a . v y_1. (c | \beta' / \beta) | ! \beta'. (\alpha | ! \alpha . (e | \rho)) \]

   \[ (\text{a simple } \alpha\text{-conversion}) \]

   \[ P' \equiv^* [c[\beta' / \beta] in \beta' + [\alpha = e] + \rho] \]

   \[ \text{indeed } s \rightarrow_{M1} c[\beta' / \beta] \text{ in } [\beta' = \alpha] + [\alpha = e] + \rho \]

2. \[ P \rightarrow^* P' = v a . v y. (Rec Y. \, \overline{\mu}(\beta).\, c + \overline{\mu} (x).! x. Y | e | \rho) \]

   \[ P' \equiv^* [< \mu\beta.c | e > in [\alpha = e] + \rho] \]

   \[ \text{indeed } s \rightarrow_{M1} [\mu\beta.c \mid e > \text{ in } [\alpha = e] + \rho \]

\[ \bullet s = < x | \mu y . c > in \rho + [x = v] \]

The proof is analogous to the previous one.

\[ \bullet s = < \mu\beta.c_1 | \mu \beta . c_2 > in \rho \]

\[ P = [\[ s \] \] = v y . (Rec Y. \, \overline{\mu}(\beta).\, c + \overline{\mu} (z) .! z . Y | Rec Y. \, \overline{\alpha} (x) . c + \mu(\alpha).! \alpha.Y | e | \rho) ] \]
Two reduction cases:
(1) \( P \xrightarrow{\pi} P' \equiv^\pi v\beta', v\check{y}.(\llbracket c_1 \rrbracket | \beta', \llbracket \check{x}.c_2 \rrbracket | \llbracket \rho \rrbracket) \)
(a simple \( \alpha \)-conversion)
\( P' \equiv^\pi \llbracket c_1[\beta'/\beta] \rrbracket in \llbracket \beta' = \check{x}.c_2 \rrbracket + \rho \)
indeed \( s \xrightarrow{M_1} S_1 \llbracket c_1[\beta'/\beta] \rrbracket in \llbracket \beta' = \check{x}.c_2 \rrbracket + \rho \)
(2) \( P \xrightarrow{\pi} P' \equiv^\pi v\check{x}' . v\check{y}.(\llbracket c_2 \rrbracket | \check{x}' . \llbracket \mu\beta.c_1 \rrbracket | \llbracket \rho \rrbracket) \)
(a simple \( \alpha \)-conversion)
\( P' \equiv^\pi \llbracket c_2[\check{x}'/x] \rrbracket in \llbracket \check{x}' = \mu\beta.c_1 \rrbracket + \rho \)
indeed \( s \xrightarrow{M_1} S_1 \llbracket c_2[\check{x}'/x] \rrbracket in \llbracket \check{x}' = \mu\beta.c_1 \rrbracket + \rho \)
\( \square \)

4 \( \llbracket \cdot \rrbracket^M \) is an operational full abstraction (correspondence \( M_1-M_2 \))

Before embarking in the proof, we define explicitly the mapping \( \llbracket \cdot \rrbracket^M \) from states of \( M_2 \) to states of \( M_1 \). Intuitively, \( \llbracket \cdot \rrbracket^M \) maps the explicit substitutions of \( M_2 \) into the global environment of \( M_1 \), caring that name clashes are avoided.

\[
\llbracket e \in \rho_1 | e \in \rho_2 \rrbracket^M = \llbracket e \rrbracket^M + \llbracket \rho_2 \rrbracket^M \\
\llbracket \rho \rrbracket^M = \sum [\text{var}_i = t_i] + \sum [\text{phi}_i] \\
\text{with } \rho = [\text{var}_1 = t_1 \text{ in } \phi_1, \text{var}_2 = t_2 \text{ in } \phi_2 \ldots \text{var}_n = t_n \text{ in } \phi_n]
\]

In what follows \( s \xrightarrow{M_1} s' \) means that there is no \( s'' \) such that \( s \xrightarrow{M_1} s'' \). We write \( q \xrightarrow{M_2} \) analogously.

**Proof:** \( \Rightarrow \)
The proof proceeds by case analysis on the states of \( M_2 \). Assume the variables \( x \) and \( \beta \) not bound by their environments when they occur.

Normal forms:

- \( < x in \rho_1 | \beta in \rho_2 > \xrightarrow{M_2} \)
  \( s = \llbracket < x in \rho_1 | \beta in \rho_2 \rrbracket^M = \llbracket x \rrbracket^M + \llbracket \rho_2 \rrbracket^M \xrightarrow{M_1} \)

- \( < x in \rho_1 | (v_2 \cdot e) in \rho_2 > \xrightarrow{M_2} \)
  \( s = \llbracket < x in \rho_1 | (v_2 \cdot e) in \rho_2 \rrbracket^M = \llbracket x \rrbracket^M + \llbracket \rho_2 \rrbracket^M \xrightarrow{M_1} \)

- \( < \lambda x. v in \rho_1 | \beta in \rho_2 > \xrightarrow{M_2} \)
  \( s = \llbracket < \lambda x. v in \rho_1 | \beta in \rho_2 \rrbracket^M = \llbracket \lambda x. v \rrbracket^M + \llbracket \rho_2 \rrbracket^M \xrightarrow{M_1} \)

Reduction cases:

- \( < \mu \beta.c in \rho_1 | e in \rho_2 > \xrightarrow{M_2} \)
  \( c in \rho_1 + [\beta = e in \rho_2] \equiv^M c[\beta'/\beta] in \rho_1 + [\beta' = e in \rho_2] \)
  \( \llbracket < \mu \beta.c in \rho_1 | e in \rho_2 \rrbracket^M = \llbracket \mu \beta.c \rrbracket^M + \llbracket \rho_2 \rrbracket^M \xrightarrow{M_1} \)
  \( c[\beta'/\beta] in \llbracket \rho_1 \rrbracket^M + \llbracket \rho_2 \rrbracket^M + [\beta' = e] \equiv^M \llbracket c[\beta'/\beta] in \rho_1 + [\beta' = e in \rho_2] \rrbracket^M \)
• $<v \in \rho_1 \ | \ \bar{\mu}x. c \ in \ \rho_2 > \longrightarrow^{M2} \ c \ in \ \rho_2 + [x = v \ in \ \rho_1]$

The proof is analogous to the previous one.

• $<x \ in \ \rho_1 + [x = v \ in \ \rho_2] \ | \ e \ in \ \rho_3 > \longrightarrow^{M2} <v \ in \ \rho_2 \ | \ e \ in \ \rho_3 >$

$[<x \ in \ \rho_1 + [x = v \ in \ \rho_2] \ | \ e \ in \ \rho_3 >]^M = <x \ | \ e > in \ \langle \rho_1 \rangle^M + \langle \rho_2 \rangle^M + [x = v] + \langle \rho_3 \rangle^M \longrightarrow^{M1} <v \ in \ \rho_2 \ | \ e \ in \ \rho_3 >^M$

Indeed $\rho_1$ and $[x = v]$ are no longer accessible.

• $<v \ in \ \rho_1 \ | \ \beta \ in \ \rho_2 + [\beta = e \ in \ \rho_3] > \longrightarrow^{M2} <v \ in \ \rho_1 \ | \ e \ in \ \rho_3 >$

The proof is analogous to the previous one.

Since we reason by cases, the proof for $\Rightarrow$ takes into account the same cases for $\Leftarrow$, thus the deterministic cases are already proved (it is easy to check that the encodings are deterministic too). The same consideration holds for normal forms. What we need to check is whether the remaining transitions lead to what we expect in $M2$ for every possible $M1$ step.

The non deterministic cases are $[<\bar{\mu} \beta. c \ in \ \rho_1 \ | \ \alpha \ in \ [\alpha = e \ in \ \rho_2] + \rho_3 >]^M$, $<[x \ in \ [x = v \ in \ \rho_1] + \rho_2] \ | \ \bar{\mu} \ y. c \ in \ \rho_3 >]^M$, $<[\bar{\mu} \beta. c_1 \ in \ \rho_1 \ | \ \bar{\mu} \ y. c_2 \ in \ \rho_2 ] >]^M$.

• $s = <\bar{\mu} \beta. c \ in \ \rho_1 \ | \ \alpha \ in \ [\alpha = e \ in \ \rho_2] + \rho_3 >$

$q = [s]^M = <\bar{\mu} \beta. c \ | \ \alpha > in \ \langle \rho_1 \rangle^M + \langle \rho_2 \rangle^M + [\alpha = e] + \langle \rho_3 \rangle^M$

Two reduction cases:

1. $q \longrightarrow^{M1} q' = c[\beta / \beta] \ in \ [\beta' = \alpha] + \langle \rho_1 \rangle^M + \langle \rho_2 \rangle^M + [\alpha = e] + \langle \rho_3 \rangle^M$

indeed $s \longrightarrow^{M2} \equiv^{M2} c[\beta' / \beta] \ in \ [\beta' = \alpha \ in \ [\alpha = e \ in \ \rho_2] + \rho_3]$

• $s = <x \ in \ [x = v \ in \ \rho_1] + \rho_2 \ | \ \bar{\mu} \ y. c \ in \ \rho_3 >$

The proof is analogous to the previous one.

• $s = <\bar{\mu} \beta. c_1 \ in \ \rho_1 \ | \ \bar{\mu} \ y. c_2 \ in \ \rho_2 >$

$q = [s]^M = <\bar{\mu} \beta. c \ | \ \bar{\mu} \ y. c > in \ \langle \rho_1 \rangle^M + \langle \rho_2 \rangle^M$

Two reduction cases:

1. $q \longrightarrow^{M1} q' = c[\beta' / \beta] \ in \ [\beta' = \bar{\mu} \ y. c_2] + \langle \rho_1 \rangle^M + \langle \rho_2 \rangle^M$

indeed $s \longrightarrow^{M2} \equiv^{M2} c[\beta' / \beta] \ in \ [\beta' = \bar{\mu} \ y. c_2 \ in \ \rho_2] + \rho_1$

(2) $q \longrightarrow^{M1} q' = c_2[x' / x] \ in \ [x' = \bar{\mu} \beta. c_1] + \langle \rho_1 \rangle^M + \langle \rho_2 \rangle^M$

$q' = [c_2[x' / x]] in \ [x' = \bar{\mu} \beta. c_1 \ in \ \rho_1] + \rho_2^M$
indeed $s \rightarrow^{M2 \equiv M2} c_2[x'/x] \ in \ [x' = \mu \beta. c_1] \ in \ \rho_1] + \rho_2$

References
