E Tests

In this section we discuss some test that we executed with the SML/NJ implementation of the algorithm [1]. The webpage also contains the output provided by the compiler when we ran the examples that we discuss below.

To improve readability of the output provided by the algorithm, in this document we omit type decorations and describe the input and output of the algorithm as π-calculus processes. In input, we use number subscripts to indicate the occurrences of a same channel that are described by distinct end points in the pair type, e.g. $a\colon (?\mathsf{bool}.\mathsf{end}, !\mathsf{bool}.\mathsf{end}) \vdash \overline{a_1}\langle \mathsf{true}\rangle.a_1(x)$. The numbers are preserved in the translation, and used to indicate the correspondence with the freshly generated forwarders (r) and semaphores (m), e.g. if $(\overline{a_1}\langle \mathsf{true}\rangle.P' \mid (\nu r_1)P'')$ is the output obtained by nested calls of $[\![\cdot]\!]$, then r_1 is the forwarder associated to a_1. We also use Q as a label for a non-specified process[6] , and base for base values (e.g. booleans, strings), $\top$ for type $(\mathsf{end}, \mathsf{end})$, and $\bot$ for type $(\mathsf{start}, \mathsf{start})$.

Example 1. The input process is

$$P_1 \stackrel{\text{def}}{=} \overline{a_1}\langle v\rangle.a_1(y).(a_2(j) \mid \overline{a_2}\langle w\rangle.Q)$$

and is typed by $\Gamma_1 \stackrel{\text{def}}{=} a\colon (?\top.?\mathsf{base}.\mathsf{end}, !\top.!\mathsf{base}.\mathsf{end}), v\colon \top, w\colon \mathsf{base}$. The output process is

$$(\nu r_1)(\overline{a_1}\langle v\rangle.(\nu r_2)(\nu m_2)(a_2(z).\overline{r_2}\langle z\rangle \mid r_1(y).r_2(j).\overline{m_2}\langle\rangle.Q \mid \overline{a_2}\langle w\rangle.m_2()) \mid a_1(z).\overline{r_1}\langle z\rangle)$$

The output context and projections are $\Gamma' \stackrel{\text{def}}{=} a\colon \top, v\colon \top, w\colon \mathsf{base}$, $\Delta' \stackrel{\text{def}}{=} a\colon \bot, v\colon \bot, w\colon \bot$.

Example 2. The input process is

$$P_2 \stackrel{\text{def}}{=} \overline{a_1}\langle v\rangle.a_1(y).\overline{a_2}\langle w\rangle.a_2(j).(\overline{ct}\langle y\rangle.Q \mid c(z).Q)$$

and is typed by $\Gamma_1, ct\colon (?\top.\mathsf{end}, !\top.\mathsf{end})$, where Γ_1 is defined in Example 1. The output process is

$$(\nu r_1)(\overline{a_1}\langle v\rangle.r_1(y).(\nu r_2)(\nu m_2)(a_2(z).\overline{r_2}\langle z\rangle \mid r_2(j).\overline{m_2}\langle\overline{ct}\langle y\rangle.Q \mid c(z).Q\rangle. \mid \overline{a_2}\langle w\rangle.m_2()) \mid a_1(z).\overline{r_1}\langle z\rangle)$$

The output context and projections are $\Gamma'' \stackrel{\text{def}}{=} a\colon \top, v\colon \top, w\colon \mathsf{base}, ct\colon \top$, $\Delta'' \stackrel{\text{def}}{=} a\colon \bot, v\colon \bot, w\colon \bot, ct\colon \bot$.

[6] In tests, Q is implemented by $!\mathbf{0}$

Example 3. The input process is

$$P_3 \stackrel{\text{def}}{=} \overline{a}\langle v\rangle.b(y).\overline{b}\langle w\rangle.a(j).Q$$

and is typed by $\Gamma_3 \stackrel{\text{def}}{=} a\colon (?\top.\mathsf{end}, !\top.\mathsf{end}), b\colon (?\top.\mathsf{end}, !\top.\mathsf{end}),\ v\colon \top, w\colon \bot$. The output process is

$$(\nu r_a)(\overline{a}\langle v\rangle.(\nu r_b)(\nu m_b)(b(u).\overline{r_b}\langle u\rangle \mid r_b(y).\overline{m_b}\langle\rangle.r_a(j).Q \mid \overline{b}\langle w\rangle.m_b()) \mid a(z).\overline{r_a}\langle z\rangle)$$

The output context and projections are $\Gamma''' \stackrel{\text{def}}{=} a\colon \top, b\colon \top, v\colon \top, w\colon \mathsf{base}$, $\Delta''' \stackrel{\text{def}}{=} a\colon \bot,\ b\colon \bot, v\colon \bot, w\colon \bot$.

Example 4. The input process is

$$a(y).\overline{y}\langle w\rangle.y(z).\overline{a}\langle v\rangle.Q$$

and is typed by $\Gamma_4 \stackrel{\text{def}}{=} a\colon (?\top.\mathsf{end}, !\top.\mathsf{end}),\ v\colon \top, w\colon \bot$. The output process is

$$(\nu r_a)(\nu m_a)(a(z).\overline{r_a}\langle z\rangle \mid r_a(y).(\nu r_y)(\overline{y}\langle w\rangle.r_y(z).\overline{m_a}\langle\rangle.Q \mid y(z).\overline{r_y}\langle z\rangle) \mid \overline{a}\langle v\rangle.m_a())$$

The output context and projections are $\Gamma'''' \stackrel{\text{def}}{=} a\colon \top, v\colon \top, w\colon \mathsf{base}$, $\Delta'''' \stackrel{\text{def}}{=} a\colon \bot,\ v\colon \bot, w\colon \bot$.