Undecidability of BPP Equivalences Revisited

Naoki Kobayashi   Takashi Suto
Tohoku University

April 23, 2007

Abstract

Hüttel [3] gave a proof of the undecidability of BPP equivalences. In this note, we point out some of the flaws in his proof, and provide a new, actually simpler proof.

1 Flaws in Hüttel's Proof

Hüttel [3] proves that all BPP equivalences between ready simulation equivalence and trace equivalence are undecidable by encoding a Minsky machine M into two BPP processes L and $\overline{L}$, so that (i) If M halts, then the trace set of L is *not* a subset of the trace set of $\overline{L}$, and (ii) If M does *not* halt, then $\overline{L}$ ready-simulates L. The proof contains the following major flaws.

1. The proof of Theorem 6 (the main theorem) says:
"If M does not halt, $L_k = L_{stop}$ is unreachable, and $R = \cdots$ is seen to be a ready-simulation."
This is wrong, because L may reach L_{stop} by invalid transitions. Therefore, the definition of the relation R must be modified, at least to include a pair having L_{stop} on the left-hand side. Moreover, the first pair:
$$(L_j \,|\, C_i^m \,|\, C_{i+1}^n \,|\, L_{c0} \,|\, L_{c1}, \overline{L_j} \,|\, \overline{C_i^m} \,|\, \overline{C_{i+1}^n} \,|\, \overline{L_{c0}} \,|\, \overline{L_{c1}})$$
does not satisfy the conditions required for R to be a ready-simulation. For example, if j is a type 1 instruction, the lefthand side can make the following transition:
$$L_j \,|\, C_i^m \,|\, C_{i+1}^n \,|\, L_{c0} \,|\, L_{c1} \xrightarrow{l''_{2i}} l''_{2i}G \,|\, C_i^m \,|\, C_{i+1}^n \,|\, L_{c0} \,|\, L_{c1}.$$
It should be matched by:
$$\overline{L_j} \,|\, \overline{C_i^m} \,|\, \overline{C_{i+1}^n} \,|\, \overline{L_{c0}} \,|\, \overline{L_{c1}} \xrightarrow{l''_{2i}} \overline{H_S} \,|\, \overline{C_i^m} \,|\, \overline{C_{i+1}^n} \,|\, \overline{L_{c0}} \,|\, \overline{L_{c1}}$$
for some $S \subseteq Act_2$. However, l'''_{2i} is in the ready set of $\overline{H_S} \,|\, \overline{C_i^m} \,|\, \overline{C_{i+1}^n} \,|\, \overline{L_{c0}} \,|\, \overline{L_{c1}}$ but not in that of $l''_{2i}G \,|\, C_i^m \,|\, C_{i+1}^n \,|\, L_{c0} \,|\, L_{c1}$.

2. The proof of Corollary 8 (which deduces the undecidability of equivalence relations from the undecidability of preorders) says:
   "As any equivalence $\sqsubseteq'$ from ready simulation downwards in the hierarchy of Figure 1 is ... and satisfies
   $$E \sqsubseteq' F \mathit{iff} E \leftrightharpoons' E + F$$
   where $\leftrightharpoons'$ is the equivalence closure of $\sqsubseteq'$"
   This does not hold in general. For example, let $\sqsubseteq'$ be the ready simulation, and $E = a$ and $F = 0$. Then, $E \leftrightharpoons' E + F$ but $E \not\sqsubseteq' F$.

The second flaw is not so hard to fix. On the other hand, we worked hard to fix the first flaw, but failed. We change (or actually *simplify*) the encoding of Minsky machines and give a new proof using the simplified encoding. In the rest of this note, we first introduce the syntax and semantics of BPP processes and Minsky machines in Sections 2 and 3. We then prove that all the equivalences between trace equivalence and ready simulation equivalence are undecidable, using the new encoding.

2 Basic Parallel Processes (BPP)

BPP [1] is a calculus of processes consisting of prefixes, parallel composition, internal choice, and recursion. Unlike in CCS [4], there is no synchronization mechanism (such as the transition $a.P \,|\, \overline{a}.Q \xrightarrow{\tau} P \,|\, Q$).

2.1 Syntax

The syntax of processes is given by:

$$P ::= 0 \mid X \mid lP \mid (P \,|\, Q) \mid P + Q \mid \mu X.P$$

Here, X and l are meta-variables ranging over the sets of *process variables* and *action labels* respectively. We write **Act** for the set of action labels, and write $\mathbf{BPP}_{\mathbf{Act}}$ for the set of BPP processes whose action labels are in the set **Act**.

The process 0 does nothing. A process lP first performs l and then behaves like P. $P \,|\, Q$ is a parallel composition of P and Q, and $P + Q$ is an internal choice of P or Q. $\mu X.P$ stands for the recursive process X usually defined by the equation $X = P$.

We often omit 0 and just write a for $a0$. We give a higher-precedence to unary prefixes, +, and | in this order, so that $l_1P_1 \,|\, l_2P_2 + l_3P_3$ means $(l_1P_1) \,|\, ((l_2P_2) + (l_3P_3))$.

We say that P is guarded by l in lP. A recursive process $\mu X.P$ is *guarded* if X appears only in guarded positions of P. In the rest of this paper, we consider only closed processes whose recursive processes are all guarded.

In this paper, we identify BPP processes up to the monoid laws: the associativity and commutativity of |, and the law $P \,|\, 0 = P$. For example, $(a \,|\, b) \,|\, b$ and $b \,|\, (b \,|\, a)$ are considered identical.

$$\frac{}{lP \xrightarrow{l} P} \quad (\textsc{Tr-Act}) \qquad \frac{[\mu X.P/X]P \xrightarrow{l} Q}{\mu X.P \xrightarrow{l} Q} \quad (\textsc{Tr-Rec})$$

$$\frac{P \xrightarrow{l} P'}{P \,|\, Q \xrightarrow{l} P' \,|\, Q} \quad (\textsc{Tr-ParL}) \qquad \frac{Q \xrightarrow{l} Q'}{P \,|\, Q \xrightarrow{l} P \,|\, Q'} \quad (\textsc{Tr-ParR})$$

$$\frac{P \xrightarrow{l} P'}{P + Q \xrightarrow{l} P'} \quad (\textsc{Tr-OrL}) \qquad \frac{Q \xrightarrow{l} Q'}{P + Q \xrightarrow{l} Q'} \quad (\textsc{Tr-OrR})$$

Figure 1: Transition rules of BPP processes

2.2 Operational semantics

The transition relation $P \xrightarrow{l} Q$ is the least relation closed under the rules in Figure 1.

2.3 Traces, simulations, and ready simulations

Definition 2.1 [trace preorder]: The set of *traces* of P, written $\mathit{trace}(P)$, is $\{a_1 a_2 \cdots a_n \mid P \xrightarrow{a_1} P_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} P_n\}$. The trace preorder $\leq_{tr}$ is defined by: $P \leq_{tr} Q$ iff $\mathit{trace}(P) \subseteq \mathit{trace}(Q)$.

The simulation preorder is defined by:

Definition 2.2 [simulation]: A binary relation $\mathcal{R}$ on BPP processes is a *simulation* if:
$\forall P, Q, l.((P\mathcal{R}Q \wedge P \xrightarrow{l} P') \Longrightarrow \exists Q'.(Q \xrightarrow{l} Q' \wedge P'\mathcal{R}Q'))$.
The simulation preorder $\leq_{sim}$ is the union of all simulations, i.e., $P \leq_{sim} Q$ if and only if there exists a simulation $\mathcal{R}$ such $P\mathcal{R}Q$. We write $P =_{sim} Q$ if $P \leq_{sim} Q \wedge Q \leq_{sim} P$.

Definition 2.3 [ready simulation]: A binary relation $\mathcal{R}$ on BPP processes is a *ready simulation* iff whenever $P\mathcal{R}Q$,

- If $P \xrightarrow{a} P'$ then there exists an Q' such that $Q \xrightarrow{a} Q'$ and $P'\mathcal{R}Q'$.
- $\mathit{readies}(P) = \mathit{readies}(Q)$

Here, $\mathit{readies}(P)$ denotes the *ready set* $\{a \mid \exists P'.P \xrightarrow{a} P'\}$.

$P \leq_{ready} Q$ if and only if there exists a ready simulation $\mathcal{R}$ such that $P\mathcal{R}Q$. We write $P =_{ready} Q$ if $P \leq_{ready} Q \wedge Q \leq_{ready} P$.

3 Minsky machine

A Minsky machine [5] has two counters, and consists of a sequence of the following instructions:

- Type 1 instruction: $c_i := c_i + 1; \textbf{goto } k$
- Type 2 instruction: $\textbf{if } c_i = 0 \textbf{ then goto } k_1 \textbf{ else } c_i := c_i - 1; \textbf{goto } k_2$
- Halt instruction

Formally, Minsky machines can be defined as follows.

Definition 3.1: A Minsky machine M is a mapping from a finite set $\mathbf{Addr}_M$ of natural numbers to the set of instructions:

$$\{\mathbf{Inc}_{i,k} \mid i \in \{0,1\}, k \in \mathbf{Addr} \cup \{\bot\}\} \cup \{\mathbf{Cond}_{i,k_1,k_2} \mid i \in \{0,1\}, k_1, k_2 \in \mathbf{Addr} \cup \{\bot\}\}.$$

A *state* of a Minsky machine is a triple $\langle k, m_0, m_1 \rangle$. The *initial state* σ_I of a Minsky machine is $\langle 0, 0, 0 \rangle$. The transition relation $\xrightarrow{l}_M$ (where $l \in \{\mathit{inc}_i, \mathit{then}_i, \mathit{else}_i \mid i \in \{0,1\}\}$) is defined by:

$$\frac{M(k) = \mathbf{Inc}_{i,k'} \qquad m'_i = m_i + 1 \qquad m'_{1-i} = m_{1-i}}{\langle k, m_0, m_1 \rangle \xrightarrow{\mathit{inc}_i}_M \langle k', m'_0, m'_1 \rangle}$$

$$\frac{M(k) = \mathbf{Cond}_{i,k_1,k_2} \qquad m_i = 0}{\langle k, m_0, m_1 \rangle \xrightarrow{\mathit{then}_i}_M \langle k_1, m_0, m_1 \rangle}$$

$$\frac{M(k) = \mathbf{Cond}_{i,k_1,k_2} \qquad m_i > 0 \qquad m'_i = m_i - 1 \qquad m'_{1-i} = m_{1-i}}{\langle k, m_0, m_1 \rangle \xrightarrow{\mathit{else}_i}_M \langle k_2, m'_0, m'_1 \rangle}$$

We write $\sigma \overset{l_1 \cdots l_n}{\Longrightarrow}_M \sigma'$ if $\sigma \xrightarrow{l_1}_M \cdots \xrightarrow{l_n}_M \sigma'$ for some $l_1, \ldots, l_n$. We also write $\sigma \Longrightarrow_M \sigma'$ if $\sigma \Longrightarrow_M \sigma'$ for some s. A Minsky machine M *halts* if there is a reduction sequence $\sigma_I \Longrightarrow_M \langle \bot, m_1, m_2 \rangle$ for some m_1, m_2.

The halting problem of Minsky machines is known to be undecidable [5]. In the rest of this paper, we omit M and write $\xrightarrow{l}$ and $\overset{s}{\Longrightarrow}$ for $\xrightarrow{l}_M$ and $\overset{s}{\Longrightarrow}_M$.

4 Undecidability of BPP Process Equivalences

In this section, we show that all the preorders between the trace preorder and the ready simulation preorder on BPP (i.e., all the preorders $\leq$ such that $\leq_{\mathit{ready}} \subseteq \leq \subseteq \leq_{\mathit{tr}}$) are undecidable. Those undecidability results have been already claimed by Hüttel [3], but we believe that our proof is simpler and more convincing (in fact, his proof seems to contain some flaws, which we found hard to fix).

The idea of our proof is the same as that of Hüttel [3]: Given the initial state σ_I of a Minsky machine M, we prepare two BPP processes $[\![M]\!]_L$ and $[\![M]\!]_R$, so that:

- If M halts, then $[\![M]\!]_L \not\leq_{\mathit{tr}} [\![M]\!]_R$.

- If M does not halt, $[\![M]\!]_L \leq_{ready} [\![M]\!]_R$.

Then, it is easy to deduce that, for any preorder $\leq$ such that $\leq_{ready} \subseteq \leq \subseteq \leq_{tr}$, a Minsky machine M halts if and only if $[\![M]\!]_L \leq [\![M]\!]_R$ holds. Since it is undecidable whether M halts, the preorder $\leq$ is also undecidable.

The basic idea of encodings is the same as that of Hirshfeld [2] used for proving the undecidability of the trace preorder (except for some new tricks to equalize the ready sets of $[\![M]\!]_L$ and $[\![M]\!]_R$). A state $\langle j, m_0, m_1\rangle$ of a Minsky machine is represented by processes of the form $L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1}$ and $\overline{L}_j \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$, where P^m denotes the parallel composition of m copies of P. Basically, L_j and $\overline{L}_j$ are defined so that $\overline{L}_j$ can do more actions than L_j, except the case $j = \bot$: $L_\bot$ can do a special action w, while $\overline{L}_\bot$ cannot. Thus, $\overline{L}_j$ can simulate the behavior of L_j as long as the Minsky machine does not halt.

We first give the definition of $[\![M]\!]_L$:

Definition 4.1: Let M be a Minsky machine. $[\![M]\!]_L$ is defined by:

$$\begin{array}{rcll}
[\![M]\!]_L & = & L_0 & \\
C_i & = & l_{d_i} & \\
L_j & = & \left\{\begin{array}{ll} l_I(C_i \,|\, L_k) + U & \text{if } M(j) = \mathbf{Inc}_{i,k} \\ l_{\langle_i} T_{i,k_1} + l_{[} E_{i,k_2} + U & \text{if } M(j) = \mathbf{Cond}_{i,k_1,k_2} \\ wU + U & \text{if } j = \bot \end{array}\right. & \\
T_{i,k} & = & l_{\rangle_i} L_k + U & E_{i,k} = l_{]_i} L_k + U \\
U & = & \displaystyle\sum_{a \in \mathbf{Act}_0} aU' & U' = \displaystyle\sum_{\mathbf{Act}_1} aU'
\end{array}$$

Here, $\mathbf{Act}_0 = \{l_I, l_{d_0}, l_{d_1}, l_{\langle_0}, l_{\langle_1}, l_{\rangle_0}, l_{\rangle_1}, l_{[}, l_{]_0}, l_{]_1}, w\}$ and $\mathbf{Act}_1 = \mathbf{Act}_0 \setminus \{w\} \cup \{l_\star\}$.

Intuitively, a transition of label inc_i of a Minsky machine is simulated by the action l_I. A transition of label $then_i$ is simulated by the action sequence $l_{\langle_i} l_{\rangle_i}$, and a transition of label $else_i$ is simulated by $l_{[} l_{d_i} l_{]_i}$. If $M(j) = \mathbf{Inc}_{i,k}$, L_j performs the action l_I and increments the value of the counter i (by spawning a fresh copy of C_i). The role of the process U is to force the ready set to be always $\mathbf{Act}_0$. In simulating the Minsky machine, U is not intended to be used for simulating the Minsky machine; if U is used, the special action $l_\star$ would be put into the ready set.

If $M(j) = \mathbf{Cond}_{i,k_1,k_2}$, then L_j executes the then-branch and the else-branch in a *non-deterministic* manner. Thus, $[\![M]\!]_L$ may execute invalid instructions that the machine M cannot execute. Such invalid transitions are simulated by the special process G of $[\![M]\!]_R$ given below.

Definition 4.2: Let M be a Minsky machine. $[\![M]\!]_R$ is defined by:

$$\begin{array}{lcll}
[\![M]\!]_R & = & \overline{L}_0 & \\
\overline{C}_i & = & l_{]_i} + l_{\langle_i} & \\
\overline{L}_j & = & \begin{cases} l_I(\overline{C}_i \,|\, \overline{L}_k) + \displaystyle\sum_{i\in\{0,1\}} l_{d_i}G + U & \text{if } M(j) = \mathbf{Inc}_{i,k} \\ l_{\langle_i}\overline{T}_{i,k_1} + l_{[}\overline{E}_{i,k_2} + \displaystyle\sum_{i\in\{0,1\}} l_{d_i}G + l_{\rangle_i}G + U & \text{if } M(j) = \mathbf{Cond}_{i,k_1,k_2} \\ 0 & \text{if } j = \bot \end{cases} & \\
\overline{T}_{i,k} & = & l_{\rangle_i}\overline{L}_k + l_{d_{1-i}}G + U & \\
\overline{E}_{i,k} & = & l_{d_i}\overline{L}_k + l_{d_{1-i}}G + l_{]_i}G + U & \\
G & = & \displaystyle\sum_{a\in\mathbf{Act}_0} aG + U &
\end{array}$$

The main differences of $[\![M]\!]_R$ from $[\![M]\!]_L$ are as follows.

- $\overline{L}_\bot$ cannot do any action, so that the process $\overline{L}_\bot \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ (which corresponds to the halting state of the Minsky machine) fails to simulate $L_\bot \,|\, C_0^{m_0} \,|\, C_1^{m_1}$.
- A process of the form lG is added to $\overline{L}_j$, where G is a kind of universal process that can simulate any behavior. G is used to simulate transitions of L_j that are invalid with respect to the Minsky machine. For example, in the state $L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1}$ where $M(j) = \mathbf{Inc}_{i,k}$, the only valid action corresponding to an action of the Minsky machine is l_I, but $L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1}$ can also perform an action l_{d_i} if $m_i > 0$. Such an action is simulated by the sub-process $l_{d_i}G$ of $\overline{L}_j$.
- The roles of $l_{]_i}$ and l_{d_i} in $\overline{C}_i$ and $E_{i,k}$ have been swapped: This is to take care of the case where $L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1}$ performs invalid transitions $l_{[}l_{]_i}$ (recall that the valid transition sequence is $l_{[}l_{d_i}l_{]_i}$). In that case, $\overline{L}_j \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ can simulate the transitions by $\overline{L}_j \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1} \xrightarrow{l_{[}} \overline{E}_{j,k} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1} \xrightarrow{l_{]_i}} G \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$.
- $\overline{C}_i$ can also perform $l_{\langle_i}$ action. This is to take care of the case where $M(j)$ is $\mathbf{Cond}_{i,k_1,k_2}$, $m_i > 0$ but $L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1}$ performs the actions $l_{\langle_i}l_{\rangle_i}$. That invalid transition sequence can be simulated by $\overline{L}_j \,|\, \overline{C}_i^{m_i} \,|\, \overline{C}_{1-i}^{m_{1-i}} \xrightarrow{l_{\langle_i}} \overline{L}_j \,|\, \overline{C}_i^{m_i-1} \,|\, \overline{C}_{1-i}^{m_{1-i}} \xrightarrow{l_{\rangle_i}} G \,|\, \overline{C}_i^{m_i-1} \,|\, \overline{C}_{1-i}^{m_{1-i}}$.

Remark 4.3: In Hirshfeld's encoding [2], a then-branch is modeled by a single action l_i (instead of the two actions $l_{\langle_i}l_{\rangle_i}$ in our case), and $\overline{C}_i$ is of the form $l_{]_i} + l_iG$. That encoding does not seem to work for the result below.

Now we state the main result.

Theorem 4.4:

1. If $\langle 0, 0, 0\rangle \Longrightarrow_M \langle \bot, m_0, m_1\rangle$, then $[\![M]\!]_L \not\lesssim_{tr} [\![M]\!]_R$.

2. If $\langle 0,0,0\rangle \not\Longrightarrow_M \langle \bot, m_0, m_1\rangle$, then $[\![M]\!]_L \leq_{\mathit{ready}} [\![M]\!]_R$.

Proof: See Section 4.1 $\square$

Corollary 4.5: If $\mathcal{R}$ is a preorder on BPP processes and $\leq_{\mathit{ready}} \subseteq \mathcal{R} \subseteq \leq_{\mathit{tr}}$, then the relation $\mathcal{R}$ is undecidable.

Proof: Trivial from the fact that M does not halt if and only if $[\![M]\!]_L \mathcal{R} [\![M]\!]_R$. $\square$

Corollary 4.6: If $\mathcal{R}$ is an equivalence relation on BPP processes and $=_{\mathit{ready}} \subseteq \mathcal{R} \subseteq =_{\mathit{tr}}$, then the relation $\mathcal{R}$ is undecidable.

Proof: It suffices to show that M does not halt if and only if $[\![M]\!]_L + [\![M]\!]_R \mathcal{R} [\![M]\!]_R$. If M halts, then $[\![M]\!]_L \not\leq_{\mathit{tr}} [\![M]\!]_R$, which implies that there is a sequence s such that $s \in \mathit{trace}([\![M]\!]_L)$ but $s \notin \mathit{trace}([\![M]\!]_R)$. Thus, $[\![M]\!]_L + [\![M]\!]_R \neq_{\mathit{tr}} [\![M]\!]_R$, which implies $\neg([\![M]\!]_L + [\![M]\!]_R \mathcal{R} [\![M]\!]_R)$.

If M does not halt, then $[\![M]\!]_L \leq_{\mathit{ready}} [\![M]\!]_R$. Let $\mathcal{R}_1$ be a ready simulation such that $([\![M]\!]_L, [\![M]\!]_R) \in \mathcal{R}_1$. Then, $\mathcal{R}_1 \cup \{([\![M]\!]_L + [\![M]\!]_R, [\![M]\!]_R)\} \cup \mathbf{Id}$ (where $\mathbf{Id}$ is the identity relation) is a ready simulation, which implies $[\![M]\!]_L + [\![M]\!]_R \leq_{\mathit{ready}} [\![M]\!]_R$. $[\![M]\!]_R \leq_{\mathit{ready}} [\![M]\!]_L + [\![M]\!]_R$ also holds, since $\{([\![M]\!]_R, [\![M]\!]_L + [\![M]\!]_R)\} \cup \mathbf{Id}$ is a ready simulation. Thus, we have $[\![M]\!]_L + [\![M]\!]_R =_{\mathit{ready}} [\![M]\!]_R$, which implies $[\![M]\!]_L + [\![M]\!]_R \mathcal{R} [\![M]\!]_R$. $\square$

4.1 Proof of Theorem 4.4

The rest of this section is devoted to the proof of Theorem 4.4.

Definition 4.7: Let $\sigma = \langle j, m_0, m_1\rangle$ be a state of a Minsky machine M. The BPP processes $[\![\sigma]\!]_L$ and $[\![\sigma]\!]_R$ are defined by:

$$[\![\langle j,m,n\rangle]\!]_L = L_j \,|\, C_0^m \,|\, C_1^n$$
$$[\![\langle j,m,n\rangle]\!]_R = \overline{L}_j \,|\, \overline{C}_0^m \,|\, \overline{C}_1^n$$

Here, L_j, $\overline{L}_j$, C_i, and $\overline{C}_i$ are those defined in Definition 4.1.

Definition 4.8: Let λ be an element of $\{\mathit{inc}_i, \mathit{then}_i, \mathit{else}_i \,|\, i \in \{0,1\}\}$. Then, $[\![\lambda]\!]$ is defined by:

$$[\![\mathit{inc}_i]\!] = l_I \quad [\![\mathit{then}_i]\!] = l_{\langle_i} l_{\rangle_i} \quad [\![\mathit{else}_i]\!] = l_{[} l_{d_i} l_{]_i}$$

The following lemma means that $[\![\sigma]\!]_L$ can simulate transitions of the Minsky machine.

Lemma 4.9: If $\sigma \xrightarrow{\lambda} \sigma'$, then $[\![\sigma]\!]_L \xrightarrow{[\![\lambda]\!]} [\![\sigma']\!]_L$

Proof: Trivial by the definition of $[\![\sigma]\!]_L$. $\square$

The following lemma means that $[\![M]\!]_R$ can simulate a transition of the Minsky machine only either by either performing valid actions, or by using the universal process U.

Lemma 4.10: If $\sigma \xrightarrow{\lambda} \sigma'$ and $[\![\sigma]\!]_R \xrightarrow{[\![\lambda]\!]} P$, then $P = [\![\sigma']\!]_R$ or $P = U' \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ for some m_0 and m_1.

Proof: Let $\sigma = \langle j, m_0, m_1 \rangle$ and $\sigma' = \langle j', m'_0, m'_1 \rangle$. Case analysis on λ.

- Case $\lambda = \mathit{inc}_i$. In this case, $M(j) = \mathbf{Inc}_{i,j'}$ and $[\![\sigma]\!]_R = \overline{L}_j \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$, with $m'_i = m_i + 1$ and $m'_{1-i} = m_{1-i}$. By the definition of $\overline{L}_j$, P must be either $(\overline{C}_i \,|\, \overline{L}_{j'}) \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1} = [\![\sigma']\!]_R$ or $U' \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$.

- Case $\lambda = \mathit{then}_i$. In this case, $M(j) = \mathbf{Cond}_{i,j',k}$ and $[\![\sigma]\!]_R = \overline{L}_j \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$, with $m_i = m'_i = 0$ and $m'_{1-i} = m_{1-i}$. Suppose $[\![\sigma]\!]_R \xrightarrow{l_{\langle_i}} P_1 \xrightarrow{l_{\rangle_i}} P$. By the definition of $\overline{L}_j$, P_1 must be either $\overline{T}_{i,j'} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ or $U' \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$. In the latter case, P must be $U' \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$. In the former case, by the definition of $\overline{T}_{i,j'}$, P must be either $\overline{L}_{j'} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ or $U' \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$.

- Case $\lambda = \mathit{else}_i$. In this case, $M(j) = \mathbf{Cond}_{i,k,j'}$ and $[\![\sigma]\!]_R = \overline{L}_j \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$, with $m'_i = m_i - 1 \geq 0$ and $m'_{1-i} = m_{1-i}$. Suppose $[\![\sigma]\!]_R \xrightarrow{l_{[}} P_1 \xrightarrow{l_{d_i}} P_2 \xrightarrow{l_{]_i}} P$. By the definition of $\overline{L}_j$, P_1 must be either $U' \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ or $\overline{E}_{i,j'} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$. In the former case, P must be of the form $U' \,|\, \overline{C}_0^{m''_0} \,|\, \overline{C}_1^{m''_1}$ for some m''_0 and m''_1. In the latter case, P_2 must be either $U' \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ or $\overline{L}_{j'} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$. In the former case, P must be $U' \,|\, \overline{C}_0^{m''_0} \,|\, \overline{C}_1^{m''_1}$ for some m''_0 and m''_1. In the latter case, P must be either $U' \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ or $\overline{L}_{j'} \,|\, \overline{C}_0^{m'_0} \,|\, \overline{C}_1^{m'_1}$.

□

We can now prove the first part of Theorem 4.4.

Lemma 4.11: If $\langle 0,0,0 \rangle \overset{s}{\Longrightarrow} \langle \bot, m_0, m_1 \rangle$, then $[\![M]\!]_L \not\leq_{tr} [\![M]\!]_R$.

Proof: Suppose $\langle 0,0,0 \rangle \overset{s}{\Longrightarrow} \langle \bot, m_0, m_1 \rangle$. By Lemma 4.9, we have $[\![s]\!]w \in \mathit{trace}([\![M]\!]_L)$. Suppose $[\![M]\!]_R \xrightarrow{[\![s]\!]} P$. Then, by Lemma 4.10, P must be $[\![\langle \bot, m_0, m_1 \rangle]\!]_R = \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ or $U' \,|\, \overline{C}_0^{m'_0} \,|\, \overline{C}_1^{m'_1}$. In either case, $P \not\xrightarrow{w}$. Therefore, $[\![s]\!]w \notin \mathit{trace}([\![M]\!]_R)$. □

To show the second part of Theorem 4.4, we use the following up-to technique.

Lemma 4.12: [up-to technique] Let $\mathcal{R}$ be a binary relation on BPP processes, which satisfies the following conditions:

- If $P \mathcal{R} Q$ and $P \xrightarrow{l} P'$, then $Q \xrightarrow{l} Q'$ and $P' \leq_{\mathit{ready}} \mathcal{R} \leq_{\mathit{ready}} Q'$ for some Q'.

- If $P \mathcal{R} Q$, then $\mathit{readies}(P) = \mathit{readies}(Q)$.

Then, $\mathcal{R} \subseteq \leq_{\mathit{ready}}$.

Proof: This follows from the fact that $\mathcal{R} \cup (\leq_{ready}\ \mathcal{R} \leq_{ready})$ is a ready simulation. $\square$

The following lemma states that G is universal in the sense that it can simulate any (even invalid) state resulting from $[\![M]\!]_L$ (except for a process of the form $U' \,|\, C_0^{m_0} \,|\, C_1^{m_1}$).

Lemma 4.13 [properties of G]: The following conditions hold for any m_0, m_1, n_0, n_1.

$$\begin{array}{cc}
L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1} \leq_{ready} G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1} & T_{i,j} \,|\, C_0^{m_0} \,|\, C_1^{m_1} \leq_{ready} G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1} \\
E_{i,j} \,|\, C_0^{m_0} \,|\, C_1^{m_1} \leq_{ready} G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1} & U \,|\, C_0^{m_0} \,|\, C_1^{m_1} \leq_{ready} G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1} \\
U' \,|\, C_0^{m_0} \,|\, C_1^{m_1} \leq_{ready} U' \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1} &
\end{array}$$

Proof: It suffices to show that the following relation $\mathcal{R}$ is a ready simulation.

$$\begin{array}{rcl}
\mathcal{R} & = & \{(L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1},\ G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}) \mid j \in \mathit{dom}(M) \cup \{\bot\}, m_0, m_1, n_0, n_1 \geq 0\} \\
& \cup & \{(T_{i,j} \,|\, C_0^{m_0} \,|\, C_1^{m_1},\ G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}) \mid i \in \{0,1\}, j \in \mathit{dom}(M) \cup \{\bot\}, m_0, m_1, n_0, n_1 \geq 0\} \\
& \cup & \{(E_{i,j} \,|\, C_0^{m_0} \,|\, C_1^{m_1},\ G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}) \mid i \in \{0,1\}, j \in \mathit{dom}(M) \cup \{\bot\}, m_0, m_1, n_0, n_1 \geq 0\} \\
& \cup & \{(U \,|\, C_0^{m_0} \,|\, C_1^{m_1},\ G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}) \mid m_0, m_1, n_0, n_1 \geq 0\} \\
& \cup & \{(U' \,|\, C_0^{m_0} \,|\, C_1^{m_1},\ U' \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}) \mid m_0, m_1, n_0, n_1 \geq 0\}
\end{array}$$

Suppose that (P, Q) is in the $\mathcal{R}$. It is easy to check that $\mathit{readies}(P) = \mathit{readies}(Q)$: If (P, Q) is in the fifth set, then $\mathit{readies}(P) = \mathit{readies}(Q) = \mathbf{Act}_1$. Otherwise, $\mathit{readies}(P) = \mathit{readies}(Q) = \mathbf{Act}_0$.

It remains to show that $P \xrightarrow{l} P'$ implies that $Q \xrightarrow{l} Q'$ and $(P', Q') \in \mathcal{R}$ for some Q'. The case where the transition $P \xrightarrow{l} P'$ comes from an action of C_i is trivial. For other cases, we perform case analysis on P. We show only the main cases; the other cases are similar or trivial.

- Case $P = L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1}$: In this case, we have one of the following conditions:
  - $l = l_I$ and $P' = L_j \,|\, C_0^{m_0'} \,|\, C_1^{m_1'}$
  - $l = l_{\langle i}$ and $P' = T_{i,k} \,|\, C_0^{m_0} \,|\, C_1^{m_1}$
  - $l = l_{[}$ and $P' = E_{i,k} \,|\, C_0^{m_0} \,|\, C_1^{m_1}$
  - $l = w$ and $P' = U \,|\, C_0^{m_0} \,|\, C_1^{m_1}$
  - $P' = U' \,|\, C_0^{m_0} \,|\, C_1^{m_1}$

  Let Q' be $U' \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}$ in the last case, and $G \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}$ in the other cases. Then, we have $Q \xrightarrow{l} Q'$ and $(P', Q') \in \mathcal{R}$ as required.

- Case $P = U \,|\, C_0^{m_0} \,|\, C_1^{m_1}$. In this case, $P' = U' \,|\, C_0^{m_0} \,|\, C_1^{m_1}$. Thus, the required result holds for $Q' = U' \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}$.

- Case $P = U' \,|\, C_0^{m_0} \,|\, C_1^{m_1}$. In this case, $Q = U' \,|\, \overline{C_0}^{n_0} \,|\, \overline{C_1}^{n_1}$ and $P' = P$. The required result holds for $Q' = Q$.

$\square$

We are ready to prove Theorem 4.4.

Proof of Theorem 4.4:

1. This has been proved in Lemma 4.11.
2. Suppose $\langle 0,0,0\rangle \not\Longrightarrow_M \langle \bot, m_0, m_1\rangle$. Let $\mathcal{R}$ be:

$$\begin{array}{ll}
 & \mathbf{Id} \\
\cup & \{([\![\sigma]\!]_L,\ [\![\sigma]\!]_R) \mid \sigma_I \Longrightarrow_M \sigma\} \\
\cup & \{(T_{i,k_1} \,|\, C_0^{m_0} \,|\, C_1^{m_1}, \overline{T}_{i,k_1} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}) \\
 & \qquad \mid \sigma_I \Longrightarrow_M \langle j, m_0, m_1\rangle \wedge M(j) = \mathbf{Cond}_{i,k_1,k_2} \wedge m_i = 0\} \\
\cup & \{(T_{i,k_1} \,|\, C_0^{m_0} \,|\, C_1^{m_1}, \overline{L}_j \,|\, \overline{C}_0^{m'_0} \,|\, \overline{C}_1^{m'_1}) \\
 & \qquad \mid \sigma_I \Longrightarrow_M \langle j, m_0, m_1\rangle \wedge M(j) = \mathbf{Cond}_{i,k_1,k_2} \wedge m'_i = m_i - 1 \wedge m'_{1-i} = m_{1-i}\} \\
\cup & \{(E_{i,k_2} \,|\, C_0^{m_0} \,|\, C_1^{m_1}, \overline{E}_{i,k_2} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}) \\
 & \qquad \mid \sigma_I \Longrightarrow_M \langle j, m_0, m_1\rangle \wedge M(j) = \mathbf{Cond}_{i,k_1,k_2}\} \\
\cup & \{(E_{i,k_2} \,|\, C_0^{m'_0} \,|\, C_1^{m'_1}, \overline{L}_{k_2} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}) \\
 & \qquad \mid \sigma_I \Longrightarrow_M \langle j, m_0, m_1\rangle \wedge M(j) = \mathbf{Cond}_{i,k_1,k_2} \wedge m'_i = m_i - 1 \wedge m'_{1-i} = m_{1-i}\}
\end{array}$$

It suffices to show that $\mathcal{R}$ is a ready simulation up to $\leq_{ready}$. Suppose (P, Q) is in $\mathcal{R}$. We can immediately obtain *readies*(P) = *readies*(Q) (note that if $(P,Q) \notin \mathbf{Id}$, then *readies*(P) = *readies*$(Q) = \mathbf{Act}_0$).

It remains to show that $P \xrightarrow{l} P'$ implies that $Q \xrightarrow{l} Q'$ and $P' \leq_{ready} \mathcal{R} \leq_{ready} Q'$ for some Q'. We perform case analysis on (P, Q). The case where $(P,Q) \in \mathbf{Id}$ is trivial. In the other cases, if U is involved in the transition $P \xrightarrow{l} P'$, then P' must be of the form $U' \,|\, C_0^{m_0} \,|\, C_1^{m_1}$. By Lemma 4.13, the result holds for $Q' = U' \,|\, \overline{C}_0^{n_0} \,|\, \overline{C}_1^{n_1}$. Suppose that U is not involved in the transition $P \xrightarrow{l} P'$.

- Case where (P,Q) is in the second set. In this case, we have $P = L_j \,|\, C_0^{m_0} \,|\, C_1^{m_1}$ and $Q = \overline{L}_j \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$, with $\sigma_I \Longrightarrow_M \langle j, m_0, m_1\rangle$ and $j \neq \bot$. There are four cases to consider.
  - Case $l = l_I$: In this case, $M(j) = \mathbf{Inc}_{i,k}$ and $P' = L_k \,|\, C_0^{m'_0} \,|\, C_1^{m'_1}$ with $m'_i = m_i + 1$ and $m'_{1-i} = m_{1-i}$. The required result holds for $Q' = \overline{L}_k \,|\, \overline{C}_0^{m'_0} \,|\, \overline{C}_1^{m'_1}$.
  - Case $l = l_{\langle i}$: In this case, $M(j) = \mathbf{Cond}_{i,k_1,k_2}$ and $P' = T_{i,k_1} \,|\, C_0^{m_0} \,|\, C_1^{m_1}$. Let Q' be $\overline{T}_{i,k_1} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$ if $m_i = 0$, and be $\overline{L}_j \,|\, \overline{C}_0^{m'_0} \,|\, \overline{C}_1^{m'_1}$ where $m'_i = m_i - 1$ and $m'_{1-i} = m_{1-i}$. Then, the result follows.
  - Case $l = l_{[}$: In this case, $M(j) = \mathbf{Cond}_{i,k_1,k_2}$ and $P' = E_{i,k_2} \,|\, C_0^{m_0} \,|\, C_1^{m_1}$. Let Q' be $\overline{E}_{i,k_2} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$. Then, the result follows, since (P', Q') is in the fifth set of $\mathcal{R}$.

  - Case $l = l_{d_i}$: In this case, $P' = L_j \,|\, C_0^{m'_0} \,|\, C_1^{m'_1}$ with $m'_i = m_i - 1$ and $m'_{1-i} = m_{1-i}$. Let $Q' = G \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$. Then, $Q \xrightarrow{l} Q'$ and $P' \leq_{ready} Q'$. Since $\mathbf{Id} \subseteq \mathcal{R}$ and $\leq_{ready}$ is transitive, we have $P' \leq_{ready} \mathcal{R} \leq_{ready} Q'$ as required.

- Case where (P, Q) is in the third set: In this case, we have $P = T_{i,k_1} \,|\, C_0^{m_0} \,|\, C_1^{m_1}$ and $Q = \overline{T}_{i,k_1} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$, with $\sigma_I \Longrightarrow_M \langle j, m_0, m_1 \rangle$, $M(j) = \mathbf{Cond}_{i,k_1,k_2}$, and $m_i = 0$. There are two cases to consider.
  - Case $l = l_{\rangle_i}$. In this case, $P' = L_{k_1} \,|\, C_0^{m_0} \,|\, C_1^{m_1}$. The required result holds for $Q' = \overline{L}_{k_1} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$.
  - Case $l = l_{d_{1-i}}$. In this case, $P' = T_{i,k_1} \,|\, C_0^{m'_0} \,|\, C_1^{m'_1}$ with $m'_i = 0$ and $m'_{1-i} = m_{1-i} - 1$. Let Q' be $G \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$. Then, the result follows.
- Case where (P, Q) is in the fourth set: In this case, we have $P = T_{i,k_1} \,|\, C_0^{m_0} \,|\, C_1^{m_1}$ and $Q = \overline{L}_j \,|\, \overline{C}_0^{m'_0} \,|\, \overline{C}_1^{m'_1}$ with $M(j) = \mathbf{Cond}_{i,k_1,k_2}$, $m'_i = m_i - 1$ and $m'_{1-i} = m_{1-i}$. l must be either $l_{\rangle_i}$ or $l_{d_{i'}}$. Let Q' be $G \,|\, \overline{C}_0^{m'_0} \,|\, \overline{C}_1^{m'_1}$. Then, the result follows, since $P' \leq_{ready} Q'$ by Lemma 4.13.
- Case where (P, Q) is in the fifth set: In this case, we have:

$$\begin{array}{cc} P = E_{i,k_2} \,|\, C_0^{m_0} \,|\, C_1^{m_1} & Q = \overline{E}_{i,k_2} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1} \\ \sigma_I \Longrightarrow_M \langle j, m_0, m_1 \rangle & M(j) = \mathbf{Cond}_{i,k_1,k_2} \end{array}$$

  There are two cases to consider.
  - Case $l = l_{d_i}$: Then, $P' = E_{i,k_2} \,|\, C_0^{m'_0} \,|\, C_1^{m'_1}$ with $m'_i = m_i - 1$ and $m'_{1-i} = m_{1-i}$. The required result holds for $Q' = \overline{L}_{k_2} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$.
  - Case l is $l_{d_{1-i}}$ or $l_{]_i}$: The required result holds for $Q' = G \,|\, C_0^{m_0} \,|\, C_1^{m_1}$.
- Case where (P, Q) is in the sixth set: In this case, we have:

$$\begin{array}{cc} P = E_{i,k_2} \,|\, C_0^{m'_0} \,|\, C_1^{m'_1} & Q = \overline{L}_{k_2} \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1} \\ \sigma_I \Longrightarrow_M \langle j, m_0, m_1 \rangle & M(j) = \mathbf{Cond}_{i,k_1,k_2} \\ m'_i = m_i - 1 & m'_{1-i} = m_{1-i} \end{array}$$

  There are two cases to consider.
  - Case $l = l_{]_i}$. In this case, $P' = L_{k_2} \,|\, C_0^{m'_0} \,|\, C_1^{m'_1}$. The required result holds for $Q' = \overline{L}_{k_2} \,|\, \overline{C}_0^{m'_0} \,|\, \overline{C}_1^{m'_1}$.
  - Case $l = l_{d_{i'}}$. In this case, the required result holds for $Q' = G \,|\, \overline{C}_0^{m_0} \,|\, \overline{C}_1^{m_1}$.

□

References

[1] S. Christensen, Y. Hirshfeld, and F. Moller. Decomposability, decidability and axiomatisability for bisimulation equivalence on basic parallel processes. In *LICS*, pages 386–396, 1993.

[2] Y. Hirshfeld. Petri nets and the equivalence problem. In *CSL 1993*, volume 832 of *Lecture Notes in Computer Science*, pages 165–174, 1993.

[3] H. Hüttel. Undecidable Equivalence for Basic Parallel Processes. In *Proceedings of TACS94, Springer LNCS 789*, pages 454–464, 1994.

[4] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[5] M. L. Minsky. *Computation: Finite and infinite Machines.* Prentice-Hall, 1967.