Place Bisimulation Equivalences for Design of Concurrent Systems *

IGOR V. TARASYUK

A.P. Ershov Institute of Informatics Systems, 6, Acad. Lavrentiev ave., Novosibirsk, 630090, Russia
Fax: +7 3832 32 34 94
E-mail: itar@iis.nsk.su

Abstract

In this paper, we supplement the set of basic and back-forth behavioural equivalences for Petri nets considered in [11] by place bisimulation ones. The relationships of all the equivalence notions are examined, and their preservation by refinements is investigated to find out which of these relations may be used in top-down design. It is demonstrated that the place bisimulation equivalences may be used for the compositional and history preserving reduction of Petri nets.

1 Introduction

The notion of equivalence is central to any theory of systems. Equivalences allow one to compare and reduce systems taking into account particular aspects of their behaviour. Petri nets became a popular formal model for design of concurrent and distributed systems. In recent years, a wide range of behavioural equivalences was proposed in the concurrency theory. The equivalences can be classified depending of semantics of concurrency they impose.

*The work is supported by Volkswagen Stiftung, grant I/70 564, INTAS-RFBR, grant 95-0378 and the Foundation for Promotion to Young Scientists of Siberian Division of the Russian Academy of Sciences
In *interleaving* semantics, a concurrent happening of actions is interpreted as their occurrence in any possible order. In *step* semantics, a concurrency of actions is a basic notion, but their causal dependencies are not respected. In *partial word* semantics, causal dependencies of actions are respected in part via partially ordered multisets (pomsets) of actions, and a pomset may be modelled by a less sequential one (i.e. having less strict partial order). In *pomset* semantics, causal dependencies of actions are fully respected, and pomsets of actions should coincide to model each other. In *process* semantics, a structure of a process (causal) net is respected.

The following basic notions of behavioural equivalences were proposed:

- **Trace equivalences** (they respect only protocols of behaviour of systems): interleaving ($\equiv_i$) [8], step ($\equiv_s$) [8], partial word ($\equiv_{pw}$) [12], pomset ($\equiv_{pom}$) [8] and process ($\equiv_{pr}$) [10].
- **Usual bisimulation equivalences** (they respect branching structure of behaviour of systems): interleaving ($\leftrightarrow_i$) [8], step ($\leftrightarrow_s$) [8], partial word ($\leftrightarrow_{pw}$) [12], pomset ($\leftrightarrow_{pom}$) [8] and process ($\leftrightarrow_{pr}$) [3].
- **ST-bisimulation equivalences** (they respect the duration or maximality of events in behaviour of systems): interleaving ($\leftrightarrow_{ST}$) [7], partial word ($\leftrightarrow_{pwST}$) [12], pomset ($\leftrightarrow_{pomST}$) [12] and process ($\leftrightarrow_{prST}$) [10].
- **History preserving bisimulation equivalences** (they respect the “history” of behaviour of systems): pomset ($\leftrightarrow_{pomh}$) [12] and process ($\leftrightarrow_{prh}$) [10].
- **Conflict preserving equivalences** (they completely respect conflicts of events in systems): multi event structure ($\equiv_{mes}$) [10] and occurrence ($\equiv_{occ}$) [7].
- **Isomorphism** ($\simeq$) (i.e. coincidence of systems up to renaming of their components).

Another important group of equivalences are back-forth bisimulation ones which are based on the idea that a bisimulation relation should not only require systems to simulate each other behaviour in the forward direction but also when going back in the history. By now, the set of all possible back-forth equivalence notions was proposed in interleaving, step, partial word and pomset semantics. Most of them coincide with basic or with other back-forth relations. The following new notions were obtained: step back step forth
(↔_{sbf}) [6], step back partial word forth (↔_{sbpfw}) [9] and step back pomset forth (↔_{sbspomf}) [9] bisimulation equivalences. In [11] we supplemented them by several new relations in process semantics: step back process forth (↔_{sbprf}) and pomset back process forth (↔_{pombprf}) bisimulation equivalences.

The third important group of equivalences are place bisimulation ones introduced in [1]. They are relations between places (instead of markings or processes). The relation on markings is obtained using the “lifting” of relation on places. The main application of place bisimulation equivalences is an effective global behaviour preserving reduction technique for Petri nets based on them. In [1], interleaving place bisimulation equivalence (∼_i) was proposed. In this paper, strict interleaving place bisimulation equivalence (∼_i) was defined also, by imposing the additional requirement stating that corresponding transitions of nets must be related by the bisimulation. In [3, 4], step (∼_s), partial word (∼_pw), pomset (∼_pom), process (∼_pr) place bisimulation equivalences and their strict analogues (∼_s, ∼_pw, ∼_pom, ∼_pr) were proposed. The coincidence of ∼_i, ∼_s and ∼_pw was established. It was shown that all strict bisimulation equivalences coincide with ∼_pr. Thus, only three different equivalences remain: ∼_i, ∼_pom and ∼_pr. In addition, in these papers the polynomial algorithm of a net reduction modulo ∼_i and ∼_pr was proposed.

To choose appropriate behavioural viewpoint on systems to be modelled, it is important to have a complete set of equivalence notions in all semantics and understand their interrelations. Treating equivalences for preservation by refinements allows one to decide which of them may be used for top-down design. In this paper, we obtain a number of results on solution these problems for place bisimulation equivalences.

The first result is a diagram of interrelations of place equivalences with basic and back-forth behavioural notions from [10, 11]. We prove that ∼_pr implies ↔_{prh} and answer the question from [1]: it is no sense to define history preserving place bisimulation equivalence. Another consequence is: the algorithm of a net reduction from [3, 4], based on ∼_pr, preserves “histories” of the behaviour of the initial net.

The second result is concerned a notion of transition refinement. In [5], SM-refinement operator for Petri nets was proposed, which “replaces” their transitions by SM-nets, a subclass of state machine nets. We treat all the considered equivalence notions for preservation by SM-refinements and establish that ∼_pr is the only place bisimulation equivalence which is preserved by SM-refinements. Thus, this equivalence may be used for the compositional
2 Basic definitions

In this section, we present some basic definitions used further.

2.1 Nets

Let $\text{Act} = \{a, b, \ldots\}$ be a set of action names.

**Definition 2.1** A labelled net is a quadruple $N = \langle P_N, T_N, F_N, l_N \rangle$, where:

- $P_N = \{p, q, \ldots\}$ is a set of places;
- $T_N = \{t, u, \ldots\}$ is a set of transitions;
- $F_N : (P_N \times T_N) \cup (T_N \times P_N) \to \mathbb{N}$ is the flow relation with weights ($\mathbb{N}$ denotes a set of natural numbers);
- $l_N : T_N \to \text{Act}$ is a labelling of transitions with action names.

Given labelled nets $N$ and $N'$ a mapping $\beta : P_N \cup T_N \to P_{N'} \cup T_{N'}$ is an isomorphism between $N$ and $N'$, denoted by $N \simeq N'$, if $\beta$ is a bijective renaming of places and transitions of $N$ s.t. the nets $N$ and $N'$ coincide up to it. Two labelled nets $N$ and $N'$ are isomorphic, denoted by $N \simeq N'$, if $\exists \beta : N \simeq N'$.

Given a labelled net $N$ and some transition $t \in T_N$, the precondition and postcondition of $t$, denoted by $\mathbf{p}t$ and $\mathbf{t}p$ respectively, are the multisets defined in such a way: $\mathbf{p}t(p) = F_N(p, t)$ and $\mathbf{t}p(p) = F_N(t, p)$. Analogous definitions are introduced for places: $\mathbf{p}t(p) = F_N(t, p)$ and $\mathbf{t}p(p) = F_N(p, t)$. Let $\circ N = \{p \in P_N | \mathbf{p}p = \emptyset\}$ is the set of input places of $N$ and $N^\circ = \{p \in P_N | \mathbf{t}p = \emptyset\}$ is the set of output places of $N$.

A labelled net $N$ is acyclic, if there exist no transitions $t_0, \ldots, t_n \in T_N$ s.t. $t_{i-1} \mathbf{p}t_i \neq \emptyset$ (1 ≤ $i$ ≤ $n$) and $t_0 = t_n$. A labelled net $N$ is ordinary, if $\forall p \in P_N \mathbf{p}p$ and $\mathbf{p}$ are proper sets (not multisets).

Let $N = \langle P_N, T_N, F_N, l_N \rangle$ be an acyclic ordinary labelled net and $x, y \in P_N \cup T_N$. Let us introduce the following notions.

- $x \prec_N y \iff xF_N^+ y$, where $F_N^+$ is a transitive closure of $F_N$ (the strict causal dependence relation);
• \( \downarrow_N x = \{ y \in P_N \cup T_N \mid y \prec_N x \} \) (the set of strict predecessors of \( x \));

A set \( T \subseteq T_N \) is left-closed in \( N \), if \( \forall t \in T \ (\downarrow_N t) \cap T_N \subseteq T \).

We denote the set of all finite multisets over a set \( X \) by \( \mathcal{M}(X) \). A marking of a labelled net \( N \) is a multiset \( M \in \mathcal{M}(P_N) \).

**Definition 2.2** A (marked) net is a tuple \( N = \langle P_N, T_N, F_N, l_N, M_N \rangle \), where

\( \langle P_N, T_N, F_N, l_N \rangle \) is a labelled net and \( M_N \in \mathcal{M}(P_N) \) is the initial marking.

Let \( M \in \mathcal{M}(P_N) \) be a marking of a net \( N \). A transition \( t \in T_N \) is fireable in \( M \), if \( \bullet t \subseteq M \). If \( t \) is fireable in \( M \), its firing yields a new marking \( \tilde{M} = M - \bullet t + \bullet t \), denoted by \( M \overset{t}{\rightarrow} \tilde{M} \).

### 2.2 Partially ordered sets

**Definition 2.3** A labelled partially ordered set (lposet) is a triple \( \rho = \langle X, \prec, l \rangle \), where:

- \( X = \{ x, y, \ldots \} \) is some set;
- \( \prec \subseteq X \times X \) is a strict partial order (irreflexive transitive relation) over \( X \);
- \( l : X \rightarrow \text{Act} \) is a labelling function.

Let \( \rho = \langle X, \prec, l \rangle \) and \( \rho' = \langle X', \prec', l' \rangle \) be lposets.

A mapping \( \beta : X \rightarrow X' \) is a homomorphism between \( \rho \) and \( \rho' \), denoted by \( \beta : \rho \subseteq \rho' \), if it is a bijection and \( \forall x, y \in X \ x \prec y \Rightarrow \beta(x) \prec' \beta(y) \), \( \forall x \in X \ l(x) = l'(\beta(x)) \). We write \( \rho \subseteq \rho' \), if \( \exists \beta : \rho \subseteq \rho' \).

A mapping \( \beta : X \rightarrow X' \) is an isomorphism between \( \rho \) and \( \rho' \), denoted by \( \beta : \rho \simeq \rho' \), if \( \beta : \rho \subseteq \rho' \) and \( \beta^{-1} : \rho' \subseteq \rho \). Two lposets \( \rho \) and \( \rho' \) are isomorphic, denoted by \( \rho \simeq \rho' \), if \( \exists \beta : \rho \simeq \rho' \).

**Definition 2.4** Partially ordered multiset (pomset) is an isomorphism class of lposets.
2.3 Processes

Definition 2.5 A causal net is an acyclic ordinary labelled net
\[ C = (P_C, T_C, F_C, l_C), \]
s.t.:

1. \( \forall r \in P_C \ \cdot |r| \leq 1 \) and \( |r^\bullet| \leq 1 \), i.e. places are unbranched;
2. \( \forall x \in P_C \cup T_C \ \downarrow_C x | \leq \infty \), i.e. a set of causes is finite.

Let us note that on the basis of any causal net \( C \) one can define lposet
\[ \rho_C = (\mathcal{M}(P_N) \cap (T_C \times T_C), l_C). \]

The fundamental property of causal nets is [3]: if \( C \) is a causal net,
then there exists a sequence of transition firings (a full execution of \( C \)) s.t.
\( \overset{\circ C}{v_1} \rightarrow \cdots \rightarrow v_n = C^o \) s.t. \( L_i \subseteq P_C \ (0 \leq i \leq n) \), \( P_C = \bigcup_{i=0}^{n} L_i \) and \( T_C = \{ v_1, \ldots, v_n \} \).

Definition 2.6 Given a net \( N \) and a causal net \( C \). A mapping \( \varphi : P_C \cup T_C \rightarrow P_N \cup T_N \) is an embedding of \( C \) into \( N \), denoted by \( \varphi : C \rightarrow N \), if:

1. \( \varphi(P_C) \in \mathcal{M}(P_N) \) and \( \varphi(T_C) \in \mathcal{M}(T_N) \), i.e. sorts are preserved;
2. \( \forall v \in T_C \ \cdot \varphi(v) = \varphi(\cdot v) \) and \( \varphi(v)^\bullet = \varphi(v^\bullet) \), i.e. flow relation is respected;
3. \( \forall v \in T_C \ l_C(v) = l_N(\varphi(v)) \), i.e. labelling is preserved.

Since embeddings respect the flow relation, if \( \overset{\circ C}{v_1} \rightarrow \cdots \rightarrow v_n = C^o \) is a full execution of \( C \), then \( M = \varphi(\circ C) \overset{\varphi(v_1)}{\longrightarrow} \cdots \overset{\varphi(v_n)}{\longrightarrow} \varphi(C^o) = \tilde{M} \) is a sequence of transition firings in \( N \).

Definition 2.7 A firable in marking \( M \) process of a net \( N \) is a pair \( \pi = (C, \varphi) \), where \( C \) is a causal net and \( \varphi : C \rightarrow N \) is an embedding s.t. \( M = \varphi(\circ C) \). A firable in \( M \) process is a process of \( N \).

We write \( \Pi(N, M) \) for the set of all firable in marking \( M \) processes of a net \( N \) and \( \Pi(N) \) for the set of all processes of a net \( N \). The initial process of a net \( N \) is \( \pi_N = (C_N, \varphi_N) \in \Pi(N) \), s.t. \( T_{C_N} = \emptyset \). If \( \pi \in \Pi(N, M) \), then firing of this process transforms a marking \( M \) into \( \tilde{M} = \varphi(C^o) \), denoted by \( M \overset{\pi}{\rightarrow} \tilde{M} \).

Let \( \pi = (C, \varphi) \), \( \tilde{\pi} = (\tilde{C}, \tilde{\varphi}) \in \Pi(N) \), \( \hat{\pi} = (\hat{C}, \hat{\varphi}) \in \Pi(N, \varphi(C^o)) \). A process \( \hat{\pi} \) is an extension of \( \pi \) by process \( \tilde{\pi} \), denoted by \( \pi \overset{\pi}{\rightarrow} \hat{\pi} \), if \( T_C \subseteq T_{\tilde{C}} \) is
a left-closed set in \( \hat{C} \) and \( T_{\hat{C}} = T_{\hat{C}} \setminus T_C \). We write \( \pi \rightarrow \hat{\pi} \), if \( \exists \hat{\pi} \pi \xrightarrow{i} \hat{\pi} \). A process \( \hat{\pi} \) is an extension of \( \pi \) by one transition, denoted by \( \pi \xrightarrow{v} \hat{\pi} \), if \( \pi \xrightarrow{i} \hat{\pi} \) and \( T_{\hat{C}} = \{v\} \).

3 Place bisimulation equivalences

In this section, place bisimulation equivalences are introduced. Let us recall the definition of usual bisimulation equivalences.

**Definition 3.1** Let \( N \) and \( N' \) be some nets. A relation \( R \subseteq M(N) \times M(N') \) is a \( * \)-bisimulation between \( N \) and \( N' \), \( * \in \{\text{interleaving, step, partial word, pomset, process}\} \), denoted by \( R : N \leftrightarrow^* N' \), \( * \in \{i, s, pw, pom, pr\} \), if:

1. \( (M_N, M_{N'}) \in R \).
2. \( (M, M') \in R \), \( M \xrightarrow{i} \hat{M} \),
   - (a) \( |T_{\hat{C}}| = 1 \), if \( * = i \);
   - (b) \( \prec_{\hat{C}} = \emptyset \), if \( * = s \);
   \( \Rightarrow \exists \hat{M}' : \ M' \xrightarrow{i} \hat{M}' \), \( (\hat{M}, \hat{M}') \in R \) and
   - (a) \( \rho_{\hat{C}} \subseteq \rho_{\hat{C}} \), if \( * = pw \);
   - (b) \( \rho_{\hat{C}} \simeq \rho_{\hat{C}} \), if \( * \in \{i, s, pom\} \);
   - (c) \( \hat{C} \simeq \hat{C}' \), if \( * = pr \).
3. As item 2, but the roles of \( N \) and \( N' \) are reversed.

Two nets \( N \) and \( N' \) are \( * \)-bisimulation equivalent, \( * \in \{\text{interleaving, step, partial word, pomset, process}\} \), denoted by \( N \leftrightarrow^* N' \), if \( \exists R : N \leftrightarrow^* N' \), \( * \in \{i, s, pw, pom, pr\} \).

*Place bisimulations* are relations between places instead of markings. A relation on markings is obtained with use of the “lifting” of a bisimulation relation on places.

Let for nets \( N \) and \( N' \) \( R \subseteq P_N \times P_{N'} \) be a relation between their places. The lifting of \( R \) is a relation \( \overline{R} \subseteq M(P_N) \times M(P_{N'}) \), defined as follows: \( (M, M') \in \overline{R} \Leftrightarrow \exists \{(p_1, p'_1), \ldots, (p_n, p'_n)\} \in M(R) : M = \{p_1, \ldots p_n\}, M' = \{p'_1, \ldots p'_n\} \).
Definition 3.2 Let $N$ and $N'$ be some nets. A relation $R \subseteq P_N \times P_{N'}$ is a $\star$-place bisimulation between $N$ and $N'$, $\star \in \{$interleaving, step, partial word, pomset, process$\}$, denoted by $R : N \sim_{\star} N'$, if $R : N \leftrightarrow_{\star} N'$, $\star \in \{$i, s, pw, pom, pr$\}$.  

Two nets $N$ and $N'$ are $\star$-place bisimulation equivalent, $\star \in \{$interleaving, step, partial word, pomset, process$\}$, denoted by $N \sim_{\star} N'$, if $\exists R : N \sim_{\star} N'$, $\star \in \{$i, s, pw, pom, pr$\}$. 

Strict place bisimulation equivalences are defined using the additional requirement stating that corresponding transitions of nets must be (as well as makings) related by $R$. This relation is defined on transitions as follows. 

Let for some nets $N$ and $N'$ $t \in T_N$, $t' \in T_{N'}$. Then $(t, t') \in R$ $\iff$ $(\bullet t, \bullet t') \in R$ $\land$ $(t \bullet, t' \bullet) \in R$ $\land$ $(l_N(t) = l_{N'}(t'))$. 

Definition 3.3 Let $N$ and $N'$ be some nets. A relation $R \subseteq P_N \times P_{N'}$ is a strict $\star$-place bisimulation between $N$ and $N'$, $\star \in \{$interleaving, step, partial word, pomset, process$\}$, denoted by $R : N \approx_{\star} N'$, $\star \in \{$i, s, pw, pom, pr$\}$, if: 

1. $R : N \leftrightarrow_{\star} N'$. 

2. In the definition of $\star$-bisimulation in item 2 (and in item 3 symmetrically) the new requirement is added: $\forall v \in T_{\tilde{C}} (\tilde{\phi}(v), \tilde{\phi}'(\beta(v))) \in R$, where:
   
   (a) $\beta : \rho_{\tilde{C}} \sqsubseteq \rho_{\tilde{C}'}$, if $\star = pw$; 
   
   (b) $\beta : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$, if $\star \in \{$i, s, pom$\}$; 
   
   (c) $\beta : \tilde{C} \simeq \tilde{C}'$, if $\star = pr$. 

Two nets $N$ and $N'$ are strict $\star$-place bisimulation equivalent, $\star \in \{$interleaving, step, partial word, pomset, process$\}$, denoted by $N \approx_{\star} N'$, if $\exists R : N \approx_{\star} N'$, $\star \in \{$i, s, pw, pom, pr$\}$. 

An important property of place bisimulations is additivity. Let for nets $N$ and $N' R : N \sim_{\star} N'$. Then $(M_1, M'_1) \in R$ and $(M_2, M'_2) \in R$ implies $((M_1 + M_2), (M'_1 + M'_2)) \in R$. In particular, if we put $n$ tokens into each of the places $p \in P_N$ and $p' \in P_{N'}$ s.t. $(p, p') \in R$, then the nets obtained as a result of such a changing of the initial markings, must be also place bisimulation equivalent. 

The following proposition establishes a coincidence of most place bisimulation equivalences.
Proposition 3.1 [3, 4] For nets $N$ and $N'$:

1. $N \sim_i N' \iff N \sim_{pw} N'$;
2. $N \sim_{pr} N' \iff N \approx_i N' \iff N \approx_{pr} N'$.

4 Interrelations of the equivalences

In this section, place bisimulation equivalences are compared with basic equivalences and back-forth bisimulation equivalences. First, recall the definition of history preserving bisimulation equivalences.

Definition 4.1 Let $N$ and $N'$ be some nets. A relation $R \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$, where $\mathcal{B} = \{\beta \mid \beta : T_C \rightarrow T_{C'}, \pi = (C, \varphi) \in \Pi(N), \pi' = (C', \varphi') \in \Pi(N')\}$, is a $\star$-history preserving bisimulation between $N$ and $N'$, $\star \in \{\text{pomset, process}\}$, denoted by $N \leftrightarrow_{\star h} N'$, $\star \in \{\text{pom, pr}\}$, if:

1. $(\pi_N, \pi_{N'}, \emptyset) \in R$.
2. $(\pi, \pi', \beta) \in R \Rightarrow$
   
   (a) $\beta : \rho_C \simeq \rho_{C'}$, if $\star \in \{\text{pom, pr}\}$;
   
   (b) $C \simeq C'$, if $\star = \text{pr}$.
3. $(\pi, \pi', \beta) \in R$, $\pi \rightarrow \tilde{\pi} \Rightarrow \exists \tilde{\beta}, \tilde{\pi}': \pi' \rightarrow \tilde{\pi}'$, $\tilde{\beta}|_{T_C} = \beta$, $(\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in R$.
4. As item 3 but the roles of $N$ and $N'$ are reversed.

Two nets $N$ and $N'$ are $\star$-history preserving bisimulation equivalent, $\star \in \{\text{pomset, process}\}$, denoted by $N \leftrightarrow_{\star h} N'$, if $\exists R : N \leftrightarrow_{\star h} N'$, $\star \in \{\text{pom, pr}\}$.

Let us note that in this definition one can use extentions of processes by one transition only. Now we are able to prove the proposition about interrelations of place and history preserving equivalences.

Proposition 4.1 For nets $N$ and $N'$: $N \sim_{pr} N' \Rightarrow N \leftrightarrow_{pr h} N'$.

Proof. See Appendix A. \qed

Below, the symbol ‘’ will denote “nothing”, and the signs of equivalences subscribed by it are considered as that of without any subscription. The following theorem collect all the results obtained here and in [11], and clarify interrelations of all the equivalences.
Theorem 4.1 Let \( \leftrightarrow, \leftrightarrow \in \{\equiv, \equiv, \sim, \simeq\}\), \( \ast, \ast \in \{i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombprf\}\). For nets \( N \) and \( N' : N \leftrightarrow \ast N' \Rightarrow N \leftrightarrow \ast \ast N' \) iff in the graph in Figure 1 there exists a directed path from \( \leftrightarrow \ast \) to \( \leftrightarrow \ast \ast \).

Proof. \((\Leftarrow)\) By Theorem 12 from [11] and the following substantiations.

- The implications \( \sim \ast \rightarrow \leftrightarrow \ast, \ast \in \{i, pom, pr\}\) are valid by the definitions.
- The implication \( \sim pr \rightarrow \leftrightarrow prh\) is valid by Proposition 3.2.
- The implication \( \sim pom \rightarrow \sim i\) is valid by the definitions.
- The implication \( \sim pr \rightarrow \sim pom\) is valid since lposets of isomorphic nets are also isomorphic.
- The implication \( \sim \rightarrow \sim pr\) is obvious.

\((\Rightarrow)\) By Theorem 12 from [11] and the following examples (dashed lines in Figure 2 connect bisimilar places).
• In Figure 2(a), $N \sim N'$, but $N \not\equiv_{pom} N'$, since only in the net $N'$ action $b$ can depend on $a$.

• In Figure 2(b), $N \sim_{pom} N'$, but $N \not\equiv_{pr} N'$, since only in the net $N'$ the transition with label $a$ has two input (and two output) places.

• In Figure 2(c), $N \equiv_{occ} N'$, but $N \not\equiv_i N'$, since any place bisimulation must relate input places of the nets $N$ and $N'$. But after putting one additional token into each of these places only in $N'$ the action $c$ can happen.

• In Figure 2(b), $N \sim_{pom} N'$, but $N \not\leftrightarrow_{iST} N'$, since only in the net $N'$ action $a$ can start so that no $b$ can begin working until finishing of $a$.

• In Figure 2(d), $N \sim_{pr} N'$, but $N \not\equiv_{mes} N'$, since only the net $N'$ has two conflict actions $a$.

• In Figure 2(b), $N \sim_{pom} N'$, but $N \not\leftrightarrow_{sbsf} N'$, since only in the net $N'$ action $a$ can happen so that $b$ must depend on $a$.

In this section, we obtained a number of important results. Before, place bisimulation equivalences have been compared with usual bisimulation ones only. Here, we clarified their interrelations with all the basic and back-forth ones. We proved that $\sim_{pom}$ does not imply neither $ST$- nor back-forth bisimulation equivalences. The situation is quite different for $\sim_{pr}$. It appears to be strict enough to imply history preserving bisimulation equivalences. This interesting result may be used in reduction of nets modulo $\sim_{pr}$ [3, 4]. Now, we can guarantee that the reduced net has the same histories of the behaviour as the initial one.

5 Preservation of the equivalences by refinements

In this section, we treat the considered equivalence notions for preservation by transition refinements. We use SM-refinement, i.e. refinement by a special subclass of state-machine nets introduced in [5].

**Definition 5.1** An SM-net is a net $D = \langle P_D, T_D, F_D, l_D, M_D \rangle$ s.t.:
Figure 2: Examples of place bisimulation equivalences
1. \( \forall t \in T_D \ |s| = |t| = 1 \), i.e. each transition has exactly one input and one output place;

2. \( \exists p_{in}, \ p_{out} \in P_D \ \text{s.t.} \ p_{in} \neq p_{out} \ \text{and} \ oD = \{p_{in}\}, \ \text{i.e. it is} \ \text{an unique input and an unique output place.} \)

3. \( M_D = \{p_{in}\}, \ \text{i.e. at the beginning there is an unique token in} \ p_{in}. \)

**Definition 5.2** Let \( N = \langle P_N, T_N, F_N, l_N, M_N \rangle \) be some net, \( a \in l_N(T_N) \) and \( D = \langle P_D, T_D, F_D, l_D, M_D \rangle \) be SM-net. An SM-refinement, denoted by \( \text{ref}(N, a, D) \), is a net \( \overline{N} = \langle P_{\overline{N}}, T_{\overline{N}}, F_{\overline{N}}, l_{\overline{N}}, M_{\overline{N}} \rangle \), where:

- \( F_{\overline{N}} = P_N \cup \{ \langle p, u \rangle \ | \ p \in P_D \setminus \{p_{in}, p_{out}\}, \ u \in l_{\overline{N}}^{-1}(a) \} \);
- \( T_{\overline{N}} = (T_N \setminus l_N^{-1}(a)) \cup \{ \langle t, u \rangle \ | \ t \in T_D, \ u \in l_{\overline{N}}^{-1}(a) \} \);
- \( F_{\overline{N}}(\bar{\bar{x}}, \bar{\bar{y}}) = \begin{cases} F_N(x, y), & \bar{\bar{x}}, \bar{\bar{y}} \in P_N \cup (T_N \setminus l_N^{-1}(a)); \\ F_D(x, y), & \bar{x} = \langle x, u \rangle, \ \bar{y} = \langle y, u \rangle, \ u \in l_{\overline{N}}^{-1}(a); \\ F_N(x, u), & \bar{\bar{x}} = \langle x, u \rangle, \ \bar{y} = \langle y, u \rangle, \ \bar{\bar{x}} \in \cdot u, \ u \in l_{\overline{N}}^{-1}(a), \ y \in p_{in}; \\ 0, & \text{otherwise}; \end{cases} \)
- \( l_{\overline{N}}(\bar{u}) = \begin{cases} l_N(\bar{u}), & \bar{u} \in T_N \setminus l_N^{-1}(a); \\ l_D(t), & \bar{u} = \langle t, u \rangle, \ t \in T_D, \ u \in l_{\overline{N}}^{-1}(a); \end{cases} \)
- \( M_{\overline{N}}(p) = \begin{cases} M_N(p), & p \in P_N; \\ 0, & \text{otherwise.} \end{cases} \)

An equivalence is preserved by refinements, if equivalent nets remain equivalent after applying any refinement operator to them accordingly. The following proposition demonstrates that some place equivalences are not preserved by SM-refinements.

**Proposition 5.1** The equivalences \( \sim_i \) and \( \sim_{pom} \) are not preserved by SM-refinements.

**Proof.** In Figure 3, \( N \sim_{pom} N' \), but \( \text{ref}(N, a, D) \not\equiv \text{ref}(N', a, D) \), since only in the net \( \text{ref}(N', a, D) \) after action \( a_1 \) action \( b \) cannot happen. Consequently, equivalences between \( \leftrightarrow \) and \( \sim_{pom} \) are not preserved by SM-refinements. \( \square \)

The following proposition proves that \( \sim_{pr} \) is preserved by refinements.
Figure 3: The equivalences between $\leftrightarrow_i$ and $\sim_{pom}$ are not preserved by SM-refinements
Proposition 5.2 For nets $N$ and $N'$ s.t. $a \in \text{l}_N(T_N) \cap \text{l}_{N'}(T_{N'})$ and SM-net $D : N \sim_{pr} N' \Rightarrow \text{ref}(N, a, D) \sim_{pr} \text{ref}(N', a, D)$.

Proof. See Appendix B. \hfill \Box

Now we can add the results obtained to that of from [11] and present the following theorem.

Theorem 5.1 Let $\leftrightarrow \in \{\equiv, \leftrightarrow, \sim, \simeq\}$ and $\star \in \{\perp, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpw, sbpom, sbpr, pombpr\}$. For nets $N$ and $N'$ s.t. $a \in \text{l}_N(T_N) \cap \text{l}_{N'}(T_{N'})$ and SM-net $D : N \leftrightarrow_{\star} N' \Rightarrow \text{ref}(N, a, D) \leftrightarrow_{\star} \text{ref}(N', a, D)$ iff the equivalence $\leftrightarrow_{\star}$ is in oval in Figure 1.

Proof. By Theorem 18 from [11] and Propositions 5.1 and 5.2. \hfill \Box

In this section, an important result has been established. From all the place bisimulation equivalences, only $\sim_{pr}$ is preserved by refinements. Thus, it can be used for the compositional refinement of Petri nets.

For example, let us consider a net modelling a concurrent system and the reduced (modulo some equivalence) version of this net. The initial and the reduced nets have similar behaviour. Thus, we can use the reduced net instead of the initial one as a model for the concurrent system. If we want to consider the system at lower abstraction level, we use a refinement operation which “replaces” several transitions of the nets to the subnets corresponding to some internal structure of the system’s components. If the equivalence used for the reduction is not preserved by refinements, we cannot use the refined reduced net as a model anymore, since its behaviour can be different with that of the refined initial net.

Hence, the preservation of $\sim_{pr}$ by refinements is a powerful property, especially if to remember that this equivalence implies the history preserving one. Consequently, the histories of behaviour of the initial net coincide with that of the reduced net, and this property is valid at different abstraction levels.

6 Conclusion

In this paper, we examined a group of place bisimulation equivalences. We compared them with basic and back-forth ones. All the considered equivalences were treated for preservation by SM-refinements to establish which of
them may be used for top-down design of concurrent systems. We proved that $\sim_{pr}$ implies $\leftrightarrow_{prh}$ and it is preserved by refinements. Hence, it may be used for the compositional and history-preserving reduction of concurrent systems modelled by Petri nets.

Further research may consist in the investigation of analogues of the considered equivalences on Petri nets with $\tau$-actions ($\tau$-equivalences). $\tau$-actions are used to abstract of internal, invisible to external observer behaviour of systems to be modelled. Let us note that a number of interleaving place $\tau$-bisimulation equivalences was proposed in [4, 2]. For other semantics, the corresponding relations have not been defined, and it would be interesting to propose them and exam their interrelations. In future, we plan to define $\tau$-analogues of all the equivalence relations considered in this paper and exam them following the same pattern.

References


By Proposition 3.1, $\exists \mathcal{R} : N \approx_{pr} N'$. Then $\overline{\mathcal{R}} : N \leftrightarrow_{prh} N'$ and transitions of $N$ and $N'$ are related by $\overline{\mathcal{R}}$. Let us define a relation $\mathcal{S}$ as follows: $\mathcal{S} = \{(\pi, \pi', \beta) | \pi = (C, \varphi) \in \Pi(N), \pi' = (C, \varphi') \in \Pi(N'), \beta = id_{TC}, \forall r \in P_C (\varphi(r), \varphi'(r)) \in \mathcal{R}, \forall v \in T_C (\varphi(v), \varphi'(v)) \in \mathcal{R}\}. Let us prove $\mathcal{S} : N \leftrightarrow_{prh} N'$. 1. Obviously, $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{S}$. 2. By definition of $\mathcal{S}$, $(\pi, \pi', \beta) \in \mathcal{S} \Rightarrow \beta : \rho_C \simeq \rho_{C'}$ and $C \simeq C'$. 3. Let $(\pi, \pi', \beta) \in \mathcal{S}$, $\pi = (C, \varphi)$, $\pi' = (C, \varphi')$ and $\pi \xrightarrow{\tau} \tilde{\pi}$, $\tilde{\pi} = (\tilde{C}, \tilde{\varphi})$. Let us consider a transition firing $\tilde{\varphi}(\bullet v) \xrightarrow{\tilde{\varphi}(v)} \tilde{\varphi}(v \bullet)$ in $N$. By definition of $\mathcal{S}$, $(\varphi(\bullet v), \varphi'(\bullet v)) \in \overline{\mathcal{R}}$. Since $\varphi(\bullet v) = \tilde{\varphi}(\bullet v)$, we have $(\tilde{\varphi}(\bullet v), \varphi'(\bullet v)) \in \overline{\mathcal{R}}.$
Since $\mathcal{R} : N \approx_{pr} N'$, we have $\exists u' \in M'$ : $\psi' \circ V \triangleright u' \in M'$, ($\hat{\psi}(v), u') \in \mathcal{R}$ and ($\hat{\psi}(v'), M'$) $\in \overline{\mathcal{R}}$.

Let $v^* = \{r_1, \ldots, r_n\}$, $M' = \{p'_1, \ldots, p'_n\}$, $\forall i \ (1 \leq i \leq n) (\hat{\psi}(r_i), p'_i) \in \mathcal{R}$. Let us define a mapping $\psi'$ as follows: $\hat{\psi}'((p_c \cup i_c)) = \psi'$, $\hat{\psi}'(v) = u'$, $\forall i \ (1 \leq i \leq n) \hat{\psi}'(r_i) = p'_i$.

Since by definition of $\hat{\psi}'$ we have $u' = \hat{\psi}'(v)$, $M' = \hat{\psi}'(v^*)$, $\psi' \circ V \triangleright \hat{\psi}'(v^*)$ is a transition firing in $N'$ and ($\hat{\psi}(v), \hat{\psi}'(v)) \in \overline{\mathcal{R}}$, ($\hat{\psi}(v^*), \hat{\psi}'(v^*)) \in \overline{\mathcal{R}}$.

Consequently, $\hat{\psi}'(v) = \hat{\psi}'(v^*)$ - $\hat{\psi}'(v) = \hat{\psi}'(v^*) - \hat{\psi}'(v)$. Because of additivity of place bisimulations and since $\hat{\psi}$ is an embedding, we have $(\emptyset, \hat{\psi}'(v)) \in \mathcal{R}$ and $(\emptyset, \hat{\psi}'(v^*)) \in \overline{\mathcal{R}}$. Consequently, $\hat{\psi}'(v^*) = \hat{\psi}'(v)$ and $\hat{\psi}'(v^*) = \hat{\psi}'(v^*)$.

Therefore $\hat{\psi}'$ is an embedding and $\hat{\pi}' = (\hat{C}, \hat{\pi}') \in \Pi(N')$. We have $\pi' \sim \hat{\pi}'$. Let us define $\hat{\beta} = id_{\hat{C}}$. Then $(\hat{\pi}, \hat{\pi}', \hat{\beta}) \in \mathcal{S}$.

4. As item 3, but the roles of $N$ and $N'$ are reversed. 

\section*{B Proof of Proposition 5.2.}

Let $\overline{N} = ref(N, a, D)$, $\overline{N'} = ref(N', a, D)$ and $\mathcal{R} : N \approx_{pr} N'$. By Proposition 3.1, $\mathcal{R} : N \approx_i N'$. It is enough to prove $\overline{N} \approx_i \overline{N'}$. Let us define a relation $\mathcal{S}$ as follows: $\mathcal{S} = \mathcal{R} \cup \{(p, u), (p, u') \} | p \in PD \setminus \{p_{in}, p_{out}\}, (u, u') \in \overline{\mathcal{R}}$.

Let us prove $\overline{S} : \overline{N} \approx_i \overline{N'}$.

1. $(M_{\overline{N}}, M_{\overline{N}'}) \in \mathcal{S}$, since $(M_N, M_{N'}) \in \mathcal{R}$.

2. Let $(M, M') \in \mathcal{S}$ and $M \overset{\hat{A}}{\rightarrow} \overline{M}$. Two cases are possible:
   
   (a) $\overline{u} = u \in T_N$;
   
   (b) $\overline{u} = (t, u)$, $t \in T_D$, $u \in T_N$, $l_N(u) = a$.

Let us consider the case (b), since the case (a) is obvious. Let $t = \{p\}$, $t^* = \{q\}$. Then we have:

\[
\begin{align*}
\bullet \langle t, u \rangle &= \{ \begin{array}{ll}
\bullet u, & t \in p_{in}^*; \\
\langle p, u \rangle, & \text{otherwise}.
\end{array} \\
\langle t, u \rangle^* &= \{ \begin{array}{ll}
u^*, & t \in p_{out}^*; \\
\langle q, u \rangle, & \text{otherwise}.
\end{array}
\end{align*}
\]
Four cases are possible:

(a) \( t \in p_{in} \cap p_{out}; \)

(b) \( t \in p_{in} \setminus p_{out}; \)

(c) \( t \in p_{out} \setminus p_{in}; \)

(d) \( t \notin p_{in} \cup p_{out}. \)

Let us consider the case (d), since the cases (a)–(c) are simpler. We have \( \langle t, u \rangle = \langle p, u \rangle \in M. \) Since \((M, M') \in \mathcal{S}, \) by definition of \( \mathcal{S} \) we have: \( \exists u' \in T_N : (u, u') \in \mathcal{R} \) and \((\langle p, u \rangle, \langle p, u' \rangle) \in \mathcal{S}, \) \( \langle p, u' \rangle \in M'. \)

Since \( \langle t, u' \rangle = \langle p, u' \rangle, \) then \( \langle t, u' \rangle \in \mathcal{S}, \) \( \langle t, u' \rangle \in M'. \)

Then \( \exists M' : M' \xrightarrow{(t, u')} M'. \) We have: \( l_{\overline{N}}((t, u)) = l_D(t) = l_{\overline{N}}((t, u')). \)

Since \( \langle t, u' \rangle^* = \langle q, u \rangle, \) by definition of \( \mathcal{S} \) we have \((\langle q, u \rangle, \langle q, u' \rangle) \in \mathcal{S}. \)

Since \( \langle t, u' \rangle^* = \langle q, u' \rangle, \) then \( \langle t, u' \rangle^* , \langle t, u' \rangle^* \in \mathcal{S}. \)

Hence, \((\langle t, u \rangle, \langle t, u' \rangle) \in \mathcal{S} \) and \((M, M') \in \mathcal{S}. \)

3. As item 2, but the roles of \( \overline{N} \) and \( \overline{N}' \) are reversed. \( \square \)