

Place Bisimulation Equivalences for Design of Concurrent and Sequential Systems

Igor V. Tarasyuk¹

*Theoretical Programming Laboratory,
A.P. Ershov Institute of Informatics Systems,
6, Acad. Lavrentiev ave.,
Novosibirsk, 630090, Russia
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. In addition, we consider all the mentioned equivalences on sequential nets which are a special subclass of general Petri nets modelling sequential systems, where no two actions can happen simultaneously. On this net subclass all pomset equivalences merge with the corresponding interleaving ones, and it allows us to simplify their check.

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. 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*

¹ The work is supported by Volkswagen Stiftung, grant I/70 564, INTAS-RFBR, grant 95-0378, the Foundation for Promotion to Young Scientists of Siberian Division of the Russian Academy of Sciences and DAAD.

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_{iST}) [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 (\leftrightarrow_{sbsf}) [6], step back partial word forth (\leftrightarrow_{sbpwf}) [9] and step back pomset forth (\leftrightarrow_{sbpomf}) [9] bisimulation equivalences. In [11] we supplemented them by several new relations in process semantics: step back process forth (\leftrightarrow_{sbprf}) and pomset back process forth ($\leftrightarrow_{pombprf}$) 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 (\sim_i) was proposed. In this paper, strict interleaving place bisimulation equivalence (\approx_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 (\sim_s), partial word (\sim_{pw}), pomset (\sim_{pom}), process (\sim_{pr}) place bisimulation equivalences and their strict analogues ($\approx_s, \approx_{pw}, \approx_{pom}, \approx_{pr}$) were proposed. The coincidence of \sim_i, \sim_s and \sim_{pw} was established. It was shown that all strict bisimulation equivalences coincide with \sim_{pr} . Thus, only three different equivalences remain: \sim_i, \sim_{pom} and \sim_{pr} . In addition, in these papers the polynomial algorithm of a net reduction modulo \sim_i and \sim_{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 which of them may be used for top-down design. To evaluate how equivalences respect concurrency, it is actual to consider correlation of these notions on concurrency-free (sequential) nets. In addition, it allows us to simplify check of the relations for such a net subclass because of merging some of them. 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 \sim_{pr} implies $\xleftrightarrow{\sim_{prh}}$ 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 \sim_{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 \sim_{pr} is the only place bisimulation equivalence which is preserved by SM-refinements. Thus, this equivalence may be used for the compositional reduction of nets.

The third result is a diagram of interrelations of all the equivalences on sequential nets. A merging of most of the relations in interleaving – pomset semantics is demonstrated. In particular, \sim_i coincide with \sim_{pom} on sequential nets.

2 Basic definitions

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

2.1 Nets

Let $Act = \{a, b, \dots\}$ 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:*

- $P_N = \{p, q, \dots\}$ is a set of places;
- $T_N = \{t, u, \dots\}$ is a set of transitions;

- $F_N : (P_N \times T_N) \cup (T_N \times P_N) \rightarrow \mathbf{N}$ is the flow relation with weights (\mathbf{N} denotes a set of natural numbers);
- $l_N : T_N \rightarrow Act$ is a labelling of transitions with action names.

Given labelled nets N and N' A mapping $\beta : P_N \cup T_N \rightarrow P_{N'} \cup T_{N'}$ is an *isomorphism* between N and N' , denoted by $\beta : N \simeq N'$, if β 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 $\bullet t$ and $t\bullet$ respectively, are the multisets defined in such a way: $(\bullet t)(p) = F_N(p, t)$ and $(t\bullet)(p) = F_N(t, p)$. Analogous definitions are introduced for places: $(\bullet p)(t) = F_N(t, p)$ and $(p\bullet)(t) = F_N(p, t)$. Let ${}^\circ N = \{p \in P_N \mid \bullet p = \emptyset\}$ is the set of *input* places of N and $N^\circ = \{p \in P_N \mid p\bullet = \emptyset\}$ is the set of *output* places of N .

A labelled net N is *acyclic*, if there exist no transitions $t_0, \dots, t_n \in T_N$ s.t. $t_{i-1}^\bullet \cap \bullet t_i \neq \emptyset$ ($1 \leq i \leq n$) and $t_0 = t_n$. A labelled net N is *ordinary*, if $\forall p \in P_N \bullet p$ and $p\bullet$ 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.

- $x \prec_N y \Leftrightarrow xF_N^+y$, 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 *firable* in M , if $\bullet t \subseteq M$. If t is firable in M , its firing yields a new marking $\widetilde{M} = M - \bullet t + t\bullet$, denoted by $M \xrightarrow{t} \widetilde{M}$. A marking M of a net N is *reachable*, if $M = M_N$ or there exists a reachable marking \widehat{M} of N s.t. $\widehat{M} \xrightarrow{t} M$ for some $t \in T_N$. $Mark(N)$ denotes a *set of all reachable* markings of a net N .

2.2 Partially ordered sets

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

- $X = \{x, y, \dots\}$ is some set;
- $\prec \subseteq X \times X$ is a *strict partial order* (irreflexive transitive relation) over X ;
- $l : X \rightarrow Act$ 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 ρ and ρ' , denoted by $\beta : \rho \sqsubseteq \rho'$, 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 $\rho \sqsubseteq \rho'$, if $\exists \beta : \rho \sqsubseteq \rho'$.

A mapping $\beta : X \rightarrow X'$ is an *isomorphism* between ρ and ρ' , denoted by $\beta : \rho \simeq \rho'$, if $\beta : \rho \sqsubseteq \rho'$ and $\beta^{-1} : \rho' \sqsubseteq \rho$. Two lposets ρ and ρ' 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 $C = \langle P_C, T_C, F_C, l_C \rangle$, s.t.:

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

Let us note that on the basis of any causal net C one can define lposet $\rho_C = \langle T_C, \prec_N \cap (T_C \times T_C), l_C \rangle$.

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. ${}^\circ C = L_0 \xrightarrow{v_1} \dots \xrightarrow{v_n} L_n = C^\circ$ s.t. $L_i \subseteq P_C$ ($0 \leq i \leq n$), $P_C = \cup_{i=0}^n L_i$ and $T_C = \{v_1, \dots, v_n\}$.

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:

- (i) $\varphi(P_C) \in \mathcal{M}(P_N)$ and $\varphi(T_C) \in \mathcal{M}(T_N)$, i.e. sorts are preserved;
- (ii) $\forall v \in T_C \ \bullet \varphi(v) = \varphi(\bullet v)$ and $\varphi(v)\bullet = \varphi(v\bullet)$, i.e. flow relation is respected;
- (iii) $\forall v \in T_C \ l_C(v) = l_N(\varphi(v))$, i.e. labelling is preserved.

Since embeddings respect the flow relation, if ${}^\circ C \xrightarrow{v_1} \dots \xrightarrow{v_n} C^\circ$ is a full execution of C , then $M = \varphi({}^\circ C) \xrightarrow{\varphi(v_1)} \dots \xrightarrow{\varphi(v_n)} \varphi(C^\circ) = \widetilde{M}$ 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_N 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 $\widetilde{M} = \varphi(C^\circ)$, denoted by $M \xrightarrow{\pi} \widetilde{M}$.

Let $\pi = (C, \varphi)$, $\tilde{\pi} = (\tilde{C}, \tilde{\varphi}) \in \Pi(N)$, $\hat{\pi} = (\hat{C}, \hat{\varphi}) \in \Pi(N, \varphi(C^\circ))$. A process $\tilde{\pi}$ is an *extension* of π by process $\hat{\pi}$, denoted by $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$, if $T_C \subseteq T_{\tilde{C}}$ is a left-closed set in \tilde{C} and $T_{\hat{C}} = T_{\tilde{C}} \setminus T_C$. We write $\pi \rightarrow \tilde{\pi}$, if $\exists \hat{\pi} \pi \xrightarrow{\hat{\pi}} \tilde{\pi}$. A process $\tilde{\pi}$ is an extension of π by one transition, denoted by $\pi \xrightarrow{v} \tilde{\pi}$, if $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ 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 $\mathcal{R} \subseteq \mathcal{M}(N) \times \mathcal{M}(N')$ is a \star -bisimulation between N and N' , $\star \in \{\text{interleaving, step, partial word, pomset, process}\}$, denoted by $\mathcal{R} : N \xleftrightarrow{\star} N'$, $\star \in \{i, s, pw, pom, pr\}$, if:*

- (i) $(M_N, M_{N'}) \in \mathcal{R}$.
- (ii) $(M, M') \in \mathcal{R}$, $M \xrightarrow{\hat{\pi}} \tilde{M}$,
 - (a) $|T_{\hat{C}}| = 1$, if $\star = i$;
 - (b) $\prec_{\hat{C}} = \emptyset$, if $\star = s$;
 - $\Rightarrow \exists \tilde{M}' : M' \xrightarrow{\hat{\pi}'} \tilde{M}'$, $(\tilde{M}, \tilde{M}') \in \mathcal{R}$ and
 - (a) $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$, if $\star = pw$;
 - (b) $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$, if $\star \in \{i, s, pom\}$;
 - (c) $\hat{C} \simeq \hat{C}'$, if $\star = pr$.

(iii) *As item 2, but the roles of N and N' are reversed.*

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

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' $\mathcal{R} \subseteq P_N \times P_{N'}$ be a relation between their places. The *lifting* of \mathcal{R} is a relation $\overline{\mathcal{R}} \subseteq \mathcal{M}(P_N) \times \mathcal{M}(P_{N'})$, defined as follows: $(M, M') \in \overline{\mathcal{R}} \Leftrightarrow \exists \{(p_1, p'_1), \dots, (p_n, p'_n)\} \in \mathcal{M}(\mathcal{R}) : M = \{p_1, \dots, p_n\}$, $M' = \{p'_1, \dots, p'_n\}$.

Definition 3.2 *Let N and N' be some nets. A relation $\mathcal{R} \subseteq P_N \times P_{N'}$ is a \star -place bisimulation between N and N' , $\star \in \{\text{interleaving, step, partial word, pomset, process}\}$, denoted by $\mathcal{R} : N \sim_{\star} N'$, if $\overline{\mathcal{R}} : N \xleftrightarrow{\star} N'$, $\star \in \{i, s, pw, pom, pr\}$.*

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

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

Let for some nets N and N' $t \in T_N$, $t' \in T_{N'}$. Then $(t, t') \in \overline{\mathcal{R}} \Leftrightarrow ((\bullet t, \bullet t') \in \overline{\mathcal{R}}) \wedge ((t^\bullet, t'^\bullet) \in \overline{\mathcal{R}}) \wedge (l_N(t) = l_{N'}(t'))$.

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

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

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

An important property of place bisimulations is *additivity*. Let for nets N and N' $\mathcal{R} : N \sim_\star N'$. Then $(M_1, M'_1) \in \overline{\mathcal{R}}$ and $(M_2, M'_2) \in \overline{\mathcal{R}}$ implies $((M_1 + M_2), (M'_1 + M'_2)) \in \overline{\mathcal{R}}$. In particular, if we put n tokens into each of the places $p \in P_N$ and $p' \in P_{N'}$ s.t. $(p, p') \in \mathcal{R}$, 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 3.4 [3,4] *For nets N and N' :*

- (i) $N \sim_i N' \Leftrightarrow N \sim_{pw} N'$;
- (ii) $N \sim_{pr} N' \Leftrightarrow N \approx_i N' \Leftrightarrow N \approx_{pr} N'$.

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 $\mathcal{R} \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$, 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 $N \xleftrightarrow{\star} N'$, $\star \in \{pom, pr\}$, if:*

- (i) $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$.
- (ii) $(\pi, \pi', \beta) \in \mathcal{R} \Rightarrow$

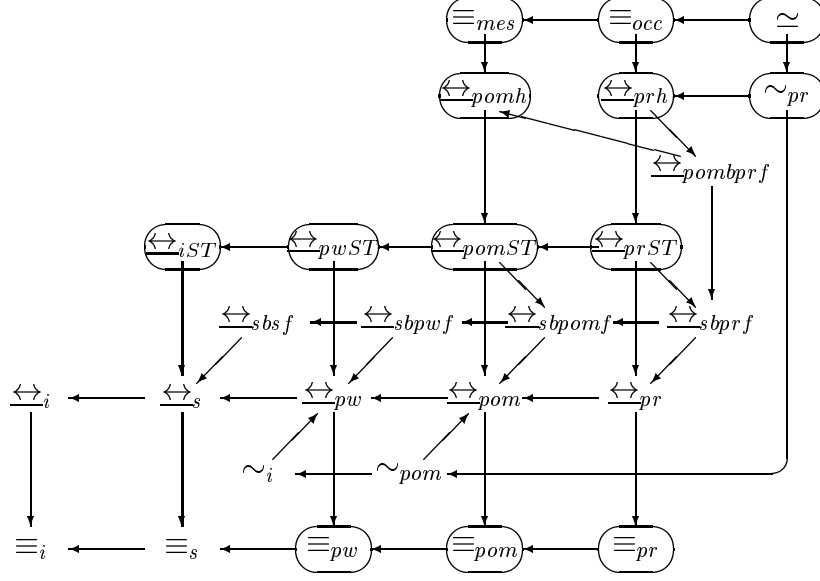


Fig. 1. Interrelations of the equivalences and their preservation by SM-refinements

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

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

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

Proposition 4.2 For nets N and $N' : N \sim_{pr} N' \Rightarrow N \underline{\leftrightarrow}_{prh} N'$.

Proof. See Appendix A. □

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 4.3 Let $\leftrightarrow, \Leftarrow \in \{\equiv, \underline{\leftrightarrow}, \sim, \simeq\}$, $\star, \star\star \in \{-, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombrf\}$. For nets N and $N' : N \leftrightarrow_{\star} N' \Rightarrow N \Leftarrow_{\star\star} N'$ iff in the graph in Figure 1 there exists a directed path from \leftrightarrow_{\star} to $\Leftarrow_{\star\star}$.

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

- The implications $\sim_{\star} \rightarrow \underline{\leftrightarrow}_{\star}$, $\star \in \{i, pom, pr\}$ are valid by the definitions.
- The implication $\sim_{pr} \rightarrow \underline{\leftrightarrow}_{prh}$ is valid by Proposition 4.2.

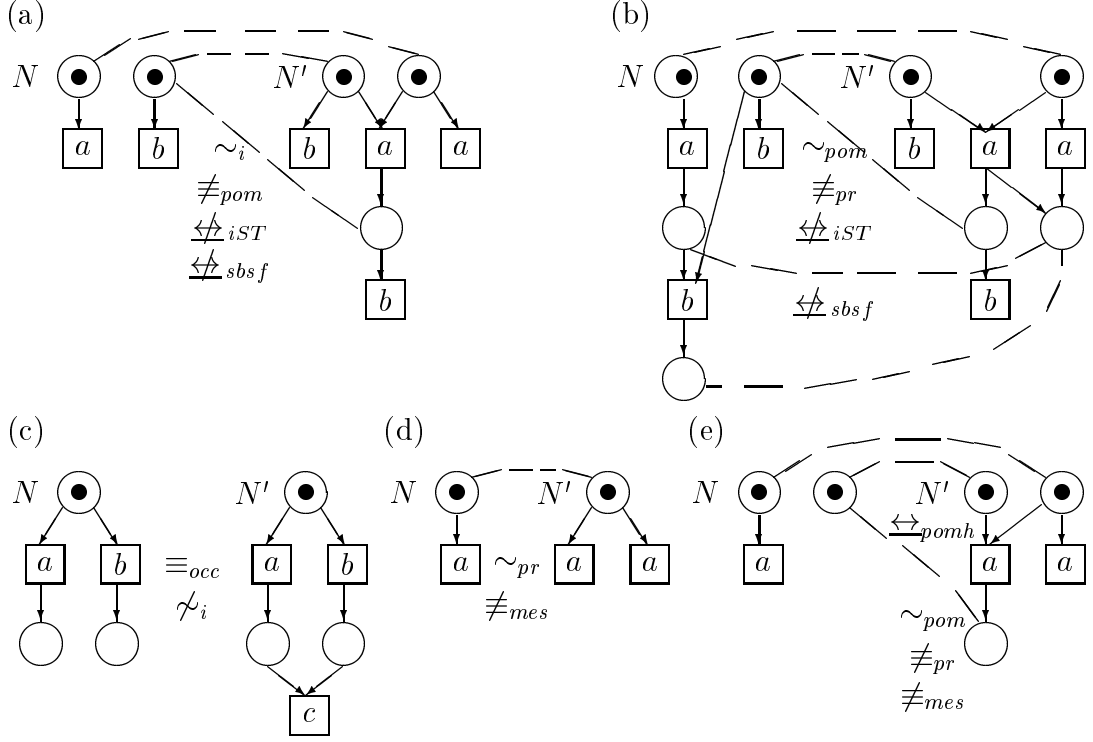


Fig. 2. Examples of place bisimulation equivalences

- 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 $\simeq \rightarrow \sim_{pr}$ is obvious.

(\Rightarrow) By Theorem 12 from [11] and the following examples (dashed lines in Figure 2 connect bisimilar places).

- In Figure 2(a), $N \sim_i N'$, 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 $N \not\sim_i N'$, 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 $N \not\leq_{iST} N'$, 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 $N \not\leq_{sbsf} N'$, 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.:*

- (i) $\forall t \in T_D \ |\bullet t| = |t\bullet| = 1$, i.e. each transition has exactly one input and one output place;
- (ii) $\exists p_{in}, p_{out} \in P_D$ s.t. $p_{in} \neq p_{out}$ and ${}^\circ D = \{p_{in}\}$, $D^\circ = \{p_{out}\}$, i.e. it is an unique input and an unique output place.
- (iii) $M_D = \{p_{in}\}$, 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 $ref(N, a, D)$, is a net $\bar{N} = \langle P_{\bar{N}}, T_{\bar{N}}, F_{\bar{N}}, l_{\bar{N}}, M_{\bar{N}} \rangle$, where:*

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

An equivalence is *preserved by refinements*, if equivalent nets remain equiv-

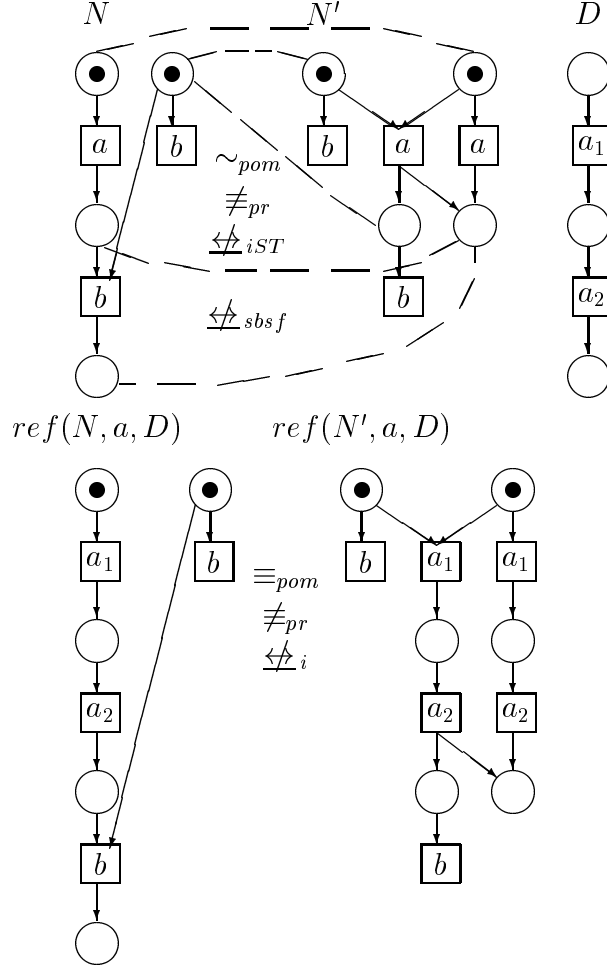


Fig. 3. The equivalences between \leftrightarrow_i and \sim_{pom} are not preserved by SM-refinements after applying any refinement operator to them accordingly. The following proposition demonstrates that some place equivalences are not preserved by SM-refinements.

Proposition 5.3 *The equivalences \sim_i and \sim_{pom} are not preserved by SM-refinements.*

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

The following proposition proves that \sim_{pr} is preserved by refinements.

Proposition 5.4 *For nets N and N' s.t. $a \in l_N(T_N) \cap l_{N'}(T_{N'})$ and SM-net $D : N \sim_{pr} N' \Rightarrow ref(N, a, D) \sim_{pr} ref(N', a, D)$.*

Proof. See Appendix B. \square

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

Theorem 5.5 *Let $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \sim, \simeq\}$ and $\star \in \{-, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombprf\}$. For nets N and N' s.t. $a \in l_N(T_N) \cap l_{N'}(T_{N'})$ and SM-net $D : N \leftrightarrow_\star N' \Rightarrow ref(N, a, D) \leftrightarrow_\star ref(N', a, D)$ iff the equivalence \leftrightarrow_\star is in oval in Figure 1.*

Proof. By Theorem 18 from [11] and Propositions 5.3 and 5.4. □

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 The equivalences on sequential nets

Let us consider the equivalences on sequential nets, where no two transitions can be fired concurrently in any reachable marking. Sequential nets are a subclass of general Petri nets used to model sequential systems in which simultaneous occurrence of several actions is impossible. In many applications, it is enough to use sequential systems, not concurrent, so it is worth investigating equivalences on the mentioned above net subclass.

Definition 6.1 *A net $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ is sequential, if $\forall M \in Mark(N) \neg \exists t, u \in T_N : \bullet t + \bullet u \subseteq M$.*

Proposition 6.2 [11] *For sequential nets N and N' :*

- (i) $N \equiv_i N' \Leftrightarrow N \equiv_{pom} N'$;
- (ii) $N \underline{\leftrightarrow}_i N' \Leftrightarrow N \underline{\leftrightarrow}_{pomh} N'$;
- (iii) $N \underline{\leftrightarrow}_{pr} N' \Leftrightarrow N \underline{\leftrightarrow}_{pombprf} N'$.

In Figure 4, the groups of equivalences which are merged on sequential nets, are marked by dashed lines.

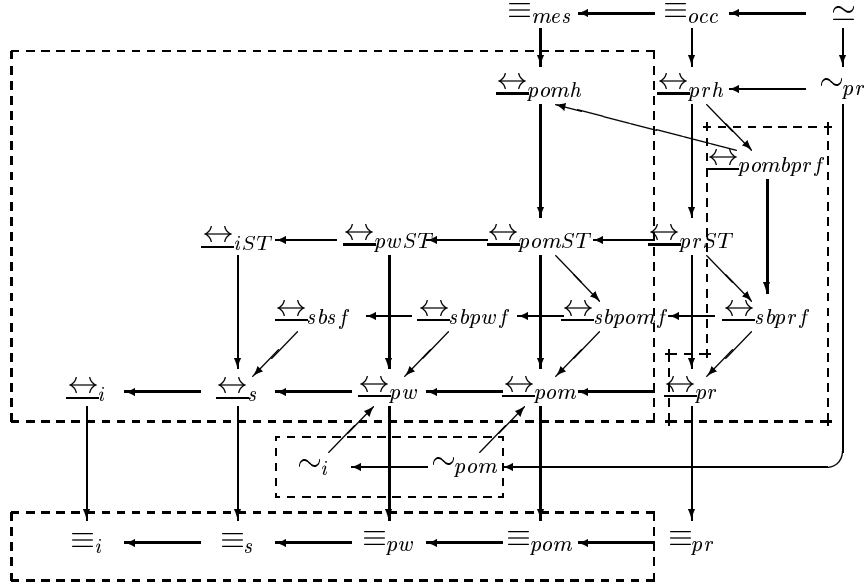


Fig. 4. Merging of the equivalences on sequential nets

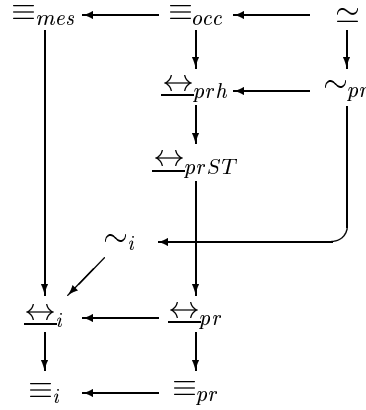


Fig. 5. Interrelations of the equivalences on sequential nets

Proposition 6.3 For sequential nets N and N' : $N \sim_i N' \Leftrightarrow N \sim_{pom} N'$.

Proof. (\Leftarrow) By Theorem 4.3.

(\Rightarrow) We have $\exists \mathcal{R} : N \sim_i N'$. By definition of place bisimulations, $\overline{\mathcal{R}} : N \xleftrightarrow{i} N'$. By Proposition 6.2, $\overline{\mathcal{R}} : N \xleftrightarrow{pom} N'$. Again by definition of place bisimulations, $\mathcal{R} : N \sim_{pom} N'$. \square

Theorem 6.4 Let $\leftrightarrow, \Leftrightarrow \in \{\equiv, \xleftrightarrow{\quad}, \sim, \simeq\}$, $\star, \star\star \in \{-, i, pr, prST, prh, mes, occ\}$. For sequential nets N and N' : $N \leftrightarrow_{\star} N' \Rightarrow N \Leftrightarrow_{\star\star} N'$ iff in the graph in Figure 5 there exists a directed path from \leftrightarrow_{\star} to $\Leftrightarrow_{\star\star}$.

Proof. (\Leftarrow) By Theorem 4.3.

(\Rightarrow) By Theorem 15 from [11] and the following examples.

- In Figure 2(e), $N \sim_i N'$, but $N \not\equiv_{pr} N'$, since only in the net N' the

transition with label a has two input places.

- In Figure 2(c), $N \equiv_{occ} N'$, but $N \not\sim_i N'$.
- In Figure 2(d), $N \sim_{pr} N'$, but $N \not\equiv_{mes} N'$.

□

Thus, we obtained the complete picture of interrelations of the equivalences on sequential nets which can be used to simplify equivalence check when modelling of sequential systems.

7 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. In addition, we proved that on sequential nets \sim_i coincide with \sim_{pom} , and it simplifies the check of the latter relation.

Further research may consist in the investigation of analogues of the considered equivalences on Petri nets with τ -actions (τ -equivalences). τ -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 τ -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 τ -analogues of all the equivalence relations considered in this paper and exam them following the same pattern.

References

- [1] AUTANT C., BELMESK Z., SCHNOEBELEN PH. *Strong bisimilarity on nets revisited. Extended abstract. LNCS 506*, p. 295–312, June 1991.
- [2] AUTANT C., PFISTER W., SCHNOEBELEN PH. *Place bisimulations for the reduction of labelled Petri nets with silent moves. Proceedings of International Conference on Computing and Information*, 1994.
- [3] AUTANT C., SCHNOEBELEN PH. *Place bisimulations in Petri nets. LNCS 616*, p. 45–61, June 1992.
- [4] AUTANT C. *Petri nets for the semantics and the implementation of parallel processes. Ph.D. thesis*, Institut National Polytechnique de Grenoble, May 1993 (in French).

- [5] BEST E., DEVILLERS R., KIEHN A., POMELLO L. *Concurrent bisimulations in Petri nets. Acta Informatica* **28**, p. 231–264, 1991.
- [6] CHERIEF F. *Back and forth bisimulations on prime event structures. LNCS* **605**, p. 843–858, June 1992.
- [7] VAN GLABBEEK R.J., VAANDRAGER F.W. *Petri net models for algebraic theories of concurrency. LNCS* **259**, p. 224–242, 1987.
- [8] POMELLO L., ROZENBERG G., SIMONE C. *A survey of equivalence notions for net based systems. LNCS* **609**, p. 410–472, 1992.
- [9] PINCHINAT S. *Bisimulations for the semantics of reactive systems. Ph.D. thesis, Institut National Polytechnique de Grenoble, January 1993 (in French).*
- [10] TARASYUK I.V. *Petri net equivalences for design of concurrent systems. Proceedings of 5th Workshop on Concurrency, Specification and Programming - 96 (CSP'96), September 25–27, 1996, Informatik-Bericht* **69**, p. 190–204, Institut für Informatik, Humboldt-Universität zu Berlin, Berlin, Germany, 1996.
- [11] TARASYUK I.V. *Back-forth equivalences for design of concurrent systems. S. Adian, A. Nerode, eds., Proceedings of 4th International Symposium on Logical Foundations of Computer Science - 97 (LFCS'97), LNCS* **1234**, p. 374–384, Yaroslavl, 1997.
- [12] VOGLER W. *Modular construction and partial order semantics of Petri nets. LNCS* **625**, 252 p., 1992.

A Proof of Proposition 4.2.

By Proposition 3.4, $\exists \mathcal{R} : N \approx_{pr} N'$. Then $\overline{\mathcal{R}} : N \leftrightarrow_{pr} N'$ and transitions of N and N' are related by $\overline{\mathcal{R}}$. Let us define a relation \mathcal{S} as follows: $\mathcal{S} = \{(\pi, \pi', \beta) \mid \pi = (C, \varphi) \in \Pi(N), \pi' = (C, \varphi') \in \Pi(N'), \beta = id_{T_C}, \forall r \in P_C (\varphi(r), \varphi'(r)) \in \mathcal{R}, \forall v \in T_C (\varphi(v), \varphi'(v)) \in \overline{\mathcal{R}}\}$. Let us prove $\mathcal{S} : N \leftrightarrow_{prh} N'$.

- (i) Obviously, $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{S}$.
- (ii) By definition of \mathcal{S} , $(\pi, \pi', \beta) \in \mathcal{S} \Rightarrow \beta : \rho_C \simeq \rho_{C'}$ and $C \simeq C'$;
- (iii) Let $(\pi, \pi', \beta) \in \mathcal{S}$, $\pi = (C, \varphi)$, $\pi' = (C, \varphi')$ and $\pi \xrightarrow{v} \tilde{\pi}$, $\tilde{\pi} = (\tilde{C}, \tilde{\varphi})$.

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 $\exists u', \widetilde{M}' : \varphi'(\bullet v) \xrightarrow{u'} \widetilde{M}'$, $(\tilde{\varphi}(v), u') \in \overline{\mathcal{R}}$ and $(\tilde{\varphi}(v\bullet), \widetilde{M}') \in \overline{\mathcal{R}}$.

Let $v\bullet = \{r_1, \dots, r_n\}$, $\widetilde{M}' = \{p'_1, \dots, p'_n\}$, $\forall i (1 \leq i \leq n) (\tilde{\varphi}(r_i), p'_i) \in \mathcal{R}$. Let us define a mapping $\tilde{\varphi}'$ as follows: $\tilde{\varphi}'|_{(P_C \cup T_C)} = \varphi'$, $\tilde{\varphi}'(v) = u'$, $\forall i (1 \leq i \leq n) \tilde{\varphi}'(r_i) = p'_i$.

Since by definition of $\tilde{\varphi}'$ we have $u' = \tilde{\varphi}'(v)$, $\widetilde{M}' = \tilde{\varphi}'(v\bullet)$, $\varphi'(\bullet v) = \tilde{\varphi}'(\bullet v)$, then $\tilde{\varphi}'(\bullet v) \xrightarrow{\tilde{\varphi}'(v)} \tilde{\varphi}'(v\bullet)$ is a transition firing in N' and $(\tilde{\varphi}(v), \tilde{\varphi}'(v)) \in \overline{\mathcal{R}}$, $(\tilde{\varphi}(v\bullet), \tilde{\varphi}'(v\bullet)) \in \overline{\mathcal{R}}$.

Consequently, $\tilde{\varphi}(\bullet v) - \bullet \tilde{\varphi}(v) = \tilde{\varphi}(v\bullet) - \tilde{\varphi}(v)\bullet$ and $\tilde{\varphi}'(\bullet v) - \bullet \tilde{\varphi}'(v) = \tilde{\varphi}'(v\bullet) - \tilde{\varphi}'(v)\bullet$. Because of additivity of place bisimulations and since $\tilde{\varphi}$ is an embedding, we have $(\emptyset, \tilde{\varphi}'(\bullet v) - \bullet \tilde{\varphi}'(v)) \in \overline{\mathcal{R}}$ and $(\emptyset, \tilde{\varphi}'(v\bullet) - \tilde{\varphi}'(v)\bullet) \in \overline{\mathcal{R}}$. Consequently, $\tilde{\varphi}'(\bullet v) = \bullet \tilde{\varphi}'(v)$ and $\tilde{\varphi}'(v\bullet) = \tilde{\varphi}'(v)\bullet$. Therefore $\tilde{\varphi}'$ is an embedding and $\tilde{\pi}' = (\tilde{C}, \tilde{\varphi}') \in \Pi(N')$. We have $\pi' \xrightarrow{v} \tilde{\pi}'$. Let us define $\tilde{\beta} = id_{T_{\tilde{C}}}$. Then $(\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in \mathcal{S}$.

(iv) As item 3, but the roles of N and N' are reversed. \square

B Proof of Proposition 5.4.

Let $\overline{N} = ref(N, a, D)$, $\overline{N}' = ref(N', a, D)$ and $\mathcal{R} : N \sim_{pr} N'$. By Proposition 3.4, $\mathcal{R} : N \approx_i N'$. It is enough to prove $\overline{N} \approx_i \overline{N}'$. Let us define a relation \mathcal{S} as follows: $\mathcal{S} = \mathcal{R} \cup \{(\langle p, u \rangle, \langle p, u' \rangle) \mid p \in P_D \setminus \{p_{in}, p_{out}\}, (u, u') \in \overline{\mathcal{R}}\}$. Let us prove $\mathcal{S} : \overline{N} \approx_i \overline{N}'$.

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

(ii) Let $(M, M') \in \mathcal{S}$ and $M \xrightarrow{\tilde{u}} \widetilde{M}$. Two cases are possible:

(a) $\tilde{u} = u \in T_N$;

(b) $\tilde{u} = \langle t, u \rangle$, $t \in T_D$, $u \in T_N$, $l_N(u) = a$.

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

$$\bullet \langle t, u \rangle = \begin{cases} \bullet u, & t \in p_{in}^\bullet; \\ \langle p, u \rangle, & \text{otherwise.} \end{cases} \quad \langle t, u \rangle \bullet = \begin{cases} u^\bullet, & t \in \bullet p_{out}; \\ \langle q, u \rangle, & \text{otherwise.} \end{cases}$$

Four cases are possible:

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

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

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

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

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

Then $\exists \widetilde{M}' : M' \xrightarrow{\langle t, u' \rangle} \widetilde{M}'$. We have: $l_{\overline{N}}(\langle t, u \rangle) = l_D(t) = l_{\overline{N}'}(\langle t, u' \rangle)$. Since $\langle t, u \rangle \bullet = \langle q, u \rangle$, by definition of \mathcal{S} we have $(\langle q, u \rangle, \langle q, u' \rangle) \in \mathcal{S}$. Since $\langle t, u' \rangle \bullet = \langle q, u' \rangle$, then $(\langle t, u \rangle \bullet, \langle t, u' \rangle \bullet) \in \overline{\mathcal{S}}$.

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

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