Back-forth Equivalences for Design of Concurrent Systems *

Igor V. Tarasyuk

Institute of Informatics Systems,
Siberian Division of the Russian Academy of Sciences,
Lavrentieva ave. 6, 630090, Novosibirsk, Russia
Phone: +7 3832 35 03 60
Fax: +7 3832 32 34 94
E-mail: itar@iis.nsk.su

Abstract. The paper is devoted to the investigation of behavioural equivalences of concurrent systems modelled by Petri nets. Back-forth bisimulation equivalences known from the literature are supplemented by new ones, and their relationship with basic behavioural equivalences is examined for the whole class of Petri nets as well as for their subclass of sequential nets. In addition, the preservation of all the equivalence notions by refinements is examined.

1 Introduction

The notion of equivalence is central in any theory of systems. It allows to compare systems taking into account particular aspects of their behaviour.

Petri nets became a popular formal model for design of concurrent and distributed systems. One of the main advantages of Petri nets is their ability for structural characterization of three fundamental features of concurrent computations: causality, nondeterminism and concurrency.

In recent years, a wide range of semantic equivalences was proposed in concurrency theory. Some of them were either directly defined or transferred from other formal models to Petri nets. The following basic notions of equivalences for Petri nets are known from the literature (some of them were introduced by the author in [15] to obtain the complete set of relations in interleaving/true concurrency and linear time/branching time semantics).

- *Trace equivalences* (respect only protocols of nets functioning): interleaving ($\equiv_i$) [8], step ($\equiv_s$) [12], pomset ($\equiv_{pom}$) [7] and process ($\equiv_{pr}$) [15].
- *(Usual) bisimulation equivalences* (respect branching structure of nets functioning): interleaving ($\underline{\leftrightarrow}_i$) [11], step ($\underline{\leftrightarrow}_s$) [10], partial word ($\underline{\leftrightarrow}_{pw}$) [16], pomset ($\underline{\leftrightarrow}_{pom}$) [3] and process ($\underline{\leftrightarrow}_{pr}$) [1].

* The work is supported by Volkswagen Fund, grant I/70 564 and Russian Fund for Basic Research, grant 96-01-01655

– *ST-bisimulation equivalences* (respect the duration of transition occurrences in nets functioning): interleaving ($\underline{\leftrightarrow}_{iST}$) [7], partial word ($\underline{\leftrightarrow}_{pwST}$) [16], pomset ($\underline{\leftrightarrow}_{pomST}$) [16] and process ($\underline{\leftrightarrow}_{prST}$) [15].
– *History preserving bisimulation equivalences* (respect the "past" or "history" of nets functioning): pomset ($\underline{\leftrightarrow}_{pomST}$) [14] and process ($\underline{\leftrightarrow}_{prST}$) [15].
– *Conflict preserving equivalences* (completely respect conflicts in nets): multi event structure ($\equiv_{mes}$) [15] and occurrence ($\equiv_{occ}$) [7].
– *Isomorphism* ($\simeq$) (i.e. coincidence of nets up to renaming of places and transitions).

Back-forth bisimulation equivalences are based on the idea that bisimulation relation do not only require systems to simulate each other behaviour in the forward direction (as usually) but also when going back in history. They are closely connected with equivalences of logics with past modalities.

These equivalence notions were initially introduced in [9] in the framework of transition systems. It was shown that back-forth variant ($\underline{\leftrightarrow}_{ibif}$) of interleaving bisimulation equivalence coincide with ordinary $\underline{\leftrightarrow}_i$.

In [4–6] the new variants of step ($\underline{\leftrightarrow}_{sbsf}$), partial word ($\underline{\leftrightarrow}_{pwbpwf}$) and pomset ($\underline{\leftrightarrow}_{pombpomf}$) back-forth bisimulation equivalences were defined in the framework of prime event structures and compared with usual, ST- and history preserving bisimulation equivalences. It was demonstrated that among all back-forth bisimulation equivalences only $\underline{\leftrightarrow}_{pombpomf}$ is preserved by refinements (it coincides with $\underline{\leftrightarrow}_{pomh}$ which has such a property).

In [13] the new idea of differentiating the kinds of back and forth simulations appeared (following this idea, it is possible, for example, to define step back – pomset forth bisimulation equivalence ($\underline{\leftrightarrow}_{sbpomf}$)). The set of all possible back-forth equivalence notions was proposed in interleaving, step, partial word and pomset semantics. Two new notions which do not coincide with known ones were proposed: step back – partial word forth ($\underline{\leftrightarrow}_{sbpwf}$) and step back – pomset forth ($\underline{\leftrightarrow}_{sbpomf}$) bisimulation equivalences. It was proved that the former is not preserved by refinements, and the question was addressed about the latter.

To choose most appropriate behavioural viewpoint on systems to be modelled, it is very important to have a complete set of equivalence notions in all semantics and understand their interrelations. This branch of research is usually called *comparative concurrency semantics*. To clarify the nature of equivalences and evaluate how they respect a concurrency, it is actual to consider also correlation of these notions on concurrency-free (sequential) nets. Treating equivalences for preservation by refinements allows one to decide which of them may be used for top-down design.

Working in the framework of Petri nets, in this paper we extend the set of back-forth equivalences from [13] by that of induced by process semantics and obtain two new notions which cannot be reduced to the known ones: step back – process forth ($\underline{\leftrightarrow}_{sbprf}$) and pomset back – process forth ($\underline{\leftrightarrow}_{pombprf}$) bisimulation equivalences. We compare all back-forth equivalences with the set of basic behavioural notions from [15] and complete the results of [6, 13].

In addition, we investigate the interrelations of all the equivalence notions on sequential nets. The merging of most of the equivalence relations in interleaving – pomset semantics is demonstrated. We prove that on sequential nets back-forth equivalences coincide with usual forth ones.

In [2], SM-refinement operator for Petri nets was proposed, which "replaces" transitions of nets by SM-nets, a special subclass of state machine nets. We treat all the considered equivalence notions for preservation by SM-refinements and establish that among back-forth relations only $\underline{\leftrightarrow}_{pombpomf}$ and $\underline{\leftrightarrow}_{prbprf}$ are preserved by SM-refinements So, we obtained the negative answer to the question from [13]: $\underline{\leftrightarrow}_{sbpomf}$ (and even more strict $\underline{\leftrightarrow}_{pombprf}$) is not preserved by refinements.

2 Basic definitions

In this section we give some basic definitions used further.

2.1 Labelled nets

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

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

Given labelled nets $N = \langle P_N, T_N, F_N, l_N \rangle$ and $N' = \langle P_{N'}, T_{N'}, F_{N'}, l_{N'} \rangle$. A mapping $\beta : N \to N'$ is an *isomorphism* between N and N', denoted by $\beta : N \simeq N'$, if:

1. β is a bijection such that $\beta(P_N) = P_{N'}$ and $\beta(T_N) = T_{N'}$;
2. $\forall t \in T_N \; l_N(t) = l_{N'}(\beta(t))$;
3. $\forall t \in T_N \; {}^\bullet\beta(t) = \beta({}^\bullet t)$ and $\beta(t)^\bullet = \beta(t^\bullet)$.

Labelled nets N and N' are *isomorphic*, denoted by $N \simeq N'$, if there exists an isomorphism $\beta : N \simeq N'$.

Given a labelled net N and some transition $t \in T_N$, the *precondition* and *postcondition* 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 a set of *initial (input)* places of N and $N^\circ = \{p \in P_N \mid p^\bullet = \emptyset\}$ is a set of *final (output)* places of N.

A labelled net N is *acyclic*, if there exist no transitions $t_0, \ldots, t_n \in T_N$ such that $t_{i-1}^\bullet \cap {}^\bullet t_i \neq \emptyset$ $(1 \leq i \leq n)$ 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 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 (*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 $\forall t \in T\ (\downarrow_N t) \cap T_N \subseteq T$.

2.2 Marked nets

A *marking* of a labelled net N is a multiset $M \in \mathcal{M}(P_N)$.

Definition 2. *A* marked net (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, firing it yields a new marking $M' = M - {}^\bullet t + t^\bullet$, denoted by $M \xrightarrow{t} M'$. A marking M of a net N is *reachable*, if $M = M_N$ or there exists a reachable marking M' of N such that $M' \xrightarrow{t} M$ for some $t \in T_N$. $Mark(N)$ denotes a *set of all reachable* markings of a net N.

2.3 Partially ordered sets

Definition 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*;*
- $l : X \to Act$ *is a* labelling *function.*

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

A mapping $\beta : X \to X'$ is a *label-preserving bijection* between ρ and ρ', denoted by $\beta : \rho \approx \rho'$, if:

1. β is a bijection;
2. $\forall x \in X\ l(x) = l'(\beta(x))$.

We write $\rho \approx \rho'$, if there exists a label-preserving bijection $\beta : \rho \approx \rho'$.

A mapping $\beta : X \to X'$ is a *homomorphism* between ρ and ρ', denoted by $\beta : \rho \sqsubseteq \rho'$, if:

1. $\beta : \rho \approx \rho'$;
2. $\forall x, y \in X\ x \prec y \;\Rightarrow\; \beta(x) \prec' \beta(y)$.

We write $\rho \sqsubseteq \rho'$, if there exists a homomorphism $\beta : \rho \sqsubseteq \rho'$.

A mapping $\beta : X \to X'$ is an *isomorphism* between ρ and ρ', denoted by $\beta : \rho \simeq \rho'$, if $\beta : \rho \sqsubseteq \rho'$ and $\beta^{-1} : \rho' \sqsubseteq \rho$. Lposets ρ and ρ' are *isomorphic*, denoted by $\rho \simeq \rho'$, if there exists an isomorphism $\beta : \rho \simeq \rho'$.

Definition 4. Partially ordered multiset (pomset) *is an isomorphism class of lposets.*

2.4 C-processes

Definition 5. *A* causal net *is acyclic ordinary labelled net* $C = \langle P_C, T_C, F_C, l_C \rangle$*, s.t:*

1. $\forall r \in P_C \ |{}^\bullet r| \leq 1$ *and* $|r^\bullet| \leq 1$*, i.e. places are unbranched;*
2. $|\downarrow_C x| < \infty$*, i.e. a set of causes is finite.*

Let us note that on the basis of any causal net $C = \langle P_C, T_C, F_C, l_C \rangle$ 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 [1]: if C is a causal net, then there exists an occurrence sequence ${}^\circ C = L_0 \stackrel{v_1}{\rightarrow} \cdots \stackrel{v_n}{\rightarrow} L_n = C^\circ$ such that $L_i \subseteq P_C$ $(0 \leq i \leq n)$, $P_C = \cup_{i=0}^n L_i$ and $T_C = \{v_1, \ldots, v_n\}$. Such a sequence is called a *full execution* of C.

Definition 6. *Given a net N and a causal net C. A mapping $\varphi : P_C \cup T_C \to P_N \cup T_N$ is an* embedding *C into N, denoted by $\varphi : C \to N$, 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. $\forall v \in T_C \ l_C(v) = l_N(\varphi(v))$*, i.e. labelling is preserved;*
3. $\forall v \in T_C \ {}^\bullet\varphi(v) = \varphi({}^\bullet v)$ *and* $\varphi(v)^\bullet = \varphi(v^\bullet)$*, i.e. flow relation is respected.*

Since embeddings respect the flow relation, if ${}^\circ C \stackrel{v_1}{\rightarrow} \cdots \stackrel{v_n}{\rightarrow} C^\circ$ is a full execution of C, then $M = \varphi({}^\circ C) \stackrel{\varphi(v_1)}{\longrightarrow} \cdots \stackrel{\varphi(v_n)}{\longrightarrow} \varphi(C^\circ) = M'$ is an occurrence sequence in N.

Definition 7. *A* fireable in marking M C-process (process) *of a net N is a pair $\pi = (C, \varphi)$, where C is a causal net and $\varphi : C \to N$ is an embedding such that $M = \varphi({}^\circ C)$. A fireable in M_N process is a* process *of N.*

We write $\Pi(N, M)$ for a *set of all fireable* 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 $\pi_N = (C_N, \varphi_N) \in \Pi(N)$, such that $T_{C_N} = \emptyset$. If $\pi \in \Pi(N, M)$, then firing of this process transforms a marking M into $M' = M - \varphi({}^\circ C) + \varphi(C^\circ) = \varphi(C^\circ)$, denoted by $M \stackrel{\pi}{\rightarrow} M'$.

Let $\pi = (C, \varphi)$, $\tilde{\pi} = (\tilde{C}, \tilde{\varphi}) \in \Pi(N)$ and $\hat{\pi} = (\hat{C}, \hat{\varphi}) \in \Pi(N, \varphi(C^\circ))$.

A process $\tilde{\pi}$ is an *extension* of π *by process* $\hat{\pi}$, denoted by $\pi \stackrel{\hat{\pi}}{\rightarrow} \tilde{\pi}$, if $T_C \subseteq T_{\tilde{C}}$ is a left-closed set in $\tilde{C}$ and $T_{\hat{C}} = T_{\tilde{C}} \setminus T_C$.

A process $\tilde{\pi}$ is an extension of a process π *by one transition* $v \in T_{\tilde{C}}$, denoted by $\pi \stackrel{v}{\rightarrow} \tilde{\pi}$, if $\pi \stackrel{\hat{\pi}}{\rightarrow} \tilde{\pi}$ and $T_{\hat{C}} = \{v\}$.

A process $\tilde{\pi}$ is an extension of a process π *by sequence of transitions* $\sigma = v_1 \cdots v_n \in T_{\tilde{C}}^*$, denoted by $\pi \stackrel{\sigma}{\rightarrow} \tilde{\pi}$, if $\exists \pi_i \in \Pi(N)$ $(1 \leq i \leq n)$ $\pi \stackrel{v_1}{\rightarrow} \pi_1 \stackrel{v_2}{\rightarrow} \ldots \stackrel{v_n}{\rightarrow} \pi_n = \tilde{\pi}$.

3 Back-forth bisimulation equivalences

In this section, in the framework of Petri nets, we supplement the definitions of back-forth bisimulation equivalences [13] by the new notions induced by process semantics and compare them with basic ones.

3.1 Definitions of back-forth bisimulation equivalences

The definitions of back-forth bisimulation equivalences are based on the following notion of sequential run.

Definition 8. *A* sequential run *of a net N is a pair (π, σ), where:*

- *a process $\pi \in \Pi(N)$ contains the information about causal dependencies of transitions which brought to this state;*
- *a sequence $\sigma \in T_C^*$ such that $\pi_N \xrightarrow{\sigma} \pi$, contains the information about the order in which the transitions occur which brought to this state.*

Let us denote the *set of all sequential runs* of a net N by $Runs(N)$.

The *initial* sequential run of a net N is a pair (π_N, ε), where ε is an empty sequence.

Let (π, σ), $(\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$. We write $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$, if $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$, $\exists \hat{\sigma} \in T_{\hat{C}}^*$ $\pi \xrightarrow{\hat{\sigma}} \tilde{\pi}$ and $\tilde{\sigma} = \sigma\hat{\sigma}$.

Definition 9. *Let N and N' be some nets. A relation $\mathcal{R} \subseteq Runs(N) \times Runs(N')$ is a* $\star$-back $\star\star$-forth bisimulation *between N and N', $\star, \star\star \in$* {interleaving, step, partial word, pomset, process}*, denoted by $\mathcal{R} : N \underline{\leftrightarrow}_{\star b \star\star f} N'$, $\star, \star\star \in \{i, s, pw, pom, pr\}$, if:*

1. *$((\pi_N, \varepsilon), (\pi_{N'}, \varepsilon)) \in \mathcal{R}$.*
2. *$((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$*
   - *(back)*
     $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$,
     (a) $|T_{\hat{C}}| = 1$, if $\star = i$;
     (b) $\prec_{\hat{C}} = \emptyset$, if $\star = s$;
     $\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') :\ (\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\hat{\pi}'} (\pi', \sigma'),\ ((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ *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 $\star = pr$;
   - *(forth)*
     $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$,
     (a) $|T_{\hat{C}}| = 1$, if $\star\star = i$;
     (b) $\prec_{\hat{C}} = \emptyset$, if $\star\star = s$;
     $\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') :\ (\pi', \sigma') \xrightarrow{\hat{\pi}'} (\tilde{\pi}', \tilde{\sigma}'),\ ((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ *and*
     (a) $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$, if $\star\star = pw$;
     (b) $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$, if $\star\star \in \{i, s, pom\}$;
     (c) $\hat{C} \simeq \hat{C}'$, if $\star\star = pr$.
3. *As item 2, but the roles of N and N' are reversed.*

Two nets N and N' $\star$-back $\star\star$-forth bisimulation equivalent*, $\star, \star\star \in$* {interleaving, step, partial word, pomset, process}*, denoted by $N \underline{\leftrightarrow}_{\star b \star\star f} N'$, if $\exists \mathcal{R} : N \underline{\leftrightarrow}_{\star b \star\star f} N'$, $\star, \star\star \in \{i, s, pw, pom, pr\}$.*

3.2 Interrelations of back-forth bisimulation equivalences

In back-forth bisimulations, it is possible to move back from a state only along the history which brought to the state. Such a determinism implies merging of some equivalences.

Proposition 1. *Let* $\star \in \{i, s, pw, pom, pr\}$*. For nets* N *and* N' *the following holds:*

1. $N \underline{\leftrightarrow}_{pwb\star f} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{pomb\star f} N'$;
2. $N \underline{\leftrightarrow}_{\star bif} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{\star b\star f} N'$.

Hence, interrelations of the remaining back-forth equivalences may be represented by the graph in Figure 1.

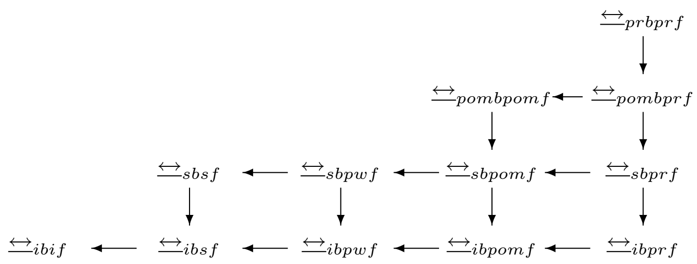

Fig. 1. Interrelations of back-forth bisimulation equivalences

3.3 Interrelations of back-forth bisimulation and basic equivalences

Let us consider how back-forth equivalences are connected with basic ones.

Proposition 2. *Let* $\star \in \{i, s, pw, pom, pr\}$, $\star\star \in \{pom, pr\}$*. For nets* N *and* N' *the following holds:*

1. $N \underline{\leftrightarrow}_{ib\star f} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{\star} N'$;
2. $N \underline{\leftrightarrow}_{\star\star b\star\star f} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{\star\star h} N'$;
3. $N \underline{\leftrightarrow}_{\star\star ST} N' \;\Rightarrow\; N \underline{\leftrightarrow}_{sb\star\star f} N'$.

Theorem 1. *Let* $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$ *and* $\star, \star\star \in \{i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombprf\}$*. For nets* N *and* $N' the following holds: N \leftrightarrow_{\star} N' \;\Rightarrow\; N \leftrightarrow_{\star\star} N'$ *iff in graph in Figure 2 there exists a directed path from* $\leftrightarrow_{\star}$ *to* $\leftrightarrow_{\star\star}$.

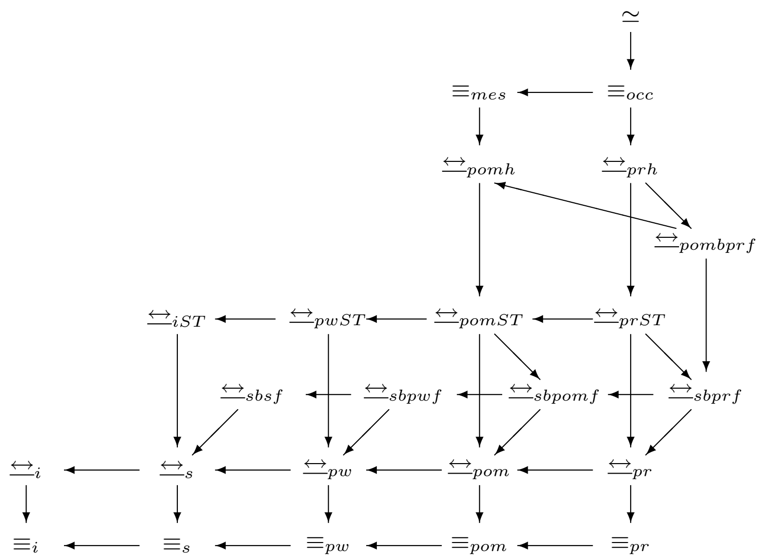

Fig. 2. Interrelations of back-forth bisimulation and basic equivalences

4 Investigation of the equivalences on sequential nets

Let us consider the influence of concurrency on interrelations of the equivalences.

Definition 10. *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 3. *For sequential nets* N *and* N' *the following holds:*

1. $N \equiv_i N' \;\Leftrightarrow\; N \equiv_{pom} N'$;
2. $N \underline{\leftrightarrow}_i N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{pomh} N'$;
3. $N \underline{\leftrightarrow}_{pr} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{pombprf} N'$.

Theorem 2. *Let* $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$, $\star, \star\star \in \{i, pr, prST, prh, mes, occ\}$. *For sequential nets* N *and* N' *the following holds:* $N \leftrightarrow_\star N' \;\Rightarrow\; N \leftrightarrow_{\star\star} N'$ *iff in graph in Figure 3 there exists a directed path from* $\leftrightarrow_\star$ *to* $\leftrightarrow_{\star\star}$.

5 Preservation of the equivalences by refinements

Let us consider which equivalences may be used for top-down design.

Definition 11. *An* SM-net *is a net* $D = \langle P_D, T_D, F_D, l_D, M_D\rangle$ *such that:*

1. $\exists p_{in},\ p_{out} \in P_D$ *such that* $p_{in} \neq p_{out}$ *and* ${}^\circ D = \{p_{in}\},\ D^\circ = \{p_{out}\}$, *i.e. net* D *has unique input and unique output place.*

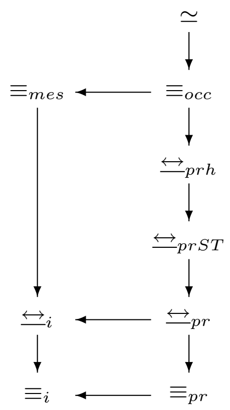

Fig. 3. The equivalences on sequential nets

2. *$M_D = \{p_{in}\}$ and $\forall M \in Mark(D)$ $(p_{out} \in M \;\Rightarrow\; M = \{p_{out}\})$, i.e. at the beginning there is unique token in p_{in}, and at the end there is unique token in p_{out};*
3. *$p_{in}^{\bullet}$ and ${}^{\bullet}p_{out}$ are proper sets (not multisets), i.e. p_{in} (respectively p_{out}) represents a set of all tokens consumed (respectively produced) for any refined transition.*
4. *$\forall t \in T_D$ $|{}^{\bullet}t| = |t^{\bullet}| = 1$, i.e. each transition has exactly one input and one output place.*

SM-refinement operator "replaces" all transitions with particular label of a net by SM-net.

Definition 12. *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 (up to isomorphism) a net $\overline{N} = \langle P_{\overline{N}}, T_{\overline{N}}, F_{\overline{N}}, l_{\overline{N}}, M_{\overline{N}}\rangle$, where:*

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

Some equivalence on nets is *preserved by refinements*, if equivalent nets remain equivalent after applying any refinement operator to them accordingly.

Theorem 3. *Let* $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$ *and* $\star \in \{i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombprf\}$. *For nets* $N = \langle P_N, T_N, F_N, l_N, M_N\rangle$, $N' = \langle P_{N'}, T_{N'}, F_{N'}, l_{N'}, M_{N'}\rangle$ *such that* $a \in l_N(T_N) \cap l_{N'}(T_{N'})$ *and SM-net* $D = \langle P_D, T_D, F_D, l_D, M_D\rangle$ *the following holds:* $N \leftrightarrow_\star N' \Rightarrow ref(N, a, D) \leftrightarrow_\star ref(N', a, D)$ *iff equivalence* $\leftrightarrow_\star$ *is in oval in Figure 4.*

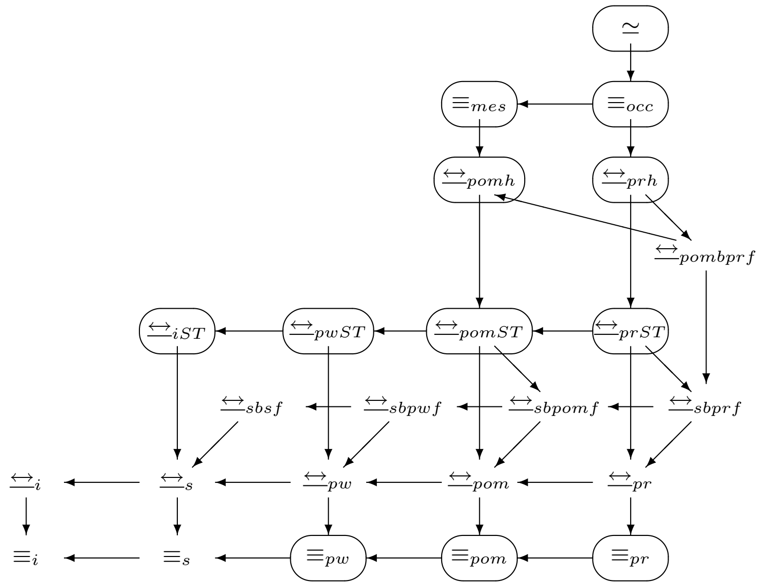

Fig. 4. Preservation of the equivalences by SM-refinements

6 Conclusion

In this paper, we examined and supplemented by new ones a group of back-forth bisimulation equivalences. We compared them with basic ones on the whole class of Petri nets as well as on their subclass of sequential nets. All the considered equivalences were treated for preservation by SM-refinements.

Further research may consist in the investigation of place bisimulation equivalences from [1] which are used for effective semantically correct reduction of nets. We intend to compare these equivalences with the ones we examined (for example, the relationship is unknown between place bisimulation equivalences and ST-, history preserving, conflict preserving and back-forth ones) and check them for preservation by refinements to establish whether they may be used for construction of multilevel concurrent systems.

Acknowledgements I would like to thank Dr. Irina B. Virbitskaite for many helpful discussions.

References

1. Autant C., Schnoebelen Ph. *Place bisimulations in Petri nets. LNCS* **616**, p. 45–61, June 1992.
2. Best E., Devillers R., Kiehn A., Pomello L. *Concurrent bisimulations in Petri nets. Acta Informatica* **28**, p. 231–264, 1991.
3. Boudol G., Castellani I. *On the semantics of concurrency: partial orders and transition systems. LNCS* **249**, p. 123–137, 1987.
4. Cherief F. *Back and forth bisimulations on prime event structures. LNCS* **605**, p.843–858, June 1992.
5. Cherief F. *Contributions à la sémantique du parallélisme: bisimulations pour le raffinement et le vrai parallélisme. Ph.D. thesis*, Institut National Politechnique de Grenoble, France, October 1992 (in French).
6. Cherief F. *Investigations of back and forth bisimulations on prime event structures. Computers and Artificial Intelligence* **11**(5), p. 481–496, 1992.
7. van Glabbeek R.J., Vaandrager F.W. *Petri net models for algebraic theories of concurrency. LNCS* **259**, p. 224–242, 1987.
8. Hoare C.A.R. *Communicating sequential processes, on the construction of programs.* (McKeag R.M., Macnaghten A.M., eds.) Cambridge University Press, p. 229–254, 1980.
9. De Nicola R., Montanari U., Vaandrager F.W. *Back and forth bisimulations. LNCS* **458**, p. 152–165, 1990.
10. Nielsen M., Thiagarajan P.S. *Degrees of non-determinizm and concurrency: A Petri net view. LNCS* **181**, p. 89–117, December 1984.
11. Park D.M.R. *Concurrency and automata on infinite sequences. LNCS* **104**, p. 167–183, March 1981.
12. Pomello L. *Some equivalence notions for concurrent systems. An overview. LNCS* **222**, p. 381–400, 1986.
13. Pinchinat S. *Bisimulations for the semantics of reactive systems. Ph.D. thesis*, Institut National Politechnique de Grenoble, January 1993 (in French).
14. Rabinovitch A., Trakhtenbrot B.A. *Behaviour structures and nets. Fundamenta Informaticae* **XI**, p. 357–404, 1988.
15. Tarasyuk I.V. *Equivalence notions for design of concurrent systems using Petri nets. Hildesheimer Informatik-Bericht* **4/96**, part 1, 19 p., Institut für Informatik, Universität Hildesheim, Hildesheim, Germany, January 1996.
16. Vogler W. *Bisimulation and action refinement. LNCS* **480**, p. 309–321, 1991.