Petri net equivalences for design of concurrent systems *

Igor V. Tarasyuk

Institute of Informatics Systems,
Siberian Division of the Russian Academy of Sciences,
6, Lavrentieva ave., 630090, Novosibirsk, Russia
e-mail: itar@iis.nsk.su

Abstract

The paper is devoted to the investigation of behavioural equivalences of concurrent systems modelled by Petri nets. The basic equivalence notions known from the literature are supplemented by new ones and examined for all class of nets as well as for their subclasses: sequential nets (nets without concurrent transitions), strictly labelled nets (which are isomorphic to unlabelled nets) and T-nets (nets without conflict transitions). A complete diagram of interrelations of the considered equivalences is obtained. In addition, the preservation of the equivalence notions by refinements is investigated, which allows one to consider the behaviour of nets on a lower abstraction level.

Keywords & phrases: concurrency, models, Petri nets, sequential nets, behavioural equivalences, bisimulation, refinement.

1 Introduction

Petri nets are 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 the framework of Petri nets. The following basic notions of behavioural equivalences for Petri nets are known from the literature.

- *Trace equivalences* (which respect only protocols of nets functioning): interleaving [5], step [8] and pomset [4].
- *(Usual) bisimulation equivalences* (which respect branching structure of nets functioning): interleaving [7], step [6], partial word [12], pomset [3] and process [1].
- *ST-bisimulation equivalences* (which respect the duration of transition occurrences in nets functioning): interleaving [4], partial word [12] and pomset [12].
- *History preserving bisimulation equivalences* (which respect the "past" or "history" of nets functioning): pomset [9] one was proposed.
- *Conflict preserving equivalences* (which fully respect conflicts in nets): occurrence [4] one was presented.
- *Isomorphism* (i.e. coincidence of nets up to renaming of places and transitions).

A *refinement* operator is used for top-down design of concurrent systems. After applying refinement, some components of the systems become having some internal structure, i.e. we consider such systems on lower abstraction level as a result. In [2], *SM-refinement* operator for Petri nets was proposed, replacing transitions of nets by SM-nets which are a special subclass of state machine nets.

In this paper, we introduce a number of the new equivalence notions in addition to the known from the literature ones to obtain a complete set of the equivalences for Petri nets: partial word and process trace

*The work is supported by Volkswagen Fund, grant N I/70 564 and Russian Fund of Fundamental Investigations, grant N 96-01-01655

equivalences, process ST-bisimulation equivalence, partial word and process history preserving bisimulation equivalences, prime event structure equivalence.

The correlation of the new and known from the literature equivalences is established on the whole class of Petri nets as well as on their subclasses: sequential nets (where no two transitions can be fired concurrently), strictly labelled nets (all transitions have different labels) and T-nets (where no two transitions have the common input or output place). In addition, all the considered behavioural equivalences are checked for preservation by SM-refinements.

2 Basic definitions

2.1 Multisets

Let X be some set. A finite *multiset* M over X is a mapping $M : X \to \mathbf{N}$ ($\mathbf{N}$ is a set of natural numbers) s.t $|\{x \in X \mid M(x) > 0\}| < \infty$. $\mathcal{M}(X)$ denotes the *set of all finite multisets* over X. When $\forall x \in X\ M(x) \leq 1,\ M$ is a proper set. *Cardinality* of multiset M is defined in such a way: $|M| = \sum_{x \in X} M(x)$. We write $x \in M$ if $M(x) > 0$ and $M \subseteq M'$, if $\forall x \in X\ M(x) \leq M'(x)$. We define $(M + M')(x) = M(x) + M'(x)$ and $(M - M')(x) = \max(0, M(x) - M'(x))$.

2.2 Labelled nets

Let $Act = \{a, b, \ldots\}$ be a set of *action names* or *labels*. 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*;
- $T_N = \{u, v, \ldots\}$ 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 a labelled net N and some transition $t \in T_N$, the *precondition* and *postcondition* t, notation respectively ${}^\bullet t$ and $t^\bullet$, 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 $(p^\bullet)(t) = F_N(p, t)$. A labelled net N is *acyclic*, if there exists no sequence $x_1, \ldots, x_n,\ x_i \in P_N \cup T_N\ (1 \leq i \leq n)$ s.t. $F_N(x_{i-1}, x_i) > 0\ (1 \leq i \leq n)$ and $x_0 = x_n$. A labelled net N is *ordinary* if $\forall p \in P_N\ {}^\bullet p$ and $p^\bullet$ are proper sets (not multisets). 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.

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', notation $\beta : N \simeq N'$, if:

1. β is a bijection s.t. $\beta(P_N) = P_{N'}$ 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*, notation $N \simeq N'$, if there exists an isomorphism $\beta : N \simeq N'$.

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.

- $x \prec_N y\ \Leftrightarrow\ xF_N^\star y$, where $F_N^\star$ is a transitive closure of F_N (*strict causal dependence* relation);
- $x \preceq_N y\ \Leftrightarrow\ (x \prec_N y) \vee (x = y)$ (*causal dependence* relation);
- $x \#_N y\ \Leftrightarrow\ \exists t, u \in T_N\ (t \neq u,\ {}^\bullet t \cap {}^\bullet u \neq \emptyset,\ t \preceq_N x,\ u \preceq_N y)$ (*conflict* relation);

2.3 Marked nets

A *marking* of a labelled net N is a multiset $M \in \mathcal{M}(P_N)$. 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 an *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, firing it yields a new marking $M' = M - {}^\bullet t + t^\bullet$, notation $M \xrightarrow{t} M'$. A marking M of a net N is *reachable*, if $M = M_N$ or there exists a reachable marking M' of N s.t. $M' \xrightarrow{t} M$ for some $t \in T_N$. $Mark(N)$ denotes a *set of all reachable* markings of a net N.

An action $a \in Act$ is *autoconcurrent* in net N, if $\exists M \in Mark(N)\ \exists t, u \in T_N :\ l_N(t) = l_N(u) = a$ and ${}^\bullet t + {}^\bullet u \subseteq M$. A net N is *autoconcurrency free*, if no action is autoconcurrent in N.

2.4 Partially ordered sets

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 $x \in X$. Then $\downarrow x = \{y \in X \mid y \prec x\}$ is a set of *strict predecessors* of x.

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', notation $\beta : \rho \approx \rho'$, 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', notation $\beta : \rho \sqsubseteq \rho'$, 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', notation $\beta : \rho \simeq \rho'$, if $\beta : \rho \sqsubseteq \rho'$ and $\beta^{-1} : \rho' \sqsubseteq \rho$. Lposets ρ and ρ' are *isomorphic*, notation $\rho \simeq \rho'$, if there exists an isomorphism $\beta : \rho \simeq \rho'$.

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

2.5 Event structures

A *labelled event structure (LES)* is a quadruple $\xi = \langle X, \prec, \#, l \rangle$, where:

- $X = \{x, y, \ldots\}$ is a set of *events*;
- $\prec \subseteq X \times X$ is a strict partial order, a *causal dependence* relation, which satisfies to the principle of *finite causes*: $\forall x \in X \; |\downarrow x| < \infty$;
- $\# \subseteq X \times X$ is an irreflexive symmetrical *conflict relation*, which satisfies to the principle of *conflict heredity*: $\forall x, y, z \in X \; x \# y \prec z \;\Rightarrow\; x \# z$;
- $l : X \to Act$ is a *labelling* function.

Let $\xi = \langle X, \prec, \#, l \rangle$ and $\xi' = \langle X', \prec', \#', l' \rangle$ be LES. A mapping $\beta : X \to X'$ is an *isomorphism* between ξ and ξ', notation $\beta : \xi \simeq \xi'$, if:

1. β is a bijection;
2. $\forall x \in X \; l(x) = l'(\beta(x))$;
3. $\forall x, y \in X \; x \prec y \;\Leftrightarrow\; \beta(x) \prec' \beta(y)$;
4. $\forall x, y \in X \; x \# y \;\Leftrightarrow\; \beta(x) \#' \beta(y)$.

LES ξ and ξ' are *isomorphic*, notation $\xi \simeq \xi'$, if there exists an isomorphism $\beta : \xi \simeq \xi'$.

A *prime event structure (PES)* is an isomorphism class of LES.

3 Equivalence notions

3.1 Equivalences based on C-processes

3.1.1 C-processes

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 a transition sequence $^\circ C = L_0 \xrightarrow{v_1} \cdots \xrightarrow{v_n} L_n = C^\circ$ s.t. $L_i \subseteq P_C$ $(0 \le i \le n)$, $P_C = \cup_{i=0}^n L_i$ and $T_C = \{v_1, \dots, v_n\}$. Such a sequence is called a *full execution* of C.

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, notation $\varphi : C \to N$, 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 $^\circ C \xrightarrow{v_1} \cdots \xrightarrow{v_n} C^\circ$ is a full execution of C, then $M = \varphi(^\circ C) \xrightarrow{\varphi(v_1)} \cdots \xrightarrow{\varphi(v_n)} \varphi(C^\circ) = M'$ is a transition sequence in N.

A *firable 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 s.t. $M = \varphi(^\circ C)$. A firable in M_N process is a *process* of N. We write $\Pi(N, M)$ for a *set of all firable* in marking M processes of a net N and $\Pi(N)$ for a *set of all processes* of a net N. An *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 $M' = M - \varphi(^\circ C) + \varphi(C^\circ) = \varphi(C^\circ)$, notation $M \xrightarrow{\pi} M'$.

Let $\pi = (C, \varphi),\ \tilde{\pi} = (\tilde{C}, \tilde{\varphi}) \in \Pi(N),\ \hat{\pi} = (\hat{C}, \hat{\varphi}) \in \Pi(N, \varphi(C^\circ)),\ C = \langle P_C, T_C, F_C, l_C\rangle,\ \tilde{C} = \langle P_{\tilde{C}}, T_{\tilde{C}}, F_{\tilde{C}}, l_{\tilde{C}}\rangle,\ \hat{C} = \langle P_{\hat{C}}, T_{\hat{C}}, F_{\hat{C}}, l_{\hat{C}}\rangle$.

We write $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$, if

1. $P_C \cup P_{\hat{C}} = P_{\tilde{C}},\ T_C \cup T_{\hat{C}} = T_{\tilde{C}},\ F_C \cup F_{\hat{C}} = F_{\tilde{C}},\ l_C \cup l_{\hat{C}} = l_{\tilde{C}}$;

2. $\varphi \cup \hat{\varphi} = \tilde{\varphi}$.

In such a case $\tilde{\pi}$ is an *extension* of π *by process* $\hat{\pi}$, and $\hat{\pi}$ is an *extending* process for π. We write $\pi \to \tilde{\pi}$, if $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ for some extending process $\hat{\pi}$.

$\tilde{\pi}$ is an extension of π *by one action*, if $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ and $|T_{\hat{C}}| = 1$. In such a case we write $\pi \xrightarrow{a} \tilde{\pi}$, if $T_{\hat{C}} = \{v\}$ and $l_{\hat{C}}(v) = a$.

$\tilde{\pi}$ is an extension of π *by multiset of actions* or *step*, if $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$ and $\prec_{\hat{C}} = \emptyset$. In such a case we write $\pi \xrightarrow{A} \tilde{\pi}$, if $T_{\hat{C}} = V$ and $l_{\hat{C}}(T_{\hat{C}}) = A,\ A \in \mathcal{M}(Act)$.

3.1.2 Trace equivalences

An *interleaving trace* of a net N is a sequence $a_1 \cdots a_n \in Act^*$ s.t. $\pi_N \xrightarrow{a_1} \pi_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} \pi_n$, where $\pi_i \in \Pi(N)$ $(1 \le i \le n)$ and π_N is an initial process of N. $SeqTraces(N)$ denotes a *set of all interleaving traces* of N. Nets N and N' are *interleaving trace equivalent*, notation $N \equiv_i N'$, if $SeqTraces(N) = SeqTraces(N')$.

A *step trace* of a net N is a sequence $A_1 \cdots A_n \in (\mathcal{M}(Act))^*$ s.t. $\pi_N \xrightarrow{A_1} \pi_1 \xrightarrow{A_2} \dots \xrightarrow{A_n} \pi_n$, where $\pi_i \in \Pi(N)$ $(0 \le i \le n)$, and π_N is an initial process of N. $StepTraces(N)$ denotes a *set of all step traces* of N. Nets N and N' are *step trace equivalent*, notation $N \equiv_s N'$, if $StepTraces(N) = StepTraces(N')$.

A *pomset trace* of a net N is a pomset ρ, an isomorphism class of lposet ρ_C for $\pi = (C, \varphi) \in \Pi(N)$. We write $\rho \sqsubseteq \rho'$, if $\rho_C \sqsubseteq \rho_{C'}$ for $\rho_C \in \rho$ and $\rho_{C'} \in \rho'$. In such a case we say that pomset ρ is *less sequential* or *more parallel* than ρ'. $Pomsets(N)$ denotes a *set of all pomset traces* of N. Nets N and N' are *partial word trace equivalent*, notation $N \equiv_{pw} N'$, if $Pomsets(N) \sqsubseteq Pomsets(N')$ and $Pomsets(N') \sqsubseteq Pomsets(N)$, i.e. for any $\rho' \in Pomsets(N')$ there exists $\rho \in Pomsets(N)$ s.t. $\rho \sqsubseteq \rho'$ and vice versa. Nets N and N' are *pomset trace equivalent*, notation $N \equiv_{pom} N'$, if $Pomsets(N) = Pomsets(N')$.

A *process trace* of a net N is an isomorphism class of causal net C for $\pi = (C, \varphi) \in \Pi(N)$. $ProcessNets(N)$ denotes a *set of all process traces* of N. Nets N and N' are *process trace equivalent*, notation $N \equiv_{pr} N'$, if $ProcessNets(N) = ProcessNets(N')$.

Example 1

- *In Figure 1(a) $N \equiv_i N'$, but $N \not\equiv_s N'$, since only in N actions a and b can happen concurrently.*

- *In Figure 1(c) $N \equiv_s N'$, but $N \not\equiv_{pw} N'$, since the pomset corresponds to the net N s.t. even less sequential pomset is not in N'.*

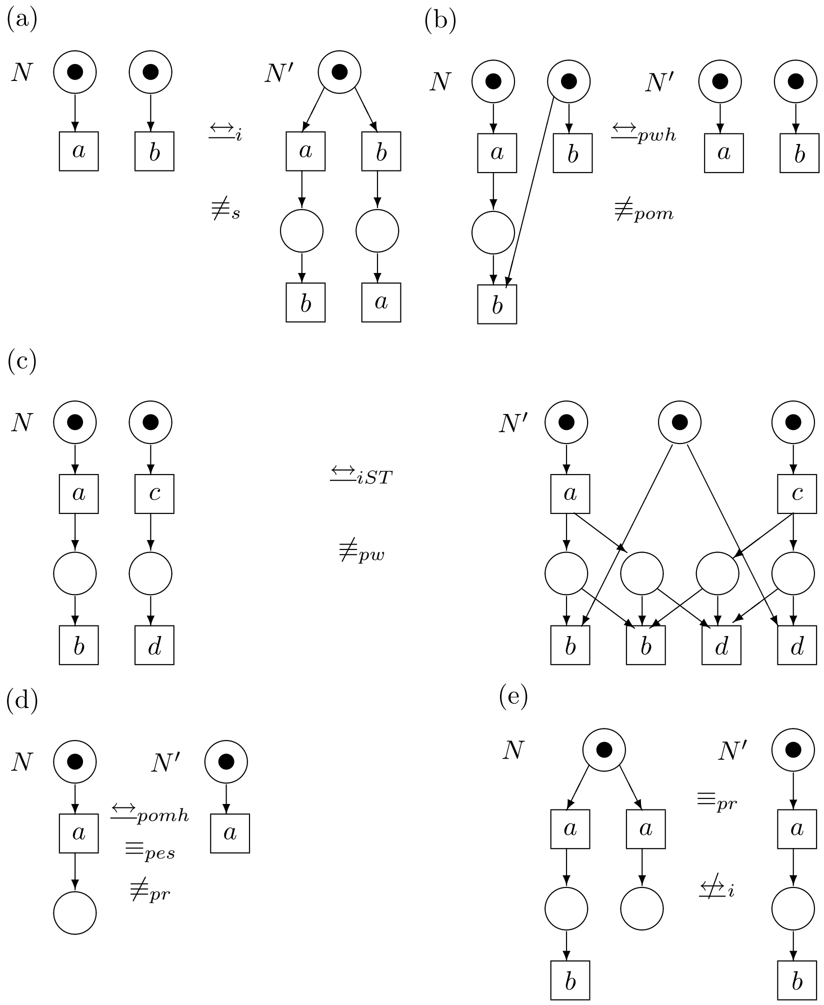

Figure 1: Examples on Petri nets

- *In Figure 1(b) $N \equiv_{pw} N'$, but $N \not\equiv_{pom} N'$, since only in net N action b can depend on action a.*
- *In Figure 1(d) $N \equiv_{pom} N'$, but $N \not\equiv_{pr} N'$, since N is causal net which is not isomorphic to N' (because of additional output place).*

3.1.3 Usual bisimilation equivalences

A notation $\mathcal{R} : N\underline{\leftrightarrow}_\star N'$ means that $\mathcal{R}$ is a bisimulation of type $\star$ ($\star$-bisimulation) between nets N and N'. Nets N and N' are called $\star$-bisimulation equivalent, notation $N\underline{\leftrightarrow}_\star N'$, if $\mathcal{R} : N\underline{\leftrightarrow}_\star N'$ for some $\star$-bisimulation $\mathcal{R}$.

Let $\mathcal{R} \subseteq \Pi(N) \times \Pi(N')$. In the following definition $\hat{\pi} = (\hat{C}, \hat{\varphi})$, $\hat{\pi}' = (\hat{C}', \hat{\varphi}')$.

$\mathcal{R}$ is a $\star$-*bisimulation* between N and N', $\star \in$\{*interleaving, step, partial word, pomset, process*\}, notation $\mathcal{R} : N\underline{\leftrightarrow}_\star N'$, $\star \in \{i, s, pw, pom, pr\}$, if:

1. $(\pi_N, \pi_{N'}) \in \mathcal{R}$;
2. $(\pi, \pi') \in \mathcal{R}$, $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$,

   (a) $|T_{\hat{C}}| = 1$, if $\star = i$;

   (b) $\prec_{\hat{C}} = \emptyset$, if $\star = s$;

   then $\exists \tilde{\pi}' : \pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}'$, $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$ and

   (a) $\rho_{\hat{C}'} \sqsubseteq \rho_{\hat{C}}$, if $\star = pw$;

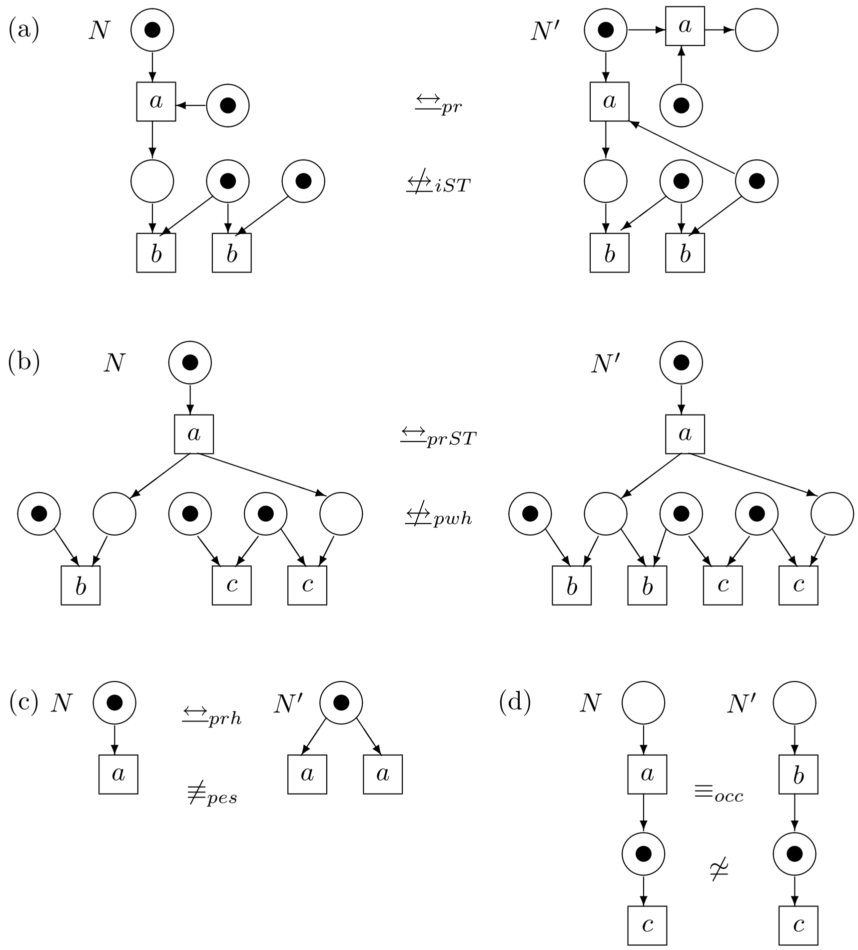

Figure 2: Examples on Petri nets (continued)

(b) $\rho_{\hat{C}} \simeq \rho_{\hat{C}'}$, if $\star \in \{i, s, pom\}$;

(c) $\hat{C} \simeq \hat{C}'$, if $\star = pr$;

3. As item 2 but the roles of N and N' are reversed.

Example 2 *In Figure 1(e) $N \equiv_{pr} N'$, but $N \not\underline{\leftrightarrow}_i N'$, since only in net N action a can happen so that action b can not happen afterwards.*

3.1.4 ST-processes

An *ST-process* of a net N is a pair (π_E, π_P) s.t. $\pi_E, \pi_P \in \Pi(N)$, $\pi_P \stackrel{\pi_W}{\to} \pi_E$ and $\forall v, w \in T_{C_E}\ v \prec_{C_E} w \ \Rightarrow v \in T_{C_P}$ In such a case π_E is a process which has started, i.e. all actions of π_E have started. A process π_P corresponds to the finished part of π_E, and π_W corresponds to the still working part. $ST - \Pi(N)$ denotes a *set of all ST-processes* of N. (π_N, π_N) is an *initial ST-process* of N. Let (π_E, π_P), $(\tilde{\pi}_E, \tilde{\pi}_P) \in ST - \Pi(N)$. We write $(\pi_E, \pi_P) \to (\tilde{\pi}_E, \tilde{\pi}_P)$, if $\pi_E \to \tilde{\pi}_E$ and $\pi_P \to \tilde{\pi}_P$.

3.1.5 ST-bisimulation equivalences

Let $\mathcal{R} \subseteq ST{-}\Pi(N) \times ST{-}\Pi(N') \times \mathcal{B}$, where $\mathcal{B} = \{\beta \mid \beta : T_C \to T_{C'},\ \pi = (C, \varphi) \in \Pi(N),\ \pi' = (C', \varphi') \in \Pi(N')\}$. In the following definition $\pi_E = (C_E, \varphi_E)$, $\pi_P = (C_P, \varphi_P)$, $\pi'_E = (C'_E, \varphi'_E)$, $\pi'_P = (C'_P, \varphi'_P)$, $\pi = (C, \varphi)$, $\pi' = (C', \varphi')$.

$\mathcal{R}$ is a $\star$-*ST-bisimulation* between N and N' $\star \in$\{*interleaving, partial word, pomset, process*\}, notation $\mathcal{R} : N \underline{\leftrightarrow}_{\star ST} N'$, $\star \in \{i, pw, pom, pr\}$, if:

1. $((\pi_N, \pi_N), (\pi_{N'}, \pi_{N'}), \emptyset) \in \mathcal{R}$;

2. $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R} \ \Rightarrow\ \beta : \rho_{C_E} \approx \rho_{C'_E}$ and $\beta(T_{C_P}) = T_{C'_P}$;

3. $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R},\ (\pi_E, \pi_P) \to (\tilde{\pi}_E, \tilde{\pi}_P) \ \Rightarrow\ \exists \tilde{\beta},\ (\tilde{\pi}'_E, \tilde{\pi}'_P) :\ (\pi'_E, \pi'_P) \to (\tilde{\pi}'_E, \tilde{\pi}'_P),\ \tilde{\beta}\lceil_{T_{C_E}} = \beta,\ ((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$, and if $\pi_P \stackrel{\pi}{\to} \tilde{\pi}_E,\ \pi'_P \stackrel{\pi'}{\to} \tilde{\pi}'_E$ then:

   (a) $(\tilde{\beta}\lceil_{T_C})^{-1} : \rho_{C'} \sqsubseteq \rho_C$, if $\star = pw$;

   (b) $\tilde{\beta}\lceil_{T_C} : \rho_C \simeq \rho_{C'}$, if $\star \in \{pom, pr\}$;

   (c) $C \simeq C'$, if $\star = pr$;

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

Example 3 *In Figure 2(a) $N \underline{\leftrightarrow}_{pr} N'$, but $N \not\underline{\leftrightarrow}_{iST} N'$, since only in net N' action a can start so that no action b can begin working until a finishes.*

3.1.6 History preserving bisimulation equivalences

Let $\mathcal{R} \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$, where $\mathcal{B} = \{\beta \mid \beta : T_C \to T_{C'},\ \pi = (C, \varphi) \in \Pi(N),\ \pi' = (C', \varphi') \in \Pi(N')\}$. In the following definition $\pi = (C, \varphi), \tilde{\pi} = (\tilde{C}, \tilde{\varphi}),\ \pi' = (C', \varphi'), \tilde{\pi}' = (\tilde{C}', \tilde{\varphi}')$.

$\mathcal{R}$ is a $\star$-*history preserving bisimulation* between N and $N', \star \in$\{*partial word, pomset, process*\}, notation $N \underline{\leftrightarrow}_{\star h} N'$,
$\star \in \{pw, pom, pr\}$, if:

1. $(\pi_N, \pi_{N'}, \emptyset) \in \mathcal{R}$;

2. $(\pi, \pi', \beta) \in \mathcal{R} \ \Rightarrow\ \beta : \rho_C \approx \rho_{C'}$;

3. $(\pi, \pi', \beta) \in \mathcal{R},\ \pi \to \tilde{\pi} \ \Rightarrow\ \exists \tilde{\beta},\ \tilde{\pi}' : \pi' \to \tilde{\pi}',\ \tilde{\beta}\lceil_{T_C} = \beta,\ (\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in \mathcal{R}$ and

   (a) $\tilde{\beta}^{-1} : \rho_{\tilde{C}'} \sqsubseteq \rho_{\tilde{C}}$, if $\star = pw$;

   (b) $\tilde{\beta} : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$, if $\star \in \{pom, pr\}$;

   (c) $\tilde{C} \simeq \tilde{C}'$, if $\star = pr$;

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

Example 4 *In Figure 2(b) $N \underline{\leftrightarrow}_{prST} N'$, but $N \not\underline{\leftrightarrow}_{pwh} N'$, since only in net N' after action a action b can happen so that action c must depend on a.*

3.2 Equivalences based on O-processes

3.2.1 O-processes

An *occurrence net* is an acyclic ordinary labelled net $O = \langle P_O, T_O, F_O, l_O \rangle$, s.t.:

1. $\forall r \in P_O \ |^\bullet r| \leq 1$, i.e. there is no forward conflict;
2. $\forall x \in P_O \cup T_O \ \neg(x \#_O x)$, i.e. conflict relation is irreflexive;
3. $\forall x \in P_O \cup T_O \ | \downarrow_O x| < \infty$, i.e. set of causes is finite.

Let us note that on the basis of any occurrence net $O = \langle P_O, T_O, F_O, l_O \rangle$ one can define LES $\xi_O = \langle T_O, \prec_O \cap (T_O \times T_O), \#_O \cap (T_O \times T_O), l_O \rangle$.

Let $O = \langle P_O, T_O, F_O, l_O \rangle$ be occurrence net and $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ be some net. A mapping $\psi : P_O \cup T_O \to P_N \cup T_N$ is an *embedding* O into N, notation $\psi : O \to N$, if:

1. $\psi(P_O) \in \mathcal{M}(P_N)$ and $\psi(T_O) \in \mathcal{M}(T_N)$. i.e. sorts are preserved;
2. $\forall v \in T_O \ l_O(v) = l_N(\psi(v))$, i.e. labelling is preserved;
3. $\forall v \in T_O \ ^\bullet\psi(v) = \psi(^\bullet v)$ and $\psi(v)^\bullet = \psi(v^\bullet)$, i.e. flow relation is respected;
4. $\forall v, w \in T_O \ (^\bullet v = {}^\bullet w) \wedge (\psi(v) = \psi(w)) \Rightarrow \ v = w$, i.e. there are no "superfluous" conflicts.

A *firable in marking M O-process* of a net N is a pair $\varpi = (O, \psi)$, where O is an occurrence net and $\psi : O \to N$ is an embedding s.t. $M = \psi(^\circ O)$. Let us note that marking M may be not reachable in general case. A firable in M_N O-process is *O-process* of a net N. We write $\wp(N, M)$ for a set of *all firable in marking M O-processes* of a net N and $\wp(N)$ for a set of *all O-processes* of a net N. An *initial* O-process of a net N coincides with its initial C-process, i.e. $\varpi_N = \pi_N$.

An extension of O-processes is idefined as well as for C-processes. An O-process ϖ of a net N is *maximal*, if it can be extended by no O-process $\hat{\varpi} = (\hat{O}, \hat{\psi})$ s.t. $T_{\hat{O}} \neq \emptyset$. A set of *all maximal O-processes* of a net N, notation $\wp_{max}(N)$, consists ot the unique (up to isomorphism) O-process $\varpi_{max} = (O_{max}, \psi_{max})$. In such a case an isomorphism class of occurrence net O_{max} is an *unfolding* of a net N, notation $\mathcal{U}(N)$. On the basis of unfolding $\mathcal{U}(N)$ of a net N one can define PES $\mathcal{E}(N) = \xi_{\mathcal{U}(N)}$ which is an isomorphism class of LES ξ_O for $O \in \mathcal{U}(N)$.

3.2.2 Conflict preserving equivalences

Nets N and N' are *PES-equivalent*, notation $N \equiv_{pes} N'$, if $\mathcal{E}(N) = \mathcal{E}(N')$.
Nets N and N' are *occurrence equivalent*, notation $N \equiv_{occ} N'$, if $\mathcal{U}(N) = \mathcal{U}(N')$.

Example 5 *In Figure 2(c) $N \underline{\leftrightarrow}_{prh} N'$, but $N \not\equiv_{pes} N'$, since only net N' has corresponding PES with two conflict actions a.*

4 Comparing the equivalence notions on the whole class of Petri nets

Theorem 1 *Let $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$ and $\star, \star\star \in \{i, s, pw, pom, pr, iST, pwST, pomST, prST, pwh, pomh, prh, pes, occ\}$. For nets N and N' $N \leftrightarrow_\star N' \ \Rightarrow \ N \leftrightarrow_{\star\star} N'$ iff there exists a directed path from $\leftrightarrow_\star$ to $\leftrightarrow_{\star\star}$ in the graph in Figure 3.*

5 Comparing the equivalence notions on subclasses of Petri nets

5.1 Sequential nets

A *sequential net* is a net $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ s.t. $\forall \pi = (C, \varphi) \in \Pi(N) \ \forall v, w \in T_C \ (v \prec_C w) \vee (w \prec_C v)$ (i.e. $\prec_C$ is a total ordering on transitions of causal net C).

Proposition 1 *For sequential nets N and N':*

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

Theorem 2 *Let $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$ and $\star, \star\star \in \{i, pr, prST, prh, pes, occ\}$. For sequential nets N and N' $N \leftrightarrow_\star N' \ \Rightarrow \ N \leftrightarrow_{\star\star} N'$ iff there exists a directed path from $\leftrightarrow_\star$ to $\leftrightarrow_{\star\star}$ in the graph in Figure 4.*

$$
\begin{tikzcd}
 & & & & \simeq \arrow[d] \\
 & & & \equiv_{pes} \arrow[d] & \equiv_{occ} \arrow[l] \arrow[d] \\
 & & \underline{\leftrightarrow}_{pwh} \arrow[d] & \underline{\leftrightarrow}_{pomh} \arrow[l] \arrow[d] & \underline{\leftrightarrow}_{prh} \arrow[l] \arrow[d] \\
 & \underline{\leftrightarrow}_{iST} \arrow[d] & \underline{\leftrightarrow}_{pwST} \arrow[l] \arrow[d] & \underline{\leftrightarrow}_{pomST} \arrow[l] \arrow[d] & \underline{\leftrightarrow}_{prST} \arrow[l] \arrow[d] \\
\underline{\leftrightarrow}_{i} \arrow[d] & \underline{\leftrightarrow}_{s} \arrow[l] \arrow[d] & \underline{\leftrightarrow}_{pw} \arrow[l] \arrow[d] & \underline{\leftrightarrow}_{pom} \arrow[l] \arrow[d] & \underline{\leftrightarrow}_{pr} \arrow[l] \arrow[d] \\
\equiv_{i} & \equiv_{s} \arrow[l] & \equiv_{pw} \arrow[l] & \equiv_{pom} \arrow[l] & \equiv_{pr} \arrow[l]
\end{tikzcd}
$$

Figure 3: Correlation of the equivalence notions on the whole class of Petri nets

$$
\begin{tikzcd}
 & \simeq \arrow[d] \\
\equiv_{pes} \arrow[ddd] & \equiv_{occ} \arrow[l] \arrow[d] \\
 & \underline{\leftrightarrow}_{prh} \arrow[d] \\
 & \underline{\leftrightarrow}_{prST} \arrow[d] \\
\underline{\leftrightarrow}_{i} \arrow[d] & \underline{\leftrightarrow}_{pr} \arrow[l] \arrow[d] \\
\equiv_{i} & \equiv_{pr} \arrow[l]
\end{tikzcd}
$$

Figure 4: Correlation of the equivalence notions on sequential nets

5.2 Strictly labelled nets

A *strictly labelled net* is a net $N = \langle P_N, T_N, F_N, l_N, M_N\rangle$ s.t. $\forall t, u \in T_N \ t \neq u \ \Rightarrow \ l_N(t) \neq l_N(u)$ (i.e. its labelling function is injective).

Proposition 2 *For strictly labelled nets N and N':*

1. $N \equiv_\star N' \Leftrightarrow \ N \underline{\leftrightarrow}_\star N', \ \star \in \{i, pw, pom, pr\}$;
2. $N \equiv_s N' \Leftrightarrow \ N \underline{\leftrightarrow}_{iST} N'$.

5.3 T-nets

A *T-net* is a net $N = \langle P_N, T_N, F_N, l_N, M_N\rangle$ s.t. $\forall p \in P_N \ |{}^\bullet p| \leq 1$ and $|p^\bullet| \leq 1$.

Proposition 3 *For autoconcurrency free T-nets $N \ \ N'$ $\ N \equiv_i N' \Leftrightarrow \ N \underline{\leftrightarrow}_{iST} N'$.*

6 Preservation of the equivalence notions by refinements

An *empty in/out net* is a net $D = \langle P_D, T_D, F_D, l_D, M_D\rangle$ s.t.:

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

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 empty in/out system. An *empty in/out refinement*, notation $ref(N, a, D)$, is (up to isomorphism) a net $\overline{N} = \langle P_{\overline{N}}, T_{\overline{N}}, F_{\overline{N}}, l_{\overline{N}}, M_{\overline{N}}\rangle$, s.t.:

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

An *SM-net* is an empty in/out net $D = \langle P_D, T_D, F_D, l_D, M_D\rangle$ s.t. $\forall t \in T_D \ |{}^\bullet t| \leq 1$ and $|t^\bullet| \leq 1$. An *SM-refinement* is an empty in/out refinement $ref(N, a, D)$ s.t. D is SM-net.

We say that some equivalence on nets is preserved by refinements, if equivalent nets remain equivalent after applying any refinement operator to them accordingly.

The following example demonstrates which of the equivalence notions are not preserved by SM-refinements.

Example 6

- *In Figure 5 $N \underline{\leftrightarrow}_s N'$, but $ref(N, c, D) \not\equiv_i ref(N', c, D)$, since only in $ref(N', c, D)$ the sequence of actions $c_1 a b c_2$ can happen. Consequently, no equivalence from $\equiv_i$ to $\underline{\leftrightarrow}_s$ is preserved by SM-refinements.*
- *In Figure 6 $N \underline{\leftrightarrow}_{pr} N'$, but $ref(N, a, D) \not\underline{\leftrightarrow}_i ref(N', a, D)$, since only in $ref(N', a, D)$ after occurrence of action a_1 action b can not happen. Consequently, no equivalence from $\underline{\leftrightarrow}_i$ to $\underline{\leftrightarrow}_{pr}$ is preserved by SM-refinements.*
- *In Figure 7 $N \underline{\leftrightarrow}_{pwh} N'$, but $ref(N, b, D) \not\underline{\leftrightarrow}_{pwh} ref(N', b, D)$, since only in $ref(N, b, D)$ after action a action b_1 can happen so that action b_2 must depend on a. Consequently, the equivalence $\underline{\leftrightarrow}_{pwh}$ is not preserved by SM-refinements.*

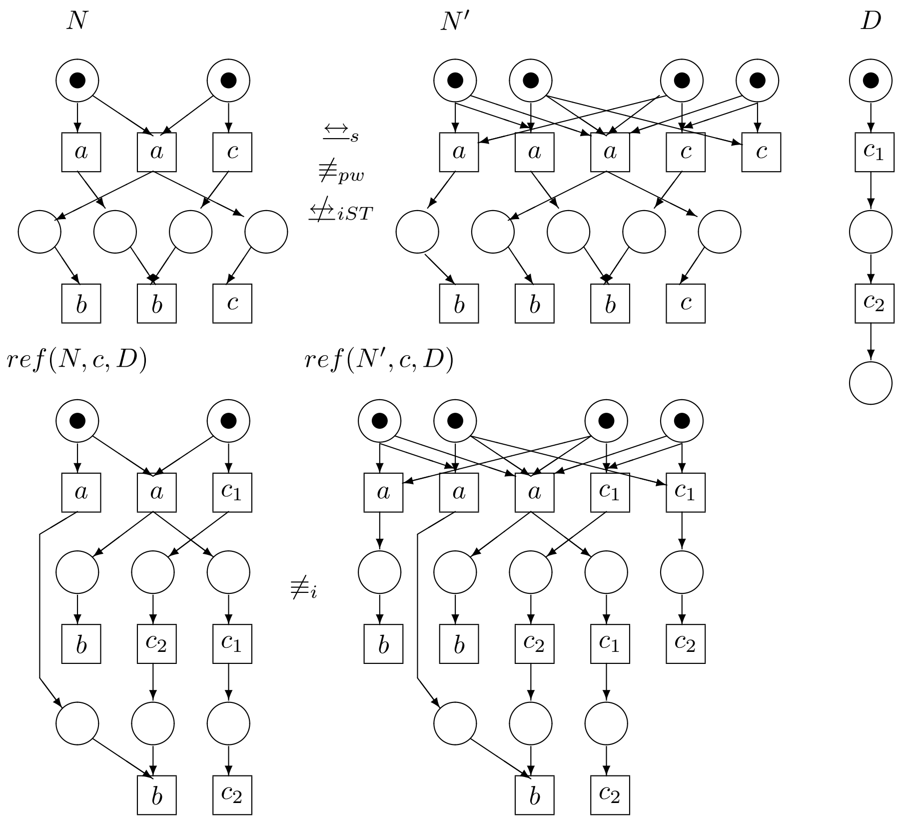

Figure 5: The equivalences from $\equiv_i$ to $\underline{\leftrightarrow}_s$ are not preserved by SM-refinements

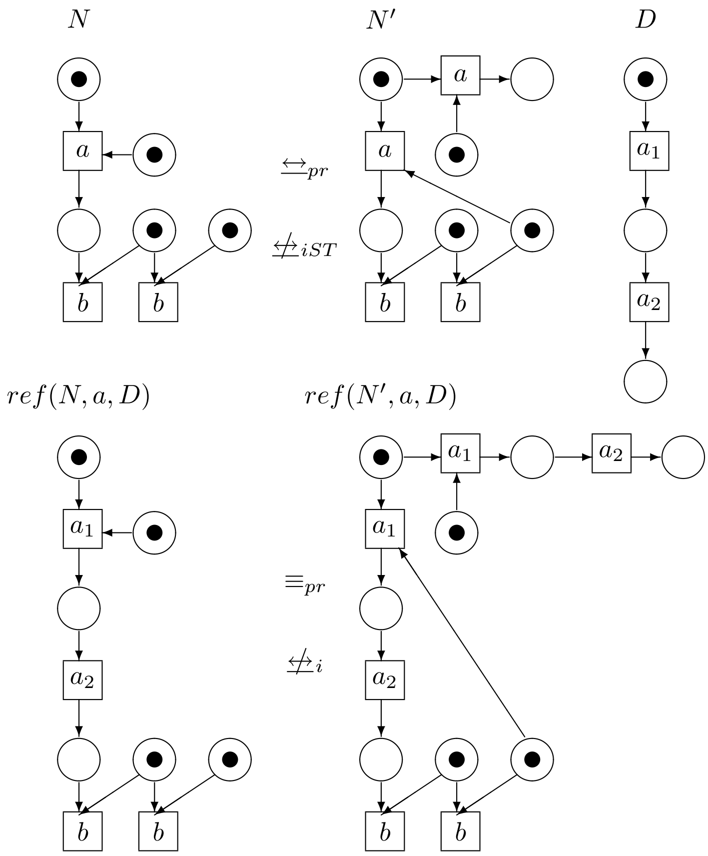

Figure 6: The equivalences from $\underline{\leftrightarrow}_i$ to $\underline{\leftrightarrow}_{pr}$ are not preserved by SM-refinements

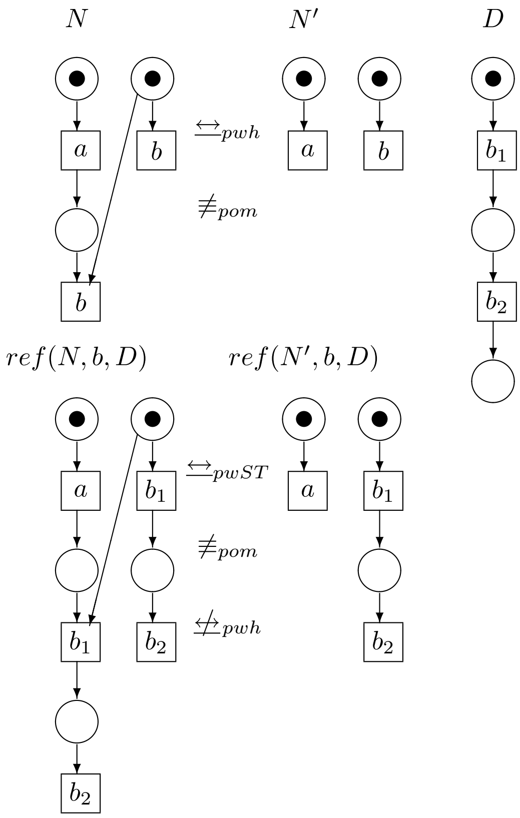

Figure 7: The equivalence $\underline{\leftrightarrow}_{pwh}$ is not preserved by SM-refinements

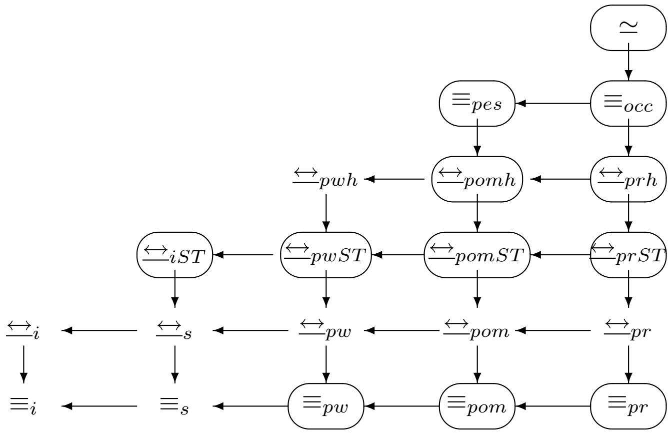

Figure 8: Preservation of the equivalences by SM-refinements

Let us consider which of the net equivalences are preserved by SM-refinements.

Theorem 3 *Let* $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$ *and* $\star \in \{i, s, pw, pom, pr, iST, pwST, pomST, prST, pwh, pomh, prh, pes, occ\}$. *For nets* $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$, $N' = \langle P_{N'}, T_{N'}, F_{N'}, l_{N'}, M_{N'} \rangle$ *s.t.* $a \in l_N(T_N) \cap l_{N'}(T_{N'})$ *and SM-net* $D = \langle P_D, T_D, F_D, l_D, M_D \rangle$ *the following is valid:* $N \leftrightarrow_\star N' \;\Rightarrow\; ref(N, a, D) \leftrightarrow_\star ref(N', a, D)$ *iff* $\leftrightarrow_\star$ *is in oval in Figure 8.*

7 Conclusion

In this paper, we examined and supplemented by new ones a group of the basic behavioural equivalences which can be used to consider systems being modelled by Petri nets, at different abstraction levels.

The main result consists in establishing correlation of all the equivalence notions on the whole class of Petri nets as well as on their subclasses of sequential, strictly labelled and T-nets. All the considered equivalences were checked for preservation by SM-refinements. So, we can use the equivalence notions that are preserved by SM-refinements, for top-down design of concurrent systems.

Let us mention some directions of further research.

One of these directions is obtaining a complete picture of correlation of the equivalence notions on strictly labelled nets and T-nets. Some early results can be found in [10, 11].

Another direction of further research consists in the investigation of place bisimulation equivalences from [1]. 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 ones). It is interesting to introduce ST- and history preserving versions of place bisimulation equivalences, that allow one to prune the structure of nets with respect to the real time aspects or "history" of functioning of nets. In addition, it is worth checking place bisimulation equivalences for preservation by refinements to establish whether they may be used for construction of multilevel concurrent systems.

Acknowledgements I would like to thank my scientific supervisor Dr. Irina B. Virbitskaite for many discussions which played an important role in the research.

This work was done within joint research project with Institute of Informatics of Hildesheim. I thank the head of the institute Prof. Dr. Eike Best for providing a good working atmosphere and for useful remarks and questions. I am indebted to researcher of the institute, Tom Thielke, for advices which improved the paper.

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] van Glabbeek R.J., Vaandrager F.W. *Petri net models for algebraic theories of concurrency.* LNCS **259**, p. 224–242, 1987.

[5] 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.

[6] Nielsen M., Thiagarajan P.S. *Degrees of non-determinizm and concurrency: A Petri net view.* LNCS **181**, p. 89–117, December 1984.

[7] Park D.M.R. *Concurrency and automata on infinite sequences.* LNCS **104**, p. 167–183, March 1981.

[8] Pomello L. *Some equivalence notions for concurrent systems. An overview.* LNCS **222**, p. 381–400, 1986.

[9] Rabinovitch A., Trakhtenbrot B.A. *Behaviour structures and nets.* Fundamenta Informaticae **XI**, p. 357–404, 1988.

[10] Tarasyuk I.V. *An investigation of equivalence notions on some subclasses of Petri nets.* Bulletin of the Novosibirsk Computing Center **3** (Series Computer Science), p. 89–101, Computing Center, Novosibirsk, 1995.

[11] Tarasyuk I.V. *Equivalence notions for design of concurrent systems using Petri nets.* Hildesheimer Informatik-Berichte **4/96**, part 1, 19 p., Institut für Informatik, Universität Hildesheim, Hildesheim, Germany, January 1996.

[12] Vogler W. *Bisimulation and action refinement.* LNCS **480**, p. 309–321, 1991.