Equivalences on Petri nets

I.V. Tarasyuk

Abstract

In this paper a set of Petri net equivalences is proposed. A correlation between all introduced equivalences is established, and a lattice of implications is obtained.

In addition equivalences are examined on sequential nets. The process equivalences are demonstrated to be well discerning on this net class.

1 Introduction

A wide class of semantic equivalences was introduced in the literature concerning Petri nets. The fundamentals of these equivalences are presented on coordinate plane in Figure 1. On the X-axis equivalences are ordered with respect to the preserved level of detail of runs of processes. On the Y-axis equivalences are ordered with respect to the preserved level of detail of the branching structure of these runs.

The following points on X-axis are known.

Sequential semantics. Another name of this semantics is interleaving semantics. A process run is represented by sequences of action occurrences.

Step semantics. A process run is simulated by multiset sequences of action occurences.

Partial word semantics. An execution of partial ordered multiset (pomset) corresponds to the run of process. One process simulates the other process in this semantics if the relation of causal dependence between actions in its pomset is less strict or the same as in the pomset of the second one.

Pomset semantics. It is analogous to partial word semantics with the exception of the fact that causal dependence on actions in the pomset is preserved.

Process-net semantics. A process run is represented by the run of acyclic net without conflicts (so-called causal or C-net). Further we will use the term "process equivalence" for equivalence associated with this semantics.

The following points on Y-axis are presented below.

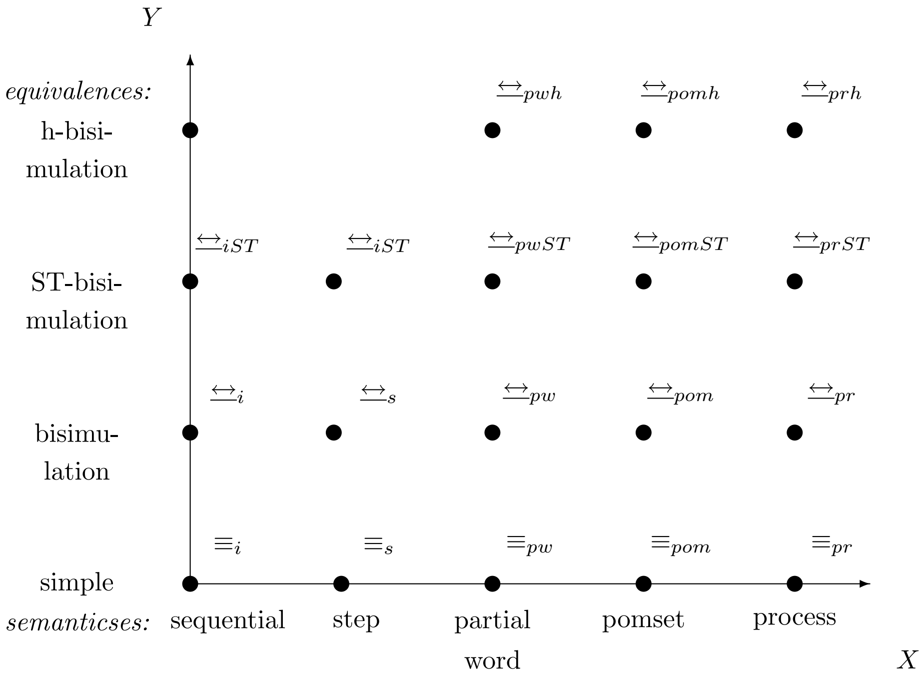

Figure 1: Classification of equivalences

Simple equivalences. A process is determined by the set of its possible runs and nondeterminism is not taken into account (i.e. we do not take into consideration a place of nondeterministic choice among several process evolutions).

Bisimulation equivalences. Nondeterminism is taken into account.

ST-bisimulation equivalences. Actions are considered to have some internal structure (or actions do not occur instantaneously but they work for some time).

History preserving bisimulation equivalences. (H-bisimulation equivalences in short). These equivalences take into account the "past" of process i.e. *how* extending process connects with process the run of which has led to the present state.

Let us examine which of the mentioned in the literature equivalences are in this coordinate plane.

- Let us consider simple equivalences.

- The simplest equivalence in the origin of coordinate is a language (sequential, interleaving) one. It is marked in $\equiv_i$ symbol. Two nets are sequentially equivalent if their languages are the same. The definition of this equivalence on event structures can be found in [12].
- Step equivalence (is marked in $\equiv_s$) associates nets having the same sets of step traces (sequences of action multisets). The definition on event structures is in [12].
- Pomset equivalence (is marked in $\equiv_{pom}$) connects nets with the same action pomsets. This equivalence was defined in [13] on Petri nets and in [12] on event structures.

• The following equivalences are known among bisimulation ones.

- Sequential bisimulation equivalence (is marked in $\underline{\leftrightarrow}_i$) was introduced in [16] on automatons. The definition can be found also in [1, 2, 3, 4, 13] on Petri nets and in [11, 12, 18, 19] on event structures.
- Step bisimulation equivalence (is marked in $\underline{\leftrightarrow}_s$) was introduced in [15]. The definition on Petri nets is in [1, 2, 13], and on event structures it is in [11, 12, 18, 19].
- Partial word bisimulation equivalence (is marked in $\underline{\leftrightarrow}_{pw}$) was introduced in [18] on event structures. The definition on Petri nets can be found in [2], and on event structures it is also in [19].
- Pomset equivalence (is marked in $\underline{\leftrightarrow}_{pom}$) was introduced in [3, 13]. The definition on Perti nets is proposed in [1, 2, 4], and on event structures it is in [11, 12, 18, 19].
- Process bisimulation equivalence (is marked in $\underline{\leftrightarrow}_{pr}$) was proposed in [2] on Petri nets.

• The following ST-bisimulation equivalences are known.

- Sequential ST-bisimulation equivalence (is marked in $\underline{\leftrightarrow}_{iST}$) was introduced on Petri nets in [13]. The definition on Petri nets can be found also in [3], and on event structures it is in [12, 18, 19]. Let us note that sequential ST-bisimulation equivalence coincides with corresponding step equivalence.
- Partial word ST-bisimulation equivalence (is marked in $\underline{\leftrightarrow}_{pwST}$) was proposed in [18] on event structures. The definition can be found also in [19].
- Pomset ST-bisimulation equivalence (is marked in $\underline{\leftrightarrow}_{pomST}$) was introduced in [18] on event structures. The definition is also in [19].

• History preserving bisimulation equivalence (is marked in $\underline{\leftrightarrow}_{pomh}$) was considered in the literature. It was introduced in [17] on behaviour structures under the name of “bisimulation equivalense of behaviour structures”. In

[11] this equivalence was defined on event structures and called "history preserving bisimulation equivalence". The definition on Petri nets was introduced in [4]. In this paper the equivalence was named "fully concurrent bisimulation equivalence". The definition on Petri nets can be found also in [3, 10], and on event structures it is in [11, 12, 18, 19].

A number of new equivalences completing known ones is introduced in this paper. These new equivalences are also presented in Figure 1. A correlation between all introduced equivalences is established, and a lattice of implications is obtained.

In addition the introduced equivalences are examined on sequential nets, one of the Petri net classes. Process equivalences are demonstrated to be well discerning on sequential nets unlike other ones, a lot of which merge on this net class.

Let us do a short review of the paper. In Section 2 the basic definitions are given. Simple net equivalences are described in Section 3, Section 4 deals with bisimulation ones. In Section 5 the theorem establishing a correlation between all introduced equivalences is proved. Section 6 is devoted to the investigation of equivalences on sequential nets. The concluding Section 7 contains some ideas about development of this work.

2 Basic definitions

2.1 Multisets

Let X be some set. A *multiset* M over X is a mapping $M : X \to \mathcal{N}$, where $\mathcal{N}$ is a set of natural numbers. For $x \in X$, $M(x)$ is a *multiplicity* x in M. We write $x \in M$ if $M(x) > 0$.

When $\forall x \in X\ M(x) \leq 1$, M is a proper set. M is *finite* if $M(x) = 0$ for all $x \in X$, except maybe a finite number of them. *Cardinality* of multiset M is defined in such a way: $|M| = \sum_{x \in X} M(x)$. From now on we will consider only finite multisets. $\mathcal{M}(X)$ denotes the *set of all finite multisets* over X.

Set-theotetic notions are extended to finite multisets in the standard way. If $M, M' \in \mathcal{M}(X)$, we define $M + M'$ by $(M + M')(x) = M(x) + M'(x)$. We write $M \subseteq M'$, if $\forall x \in X\ M(x) \leq M'(x)$. When $M' \subseteq M$, we define $M - M'$ by $(M - M')(x) = M(x) - M'(x)$. Notation $M + x - y$ is used instead of $M + \{x\} - \{y\}$. We write symbol $\emptyset$ for empty multiset.

2.2 Marked nets

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;
- $F_N : (P_N \times T_N) \cup (T_N \times P_N) \to \mathcal{N}$ is the flow relation with weights;

- $l_N : T_N \to Act$ is a labelling of transitions with action names.

It is believed that $P_N \cap T_N = \emptyset$.

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

A *marked net* is a tuple $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ so that $\langle P_N, T_N, F_N, l_N \rangle$ is a labelled net and $M_N \in \mathcal{M}(P_N)$ is an initial marking. We write "net" instead of "marked net". Given a net N and some transition $u \in T_N$, the *precondition* and *postcondition* u, written respectively ${}^\bullet u$ and $u^\bullet$, are the multisets defined in such a way: $({}^\bullet u)(p) = F_N(p, u)$ and $(u^\bullet)(p) = F_N(u, p)$. Analogous definitions are introduced for places: $({}^\bullet p)(u) = F_N(u, p)$ and $(p^\bullet)(u) = F_N(p, u)$. A transition u is *unstable* if ${}^\bullet u = \emptyset$. A net is *stable* if it has no unstable transitions. A net N is *finite* if $P_N \cup T_N$ is.

Let $M \in \mathcal{M}(P_N)$ be a marking of a net N. A transition $u \in T_N$ is *firable* in M if ${}^\bullet u \subseteq M$. If u is firable in M, firing it yields a new marking $M' = M - {}^\bullet u + u^\bullet$, written $M \stackrel{u}{\to} M'$ or $M \stackrel{a}{\to} M'$ if $l_N(u) = a$. We write $M \to M'$ if $M \stackrel{u}{\to} M'$ for some u.

2.3 Processes

A *labelled C-net (causal net)* is a labelled net $C = \langle P_C, T_C, F_C, l_C \rangle$, where:

1. $\forall v \in T_C$ ${}^\bullet v$ and $v^\bullet$ are proper sets;
2. places are unbranched, i.e. $\forall p \in P_C$ $|{}^\bullet p| \leq 1$ and $|p^\bullet| \leq 1$;
3. F_C is well-founded, i.e. there is no backward infinite chain $\cdots (p_n, v_n)(v_n, p_{n-1}) \cdots (p_1, v_1)(v_1, p_0)$ in F_C.

Let us introduce the following notations. ${}^\circ C = \{p \in P_C | {}^\bullet p = \emptyset\}$ is a set of *initial* places in C and $C^\circ = \{p \in P_C | p^\bullet = \emptyset\}$ is a set of *final* places in C. The fundamental property of causal nets is known: for C-net there exists a transition sequence ${}^\circ C = L_0 \stackrel{v_1}{\to} \cdots \stackrel{v_n}{\to} L_n = C^\circ$ so 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.

Given a net N and a labelled C-net C. A mapping $f : P_C \cup T_C \to P_N \cup T_N$ is an *embedding* C into N, written $f : C \to N$, if:

1. $f(P_C) \in \mathcal{M}(P_N)$ and $f(T_C) \in \mathcal{M}(T_N)$;
2. $\forall v \in T_C$ $l_C(v) = l_N(f(v))$;
3. $\forall v \in T_C$ ${}^\bullet f(v) = f({}^\bullet v)$ and $f(v)^\bullet = f(v^\bullet)$.

Point 3 means that embedding respects the flow relation. Consequently, if ${}^\circ C \stackrel{v_1}{\to} \cdots \stackrel{v_n}{\to} C^\circ$ is a full execution of C, then $M = f({}^\circ C) \stackrel{f(v_1)}{\longrightarrow} \cdots \stackrel{f(v_n)}{\longrightarrow} f(C^\circ) = M'$ is a transition sequence in N, *corresponding* to this full execution, written $M \stackrel{C,f}{\to} M'$. Conversely, for any transition sequence $M \stackrel{u_1}{\to} \cdots \stackrel{u_n}{\to} M'$ of a net N there exists a labelled C-net C and an embedding $f : C \to N$ so that $M = f({}^\circ C)$, $M' = f(C^\circ)$, $u_i = f(v_i)$ $(0 \leq i \leq n)$ and ${}^\circ C \stackrel{v_1}{\to} \cdots \stackrel{v_n}{\to} C^\circ$ is a full execution of C.

A *firable process in marking* M of a net N is a pair $\pi = (C, f)$, where C is a labelled C-net (we will write "C-net" in short in this case) and $f : C \to N$ is an embedding so that $M = f({}^\circ C)$. A process firable in M_N is a *process* of N. We write $\Pi(N, M)$ for a *set of all firable* in M processes of N and $\Pi(N)$ for a *set of all processes* of N. Further we will deal only with *finite* processes, i.e. with processes having finite C-nets.

If $\pi \in \Pi(N, M)$, then firing of this process transforms a marking M into $M' = M - f({}^\circ C) + f(C^\circ) = f(C^\circ)$, written $M \xrightarrow{\pi} M'$. A C-net sets an ordering on transitions (a *precedence, causal dependence* relation) $\prec_C$, defined in such a way: $\prec_C = F_C^+\lceil_{T_C \times T_C}$, where F_C^+ is a transitive closure of F_C. The *initial* process of a net N is $\pi_N = (C_N, f_N) \in \Pi(N)$, where $T_{C_N} = \emptyset$. Let $\pi = (C, f)$, $\tilde{\pi} = (\tilde{C}, \tilde{f}) \in \Pi(N)$, $\hat{\pi} = (\hat{C}, \hat{f}) \in \Pi(N, f(C^\circ))$, $C = \langle P_C, T_C, F_C, l_C \rangle$, $\tilde{C} = \langle P_{\tilde{C}}, T_{\tilde{C}}, F_{\tilde{C}}, l_{\tilde{C}} \rangle$, $\hat{C} = \langle P_{\hat{C}}, T_{\hat{C}}, F_{\hat{C}}, l_{\hat{C}} \rangle$.

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. $f \cup \hat{f} = \tilde{f}$.

In such a case $\tilde{\pi}$ is an extension of π *by process* $\hat{\pi}$, and $\hat{\pi}$ is an *extending* process for π. Note that for all $\pi \in \Pi(N)$ $\pi_N \xrightarrow{\pi} \pi$. 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 $\pi \to \tilde{\pi}$ and $|T_{\tilde{C}} \setminus T_C| = 1$. In such a case we write $\pi \xrightarrow{v} \tilde{\pi}$ or $\pi \xrightarrow{a} \tilde{\pi}$, if $T_{\tilde{C}} \setminus T_C = \{v\}$ and $l_{\tilde{C}}(v) = a$.

So, we can write a full execution of C in "context" of a net N using processes as $\pi_0 \xrightarrow{v_1} \cdots \xrightarrow{v_n} \pi_n$, where $\pi_i = (C_i, f_i)$, $C_i^\circ = L_i$, $(0 \le i \le n)$ and ${}^\circ C = L_0 \xrightarrow{v_1} \cdots \xrightarrow{v_n} L_n = C^\circ$ is a full execution of C.

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

2.4 Mappings

Given nets $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ and $N' = \langle P_{N'}, T_{N'}, F_{N'}, l_{N'}, M_{N'} \rangle$. We call β a *mapping* of N *into* N', written $\beta : N \to N'$, if $\beta : P_N \cup T_N \to P_{N'} \cup T_{N'}$, $\beta(P_N) \subseteq P_{N'}$ and $\beta(T_N) \subseteq T_{N'}$. We write $\beta(N) = N'$, when $\beta(P_N) = P_{N'}$ and $\beta(T_N) = T_{N'}$.

A mapping $\beta : N \to N'$ is an *isomorphism* between N and N', written $\beta : N \simeq N'$, if:

1. β is a bijection and $\beta(N) = N'$;

2. $\forall u \in T_N \; l_N(u) = l_{N'}(\beta(u))$;

3. $\forall u \in T_N \; {}^\bullet\beta(u) = \beta({}^\bullet u)$ and $\beta(u)^\bullet = \beta(u^\bullet)$.

Nets N and N' are *isomorphic*, written $N \simeq N'$, if there exists an isomorphism $\beta : N \simeq N'$.

Given two labelled C-nets $C = \langle P_C, T_C, F_C, l_C \rangle$ and $C' = \langle P_{C'}, T_{C'}, F_{C'}, l_{C'} \rangle$.

A mapping $\beta : T_C \to T_{C'}$ is a *label preserving bijection* between $_C$ and $_{C'}$, written $\beta : T_C \approx T_{C'}$, if:

1. β is a bijection and $\beta(T_C) = T_{C'}$;
2. $\forall v \in T_C \; l_C(v) = l_{C'}(\beta(v))$.

We write $T_C \approx T_{C'}$, if there exists a label-preserving bijection $\beta : T_C \approx T_{C'}$.

A mapping $\beta : T_C \to T_{C'}$ is a *homomorphism* between T_C and $T_{C'}$, written $\beta : T_C \sqsubseteq T_{C'}$, if:

1. $\beta : T_C \approx T_{C'}$;
2. $\forall v, \; w \in T_C \; v \prec_C w \;\Rightarrow\; \beta(v) \prec_{C'} \beta(w)$.

We write $T_C \sqsubseteq T_{C'}$, if there exists a homomorphism $\beta : T_C \sqsubseteq T_{C'}$.

A mapping $\beta : T_C \to T_{C'}$ is an *isomorphism* between T_C and $T_{C'}$, written $\beta : T_C \simeq T_{C'}$, if $\beta : T_C \sqsubseteq T_{C'}$ and $\beta^{-1} : T_{C'} \sqsubseteq T_C$. We write $T_C \simeq T_{C'}$, if there exists an isomorphism $\beta : T_C \simeq T_{C'}$.

3 Simple net equivalences

A *sequential trace* of a net N is a sequence $a_1 \cdots a_n \in Act^*$ so that $\pi_N \stackrel{a_1}{\to} \pi_1 \stackrel{a_2}{\to} \ldots \stackrel{a_n}{\to} \pi_n$, where $\pi_i \in \Pi(N)$ $(1 \leq i \leq n)$ and π_N is an initial process of N. $SeqTraces(N)$ denotes a *set of all sequential traces* of N. Two nets N and N' are *sequentially equivalent*, written $N \equiv_i N'$, if $SeqTraces(N) = SeqTraces(N')$.

A *step trace* of a net N is a sequence $A_1 \cdots A_n \in (\mathcal{M}(Act))^*$ so that $\pi_N \stackrel{A_1}{\to} \pi_1 \stackrel{A_2}{\to} \ldots \stackrel{A_n}{\to} \pi_n$, where $\pi_i \in \Pi(N)$ $(0 \leq i \leq n)$, and π_N is an initial process of N. $StepTraces(N)$ denotes a *set of all step traces* of N. Two nets N and N' are *step equivalent*, written $N \equiv_s N'$, if $StepTraces(N) = StepTraces(N')$.

A *pomset trace* of a net N is a pomset ρ, an isomorphism class of T_C for $\pi = (C, f) \in \Pi(N)$, where $C = \langle P_C, T_C, F_C, l_C \rangle$. We write $\rho \sqsubseteq \rho'$, if $T_C \sqsubseteq T_{C'}$ for $T_C \in \rho$ and $T_{C'} \in \rho'$. In such a case we say that pomset ρ is *less sequential* or *more parallel* than ρ'. Let us denote a *set of all pomset traces* of N by $Pomsets(N)$. Two nets N and N' are *partial word equivalent*, written $N \equiv_{pw} N'$, if $Pomsets(N) \sqsubseteq Pomsets(N')$ and $Pomsets(N') \sqsubseteq Pomsets(N)$, i.e. for any $\rho' \in Pomsets(N')$ there exists $\rho \in Pomsets(N)$ so that $\rho \sqsubseteq \rho'$ and vice versa. Two nets N and N' are *pomset equivalent*, written $N \equiv_{pom} N'$, if $Pomsets(N) = Pomsets(N')$.

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

4 Bisimulation equivalences

Bisimulation is a fundamental behavioural equivalence. For two nets to be bisimulation equivalent there must be some relation R (bisimulation) on their states so that:

- Initial states of both nets are related by R.
- If nets are in states related by R, and one of these nets evolved into new state, then the other net is able to simulate a behaviour of the first one, evolving in a new state too. In addition new states of nets have to be related by R.

States of nets may be, for example, markings or processes. There exist other types of states (for example, ST-processes we will consider further).

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

4.1 Simple bisimulations

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

R is a *sequential bisimulation* between N and N', written $R : N \underline{\leftrightarrow}_i N'$, if:

1. $(\pi_N, \pi_{N'}) \in R$;
2. $(\pi, \pi') \in R,\ \pi \xrightarrow{a} \tilde{\pi},\ a \in Act\ \Rightarrow\ \exists \pi' :\ \pi' \xrightarrow{a} \tilde{\pi}'$ and $(\tilde{\pi}, \tilde{\pi}') \in R$;
3. As previous item but N and N' are transposed.

R is a *step bisimulation* between N and N', written $R : N \underline{\leftrightarrow}_s N'$, if:

1. $(\pi_N, \pi_{N'}) \in R$;
2. $(\pi, \pi') \in R,\ \pi \xrightarrow{A} \tilde{\pi},\ a \in \mathcal{M}(Act)\ \Rightarrow\ \exists \pi' :\ \pi' \xrightarrow{A} \tilde{\pi}'$ and $(\tilde{\pi}, \tilde{\pi}') \in R$;
3. As previous item but N and N' are transposed.

R is a *partial word bisimulation* between N and N', written $R : N \underline{\leftrightarrow}_{pw} N'$, if:

1. $(\pi_N, \pi_{N'}) \in R$;
2. $(\pi, \pi') \in R,\ \pi \xrightarrow{\hat{\pi}} \tilde{\pi},\ \Rightarrow\ \exists \pi' :\ \pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}',\ T_{\hat{C}'} \sqsubseteq T_{\hat{C}}$ and $(\tilde{\pi}, \tilde{\pi}') \in R$;
3. As previous item but N and N' are transposed.

R is a *pomset bisimulation* between N and N', written $R : N \underline{\leftrightarrow}_{pom} N'$, if:

1. $(\pi_N, \pi_{N'}) \in R$;
2. $(\pi, \pi') \in R,\ \pi \xrightarrow{\hat{\pi}} \tilde{\pi},\ \Rightarrow\ \exists \pi' :\ \pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}',\ T_{\hat{C}} \simeq T_{\hat{C}'}$ and $(\tilde{\pi}, \tilde{\pi}') \in R$;

3. As previous item but N and N' are transposed.

R is a *process bisimulation* between N and N', written $R : N \underline{\leftrightarrow}_{pr} N'$, if:

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

2. $(\pi, \pi') \in R,\ \pi \xrightarrow{\hat{\pi}} \tilde{\pi},\ \Rightarrow\ \exists \pi' :\ \pi' \xrightarrow{\hat{\pi}'} \tilde{\pi}',\ \hat{C} \simeq \hat{C}'$ and $(\tilde{\pi}, \tilde{\pi}') \in R$;

3. As previous item but N and N' are transposed.

4.2 ST-processes

ST-processes are introduced for description of the timed net states. In such nets each transition (and action labelling it) does not occur instantaneously but, starting work, it works for some time and then it terminates.

A *ST-process* of a net N is a pair (π_E, π_P) so that $\pi_E, \pi_P \in \Pi(N),\ \pi_P \xrightarrow{\pi_W} \pi_E$ 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 began to work, i.e. all actions of π_E began working. A process π_P corresponds to the terminated part of π_E, and π_W corresponds to the still working part. Clearly, $\prec_{C_W} = \emptyset$. $ST - \Pi(N)$ denotes a *set of all ST-processes* of N.

(π_N, π_N) will be an *initial ST-process* of N. Let $(\pi_E, \pi_P),\ (\tilde{\pi}_E, \tilde{\pi}_P) \in ST - \Pi(N)$. We write $(\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)$, if $\pi_E \rightarrow \tilde{\pi}_E$ and $\pi_P \rightarrow \tilde{\pi}_P$.

4.3 ST-bisimulations

Let $R \subseteq ST - \Pi(N) \times ST - \Pi(N') \times \mathcal{B}$, where $\mathcal{B} = \{\beta | \beta : T_C \rightarrow T_{C'},\ \pi = (C, f) \in \Pi(N),\ \pi' = (C', f') \in \Pi(N')\}$.

In the following definitions $\pi_E = (C_E, f_E),\ \pi_P = (C_P, f_P),\ \pi'_E = (C'_E, f'_E),\ \pi'_P = (C'_P, f'_P),\ \pi = (C, f),\ \pi' = (C', f')$.

R is a *sequential ST-bisimulation* between N and N', written $R : N \underline{\leftrightarrow}_{iST} N'$, if:

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

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

3. $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in R,\ (\pi_E, \pi_P) \rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)\ \Rightarrow\ \exists \tilde{\beta},\ (\tilde{\pi}'_E, \tilde{\pi}'_P) : (\pi'_E, \pi'_P) \rightarrow (\tilde{\pi}'_E, \tilde{\pi}'_P),\ \tilde{\beta}\lceil_{T_{C_E}} = \beta$ and $((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in R$;

4. As previous item but N and N' are transposed.

R is a *partial word ST-bisimulation* between N and N', written $R : N \underline{\leftrightarrow}_{pwST} N'$, if:

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

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

3. $((\pi_E,\pi_P),(\pi'_E,\pi'_P),\beta) \in R,\ (\pi_E,\pi_P) \to (\tilde{\pi}_E,\tilde{\pi}_P) \ \Rightarrow\ \exists\tilde{\beta},\ (\tilde{\pi}'_E,\tilde{\pi}'_P):$ $(\pi'_E,\pi'_P) \to (\tilde{\pi}'_E,\tilde{\pi}'_P),\ \tilde{\beta}\lceil_{T_{C_E}} = \beta,\ \tilde{\beta}^{-1} : T_{C'} \sqsubseteq T_C$, where $\pi_P \xrightarrow{\pi} \tilde{\pi}_E,\ \pi'_P \xrightarrow{\pi'} \tilde{\pi}'_E$ and $((\tilde{\pi}_E,\tilde{\pi}_P),(\tilde{\pi}'_E,\tilde{\pi}'_P),\tilde{\beta}) \in R$;
4. As previous item but N and N' are transposed.

R is a *pomset ST-bisimulation* between N and N', written $R : N\underline{\leftrightarrow}_{pomST}N'$, if:

1. $((\pi_N,\pi_N),(\pi_{N'},\pi_{N'}),\emptyset) \in R$;
2. $((\pi_E,\pi_P),(\pi'_E,\pi'_P),\beta) \in R \ \Rightarrow\ \beta : T_{C_E} \approx T_{C'_E}$ and $\beta(T_{C_P}) = T_{C'_P}$;
3. $((\pi_E,\pi_P),(\pi'_E,\pi'_P),\beta) \in R,\ (\pi_E,\pi_P) \to (\tilde{\pi}_E,\tilde{\pi}_P) \ \Rightarrow\ \exists\tilde{\beta},\ (\tilde{\pi}'_E,\tilde{\pi}'_P):$ $(\pi'_E,\pi'_P) \to (\tilde{\pi}'_E,\tilde{\pi}'_P),\ \tilde{\beta}\lceil_{T_{C_E}} = \beta,\ \tilde{\beta} : T_C \simeq T_{C'}$, where $\pi_P \xrightarrow{\pi} \tilde{\pi}_E,\ \pi'_P \xrightarrow{\pi'} \tilde{\pi}'_E$ and $((\tilde{\pi}_E,\tilde{\pi}_P),(\tilde{\pi}'_E,\tilde{\pi}'_P),\tilde{\beta}) \in R$;
4. As previous item but N and N' are transposed.

R is a *process ST-bisimulation* between N and N', written $R : N\underline{\leftrightarrow}_{prST}N'$, if:

1. $((\pi_N,\pi_N),(\pi_{N'},\pi_{N'}),\emptyset) \in R$;
2. $((\pi_E,\pi_P),(\pi'_E,\pi'_P),\beta) \in R \ \Rightarrow\ \beta : T_{C_E} \approx T_{C'_E}$ and $\beta(T_{C_P}) = T_{C'_P}$;
3. $((\pi_E,\pi_P),(\pi'_E,\pi'_P),\beta) \in R,\ (\pi_E,\pi_P) \to (\tilde{\pi}_E,\tilde{\pi}_P) \ \Rightarrow\ \exists\tilde{\beta},\ (\tilde{\pi}'_E,\tilde{\pi}'_P):$ $(\pi'_E,\pi'_P) \to (\tilde{\pi}'_E,\tilde{\pi}'_P),\ \tilde{\beta}\lceil_{T_{C_E}} = \beta,\ \tilde{\beta} : T_C \simeq T_{C'}, C \simeq C'$, where $\pi_P \xrightarrow{\pi} \tilde{\pi}_E,\ \pi'_P \xrightarrow{\pi'} \tilde{\pi}'_E$ and $((\tilde{\pi}_E,\tilde{\pi}_P),(\tilde{\pi}'_E,\tilde{\pi}'_P),\tilde{\beta}) \in R$;
4. As previous item but N and N' are transposed.

4.4 History preserving bisimulations

Let $R \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$, where $\mathcal{B} = \{\beta | \beta : T_C \to T_{C'},\ \pi = (C,f) \in \Pi(N),\ \pi' = (C',f') \in \Pi(N')\}$.

In the following definitions $\pi = (C,f), \tilde{\pi} = (\tilde{C},\tilde{f}),\ \pi' = (C',f'), \tilde{\pi}' = (\tilde{C}',\tilde{f}')$.

R is a *partial word history preserving bisimulation* between N and N', written $N\underline{\leftrightarrow}_{pwh}N'$, if:

1. $(\pi_N,\pi_{N'},\emptyset) \in R$;
2. $(\pi,\pi',\beta) \in R \ \Rightarrow\ \beta : T_C \approx T_{C'}$;
3. $(\pi,\pi',\beta) \in R,\ \pi \to \tilde{\pi} \ \Rightarrow\ \exists\tilde{\beta},\ \tilde{\pi}' : \pi' \to \tilde{\pi}',\ \tilde{\beta}\lceil_{T_C} = \beta,\ \tilde{\beta}^{-1} : T_{\tilde{C}'} \sqsubseteq T_{\tilde{C}}$ and $(\tilde{\pi},\tilde{\pi}',\tilde{\beta}) \in R$;
4. As previous item but N and N' are transposed.

R is a *pomset history preserving bisimulation* between N and N', written $N \underline{\leftrightarrow}_{pomh} N'$, if:

1. $(\pi_N, \pi_{N'}, \emptyset) \in R$;
2. $(\pi, \pi', \beta) \in R \ \Rightarrow \ \beta : T_C \simeq T_{C'}$;
3. $(\pi, \pi', \beta) \in R, \ \pi \to \tilde{\pi} \ \Rightarrow \ \exists \tilde{\beta}, \ \tilde{\pi}' : \pi' \to \tilde{\pi}', \ \tilde{\beta}\lceil_{T_C} = \beta$ and $(\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in R$;
4. As previous item but N and N' are transposed.

R is a *process history preserving bisimulation* between N and N', written $N \underline{\leftrightarrow}_{prh} N'$, if:

1. $(\pi_N, \pi_{N'}, \emptyset) \in R$;
2. $(\pi, \pi', \beta) \in R \ \Rightarrow \ \beta : T_C \simeq T_{C'}$;
3. $(\pi, \pi', \beta) \in R, \ \pi \to \tilde{\pi} \ \Rightarrow \ \exists \tilde{\beta}, \ \tilde{\pi}' : \pi' \to \tilde{\pi}', \ \tilde{\beta}\lceil_{T_C} = \beta, \ \tilde{C} \simeq \tilde{C}'$ and $(\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in R$;
4. As previous item but N and N' are transposed.

Obviously, we can simplify these definitions by dealing with one-action process extensions. An iteration of such extensions produces an extension by process.

5 A comparison of net equivalences

In this Section a theorem establishing correlation between all introduced equivalences is proved.

Theorem 1 *Let* $\sim \in \{\equiv, \underline{\leftrightarrow}\}$ *and* $\alpha, \beta \in \{i, \ s, \ pw, \ pom, \ pr, \ iST, \ pwST, \ pomST, \ prST, \ pwh, \ pomh, \ prh\}$. *For nets* N *and* N' $N \sim_\alpha N' \ \Rightarrow \ N \sim_\beta N'$ *iff there exists a directed path in a graph in Figure 2* $\sim_\alpha \to \cdots \to \sim_\beta$.

Proof.
$\Leftarrow$ Let us check that all implications in Figure 2 are valid.

- Firstly we check horizontal implications.
  - The connection $\equiv_s \to \equiv_i$ follows from the fact that sequential trace $a_1 \cdots a_n \in Act^*$ is a step trace $A_1 \cdots A_n \in (\mathcal{M}(Act))^*$ so that $A_1 = \{a_1\}, \ldots, A_n = \{a_n\}$.
  - The connection $\equiv_{pw} \to \equiv_s$ follows from the fact that trace $A_1 \cdots A_n \in (\mathcal{M}(Act))^*$, where $\pi_N \stackrel{A_1}{\to} \pi_1 \stackrel{A_2}{\to} \ldots \stackrel{A_n}{\to} \pi_n$, is corresponded by pomset ρ, an isomorphism class of T_{C_n}.

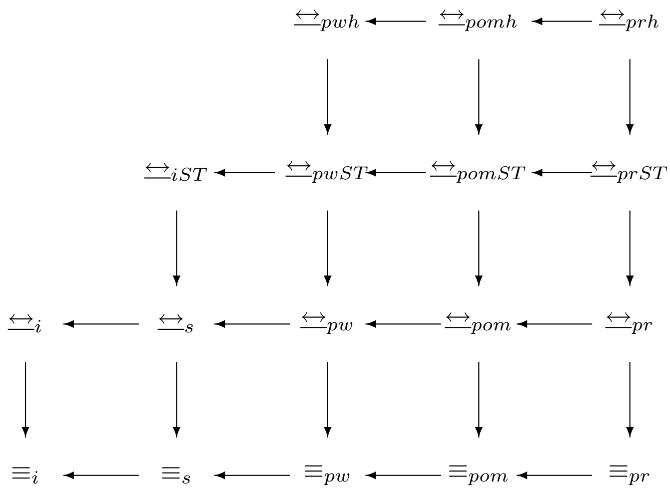

Figure 2: Correlation of equivalences

- The connection $\underline{\leftrightarrow}_s \rightarrow \underline{\leftrightarrow}_i$ is easy to prove if to deal with one-element multiset extensions in the step bisimulation definition.
- The connection $\underline{\leftrightarrow}_{pw} \rightarrow \underline{\leftrightarrow}_s$ becomes obvious if in the partial word bisimulation definition we use extending processes with unordered by precedence relation C-net transitions.
- The connection $\underline{\leftrightarrow}_{pwST} \rightarrow \underline{\leftrightarrow}_{iST}$ is established due to the fact that homomorphism on process C-net transitions is a label-preserving bijection.
- The implications between pomset and partial word equivalences are sequences of the fact that an equivalence is more strict than a label-preserving bijection on process C-net transitions.
- The implications between process and pomset equivalences are sequences of the fact that isomorphic C-nets of processes have isomorphic transition sets.

- Let us prove now the validity of vertical implications.
  - The connection $\underline{\leftrightarrow}_i \rightarrow \equiv_i$ is established as follows. Let $R : N\underline{\leftrightarrow}_i N'$. If $\pi_N \stackrel{a_1}{\rightarrow} \pi_1 \stackrel{a_2}{\rightarrow} \ldots \stackrel{a_n}{\rightarrow} \pi_n$, then there exists a sequence $(\pi_N, \pi_{N'}),\ (\pi_1, \pi_1'), \ldots, (\pi_n, \pi_n') \in R$ so that $\pi_{N'} \stackrel{a_1}{\rightarrow} \pi_1' \stackrel{a_2}{\rightarrow} \ldots \stackrel{a_n}{\rightarrow} \pi_n'$, and vice versa due to the symmetry of a bisimulation defintion.
  - The connection $\underline{\leftrightarrow}_s \rightarrow \equiv_s$ is proved as in the previous item but we use $A_1, .., A_n \in \mathcal{M}(Act)$ instead of $a_1, \ldots, a_n \in Act$.
  - The connection $\underline{\leftrightarrow}_{pw} \rightarrow \equiv_{pw}$ is proved as follows. Let $R : N\underline{\leftrightarrow}_{pw} N'$ and ρ be an isomorphism class of T_C for $\pi = (C, f) \in \Pi(N)$. Since

for any net N, its initial process π_N and a process π $\pi_N \xrightarrow{\pi} \pi$ is valid then there exists a pair $(\pi, \pi') \in R$ so that $\pi' = (C', f')$ and $T_{C'} \sqsubseteq T_C$. If ρ' is an isomorphism class of $T_{C'}$ then $\rho' \sqsubseteq \rho$. It means that $Pomsets(N') \sqsubseteq Pomsets(N)$. The fact that $Pomsets(N) \sqsubseteq Pomsets(N')$ is proved similarly, using a symmetry of the bisimulation definition.

- The connection $\underline{\leftrightarrow}_{pom} \rightarrow \equiv_{pom}$ is proved as in the previous item but using isomorphism instead of homomorphism on process C-net transitions.
- The connection $\underline{\leftrightarrow}_{pr} \rightarrow \equiv_{pr}$ is proved analogously to the previous item using process traces instead of pomset traces and isomorphism on C-nets of processes instead of isomorphism on their transitions.
- The implications $\underline{\leftrightarrow}_{\alpha ST} \rightarrow \underline{\leftrightarrow}_{\alpha}$, $\alpha \in \{pw, pom, pr\}$, are proved by construction of the relation $S : N \underline{\leftrightarrow}_{\alpha} N'$ on the base of relation $R : N \underline{\leftrightarrow}_{\alpha ST} N'$ defined as follows:
$\exists \beta \ ((\pi, \pi), (\pi', \pi'), \beta) \in R \ \Leftrightarrow \ (\pi, \pi') \in S$.
- The connection $\underline{\leftrightarrow}_{iST} \rightarrow \underline{\leftrightarrow}_{s}$ is checked as in the previous item but taking into account the fact that the step $\pi \xrightarrow{A} \tilde{\pi}$, where
$A = \{a_1, \ldots, a_n\} \in \mathcal{M}(Act)$, is corresponded by a sequence of ST-processes $(\pi, \pi), \ldots, (\tilde{\pi}, \pi), \ldots, (\tilde{\pi}, \tilde{\pi})$ based on two equal process extensions $\pi \xrightarrow{a_1} \ldots \xrightarrow{a_n} \tilde{\pi}$.
- The implications $\underline{\leftrightarrow}_{\alpha h} \rightarrow \underline{\leftrightarrow}_{\alpha ST}$, $\alpha \in \{pw, pom, pr\}$, are proved by construction of the relation $S : N \underline{\leftrightarrow}_{\alpha ST} N'$ on the base of relation $R : N \underline{\leftrightarrow}_{\alpha h} N'$ as follows: $(\pi_E, \pi'_E, \beta) \in R$, $(\pi_E, \pi_P) \in ST - \Pi(N)$, $(\pi'_E, \pi'_P) \in ST - \Pi(N')$, $\beta(T_{C_P}) = T_{C'_P} \ \Leftrightarrow \ ((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in S$.

$\Rightarrow$ Let us prove that it is impossible to draw any arrow from one equivalence to the other in Figure 2 such that there exists no directed path from the first equivalence to the second one in a graph in this Figure. For this, it is enough to prove the absence of the following connections: $\underline{\leftrightarrow}_i \rightarrow \equiv_s$, $\equiv_{pr} \rightarrow \underline{\leftrightarrow}_i$, $\underline{\leftrightarrow}_{pwh} \rightarrow \equiv_{pom}$, $\underline{\leftrightarrow}_{pomh} \rightarrow \equiv_{pr}$, $\underline{\leftrightarrow}_{iST} \rightarrow \equiv_{pw}$, $\underline{\leftrightarrow}_{pr} \rightarrow \underline{\leftrightarrow}_{iST}$, $\underline{\leftrightarrow}_{prST} \rightarrow \underline{\leftrightarrow}_{pwh}$.

- In Figure 3.1 $N \underline{\leftrightarrow}_i N'$ but $N \not\equiv_s N'$ since there exists a step trace $\{a, b\}$ in N which is not in N'.
- In Figure 3.2 $N \equiv_{pr} N'$ but $N \underline{\not\leftrightarrow}_i N'$ since only in N it is possible to execute an action a so that it is impossible to run b after it.
- In Figure 3.3 $N \underline{\leftrightarrow}_{pwh} N'$ but $N \not\equiv_{pom} N'$ since b can depend on a in N.
- In Figure 3.4 $N \underline{\leftrightarrow}_{pomh} N'$ but $N \not\equiv_{pr} N'$ since N is a C-net which is not isomorphic to C-net N'.
- In Figure 3.5 $N \underline{\leftrightarrow}_{iST} N'$ but $N \not\equiv_{pw} N'$ since a net N is corresponded by a pomset such that we can not execute even less sequential one in N'.

- In Figure 4.1 $N \underline{\leftrightarrow}_{pr} N'$ but $N \underline{\not\leftrightarrow}_{iST} N'$ since an action a is able to begin working in N' so that no b can start later.
- In Figure 4.2 $N \underline{\leftrightarrow}_{prST} N'$ but $N \underline{\not\leftrightarrow}_{pwh} N'$ since only N' can execute a and b so that the next action, c, must depend on a. □

6 Equivalences on sequential nets

A net $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ is *sequential* if $\forall \pi = (C, f) \in \Pi(N)$ $\forall v, w \in T_C$ $(v \prec_C w) \vee (w \prec_C v)$, i.e. $\prec_C$ is a linear (total) ordering on C-net transitions of any process $\pi = (C, f)$ of a net N.

Let $\pi = (C, f) \in \Pi(N)$ for some net N and $v \in T_C$. *A set of predecessors* of v is defined as follows: $\downarrow v = \{w \in T_C | w \prec_C v\}$.

Lemma 1 *A transition $v \in T_C$ may occur in a full execution of a C-net C iff all transitions from $\downarrow v$ have already been fired.*

Proof.

$\Rightarrow$ If transition $v \in T_C$ may be fired then there are tokens in places from ${}^\bullet v$. We can divide these places for two groups.

1. $q \in {}^\bullet v$ ${}^\bullet q = \emptyset$. Then $q \in {}^\circ C$ and by definition of full execution of C-net there is a token in q.
2. $q \in {}^\bullet v$ and ${}^\bullet q \neq \emptyset$. Because of the unbranching of places in C-nets, a token can appear in q only after firing of the only transition $w \in T_C$, where ${}^\bullet q = \{w\}$.

So, for firing of transition v all transitions from ${}^\bullet({}^\bullet v) = {}^{2\bullet} v$ have to be fired, and for firing of these transitions from ${}^{4\bullet} v$ must occur and so on. Since by definition of C-nets there are no backward infinite chains in F_C then there exists $n \in \mathcal{N}$ such that ${}^{2n\bullet} v = \emptyset$. Consequently for firing of v the firing of all transitions from $\cup_{i=1}^{n-1} {}^{2i\bullet} v = \downarrow v$ is necessary.

$\Leftarrow$ If all transitions from $\downarrow v$ have been fired in a full execution of C for $v \in T_C$ then transitions from ${}^{2\bullet} v$ have been already fired too. Therefore, all places from ${}^\bullet v$ have tokens, and v can be fired. □

Lemma 2 *Let C be a C-net and $v, w \in T_C$. In this case $v \prec_C w$ iff v occurs before w in any full execution of C.*

Proof.

$\Rightarrow$ By lemma 1 a transition $w \in T_C$ may be fired only then all transitions from $\downarrow w$ occured but $v \in \downarrow w$.

$\Leftarrow$ Let us prove by contradiction. Let $\neg(v \prec_C w)$. Two cases are possible.

1. $w \prec_C v$. Then by proved upper w occurs before v in any full execution of C. It is in contrary with our initial assumption.

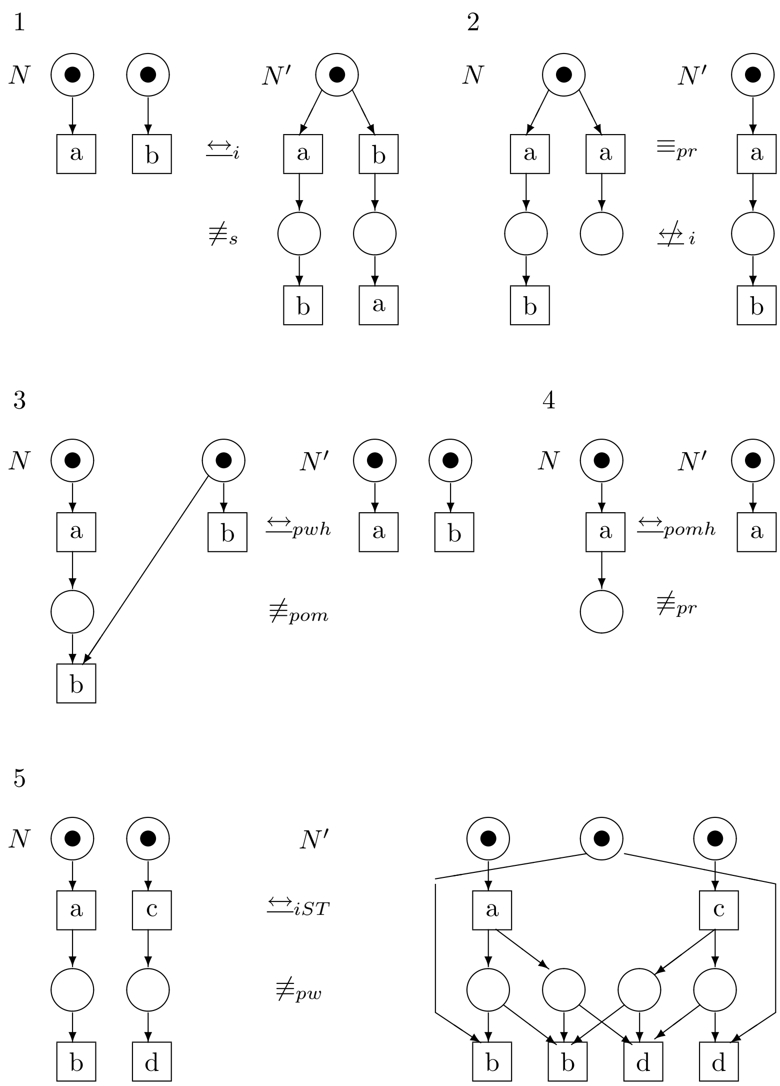

Figure 3: Examples of nets

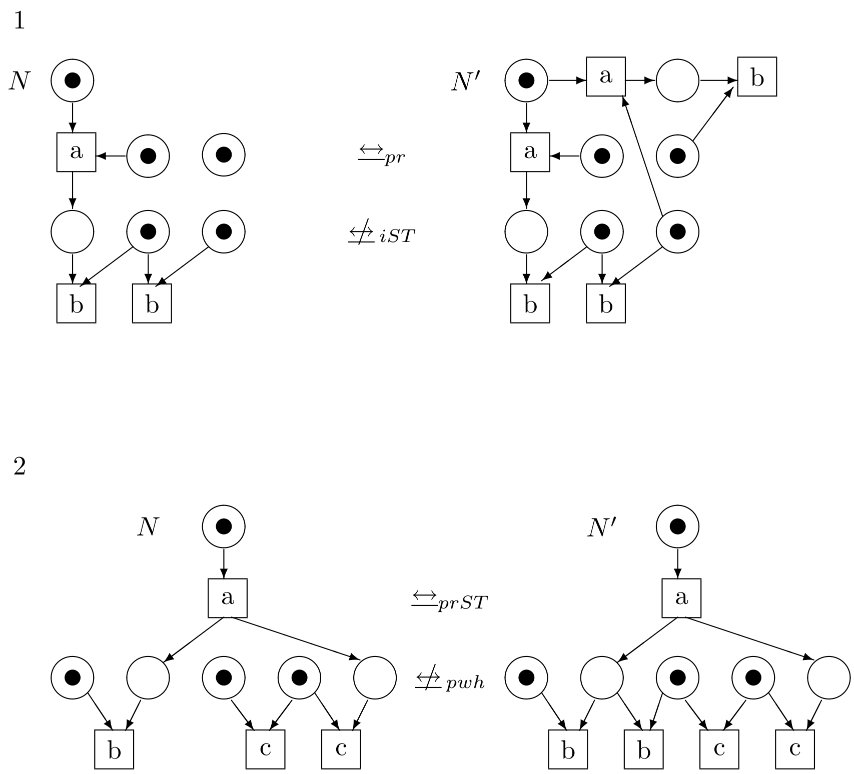

Figure 4: Examples of nets (continued)

2. $\neg(w \prec_C v)$, i.e. v and w are not related by $\prec_C$. Then there exists a full execution of C where all transitions from $\downarrow w$ occur from the very beginning and w does after it. Since $v \notin \downarrow w$, w occurs before v in this full execution.

So both cases must be rejected. Consequently, $v \prec_C w$. □

Proposition 1 *For sequential nets N and N' $N \underline{\leftrightarrow}_i N' \Leftrightarrow N \underline{\leftrightarrow}_{pomh} N'$.*

Proof.

$\Leftarrow$ Sequence of theorem 1.

$\Rightarrow$ Let $R : N \underline{\leftrightarrow}_i N'$ for sequential nets N and N'. Let us prove that $S : N \underline{\leftrightarrow}_{pomh} N'$ where S is defined as follows: $(\pi, \pi') \in R,\ \beta : T_C \simeq T_{C'} \Leftrightarrow (\pi, \pi', \beta) \in S$ where $\pi = (C, f),\ \pi' = (C', f')$. Items 1 and 2 of the pomset h-bisimulation definition are valid automatically. By force of symmetry of items 3 and 4 of this definition it is enough to prove only item 3. Also we may deal only with one-action extensions of processes.

Let $(\pi, \pi', \beta) \in S,\ \pi \stackrel{v}{\rightarrow} \tilde{\pi},\ \tilde{\pi} = (\tilde{C}, \tilde{f})$. By definition of sequential nets all transitions in C-nets of their processes are linear ordered by precedence relation. Therefore full executions of such nets are singletons by lemma 2. Let us consider a full execution of $\tilde{C}$ in N $\pi_N = \pi_0 \stackrel{v_1}{\rightarrow} \dots \stackrel{v_n}{\rightarrow} \pi_n = \pi \stackrel{v}{\rightarrow} \tilde{\pi}$. By sequential bisimulation definition there exist $\tilde{\pi}',\ v'$ so that $\pi_{N'} = \pi'_0 \stackrel{\beta(v_1)}{\rightarrow} \dots \stackrel{\beta(v_n)}{\rightarrow} \pi'_n = \pi' \stackrel{v'}{\rightarrow} \tilde{\pi}',\ \tilde{\pi}' = (\tilde{C'}, \tilde{f'}),\ l_{\tilde{C}}(v) = l_{\tilde{C'}}(v'),\ (\tilde{\pi}, \tilde{\pi}') \in R$. It is the only full execution of $\tilde{C'}$ in N'.

Let us consider the mapping $\tilde{\beta}$ where $\tilde{\beta}\lceil_{T_C} = \beta$ and $\tilde{\beta}(v) = v'$. Obviously, $\tilde{\beta} : T_{\tilde{C}} \approx T_{\tilde{C'}}$. Let us prove that $\tilde{\beta} : T_{\tilde{C}} \simeq T_{\tilde{C'}}$. For this it is sufficient to prove the following: $\forall w \in T_C\ w \prec_{\tilde{C}} v \Leftrightarrow \tilde{\beta}(w) \prec_{\tilde{C'}} \tilde{\beta}(v)$. Let $w \prec_{\tilde{C}} v$. Then w occurs before v in any full execution of $\tilde{C}$. Consequently, it is in the only full execution of this net. By definition of $\tilde{\beta}$ $\tilde{\beta}(w)$ occurs before $\tilde{\beta}(v)$ in the only full execution of $\tilde{C'}$. So we can assert that $\tilde{\beta}(w)$ occurs before $\tilde{\beta}(v)$ in any full execution of $\tilde{C'}$. Then by lemma 2 $\tilde{\beta}(w) \prec_{\tilde{C'}} \tilde{\beta}(v)$. Preserving of precedence relation in the opposite direction is proved analogously. □

Proposition 2 *For sequential nets N and N' $N \equiv_i N' \Leftrightarrow N \equiv_{pom} N'$.*

Proof.

$\Leftarrow$ Sequence of theorem 1.

$\Rightarrow$ Let $N \equiv_i N'$. Let us prove that $N \equiv_{pom} N'$, i.e. that $Pomsets(N) = Pomsets(N')$. Let $\pi = (C, f) \in \Pi(N)$. Since N is a sequential net then there exists the only full execution of C in N $\pi_N = \pi_0 \stackrel{v_1}{\rightarrow} \dots \stackrel{v_n}{\rightarrow} \pi_n = \pi$. Since $N \equiv_i N'$ then there exists $\pi'_i,\ v'_i\ (1 \le i \le n)$ so that $\pi_{N'} = \pi'_0 \stackrel{v'_1}{\rightarrow} \dots \stackrel{v'_n}{\rightarrow} \pi'_n = \pi'$ $\pi' = (C', f'),\ l_C(v_i) = l_{C'}(v'_i)$. It is the only full execution of C' in N'. Let us define a mapping β as follows: $\beta(v_i) = v'_i$. Clearly, $\beta : T_C \approx T_{C'}$. By lemma 2 and due to the uniqueness of C and C' full executions the order of firing of these net transitions determines a total ordering on these transitions.

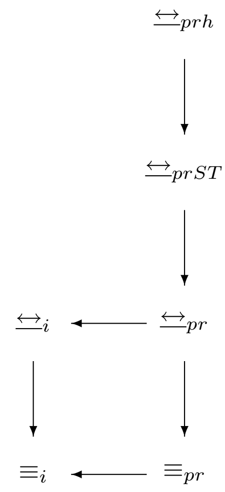

Figure 5: Equivalences on sequential nets

Therefore $\forall v, w \in T_C \ v \prec_C w \ \Leftrightarrow \ \beta(v) \prec_{C'} \beta(w)$, i.e. $\beta : T_C \simeq T_{C'}$. Hence $Pomsets(N) \subseteq Pomsets(N')$. The fact that $Pomsets(N') \subseteq Pomsets(N)$ is proved analogously. □

Theorem 2 *Let* $\sim \in \{\equiv, \underline{\leftrightarrow}\}$, $\alpha, \beta \in \{i, pr, prST, prh\}$. *For sequential nets* N *and* N' $N \sim_\alpha N' \ \Rightarrow \ N \sim_\beta N'$ *iff thete exists a directed path* $\sim_\alpha \rightarrow \cdots \ \rightarrow \sim_\beta$ *in graph in Figure 5.*

Proof.

$\Leftarrow$ Sequence of theorem 1.

$\Rightarrow$ Let us prove that it is impossible to draw any arrow from one equivalence to the other in Figure 4 such that there exists no directed path from the first equivalence to the second one. For proving it is sufficient to show the absence of the following connections: $\underline{\leftrightarrow}_i \rightarrow \equiv_{pr}$, $\equiv_{pr} \rightarrow \underline{\leftrightarrow}_i$, $\underline{\leftrightarrow}_{pr} \rightarrow \underline{\leftrightarrow}_{prST}$, $\underline{\leftrightarrow}_{prST} \rightarrow \underline{\leftrightarrow}_{prh}$.

- In Figure 3.4 $N \underline{\leftrightarrow}_i N'$ but $N \not\equiv_{pr} N'$.
- In Figure 3.2 $N \equiv_{pr} N'$ but $N \underline{\not\leftrightarrow}_i N'$.
- In Figure 6.1 $N \underline{\leftrightarrow}_{pr} N'$ but $N \underline{\not\leftrightarrow}_{prST} N'$ since only in N' we can begin running a process with action a so that it may be extended by action b in the only way (i.e. so that extended process be only one).
- In Figure 6.2 $N \underline{\leftrightarrow}_{prST} N'$ but $N \underline{\not\leftrightarrow}_{prh} N'$ since only in N' it is possible to run a process with sequential occuring actions a and b so that the next action, c, may extend this process only in one way (i.e. C-net with

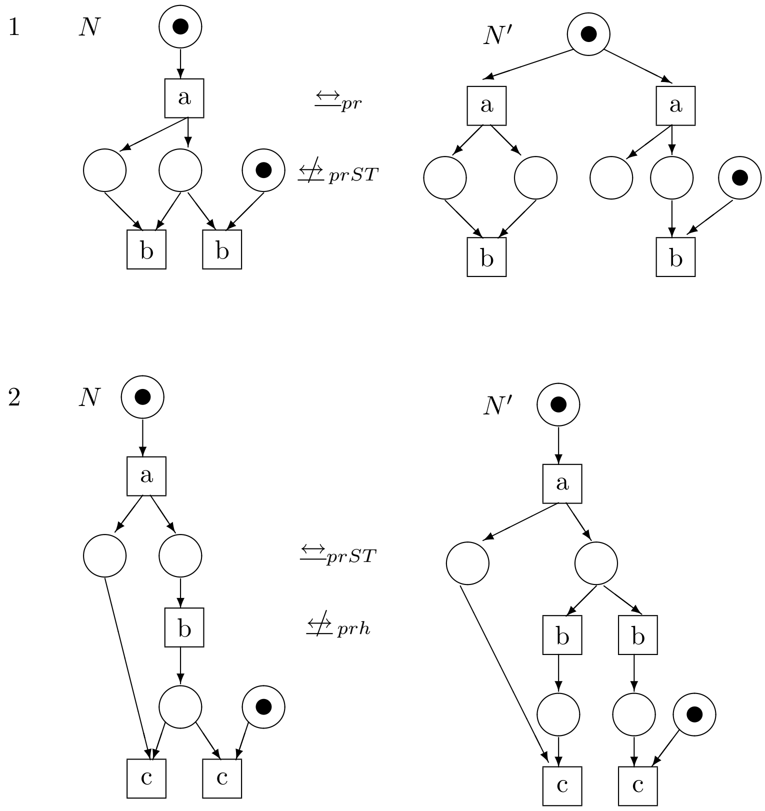

Figure 6: Examples of sequential nets

action c, extending a C-net corresponding to sequence ab, connects with its subnet containing a, in the only way). □

7 Conclusion

A large group of the Petri net equivalences is introduced in the paper. A correlation between these equivalences on nets with finite processes without λ-actions is found out. In addition it is considered which equivalences coincide on sequential nets.

Further development of the subject consist in exploration of introduced equivalences on C-nets and on strict labelled nets (for C-nets without auto-concurrency the merging of sequential equivalence and step bisimulation equivalence was proved by author). Nets with strict labelling are interesting by the fact that A-nets denoted by formulas of the algebra AFP_0 introduced by V.E.Kotov in [14] (the description is in [5, 7]) are a subclass of these nets.

It is interesting to find out how formula representations of equivalent A-nets are connected. Also it is possible to compare the introduced equivalences with those considered in [5, 6, 8, 9] devoted to the algebras AFP_1 and AFP_2.

The next direction of the development of this theme may be the examination of the proposed equivalences on the wider net class, exactly, on nets with λ-actions. Probably some equivalences would stop to be connected on such nets. In [18, 19] the example of event structures with λ-actions was considered which demonstrated the independence of ST-bisimulation equivalences and h-bisimulation equivalence on such event structures.

Finally we would find out how ST- and h-equivalences are connected with place bisimulation equivalences introduced in [1, 2].

References

[1] C.Autant, Z.Belmesk, Ph.Schnoebelen: *Strong bisimularity on nets revisited.* Research Report 847-I, LIFIA-IMAG, Grenoble, France, 28 p., March 1991.

[2] C.Autant, Ph.Schnoebelen: *Place bisimulations in Petri nets.* LNCS 616, p.45–61, June 1992.

[3] Luca Bernardinello, Fiorella de Cindio: *A survey of basic net models and modular net classes.* Design methods based on nets, Esprit basic research action 3148, June 1989–June 1990.

[4] Eike Best, Raymond Devillers, Astrid Kiehn, Lucia Pomello: *Concurrent bisimulations in Petri nets.* Springer Verlag, Acta Informatica 28, p.231–264, 1991.

[5] Ludmila Cherkasova: *Posets with non-actions: A model for concurrent nondeterministic processes.* Arbeitspapiere der GMD 403, 68p., July 1989.

[6] Ludmila Cherkasova: *Algebra AFP_2 for concurrent nondeterministic processes: Fully abstract model and complete axiomatization.* Reihe Informatic 72, 28p., July 1990.

[7] Ludmila Cherkasova: *A fully abstract model for concurrent nondeterministic processes based on posets with non-actions.* Computer Science, Department of Software Technology, Report CS-R 9031, Center for Mathematics and Computer Science, Amsterdam, Netherlands, 42p., July 1990.

[8] Ludmila Cherkasova, Vadim Kotov: *Concurrent nondeterministic processes: Adequacy of structure and behaviour.* LNCS 379, p.67–87, 1989.

[9] Ludmila Cherkasova, Vadim Kotov: *Descriptive and analitical process algebras.* LNCS 424, p.77–104, 1989.

[10] Raymond Devillers: *Maximality preserving bisimulation.* Theoretical Computer Science 102, p.165–184, 1992.

[11] Rob van Glabbeek, Ursula Goltz: *Equivalence notions for concurrent systems and refinement of actions.* LNCS 379, p.237–248, 1989.

[12] Rob van Glabbeek: *The refinement theorem for ST-bisimulation semantics.* In: M.Broy and C.B.Jones, editors. Proc. IFIP Working Conference of Programming Concepts and Methods, Sea of Galilee, Israel, 1990, to appear.

[13] Rob van Glabbeek, Frits Vaandrager: *Petri net models for algebraic theories of concurrency.* LNCS 259, p.224–242, 1987.

[14] Vadim Kotov: *An algebra for parallelism based on Petri nets.* LNCS 64, p.39–55, 1978.

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

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

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

[18] Walter Vogler: *Bisimulation and action refinement.* LNCS 480, p.309–321, 1991.

[19] Walter Vogler: *Modular construction and partial order semantics of Petri nets.* Springer Verlag, LNCS 625, 252p., 1992.