Equivalence Relations for Net and Algebraic Models of Concurrency

Igor V. Tarasyuk

A.P. Ershov Institute of Informatics Systems
Siberian Division of the Russian Academy of Sciences
6, Acad. Lavrentiev pr., Novosibirsk 630090, Russia

itar@iis.nsk.su

www.iis.nsk.su\persons\itar\index.html

1. Equivalences for Petri nets
2. Equivalences for Petri nets with silent transitions
3. Equivalences for process algebras: calculus AFP_2

Equivalences for Petri Nets

Abstract: Behavioural equivalences of concurrent systems modeled by Petri nets are considered.

Known basic, back-forth and place bisimulation equivalences are supplemented by new ones.

The equivalence interrelations are examined for the general Petri nets as well as for their subclasses of sequential nets (no concurrent transitions), strictly labeled nets (unlabeled) and T-nets (no place branching).

A logical characterization of back-forth bisimulation equivalences in terms of logics with past modalities is proposed.

An effective net reduction method based on place bisimulation relations is presented.

A preservation of all the equivalences by refinements is investigated to find out their appropriateness for top-down design.

Keywords: Petri nets, sequential nets, strictly labeled nets, T-nets, basic equivalences, back-forth bisimulations, place bisimulations, logical characterization, net reduction, refinement.

Contents

- Back-forth simulation and logics
  - Sequential runs
  - Back-forth bisimulation equivalences
  - Comparing back-forth bisimulation equivalences
  - Comparing back-forth bisimulation equivalences with basic ones
  - Logic HML
  - Logic $PBFL$
  - Logic $PrBFL$
- Place simulation and net reduction
  - Place bisimulation equivalences
  - Comparing place bisimulation equivalences
  - Comparing place bisimulation equivalences with basic and back-forth ones
  - Net reduction based on place bisimulation equivalences
- Refinements
  - SM-refinements
- Net subclasses
  - The equivalences on sequential nets
  - The equivalences on strictly labeled nets
  - The equivalences on T-nets
- Decidability
  - Decidability results for the equivalences

Previous work

The following basic equivalences are known:

- *Trace equivalences* (respect protocols of behaviour):

  interleaving ($\equiv_i$) [Hoa80], step ($\equiv_s$) [Pom86], partial word ($\equiv_{pw}$) [Gra81] and pomset ($\equiv_{pom}$) [Pra86].

- *Usual bisimulation equivalences* (respect branching structure of behaviour):

  interleaving ($\leftrightarrow_i$) [Par81], step ($\leftrightarrow_s$) [NT84], partial word ($\leftrightarrow_{pw}$)[Vog91a], pomset ($\leftrightarrow_{pom}$) [BCa87] and process ($\leftrightarrow_{pr}$) [AS92].

- *ST-bisimulation equivalences* (respect the duration or maximality of events in behaviour):

  interleaving ($\leftrightarrow_{iST}$) [GV87], partial word ($\leftrightarrow_{pwST}$) [Vog91a] and pomset ($\leftrightarrow_{pomST}$) [Vog91a].

- *History preserving bisimulation equivalences* (respect the "history" of behaviour):

  pomset ($\leftrightarrow_{pomh}$) [RT88].

- *Conflict preserving equivalences* (respect conflicts of events):

  occurrence ($\equiv_{occ}$) [NPW81].

- *Isomorphism* (coincidence up to renaming of components):

  ($\simeq$).

Back-forth bisimulation equivalences: bisimulation relation do not only require systems to simulate each other behavior in the forward direction but also when going back in history, i.e. backward.

They are connected with equivalences of logics with past modalities.

Interleaving *back* interleaving *forth bisimulation equivalence* ($\leftrightarrow_{ibif} = \leftrightarrow_i$) [NMV90].

Step *back* step *forth* ($\leftrightarrow_{sbsf}$), partial word *back* partial word *forth* ($\leftrightarrow_{pwbpwf}$) and pomset *back* pomset *forth* ($\leftrightarrow_{pombpomf}$) *bisimulation equivalences* [Che92a, Che92b, Che92c].

All possible *back-forth equivalences* in interleaving, step, partial word and pomset semantics s.t. types of backward and forward simulations may differ. New relations: step *back* partial word *forth* ($\leftrightarrow_{sbpwf}$) and step *back* pomset *forth* ($\leftrightarrow_{sbpomf}$) *bisimulation equivalences* [Pin93].

Place bisimulation equivalences [ABS91] are based on definition from [Old89,Old91]. They are relations over places instead of markings or processes. The relation on markings is obtained via *"lifting"* that on places.

The main application of the place equivalences is effective behaviour preserving reduction of Petri nets.

Interleaving *place bisimulation equivalence* ($\sim_i$) and interleaving *strict place bisimulation equivalence* ($\approx_i$) [ABS91].

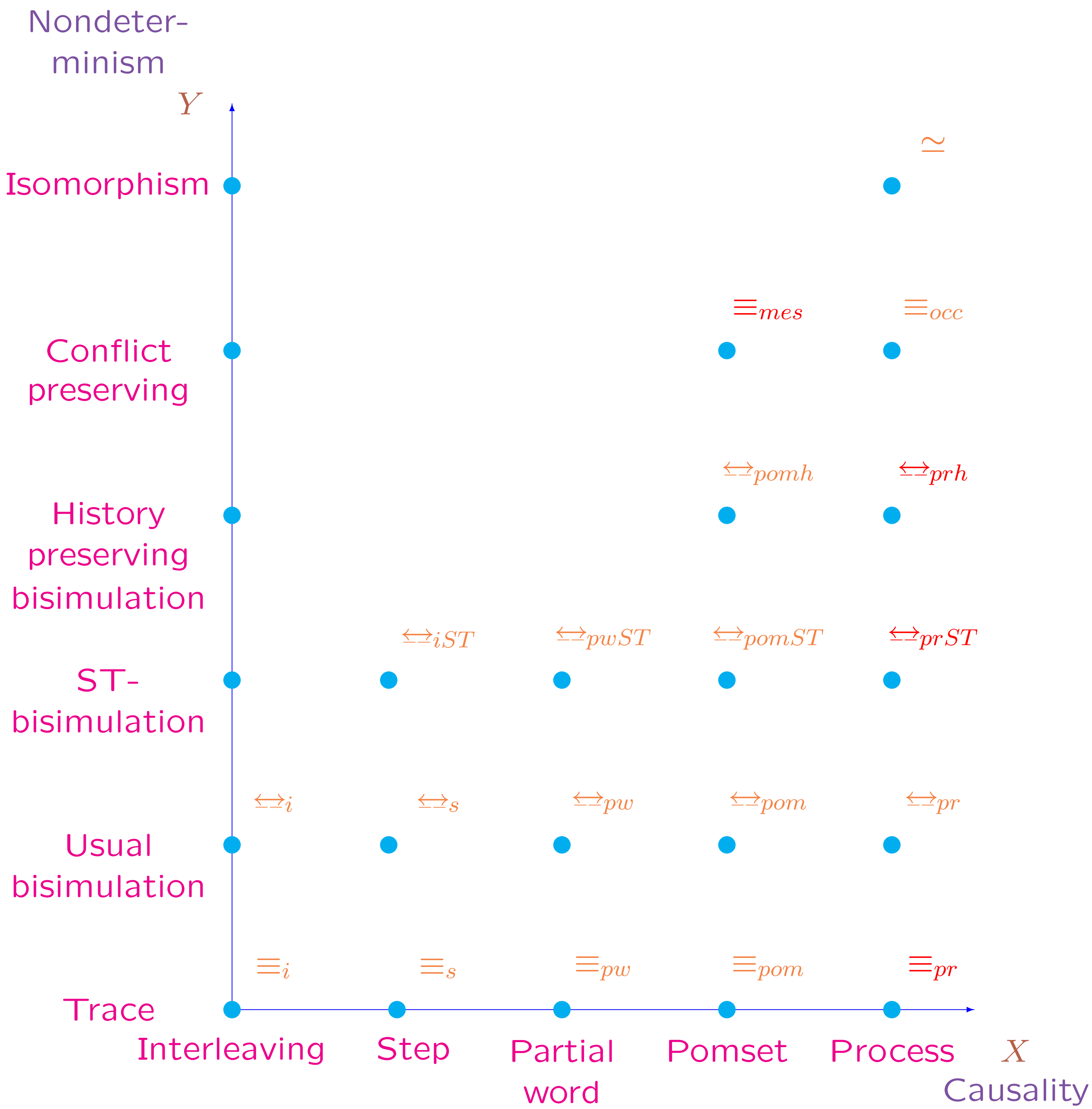
Step ($\sim_s$), partial word ($\sim_{pw}$), pomset ($\sim_{pom}$) and process ($\sim_{pr}$) *place bisimulation equivalences*. Their *strict analogues*: ($\approx_s$, $\approx_{pw}$, $\approx_{pom}$, $\approx_{pr}$).

Merging: $\sim_i = \sim_s = \sim_{pw}$ and $\approx_s = \approx_{pw} = \approx_{pom} = \approx_{pr} = \sim_{pr}$. Three different relations remain: $\sim_i$, $\sim_{pom}$ and $\sim_{pr}$ [AS92].

New equivalences

- Basic equivalences:
  process *trace* ($\equiv_{pr}$),
  process *ST-bisimulation* ($\underline{\leftrightarrow}_{prST}$),
  process *history preserving bisimulation* ($\underline{\leftrightarrow}_{prh}$) and
  multi event structure ($\equiv_{mes}$).

- Back-forth bisimulation equivalences:
  step *back* process *forth* ($\underline{\leftrightarrow}_{sbprf}$) and
  pomset *back* process *forth* ($\underline{\leftrightarrow}_{pombprf}$).



Classification of basic equivalences

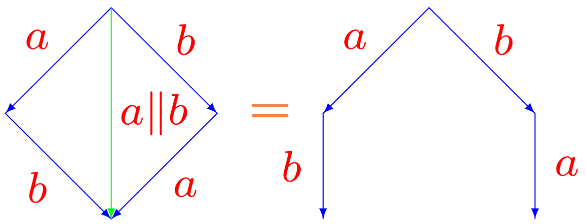
Basic equivalences are positioned on coordinate plane. New relations are depicted in red colour.

Moving along X axis: a degree of causality grows.

Moving along Y axis: a degree of non-determinism grows.

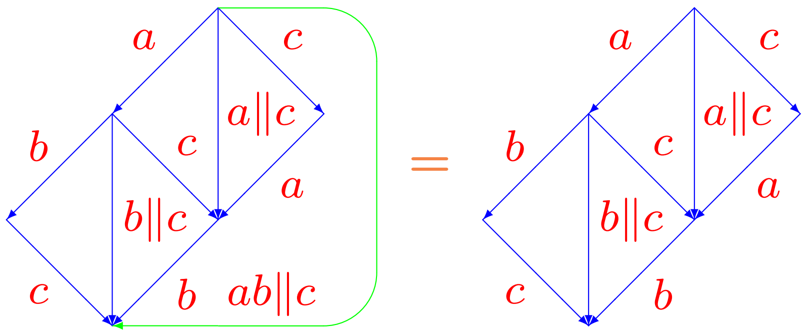
Interleaving

$a||b = ab + ba$



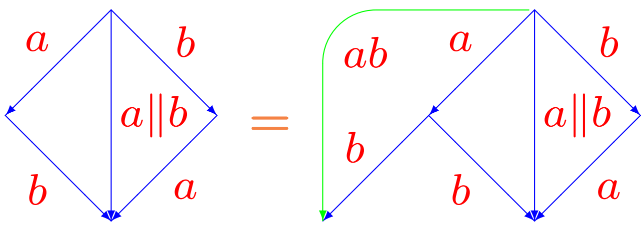
Step

$ab||c = a(b||c) + (a||c)b$



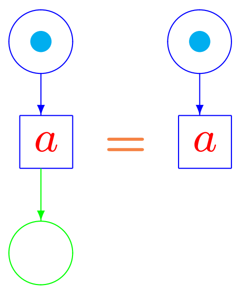
Partial word

$a||b = a||b + ab$



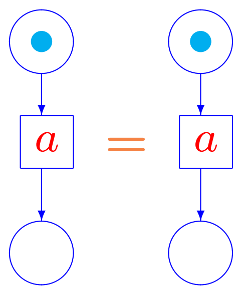
Pomset

$a = a$



Process

$a = a$

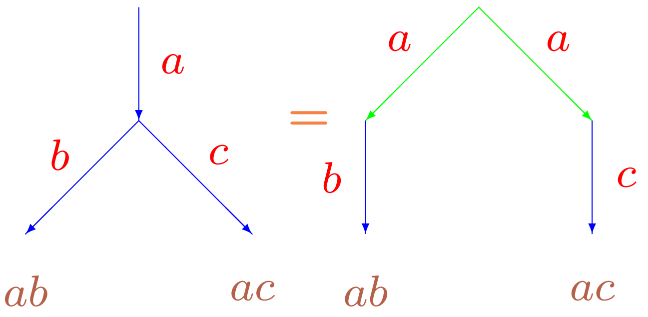


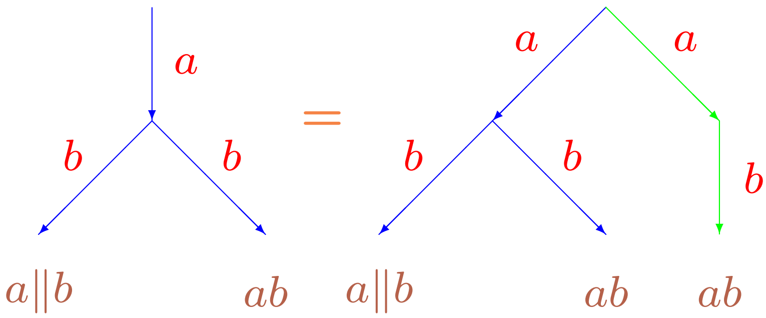
Causality degrees

Trace

Usual bisimulation

$a(b+c) = ab + ac$

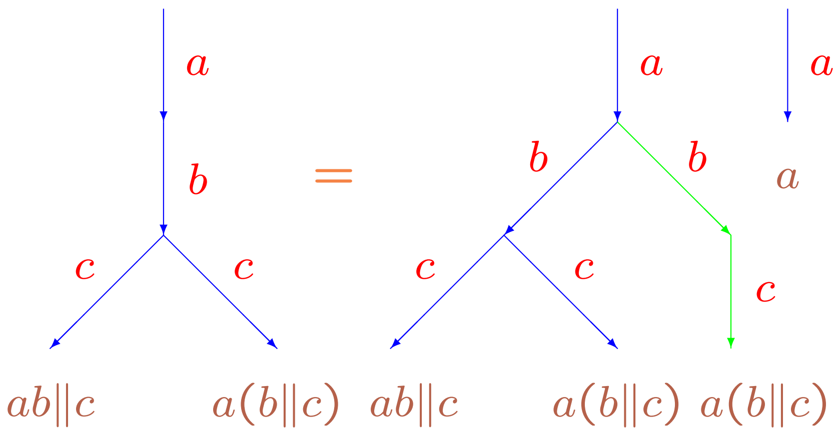


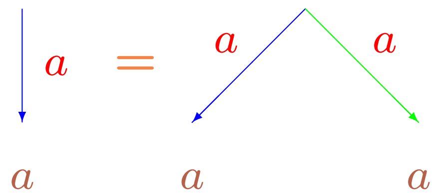


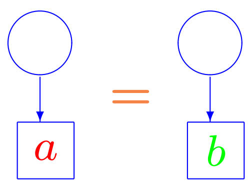
ST-bisimulation

History preserving bisimulation

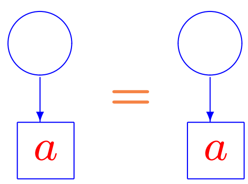
Conflict preserving







Isomorphism



Nondeterminism degrees

Multisets

Definition 1 *A* finite multiset (bag) M *over a set* X *is a mapping* $M : X \to \mathbb{N}$ *s.t.* $|\{x \in X \mid M(x) > 0\}| < \infty$.

$\mathcal{M}(X)$ is the *set of all finite multisets* over X.

For $x \in X$, $M(x)$ is a number of elements x in M.

When $\forall x \in X \; M(x) \leq 1$, M is a proper set.

The *cardinality* of a multiset M: $|M| = \sum_{x \in X} M(x)$.

Let $M_1, M_2 \in \mathcal{M}(X)$ and $x \in X$, then:

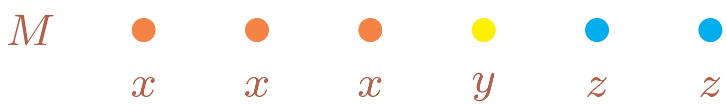
$$\begin{array}{rcl}
(M_1 + M_2)(x) & = & M_1(x) + M_2(x); \\
(M_1 - M_2)(x) & = & \max\{M_1(x) - M_2(x), 0\}; \\
(M_1 \cup M_2)(x) & = & \max\{M_1(x), M_2(x)\}; \\
(M_1 \cap M_2)(x) & = & \min\{M_1(x), M_2(x)\}; \\
M_1 \subseteq M_2 & \Leftrightarrow & \forall x \in X \; M_1(x) \leq M_2(x); \\
x \in M & \Leftrightarrow & M(x) > 0.
\end{array}$$

We write $M + x - y$ for $M + \{x\} - \{y\}$.

The *empty multiset*: $\emptyset$.

Multisets: sets with identical elements.

$M = \{x, x, x, y, z, z\}$ denotes the multiset M s.t. $M(x) = 3$, $M(y) = 1$, $M(z) = 2$, and for other elements M is equal to 0.



Example of multiset

Labeled nets

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

$\tau \notin Act$ denotes *silent* action that represents an internal activity. Let $Act_\tau = Act \cup \{\tau\}$.

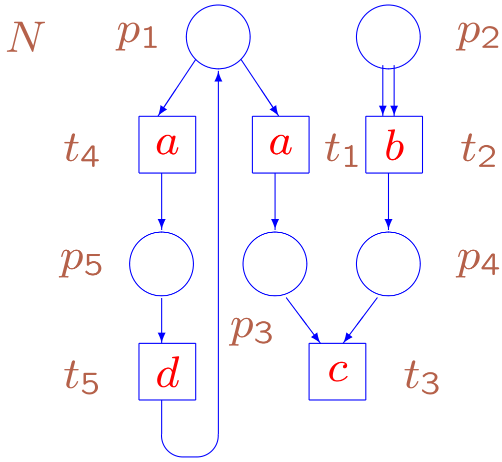
Definition 2 *A* labeled net *is a quadruple* $N = \langle P_N, T_N, F_N, l_N \rangle$*:*

- $P_N = \{p, q, \ldots\}$ *is a set of* places*;*
- $T_N = \{t, u, \ldots\}$ *is a set of* transitions*;*
- $F_N : (P_N \times T_N) \cup (T_N \times P_N) \to \mathbb{N}$ *is the* flow relation *with weights;*
- $l_N : T_N \to Act_\tau$ *is a* labeling *of transitions with action names.*

Given labeled 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 $\beta : P_N \cup T_N \to P_{N'} \cup T_{N'}$ is an *isomorphism* between N and N', denoted by $\beta : N \simeq N'$, if:

1. β is a bijection s.t. $\beta(P_N) = P_{N'}$ and $\beta(T_N) = T_{N'}$;
2. $\forall p \in P_N \ \forall t \in T_N \ F_N(p, t) = F_{N'}(\beta(p), \beta(t))$ and $F_N(t, p) = F_{N'}(\beta(t), \beta(p))$;
3. $\forall t \in T_N \ l_N(t) = l_{N'}(\beta(t))$.

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



Example of labeled net

Let N be a labeled net and $t \in T_N$.

The *precondition* ${}^\bullet t$ and the *postcondition* $t^\bullet$ of t are the multisets: $({}^\bullet t)(p) = F_N(p,t)$ and $(t^\bullet)(p) = F_N(t,p)$. Similar for places: $({}^\bullet p)(t) = F_N(t,p)$ and $(p^\bullet)(t) = F_N(p,t)$.

${}^\circ N = \{p \in P_N \mid {}^\bullet p = \emptyset\}$ is the set of *initial (input)* and $N^\circ = \{p \in P_N \mid p^\bullet = \emptyset\}$ is the set of *final (output)* places of N.

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

A labeled 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 labeled net and $x, y \in P_N \cup T_N$. Then:

- $x \prec_N y \;\Leftrightarrow\; x F_N^+ y$, where F_N^+ is a transitive closure of F_N (*strict causal dependence* relation);
- $x \preceq_N y \;\Leftrightarrow\; (x \prec_N y) \vee (x = y)$ (a relation of *causal dependence*);
- $x \#_N y \;\Leftrightarrow\; \exists t, u \in T_N \; (t \neq u,\; {}^\bullet t \cap {}^\bullet u \neq \emptyset,\; t \preceq_N x,\; u \preceq_N y)$ (a relation of *conflict*);
- $\downarrow_N x = \{y \in P_N \cup T_N \mid y \prec_N x\}$ (the set of *strict predecessors* of x).

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

Marked nets

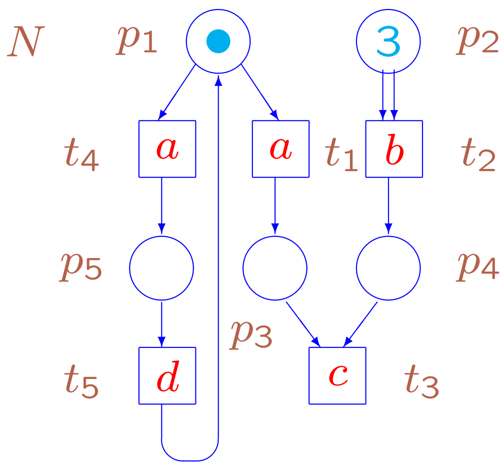
A *marking* of a labeled net N is $M \in \mathcal{M}(P_N)$.

Definition 3 *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 labeled net and* $M_N \in \mathcal{M}(P_N)$ *is the* initial *marking.*

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

1. $\beta : \langle P_N, T_N, F_N, l_N \rangle \simeq \langle P_{N'}, T_{N'}, F_{N'}, l_{N'} \rangle$;
2. $\forall p \in P_N \;\; M_N(p) = M_{N'}(\beta(p))$.

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



Example of marked net

Let $M \in \mathcal{M}(P_N)$ be a marking of a net N. A transition $t \in T_N$ is *fireable* in M, if ${}^{\bullet}t \subseteq M$. If t is fireable in M, its firing yields a new marking $\widetilde{M} = M - {}^{\bullet}t + t^{\bullet}$, $M \xrightarrow{t} \widetilde{M}$.

A marking M of a net N is *reachable*, if $M = M_N$ or there exists a reachable marking $\widehat{M}$ of N s.t. $\widehat{M} \xrightarrow{t} M$ for some $t \in T_N$. $Mark(N)$ is a *set of all reachable* markings of a net N.

A net N is *n-bounded* ($n \in I\!N$), if $\forall M \in Mark(N)$ $\forall p \in P_N$ $M(p) \leq n$. A net N is *bounded*, if $\exists n \in I\!N$ s.t. N is n-bounded. A net N is *safe*, if it is 1-bounded.

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

An action $a \in Act$ is *self-concurrent* in N, if $\exists M \in Mark(N)$ $\exists t \in T_N$ s.t $l_N(t) = a$ and ${}^\bullet t + {}^\bullet t \subseteq M$. A net N is *self-concurrency free*, if no action is self-concurrent in N.

Partially ordered sets [Pra86]

Definition 4 *A partially ordered set (poset) is a pair $\rho = \langle X, \prec \rangle$:*

- $X = \{x, y, \ldots\}$ *is an underlying set;*
- $\prec \subseteq X \times X$ *is a strict partial order (irreflexive transitive relation) over X.*

Let $\rho = \langle X, \prec \rangle$ be a poset. A *restriction* of ρ to the set $Y \subseteq X$ is $\rho|_Y = \langle Y, \prec \cap (Y \times Y) \rangle$. A set of *strict predecessors* of $x \in X$ is $\downarrow x = \{y \in X \mid y \prec x\}$. A set $Y \subseteq X$ is *left-closed*, if $\forall y \in Y \;\; \downarrow y \subseteq Y$.

Let $\rho_1 = \langle X_1, \prec_1 \rangle$ and $\rho_2 = \langle X_2, \prec_2 \rangle$ be posets. ρ_1 is a *strict prefix* of ρ_2, $\rho_1 \lhd \rho_2$, if $\rho_1 = \rho_2|_Y$ s.t. $Y \subset X$ is a finite left-closed set. ρ_1 is a *prefix* of ρ_2, notation $\rho_1 \unlhd \rho_2$, if $\rho_1 \lhd \rho_2$ or $\rho_1 = \rho_2$.

Definition 5 *A labeled partially ordered set (lposet, causal structure) is a triple $\rho = \langle X, \prec, l \rangle$:*

- $\langle X, \prec \rangle$ *is a poset;*
- $l : X \to Act_\tau$ *is a labeling function.*

The notions defined for posets are transferred to lposets.

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', $\beta : \rho \asymp \rho'$, if:

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

We write $\rho \asymp \rho'$, if $\exists \beta : \rho \asymp \rho'$.

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

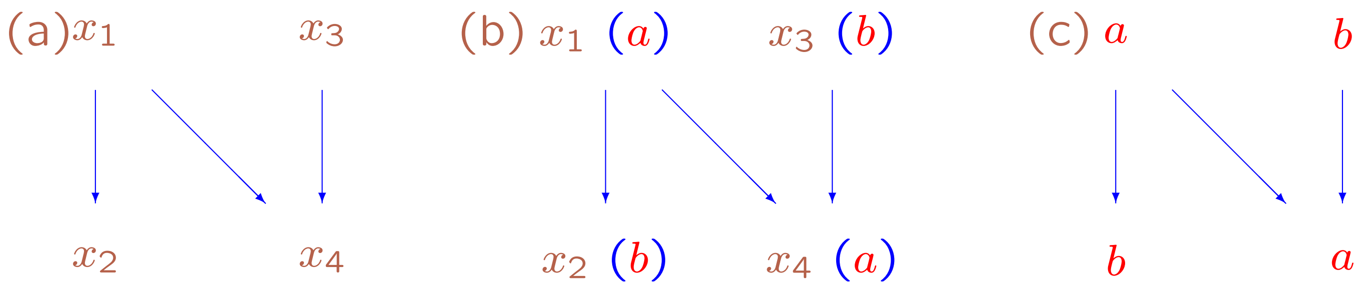
1. $\beta : \rho \asymp \rho'$;

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

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

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

Definition **6** Partially ordered multiset (pomset) *is an isomorphism class of lposets.*



Examples of poset, lposet and pomset

Event structures [NPW81]

Definition **7** *An* event structure (ES) *is a triple* $\xi = \langle X, \prec, \# \rangle$*:*

- $X = \{x, y, \ldots\}$ *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*:* $\forall x, y, z \in X \ x\#y \prec z \ \Rightarrow \ x\#z$*.*

Let $\xi = \langle X, \prec, \# \rangle$ be LES and $Y \subseteq X$. A *restriction* of ξ to the set Y is: $\xi|_Y = \langle Y, \prec \cap (Y \times Y), \# \cap (Y \times Y) \rangle$.

Definition **8** *A* labeled event structure (LES) *is a quadruple* $\xi = \langle X, \prec, \#, l \rangle$*:*

- $\langle X, \prec, \# \rangle$ *is an event structure;*
- $l : X \to Act_\tau$ *is a* labeling *function.*

The notions defined for ES's are transferred to LES's.

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

1. β is a bijection;

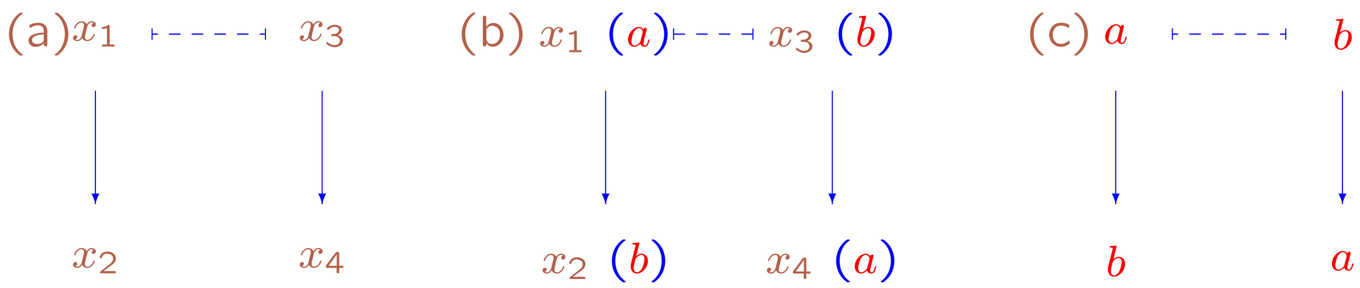
2. $\forall x \in X \ \ l(x) = l'(\beta(x))$;

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's ξ and ξ' are *isomorphic*, $\xi \simeq \xi'$, if $\exists \beta : \xi \simeq \xi'$.

Definition 9 *A* multi-event structure (MES) *is an isomorphism class of LES's.*



Examples of ES, LES and MES

Processes [BD87]

Definition 10 *A* causal net *is an acyclic ordinary labeled 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. $\forall x \in P_C \cap T_C \; |\downarrow_C x| < \infty$*, i.e. a set of causes is finite.*

Based on causal net $C = \langle P_C, T_C, F_C, l_C \rangle$, one can define lposet $\rho_C = \langle T_C, \prec_N \cap (T_C \times T_C), l_C \rangle$.

For any causal net C there is a sequence of transition firings: ${}^\circ C = L_0 \xrightarrow{v_1} \cdots \xrightarrow{v_n} L_n = C^\circ$ s.t. $L_i \subseteq P_C$ $(0 \leq i \leq n)$, $P_C = \cup_{i=0}^{n} L_i$ and $T_C = \{v_1, \ldots, v_n\}$. It is called a *full execution* of C.

Definition 11 *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* homomorphism *of* C *into* N*,* $\varphi : C \to N$*, 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. $\forall v \in T_C \; {}^\bullet\varphi(v) = \varphi({}^\bullet v)$ *and* $\varphi(v)^\bullet = \varphi(v^\bullet)$*, i.e. flow relation is respected;*

3. $\forall v \in T_C \; l_C(v) = l_N(\varphi(v))$*, i.e. labeling is preserved.*

Since homomorphisms respect the flow relation, if ${}^\circ C \xrightarrow{v_1} \cdots \xrightarrow{v_n} C^\circ$ is a full execution of C, then $M = \varphi({}^\circ C) \xrightarrow{\varphi(v_1)} \cdots \xrightarrow{\varphi(v_n)} \varphi(C^\circ) = \widetilde{M}$ is a sequence of transition firings in N.

Definition 12 *A fireable in marking M process of a net N is a pair $\pi = (C, \varphi)$, where C is a causal net and $\varphi : C \to N$ is an homomorphism s.t. $M = \varphi({}^\circ C)$. A fireable in M_N process is a process of N.*

$\Pi(N, M)$ is a *set of all fireable* in marking M, and $\Pi(N)$ is the *set of all* processes of a net N.

The *initial* process of a net N is $\pi_N = (C_N, \varphi_N) \in \Pi(N)$, s.t. $T_{C_N} = \emptyset$.

If $\pi \in \Pi(N, M)$, then firing of this process transforms a marking M into $\widetilde{M} = M - \varphi({}^\circ C) + \varphi(C^\circ) = \varphi(C^\circ)$, $M \xrightarrow{\pi} \widetilde{M}$.

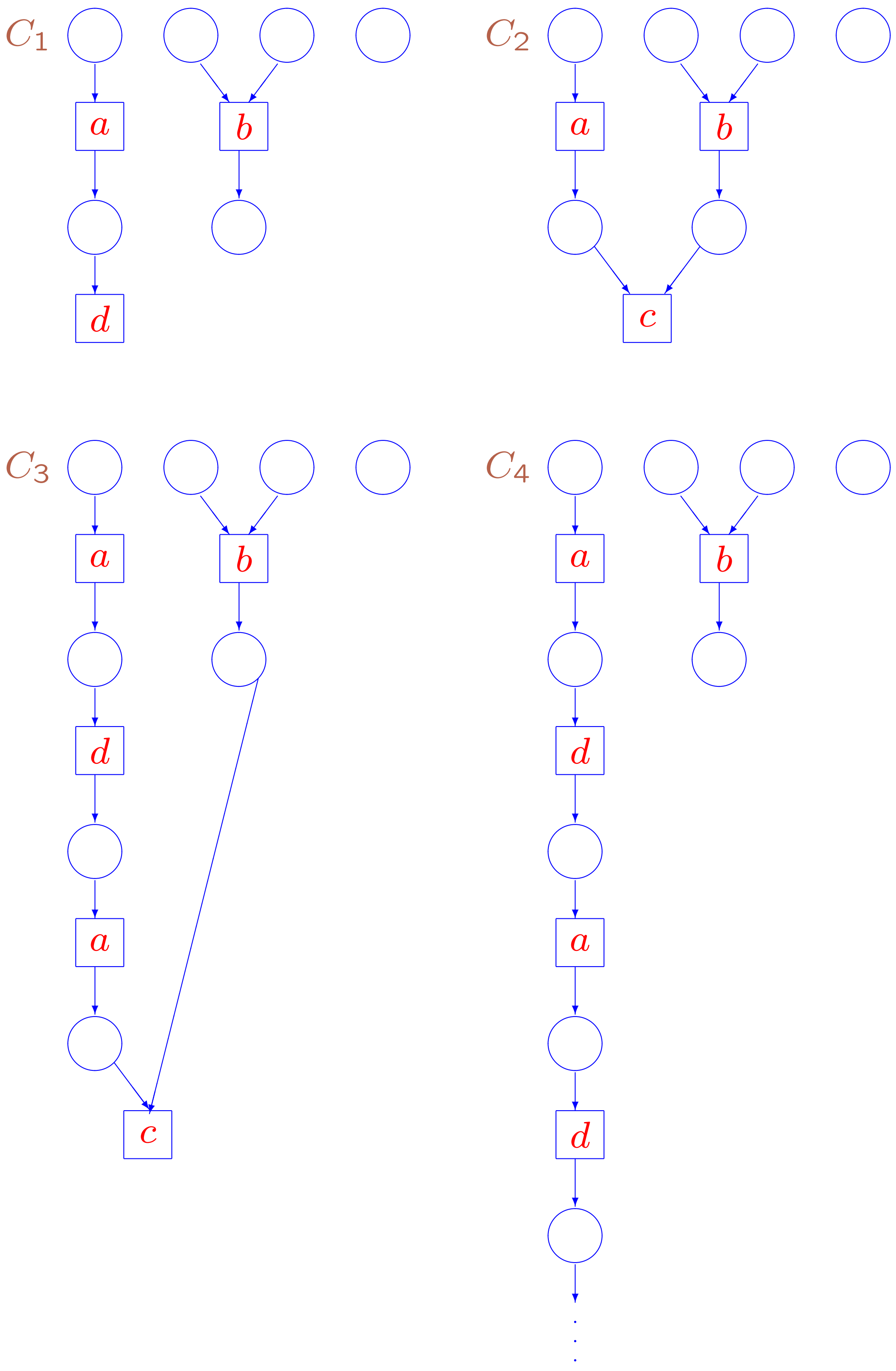
Let $\pi = (C, \varphi)$, $\tilde{\pi} = (\widetilde{C}, \tilde{\varphi}) \in \Pi(N)$, $\hat{\pi} = (\widehat{C}, \hat{\varphi}) \in \Pi(N, \varphi(C^\circ))$. A process π is a *prefix* of a process $\tilde{\pi}$, if $T_C \subseteq T_{\widetilde{C}}$ is a left-closed set in $\widetilde{C}$. A process $\hat{\pi}$ is a *suffix* of a process $\tilde{\pi}$, if $T_{\widehat{C}} = T_{\widetilde{C}} \setminus T_C$.

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

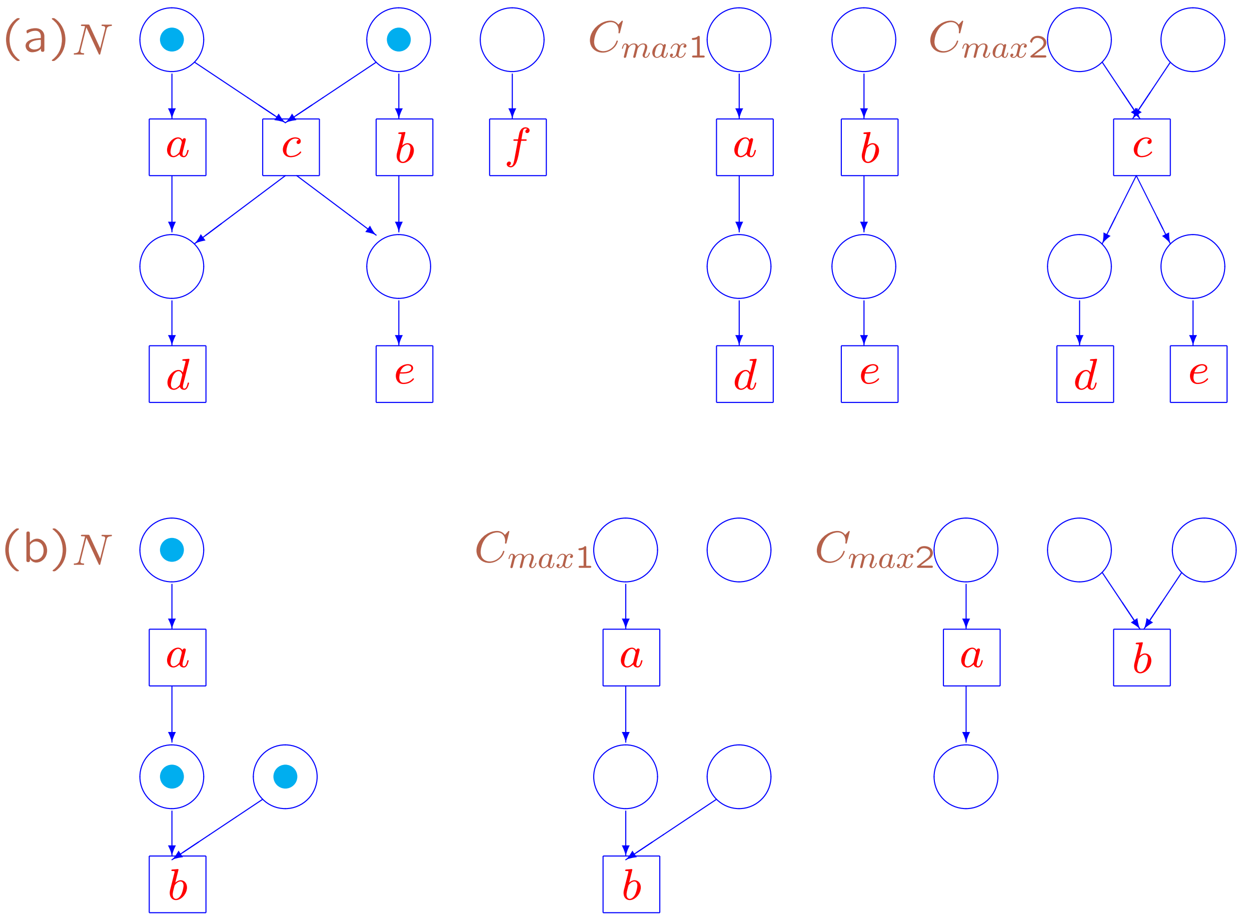
A process $\tilde{\pi}$ is an extension of a process π *by one transition*, $\pi \xrightarrow{v} \tilde{\pi}$ or $\pi \xrightarrow{a} \tilde{\pi}$, if $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$, $T_{\widehat{C}} = \{v\}$ and $l_{\widehat{C}}(v) = a$.

A process $\tilde{\pi}$ is an extension of a process π *by sequence of transitions*, $\pi \xrightarrow{\sigma} \tilde{\pi}$ or $\pi \xrightarrow{\omega} \tilde{\pi}$, if $\exists \pi_i \in \Pi(N)$ $(1 \le i \le n)$ $\pi \xrightarrow{v_1} \pi_1 \xrightarrow{v_2} \ldots \xrightarrow{v_n} \pi_n = \tilde{\pi}$, $\sigma = v_1 \cdots v_n$ and $l_{\widehat{C}}(\sigma) = \omega$.

A process $\tilde{\pi}$ is an extension of a process π *by multiset of transitions*, $\pi \xrightarrow{V} \tilde{\pi}$ or $\pi \xrightarrow{A} \tilde{\pi}$, if $\pi \xrightarrow{\hat{\pi}} \tilde{\pi}$, $\prec_{\widehat{C}} = \emptyset$, $T_{\widehat{C}} = V$ and $l_{\widehat{C}}(V) = A$.



Causal nets of processes



Causal nets of maximal processes

Branching processes [Eng91]

Definition 13 *An occurrence net is an acyclic ordinary labeled net* $O = \langle P_O, T_O, F_O, l_O \rangle$*, s.t.:*

1. $\forall r \in P_O\ |{}^\bullet r| \le 1$, *i.e. there are no backwards conflicts;*

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 $O = \langle P_O, T_O, F_O, l_O \rangle$ 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 *homomorphism* O into N, $\psi : O \to N$, 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. labeling 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.

Based on occurrence net $O = \langle P_O, T_O, F_O, l_O \rangle$, one can define LES $\xi_O = \langle T_O, \prec_O \cap (T_O \times T_O), \#_O \cap (T_O \times T_O), l_O \rangle$.

Definition 14 *A branching process of a net* N *is a pair* $\varpi = (O, \psi)$*, where* O *is an occurrence net and* $\psi : O \to N$ *is an homomorphism s.t.* $M_N = \psi({}^\circ O)$*.*

$\wp(N)$ is the set of *all branching processes* of a net N. The *initial* branching process of a net N coincides with its initial process, i.e. $\varpi_N = \pi_N$.

Let $\varpi = (O, \psi)$, $\tilde{\varpi} = (\widetilde{O}, \tilde{\psi}) \in \wp(N)$, $O = \langle P_O, T_O, F_O, l_O \rangle$, $\widetilde{O} = \langle P_{\widetilde{O}}, T_{\widetilde{O}}, F_{\widetilde{O}}, l_{\widetilde{O}} \rangle$. ϖ is a *prefix* of $\tilde{\varpi}$, if $T_O \subseteq T_{\widetilde{O}}$ is a left-closed set in $\widetilde{O}$.

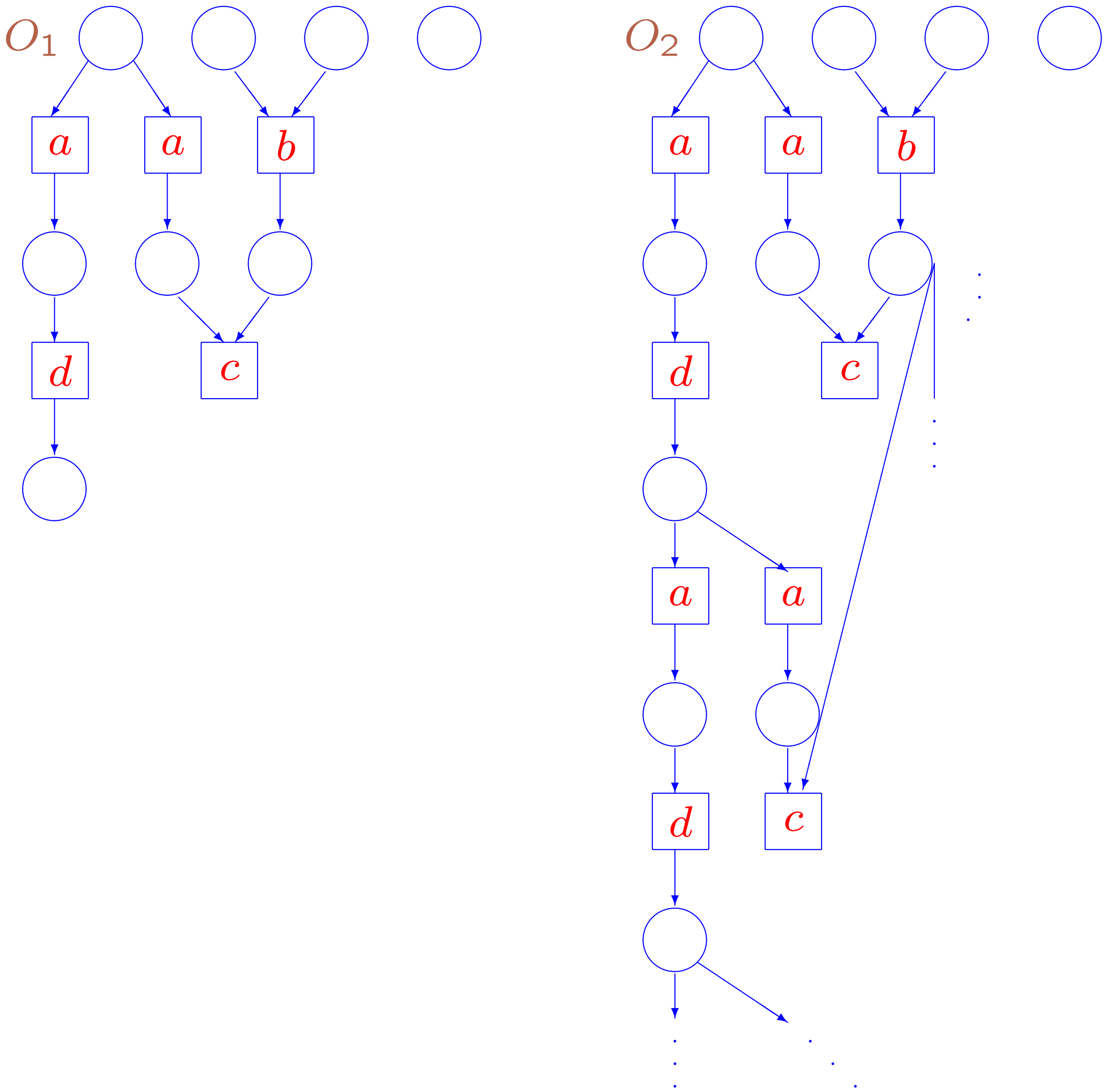
Then $\tilde{\varpi}$ is an *extension* of ϖ, and $\hat{\varpi}$ is an *extending* branching process for ϖ, $\varpi \rightarrow \tilde{\varpi}$.

A branching process $\varpi = (O, \psi)$ of a net N is *maximal*, if it cannot be extended, i.e. $\forall \tilde{\varpi} = (\widetilde{O}, \tilde{\psi})$ s.t. $\varpi \rightarrow \tilde{\varpi}$: $T_{\widetilde{O}} \setminus T_O = \emptyset$.

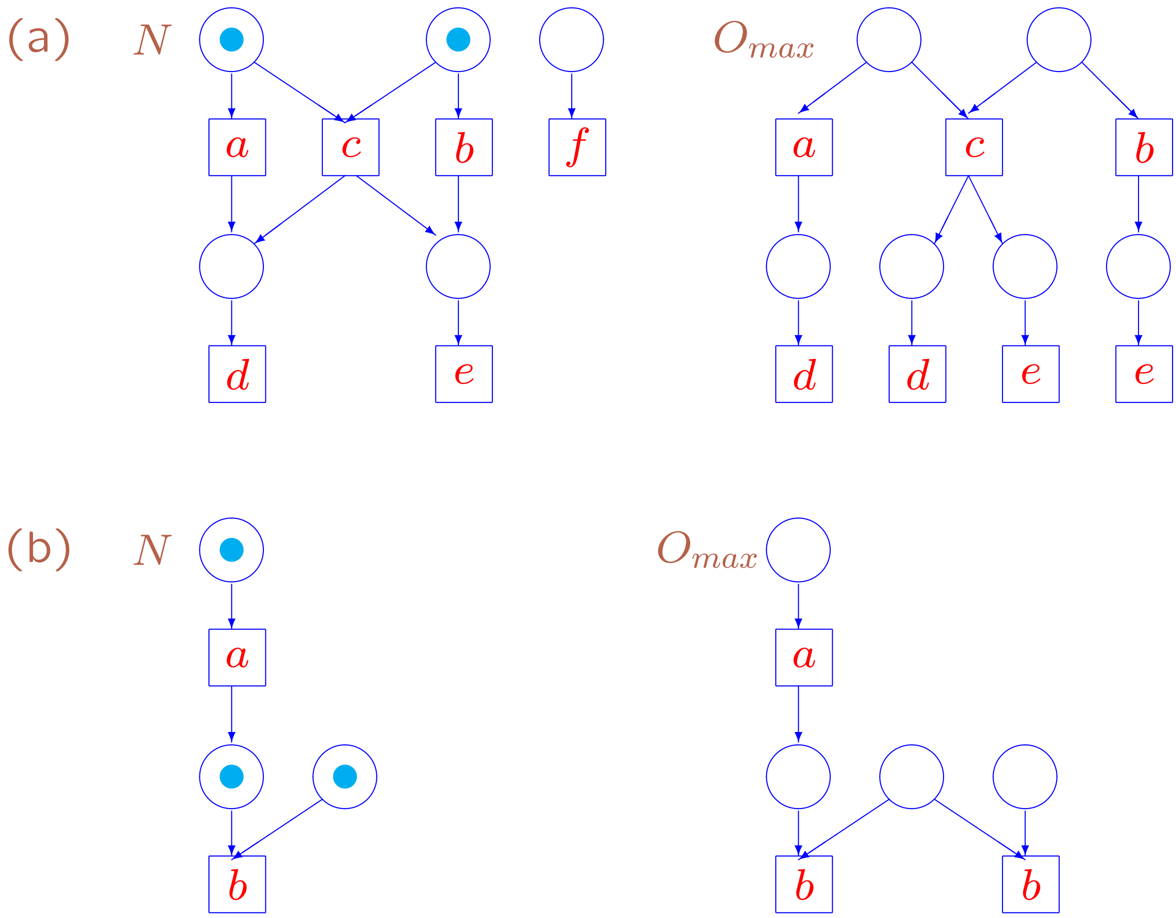
The set of all maximal branching processes of a net N consists of the unique (up to isomorphism) branching process $\varpi_{max} = (O_{max}, \psi_{max})$.

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 MES $\mathcal{E}(N) = \xi_{\mathcal{U}(N)}$ which is an isomorphism class of LES ξ_O for $O \in \mathcal{U}(N)$.



Occurrence nets of branching processes



Occurrence nets of maximal branching processes

Trace equivalences

Definition 15 *An* interleaving trace *of a net* N *is a sequence* $a_1 \cdots a_n \in Act^*$ *s.t.* $\pi_N \xrightarrow{a_1} \pi_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} \pi_n$, $\pi_i \in \Pi(N)$ $(1 \leq i \leq n)$.

The set of all interleaving traces *of* N *is* $IntTraces(N)$.

N *and* N' *are* interleaving trace equivalent, $N \equiv_i N'$, *if* $IntTraces(N) = IntTraces(N')$.

Definition 16 *A* step trace *of a net* N *is a sequence* $A_1 \cdots A_n \in (\mathcal{M}(Act))^*$ *s.t.* $\pi_N \xrightarrow{A_1} \pi_1 \xrightarrow{A_2} \ldots \xrightarrow{A_n} \pi_n$, $\pi_i \in \Pi(N)$ $(1 \leq i \leq n)$.

The set of all step traces *of* N *is* $StepTraces(N)$.

N *and* N' *are* step trace equivalent, $N \equiv_s N'$, *if* $StepTraces(N) = StepTraces(N')$.

Definition 17 *A* pomset trace *of a net* N *is a pomset* ρ, *an isomorphism class of lposet* ρ_C *for* $\pi = (C, \varphi) \in \Pi(N)$.

The set of all pomset traces *of* N *is* $Pomsets(N)$.

N *and* N' *are* partial word trace equivalent, $N \equiv_{pw} N'$, *if* $Pomsets(N) \sqsubseteq Pomsets(N')$ *and* $Pomsets(N') \sqsubseteq Pomsets(N)$.

Definition 18 N *and* N' *are* pomset trace equivalent, $N \equiv_{pom} N'$, *if* $Pomsets(N) = Pomsets(N')$.

Definition 19 *A* process trace *of a net* N *is an isomorphism class of causal net* C *for* $\pi = (C, \varphi) \in \Pi(N)$.

The set of all process traces *of* N *is* $ProcessNets(N)$.

N *and* N' *are* process trace equivalent, $N \equiv_{pr} N'$, *if* $ProcessNets(N) = ProcessNets(N')$.

	1	2	3
$System$ $States$	s_{in}	$s \xrightarrow{Activity} \tilde{s}$	$s \xrightarrow{Sim(Activity)} \tilde{s}$
$\mathcal{R} \subseteq States \times States'$	$\mathcal{R}$	$\mathcal{R} \qquad\qquad \mathcal{R}$	$\mathcal{R} \qquad\qquad \mathcal{R}$
$System'$ $States'$	s'_{in}	$s' \xrightarrow{Sim(Activity)} \tilde{s}'$	$s' \xrightarrow{Activity} \tilde{s}'$

Bisimulation equivalence

Usual bisimulation equivalences

Definition 20 $\mathcal{R} \subseteq \Pi(N) \times \Pi(N')$ *is a* $\star$-bisimulation *between nets* N *and* N', $\star \in$\{interleaving, step, partial word, pomset, process\}, $\mathcal{R} : N \underline{\leftrightarrow}_\star N'$, $\star \in \{i, s, pw, pom, pr\}$, *if:*

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

2. $(\pi, \pi') \in \mathcal{R}$, $\pi \xrightarrow{\widehat{\pi}} \tilde{\pi}$,

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

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

   $\Rightarrow$ $\exists \tilde{\pi}' : \ \pi' \xrightarrow{\widehat{\pi}'} \tilde{\pi}'$, $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$ *and*

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

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

   (c) $\widehat{C} \simeq \widehat{C}'$, if $\star = pr$.

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

N *and* N' *are* $\star$-bisimulation equivalent, $\star \in$\{interleaving, step, partial word, pomset, process\}, $N \underline{\leftrightarrow}_\star N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}_\star N'$, $\star \in \{i, s, pw, pom, pr\}$.

ST-bisimulation equivalences

Definition 21 *[Vog92] An ST-marking of a net N is a pair (M, U):*

- *$M \in \mathcal{M}(P_N)$ is the current marking;*
- *$U \in \mathcal{M}(T_N)$ are the working transitions.*

$(M_N, \emptyset)$ is the *initial ST-marking* of a net N.

$T_N^{\pm} = \{t^+, t^- \mid t \in T_N\}$ is a set of *transition parts*.

t^+ is the *beginning*, and t^- is the *end* of t.

A transition part $q \in T_N^{\pm}$ is *enabled* in ST-marking $Q = (M, U)$, $Q \xrightarrow{q}$, if:

1. $M \xrightarrow{t}$, if $q = t^+$ or
2. $t \in U$, if $q = t^-$.

If q is enabled in M, its occurrence transforms ST-marking Q into $\widetilde{Q}$, $Q \xrightarrow{q} \widetilde{Q}$, as:

1. $\widetilde{M} = M - {}^\bullet t$ and $\widetilde{U} = U + t$, if $q = t^+$ or
2. $\widetilde{M} = M + t^\bullet$ and $\widetilde{U} = U - t$, if $q = t^-$.

We write $Q \rightarrow \widetilde{Q}$, if $Q \xrightarrow{q} \widetilde{Q}$ for some q.

An ST-marking $\widetilde{Q}$ of N is *reachable from* Q, if:

1. $\widetilde{Q} = Q$ or
2. there is a reachable from Q ST-marking $\widehat{Q}$ s.t. $\widehat{Q} \rightarrow \widetilde{Q}$.

An ST-marking Q of N is *reachable*, if it is reachable from M_N.

$ST-Mark(N)$ is the set of *all reachable* ST-markings of N.

Definition 22 *An ST-process of a net N is a pair (π_E, π_P):*

1. $\pi_E, \pi_P \in \Pi(N)$, $\pi_P \xrightarrow{\pi_W} \pi_E$;
2. $\forall v, w \in T_{C_E}\ v \prec_{C_E} w \ \Rightarrow\ v \in T_{C_P}$.

- π_E is the current process;
- π_P is the completed part;
- π_W is the still working part.

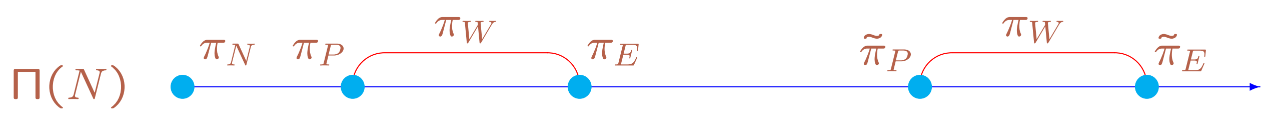
Obviously, $\prec_{C_W} = \emptyset$.

$ST-\Pi(N)$ is the set of *all ST-processes* of a net N.

(π_N, π_N) is the *initial ST-process* of a net N.

Let (π_E, π_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 $\pi_P \rightarrow \tilde{\pi}_P$.



ST-processes

Definition **23** $\mathcal{R} \subseteq ST-\Pi(N) \times ST-\Pi(N') \times \mathcal{B}$, *where* $\mathcal{B} = \{\beta \mid \beta : T_C \to T_{C'},\ \pi = (C, \varphi) \in \Pi(N),$ $\pi' = (C', \varphi') \in \Pi(N')\}$, *is a* $\star$-ST-bisimulation *between nets* N *and* N', $\star \in$ {interleaving, partial word, pomset, process}, $\mathcal{R} : N \underline{\leftrightarrow}_{\star ST} N'$, $\star \in \{i, pw, pom, pr\}$, *if:*

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

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

3. $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R},\ (\pi_E, \pi_P) \to (\tilde{\pi}_E, \tilde{\pi}_P) \ \Rightarrow$ $\exists \tilde{\beta},\ (\tilde{\pi}'_E, \tilde{\pi}'_P) :\ (\pi'_E, \pi'_P) \to (\tilde{\pi}'_E, \tilde{\pi}'_P),\ \tilde{\beta}|_{T_{C_E}} = \beta,$ $((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$, *and if* $\pi_P \xrightarrow{\pi} \tilde{\pi}_E$, $\pi'_P \xrightarrow{\pi'} \tilde{\pi}'_E,\ \gamma = \tilde{\beta}|_{T_C}$, *then:*

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

   (b) $\gamma : \rho_C \simeq \rho_{C'}$, *if* $\star = pom$;

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

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

N *and* N' *are* $\star$-ST-bisimulation equivalent, $\star \in$ {interleaving, partial word, pomset, process}, $N \underline{\leftrightarrow}_{\star ST} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}_{\star ST} N'$, $\star \in \{i, pw, pom, pr\}$.

History preserving bisimulation equivalences

Definition 24 $\mathcal{R} \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$, *where* $\mathcal{B} = \{\beta \mid \beta : T_C \to T_{C'},\ \pi = (C, \varphi) \in \Pi(N),$ $\pi' = (C', \varphi') \in \Pi(N')\}$, *is a* $\star$-history preserving bisimulation *between nets* N *and* N', $\star \in$\{pomset, process\}, $N \underline{\leftrightarrow}_{\star h} N'$, $\star \in \{pom, pr\}$, *if:*

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

2. $(\pi, \pi', \beta) \in \mathcal{R} \ \Rightarrow$

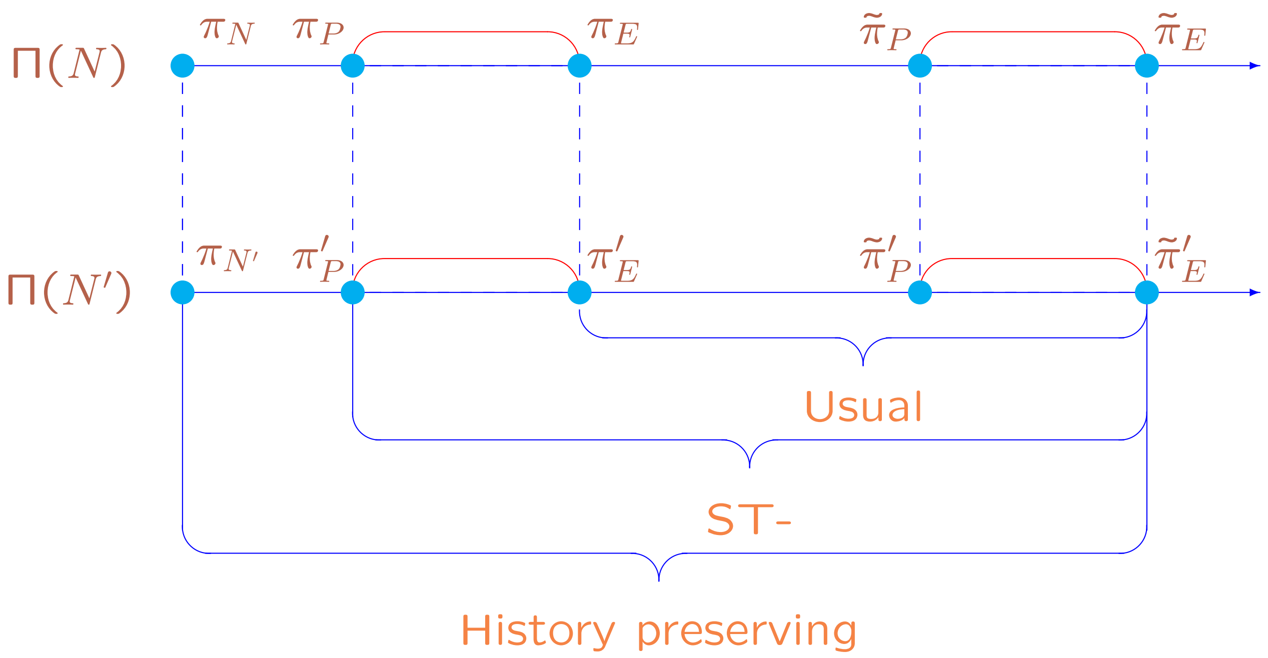
   (a) $\tilde{\beta} : \rho_{\tilde{C}} \simeq \rho_{\tilde{C}'}$, *if* $\star \in \{pom, pr\}$;

   (b) $\tilde{C} \simeq \tilde{C}'$, *if* $\star = pr$.

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

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

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



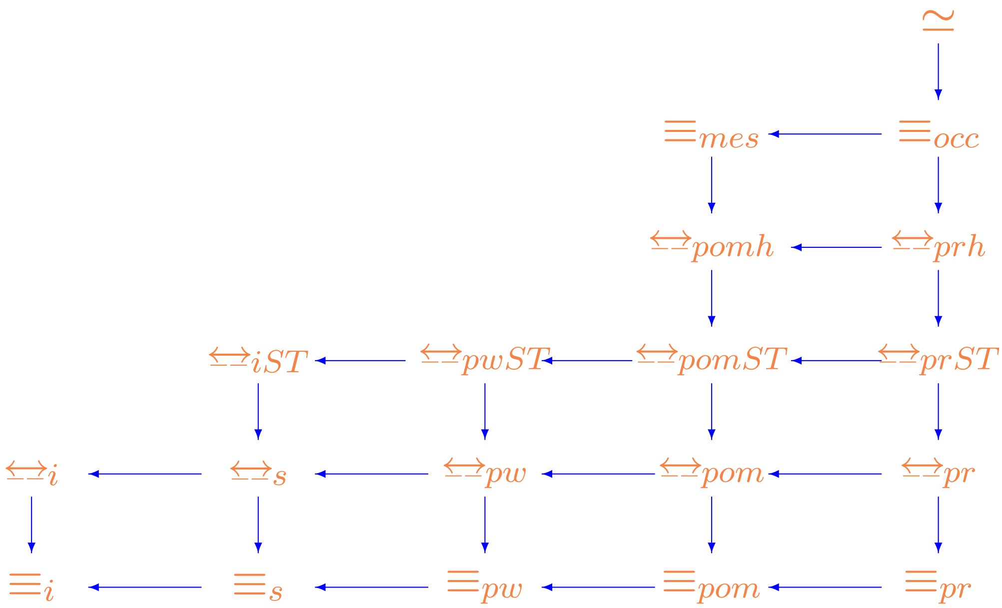
A distinguish ability of the bisimulation equivalences

Conflict preserving equivalences

Definition 25 N *and* N' *are* MES conflict preserving equivalent, $N \equiv_{mes} N'$, *if* $\mathcal{E}(N) = \mathcal{E}(N')$.

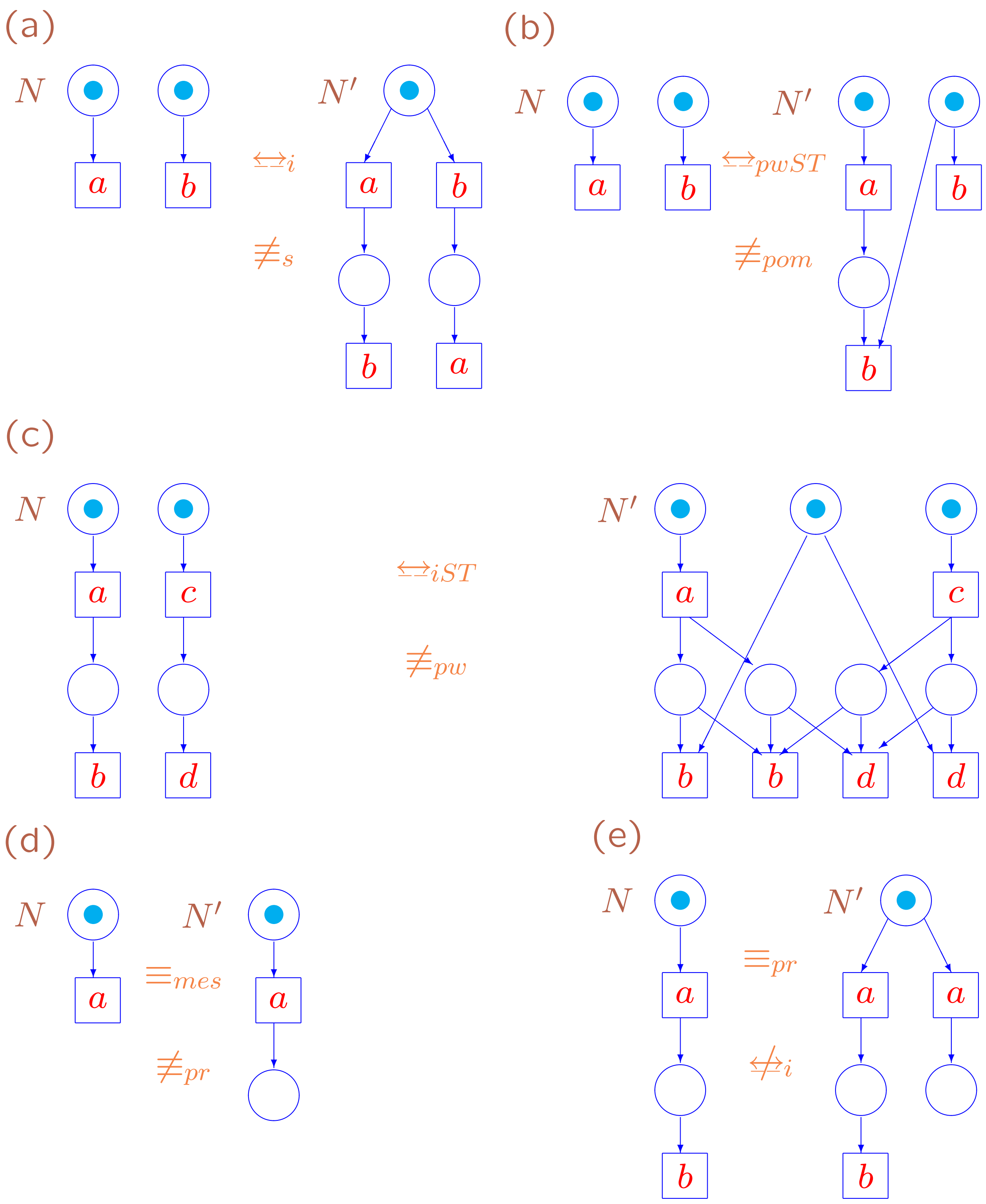
Definition 26 N *and* N' *are* occurrence conflict preserving equivalent, $N \equiv_{occ} N'$, *if* $\mathcal{U}(N) = \mathcal{U}(N')$.

Comparing basic equivalences

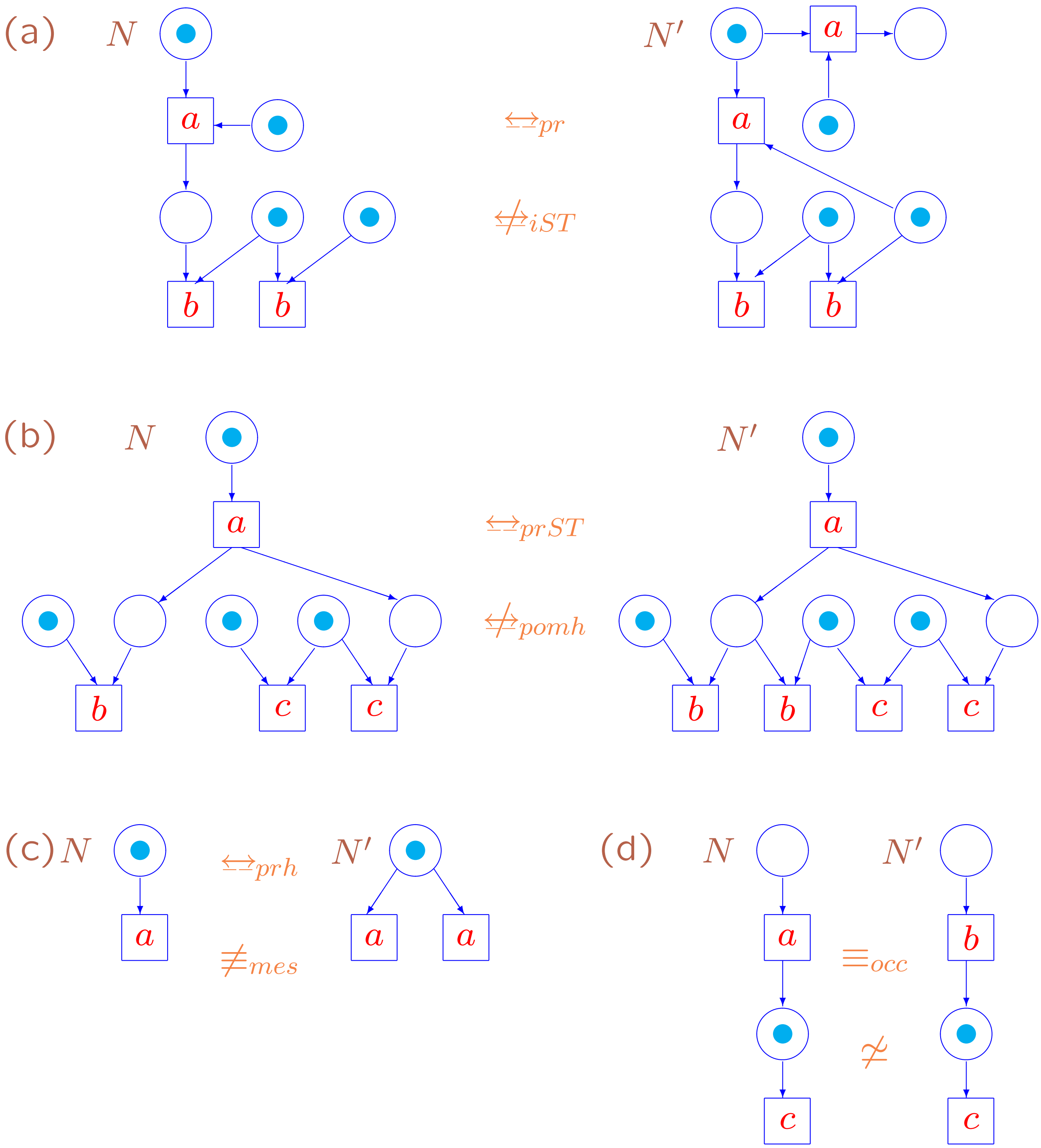


Interrelations of basic equivalences

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



B: Examples of basic equivalences



B1: Examples of basic equivalences (continued)

- In Figure B(a), $N \underline{\leftrightarrow}_i N'$, but $N \not\equiv_s N'$, since only in the net N' actions a and b can happen concurrently.

- In Figure B(c), $N \underline{\leftrightarrow}_{iST} N'$, but $N \not\equiv_{pw} N'$, since for the pomset corresponding to the net N there is no even less sequential pomset in N'.

- In Figure B(b), $N \underline{\leftrightarrow}_{pwST} N'$, but $N \not\equiv_{pom} N'$, since only in the net N' an action b can depend on action a.

- In Figure B(d), $N \equiv_{mes} N'$, but $N \not\equiv_{pr} N'$, since N' is a causal net which is not isomorphic to N (because of additional output place).

- In Figure B(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 cannot happen afterwards.

- In Figure B1(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.

- In Figure B1(b), $N \underline{\leftrightarrow}_{prST} N'$, but $N \not\underline{\leftrightarrow}_{pomh} N'$, since only in net N' actions a and b can happen so that action c must depend on a.

- In Figure B1(c), $N \underline{\leftrightarrow}_{prh} N'$, but $N \not\equiv_{mes} N'$, since only net N' has corresponding MES with two conflict actions a.

- In Figure B1(d), $N \equiv_{occ} N'$, but $N \not\simeq N'$, since upper transitions of nets N and N' are labeled by different actions (a and b).

Sequential runs [Che92a,Tar97]

Definition 27 *A* sequential run *of a net N is a pair (π, σ):*

- *a process $\pi \in \Pi(N)$:*

  causal dependencies of transitions;

- *a sequence $\sigma \in T_C^*$ s.t. $\pi_N \xrightarrow{\sigma} \pi$:*

  occurrence order of transitions.

The *set of all sequential runs* of a net N is $Runs(N)$.

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

Let $(\pi, \sigma),\ (\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$.

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

We write $(\pi, \sigma) \rightarrow (\tilde{\pi}, \tilde{\sigma})$, if $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$ for some $\hat{\pi}$.

$|\sigma|$ is the *length* of a sequence σ.

Let $(\pi, \sigma) \in Runs(N)$, $(\pi', \sigma') \in Runs(N')$ and $\sigma = v_1 \cdots v_n$, $\sigma' = v'_1 \cdots v'_n$.

We define a mapping $\beta_\sigma^{\sigma'} : T_C \to T_{C'}$:

- $\beta_\varepsilon^\varepsilon = \emptyset$;
- $\beta_\sigma^{\sigma'} = \{(v_i, v'_i) \mid 1 \leq i \leq n\}$.

Let $(\pi, \sigma) \in Runs(N)$ and $\sigma = v_1 \cdots v_n$, $\pi_N \xrightarrow{v_1} \ldots \xrightarrow{v_i} \pi_i$ $(1 \leq i \leq n)$. Then:

- $\pi(0) = \pi_N$,
  $\pi(i) = \pi_i$ $(1 \leq i \leq n)$;
- $\sigma(0) = \varepsilon$,
  $\sigma(i) = v_1 \cdots v_i$ $(1 \leq i \leq n)$.

Back-forth bisimulation equivalences

Definition 28 $\mathcal{R} \subseteq Runs(N) \times Runs(N')$ is a $\star$-back $\star\star$-forth bisimulation between nets N and N', $\star, \star\star \in$\{interleaving, step, partial word, pomset,process\}, $\mathcal{R} : N \underline{\leftrightarrow}_{\star b \star\star f} N'$, $\star, \star\star \in \{i, s, pw, pom, pr\}$, if:

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

2. $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$

   - (back) $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\widehat{\pi}} (\pi, \sigma)$,

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

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

     $\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') : \; (\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\widehat{\pi}'} (\pi', \sigma'), \; ((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ and

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

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

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

   - (forth) $(\pi, \sigma) \xrightarrow{\widehat{\pi}} (\tilde{\pi}, \tilde{\sigma})$,

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

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

     $\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') : \; (\pi', \sigma') \xrightarrow{\widehat{\pi}'} (\tilde{\pi}', \tilde{\sigma}'), \; ((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ and

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

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

     (c) $\widehat{C} \simeq \widehat{C}'$, if $\star\star = pr$.

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

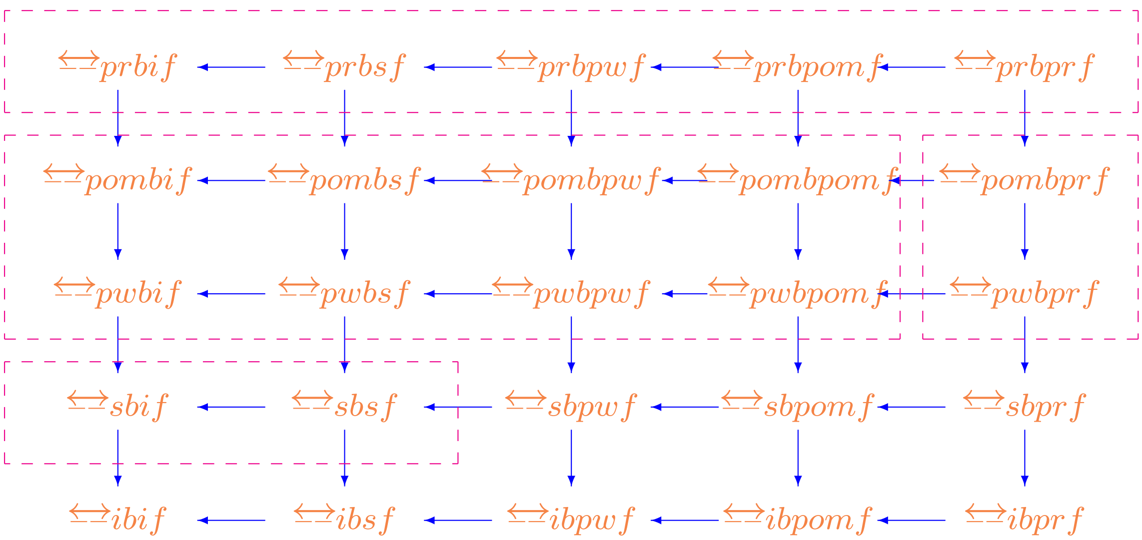
N and N' are $\star$-back $\star\star$-forth bisimulation equivalent, $\star, \star\star \in$\{interleaving, step, partial word, pomset, process\}, $N \underline{\leftrightarrow}_{\star b \star\star f} N'$, if $\exists \mathcal{R} : N \underline{\leftrightarrow}_{\star b \star\star f} N'$, $\star, \star\star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

Comparing back-forth bisimulation equivalences

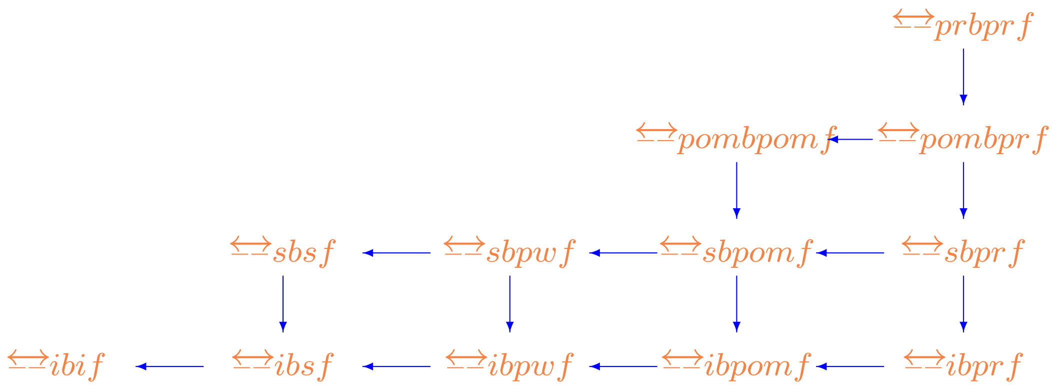
Proposition 1 [Pin93, Tar97] *Let* $\star \in \{i, s, pw, pom, pr\}$*. For nets* N *and* N'*:*

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

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



Merging of back-forth bisimulation equivalences



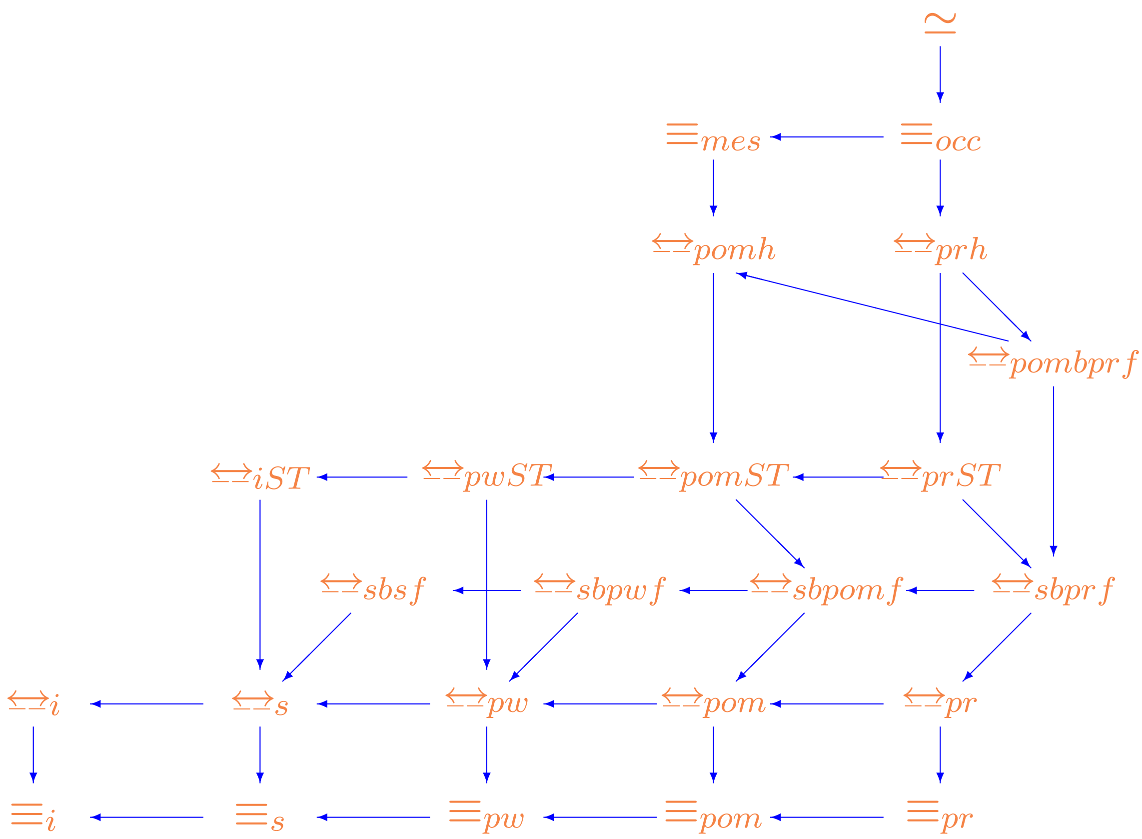
Interrelations of back-forth bisimulation equivalences

Comparing back-forth bisimulation equivalences with basic ones

Proposition 2 *[Pin93, Tar97] Let $\star \in \{i, s, pw, pom, pr\}$ and $\star\star \in \{pom, pr\}$. For nets N and N':*

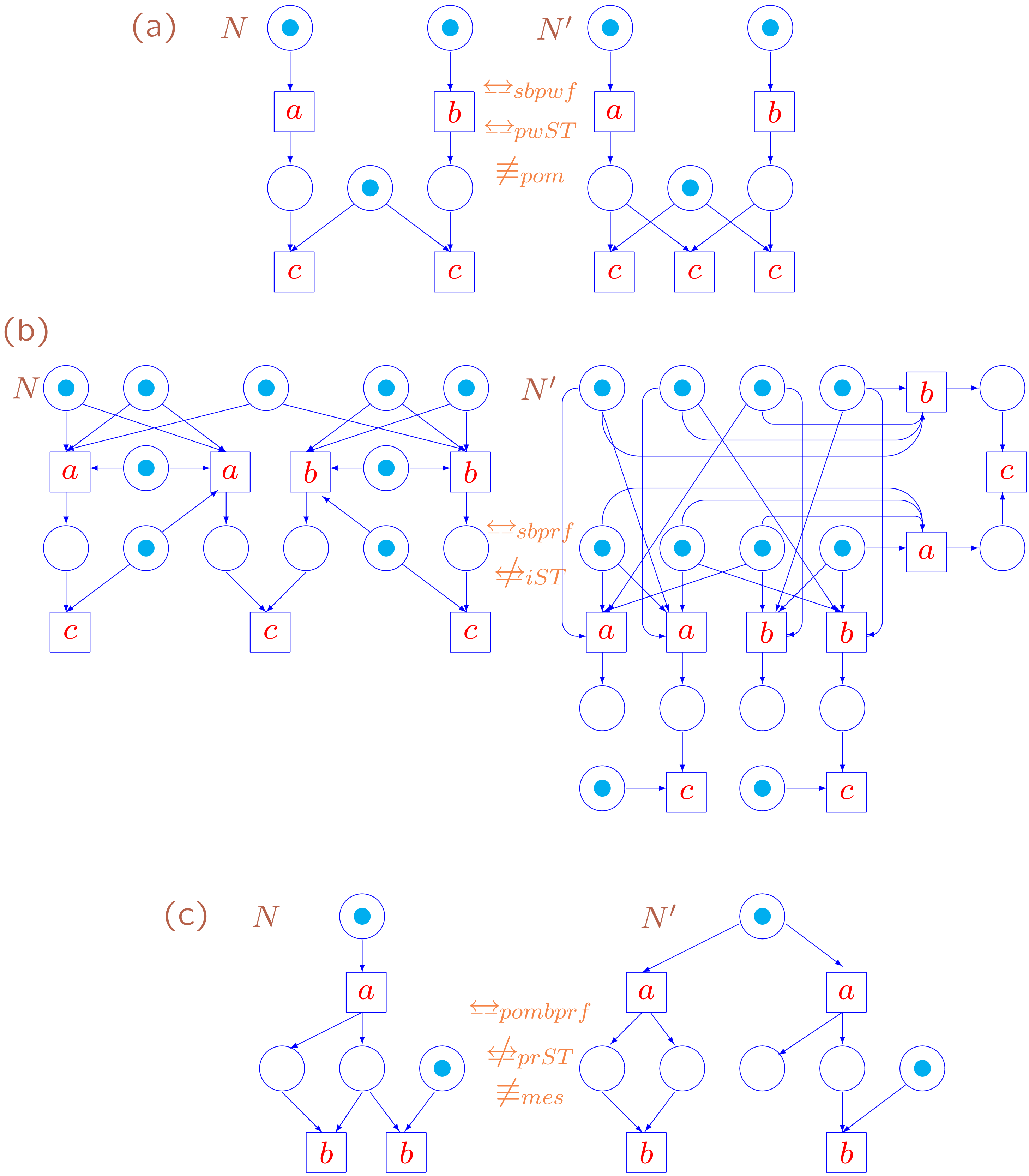
1. $N \underline{\leftrightarrow}_{ib\star f} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{\star} N'$;

2. $N \underline{\leftrightarrow}_{\star\star ST} N' \;\Rightarrow\; N \underline{\leftrightarrow}_{sb\star\star f} N'$.



Interrelations of back-forth bisimulation equivalences with basic ones

Theorem 2 *Let $\leftrightarrow, \leftrightarrow\!\!\!\!\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$ 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' $N \leftrightarrow_{\star} N' \Rightarrow N \leftrightarrow\!\!\!\!\leftrightarrow_{\star\star} N'$ iff in the graph above there exists a directed path from $\leftrightarrow_{\star}$ to $\leftrightarrow\!\!\!\!\leftrightarrow_{\star\star}$.*



BF: Examples of back-forth bisimulation equivalences

- In Figure B(c), $N \underline{\leftrightarrow}_{sbsf} N'$, but $N \not\equiv_{pw} N'$.

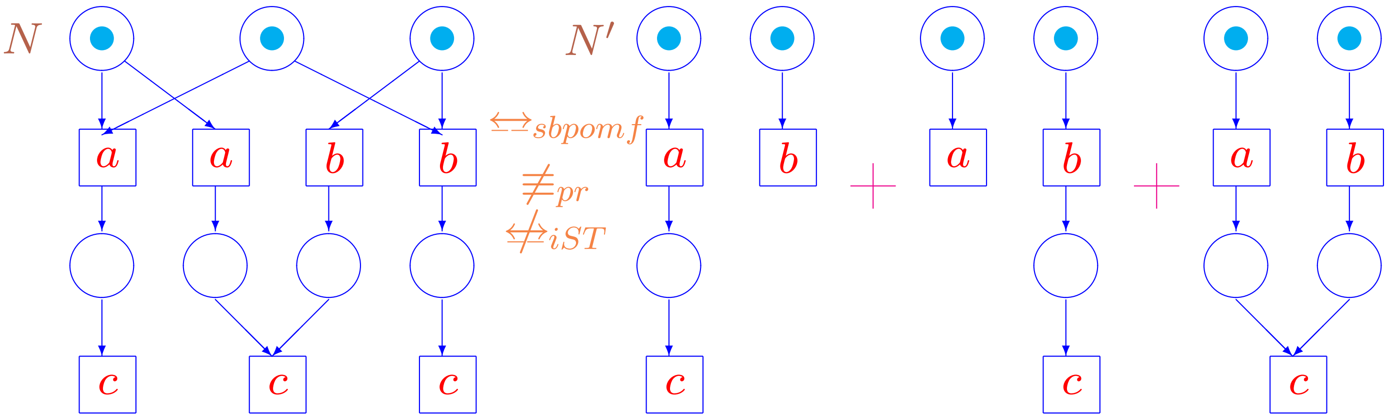
- In Figure BF(a), $N \underline{\leftrightarrow}_{sbpwf} N'$, but $N \not\equiv_{pom} N'$, since only in the net N' action c can depend on actions a and b.

- In Figure BF(b), $N \underline{\leftrightarrow}_{sbprf} N'$, but $N \not\underline{\leftrightarrow}_{iST} N'$, since only in the net N' action a can start so that:
  1. until finishing of a the sequence of actions bc cannot happen, and
  2. immediately after finishing of a action c cannot happen.

- In Figure BF(c), $N \underline{\leftrightarrow}_{pombprf} N'$, but $N \not\underline{\leftrightarrow}_{prST} N'$, since only in the net N' the process with action a can start so that it can be extended by process with action b in the only way (i.e. so that extended process be unique).

- In Figure B(b), $N \underline{\leftrightarrow}_{pwST} N'$, but $N \not\underline{\leftrightarrow}_{sbsf} N'$, since only in the net N' the sequence of actions ab can happen so that b must depend on a.

- In Figure B1(a), $N \underline{\leftrightarrow}_{pr} N'$, but $N \not\underline{\leftrightarrow}_{sbsf} N'$, since only in the net N' action a can happen so that action b must depend on a.



More clear, but weaker example of back-forth bisimulation equivalences

Logic HML [HM85]

Definition 29 $\top$ *denotes the truth,* $a \in Act$.

A formula *of* HML:

$$\Phi ::= \top \mid \neg\Phi \mid \Phi\wedge\Psi \mid \langle a\rangle\Phi$$

HML is the set of *all formulas* of HML.

Definition 30 *Let* N *be a net and* $\pi \in \Pi(N)$. *The* satisfaction relation $\models_N \in \Pi(N) \times \mathbf{HML}$:

1. $\pi \models_N \top$ — *always;*
2. $\pi \models_N \neg\Phi$, *if* $\pi \not\models_N \Phi$;
3. $\pi \models_N \Phi\wedge\Psi$, *if* $\pi \models_N \Phi$ *and* $\pi \models_N \Psi$;
4. $\pi \models_N \langle a\rangle\Phi$, *if* $\exists\tilde{\pi} \in \Pi(N)\ \pi \stackrel{a}{\to} \tilde{\pi}$ *and* $\tilde{\pi} \models_N \Phi$.

$[a]\Phi = \neg\langle a\rangle\neg\Phi$.

$N \models_N \Phi$, if $\pi_N \models_N \Phi$.

Definition 31 N *and* N' *are* are logical equivalent *in* HML, $N {=}_{HML} N'$, *if* $\forall\Phi \in \mathbf{HML}\ N \models_N \Phi \ \Leftrightarrow\ N' \models_{N'} \Phi$.

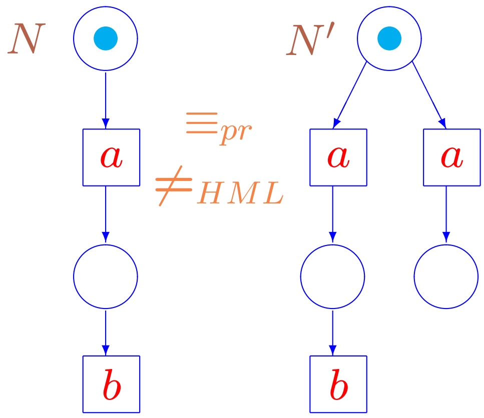
Let for a net N $\pi \in \Pi(N)$, $a \in Act$.

The set of *extensions* of a process π by action a (*image set*) is $Image(\pi, a) = \{\tilde{\pi} \mid \pi \stackrel{a}{\to} \tilde{\pi}\}$.

A net N is a *finite-image* one, if $\forall\pi \in \Pi(N)\ \forall a \in Act$ $|Image(\pi, a)| < \infty$.

Theorem 3 *For image-finite nets* N *and* N'
$N \underline{\leftrightarrow}_i N' \ \Leftrightarrow\ N \underline{\leftrightarrow}_{ibif} N' \ \Leftrightarrow\ N {=}_{HML} N'$.

Example on logical equivalence of HML



Differentiating power of $=_{HML}$

$N \equiv_{pr} N'$, but $N \not\equiv_{HML} N'$, because for $\Phi = [a]\langle b\rangle\top$, $N \models_N \Phi$, but $N' \not\models_{N'} \Phi$ since only in the net N' an action a can happen so that no b is possible afterwards.

Logic $PBFL$ [CLP92]

Definition 32 *$\top$ denotes the truth, $a \in Act$ and ρ is a pomset with labeling into Act.*

A formula of $PBFL$:

$$\Phi ::= \top \mid \neg\Phi \mid \Phi \wedge \Psi \mid \langle\leftarrow \rho\rangle\Phi \mid \langle a\rangle\Phi$$

PBFL is the set of *all formulas* of $PBFL$.

Definition 33 *Let $(\pi, \sigma) \in Runs(N)$ for a net N. The satisfaction relation $\models_N \in Runs(N) \times \mathbf{PBFL}$:*

1. $(\pi, \sigma) \models_N \top$ — *always;*

2. $(\pi, \sigma) \models_N \neg\Phi$, *if* $(\pi, \sigma) \not\models_N \Phi$;

3. $(\pi, \sigma) \models_N \Phi \wedge \Psi$, *if* $(\pi, \sigma) \models_N \Phi$ *and* $(\pi, \sigma) \models_N \Psi$;

4. $(\pi, \sigma) \models_N \langle\leftarrow \rho\rangle\Phi$, *if* $\exists(\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$ $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$, *where* $\hat{\pi} = (\widehat{C}, \hat{\varphi})$, $\rho_{\widehat{C}} \in \rho$ *and* $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$;

5. $(\pi, \sigma) \models_N \langle a\rangle\Phi$, *if* $\exists(\tilde{\pi}, \tilde{\sigma}) \in Runs(N)$ $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$, *where* $\hat{\pi} = (\widehat{C}, \hat{\varphi})$, $l_{\widehat{C}}(T_{\widehat{C}}) = a$ *and* $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$.

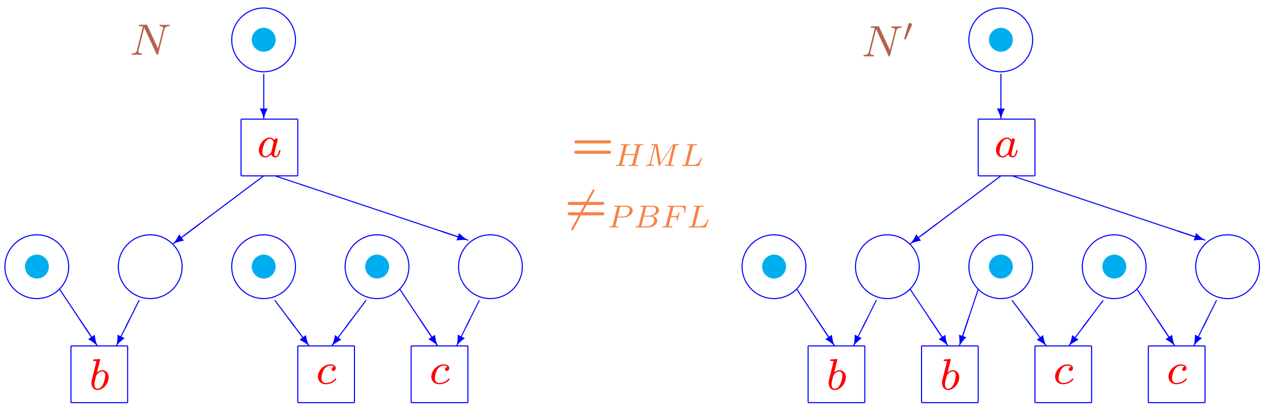
$[a]\Phi = \neg\langle a\rangle\neg\Phi$, $[\leftarrow \rho]\Phi = \neg\langle\leftarrow \rho\rangle\neg\Phi$

$N \models_N \Phi$, if $(\pi_N, \varepsilon) \models_N \Phi$.

Definition 34 *N and N' are logical equivalent in $PBFL$, $N =_{PBFL} N'$, if $\forall\Phi \in \mathbf{PBFL}$ $N \models_N \Phi \Leftrightarrow N' \models_{N'} \Phi$.*

Theorem 4 *For image-finite nets N and N' $N \leftrightarrows_{pomh} N' \Leftrightarrow N \leftrightarrows_{pombpomf} N' \Leftrightarrow N =_{PBFL} N'$.*

Example on logical equivalence of $PBFL$



Differentiating power of $=_{PBFL}$

$N =_{HML} N'$, but $N \neq_{PBFL} N'$, because for $\Phi = [a][b]\langle c\rangle\langle\leftarrow (a;b)\|c\rangle\top$, $N \models_N \Phi$, but $N' \not\models_{N'} \Phi$ since only in the net N' after action a an action b can happen so that c must depend on a.

Here $(a;b)\|c$ denotes the pomset where b depends on a, and $a,\ b$ are independent with c.

Logic $PrBFL$ [Tar97]

Definition 35 *$\top$ denotes the truth, $a \in Act$ and $\mathbf{C}$ is the isomorphism class of a causal net C.*

A formula of $PrBFL$:

$$\Phi ::= \top \mid \neg\Phi \mid \Phi \wedge \Psi \mid \langle\leftarrow \mathbf{C}\rangle\Phi \mid \langle a\rangle\Phi$$

PrBFL is the set of *all formulas* of $PrBFL$.

Definition 36 *Let $(\pi, \sigma) \in \mathit{Runs}(N)$ for a net N. The satisfaction relation $\models_N \in \mathit{Runs}(N) \times \mathbf{PrBFL}$:*

1. $(\pi, \sigma) \models_N \top$ — *always;*
2. $(\pi, \sigma) \models_N \neg\Phi$, *if* $(\pi, \sigma) \not\models_N \Phi$;
3. $(\pi, \sigma) \models_N \Phi \wedge \Psi$, *if* $(\pi, \sigma) \models_N \Phi$ *and* $(\pi, \sigma) \models_N \Psi$;
4. $(\pi, \sigma) \models_N \langle\leftarrow \mathbf{C}\rangle\Phi$, *if* $\exists(\tilde{\pi}, \tilde{\sigma}) \in \mathit{Runs}(N)$ $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\hat{\pi}} (\pi, \sigma)$, *where* $\hat{\pi} = (\widehat{C}, \hat{\varphi})$, $\widehat{C} \in \mathbf{C}$ *and* $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$;
5. $(\pi, \sigma) \models_N \langle a\rangle\Phi$, *if* $\exists(\tilde{\pi}, \tilde{\sigma}) \in \mathit{Runs}(N)$ $(\pi, \sigma) \xrightarrow{\hat{\pi}} (\tilde{\pi}, \tilde{\sigma})$, *where* $\hat{\pi} = (\widehat{C}, \hat{\varphi})$, $l_{\widehat{C}}(T_{\widehat{C}}) = a$ *and* $(\tilde{\pi}, \tilde{\sigma}) \models_N \Phi$.

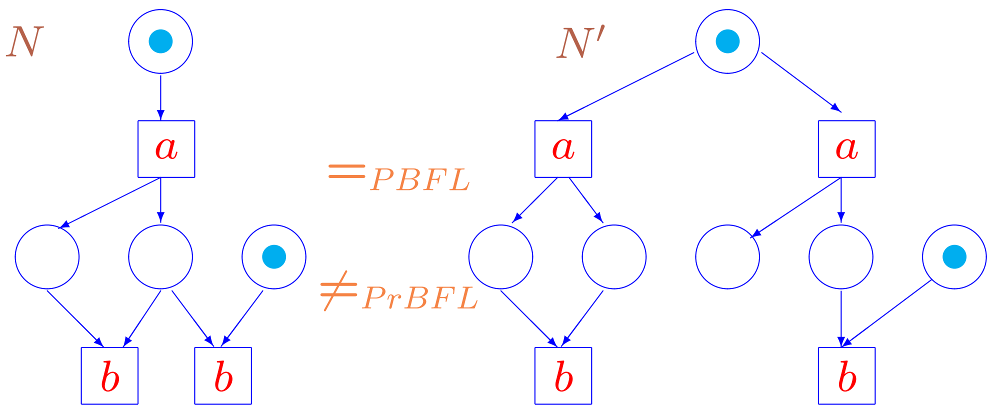
$[a]\Phi = \neg\langle a\rangle\neg\Phi$, $[\leftarrow \mathbf{C}]\Phi = \neg\langle\leftarrow \mathbf{C}\rangle\neg\Phi$.

$N \models_N \Phi$, if $(\pi_N, \varepsilon) \models_N \Phi$.

Definition 37 *N and N' are logical equivalent in $PrBFL$, $N =_{PrBFL} N'$, if $\forall\Phi \in \mathbf{PrBFL}$ $N \models_N \Phi \ \Leftrightarrow\ N' \models_{N'} \Phi$.*

Theorem 5 *For image-finite nets N and N'*
$N \leftrightarrows_{prh} N' \ \Leftrightarrow\ N \leftrightarrows_{prbprf} N' \ \Leftrightarrow\ N =_{PrBFL} N'$.

Example on logical equivalence of $PrBFL$



Differentiating power of $=_{PrBFL}$

$N =_{PBFL} N'$, but $N \neq_{PrBFL} N'$, because for $\Phi = [a]\langle b\rangle\langle\leftarrow \mathbf{C}\rangle\top$, $N \models_N \Phi$, but $N' \not\models_{N'} \Phi$, since only in the net N' a process with action a can start so that it can be extended by b in the only way (connecting pairwise output and input places).

Here $\mathbf{C}$ is an isomorphism class of causal net where two output places of an a-labeled transition are both the input places of b-labeled one.

Place bisimulation equivalences

Definition 38 $\mathcal{R} \subseteq \mathcal{M}(N) \times \mathcal{M}(N')$ *is a* $\star$-bisimulation *between nets* N *and* N', $\star \in$ {interleaving, step, partial word, pomset, process}, $\mathcal{R} : N \underline{\leftrightarrow}_\star N'$, $\star \in \{i, s, pw, pom, pr\}$, *if:*

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

2. $(M, M') \in \mathcal{R}$, $M \xrightarrow{\widehat{\pi}} \widetilde{M}$,

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

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

   $\Rightarrow$ $\exists \widetilde{M}' : \ M' \xrightarrow{\widehat{\pi}'} \widetilde{M}'$, $(\widetilde{M}, \widetilde{M}') \in \mathcal{R}$ *and*

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

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

   (c) $\widehat{C} \simeq \widehat{C}'$, *if* $\star = pr$.

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

N *and* N' *are* $\star$-bisimulation equivalent, $\star \in$ {interleaving, step, partial word, pomset, process}, $N \underline{\leftrightarrow}_\star N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}_\star N'$, $\star \in \{i, s, pw, pom, pr\}$.

Definition 39 *Let for nets* N *and* N' $\mathcal{R} \subseteq P_N \times P_{N'}$.
A lifting *of* $\mathcal{R}$ *is* $\overline{\mathcal{R}} \subseteq \mathcal{M}(P_N) \times \mathcal{M}(P_{N'})$, *defined as:*

$$(M, M') \in \overline{\mathcal{R}} \;\Leftrightarrow\; \begin{cases} \exists\{(p_1, p'_1), \ldots, (p_n, p'_n)\} \in \mathcal{M}(\mathcal{R}) : \\ M = \{p_1, \ldots p_n\},\;\; M' = \{p'_1, \ldots p'_n\} \end{cases}$$

Definition 40 $\mathcal{R} \subseteq P_N \times P_{N'}$ *is a* $\star$-place bisimulation *between nets* N *and* N', $\star \in${interleaving, step, partial word, pomset, process}, $\mathcal{R} : N \sim_\star N'$, *if* $\overline{\mathcal{R}} : N \underline{\leftrightarrow}_\star N'$, $\star \in \{i, s, pw, pom, pr\}$.

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

Strict place bisimulations require additionally the corresponding transitions to be related by $\overline{\mathcal{R}}$.

Definition 41 *Let for nets N and N' $t \in T_N$, $t' \in T_{N'}$. Then:*

$$(t, t') \in \overline{\mathcal{R}} \;\Leftrightarrow\; \begin{cases} ({}^\bullet t, {}^\bullet t') \in \overline{\mathcal{R}} \;\wedge \\ (t^\bullet, t'^\bullet) \in \overline{\mathcal{R}} \;\wedge \\ l_N(t) = l_{N'}(t') \end{cases}$$

Definition 42 $\mathcal{R} \subseteq P_N \times P_{N'}$ *is a* strict $\star$-place bisimulation *between nets N and N'*, $\star \in${interleaving, step, partial word, pomset, process}, $\mathcal{R} : N \approx_\star N'$, $\star \in \{i, s, pw, pom, pr\}$, *if:*

1. $\overline{\mathcal{R}} : N \underline{\leftrightarrow}_\star N'$.

2. *The new requirement is added to item 2 (and to 3) of the definition of $\star$-bisimulation:*

   $\forall v \in T_{\widehat{C}}\ (\widehat{\varphi}(v), \widehat{\varphi}'(\beta(v))) \in \overline{\mathcal{R}}$, *where:*

   (a) $\beta : \rho_{\widehat{C'}} \sqsubseteq \rho_{\widehat{C}}$, *if* $\star = pw$;

   (b) $\beta : \rho_{\widehat{C}} \simeq \rho_{\widehat{C'}}$, *if* $\star \in \{i, s, pom\}$;

   (c) $\beta : \widehat{C} \simeq \widehat{C}'$, *if* $\star = pr$.

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

An important property of place bisimulations: *additivity*.

Let for nets N and N' $\mathcal{R} : N \sim_{\star} N'$, $\star \in \{i, s, pw, pom, pr\}$.

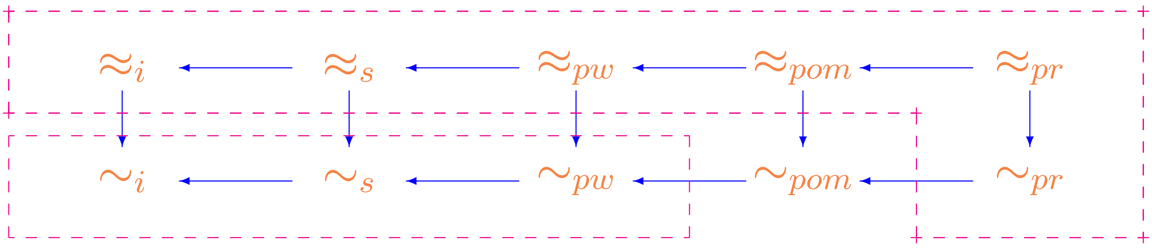
Then $(M_1, M_1') \in \overline{\mathcal{R}}$ and $(M_2, M_2') \in \overline{\mathcal{R}}$ implies $((M_1 + M_2), (M_1' + M_2')) \in \overline{\mathcal{R}}$.

If we add n tokens in each of the places $p \in P_N$ and $p' \in P_{N'}$ s.t. $(p, p') \in \mathcal{R}$, then the resulting nets must be also place bisimulation equivalent.

Comparing place bisimulation equivalences

Proposition 3 *[AS92] For nets N and N':*

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

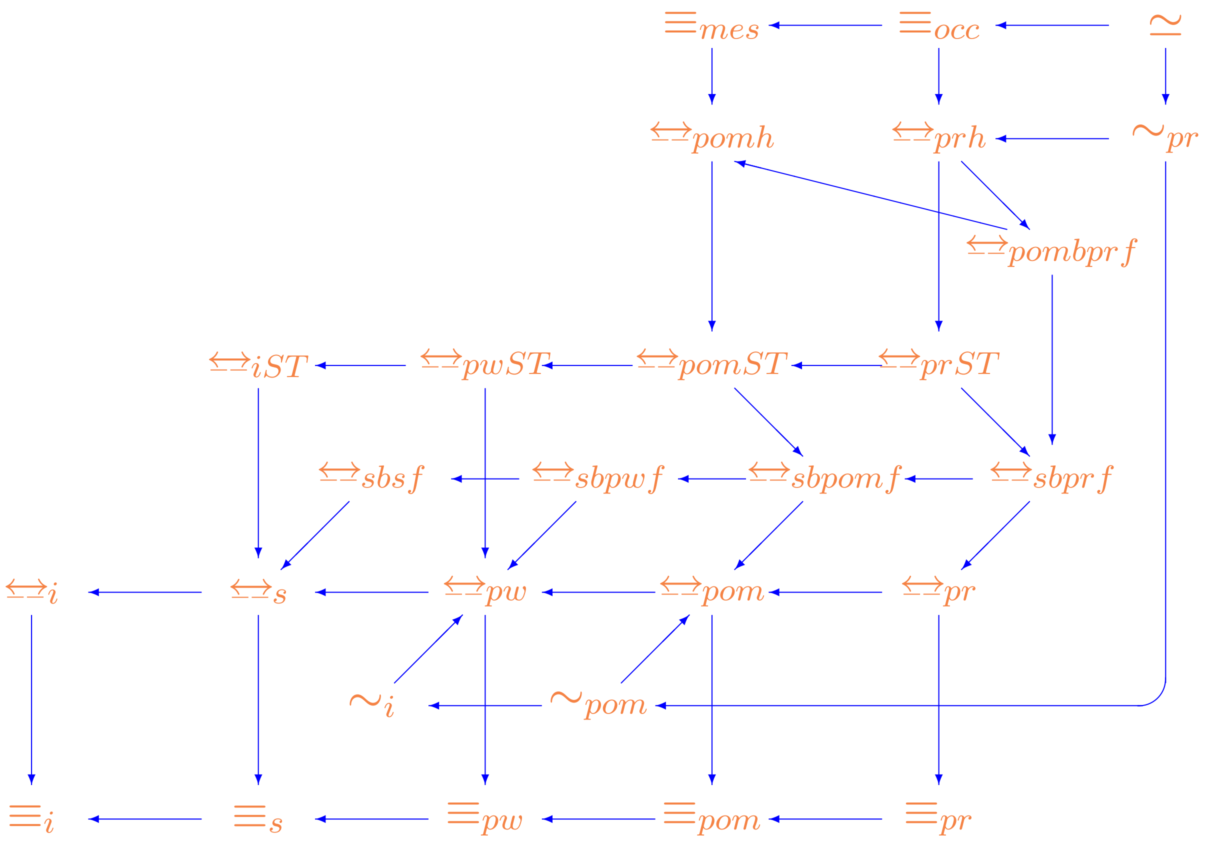


Merging of place bisimulation equivalences

Interrelations of place bisimulation equivalences

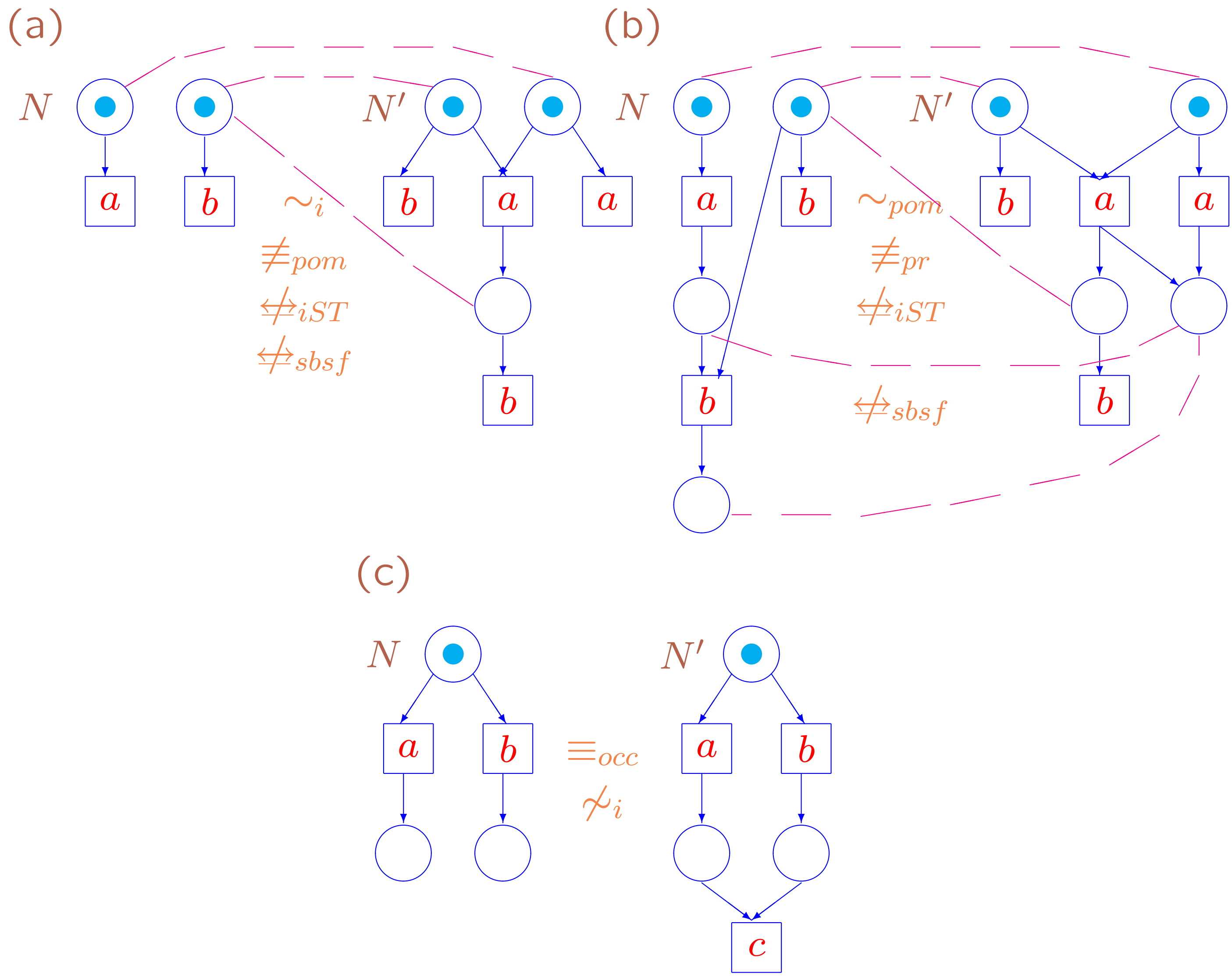
Comparing place bisimulation equivalences with basic and back-forth ones

Proposition 4 [Tar97, Tar98b] *For nets N and N' $N \sim_{pr} N' \ \Rightarrow \ N \leftrightarrow_{prh} N'$.*



Interrelations of place bisimulation equivalences with basic and back-forth ones

Theorem 6 *Let $\leftrightarrow, \leftrightarrow\!\!\!\!\leftrightarrow \in \{\equiv, \leftrightarrow, \sim, \simeq\}$, $\star, \star\star \in \{\_, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombprf\}$. For nets N and N' $N \leftrightarrow_{\star} N'$ $\Rightarrow$ $N \leftrightarrow\!\!\!\!\leftrightarrow_{\star\star} N'$ iff in the graph above there exists a directed path from $\leftrightarrow_{\star}$ to $\leftrightarrow\!\!\!\!\leftrightarrow_{\star\star}$.*



P: Examples of place bisimulation equivalences

- In Figure P(a), $N \sim_i N'$, but $N \not\equiv_{pom} N'$, since only in the net N' action b can depend on a.

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

- In Figure P(c), $N \equiv_{occ} N'$, but $N \not\sim_i N'$, since any place bisimulation must relate input places of the nets N and N'. But if we add one additional token in each of these places, then only in N' the action c can happen.

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

- In Figure B1(c), $N \sim_{pr} N'$, but $N \not\equiv_{mes} N'$, since only the MES corresponding to the net N' has two conflict actions a.

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

Net reduction based on place bisimulation equivalences

An *autobisimulation* is a bisimulation between a net and itself.

An *equibisimulation* is an autobisimulation that is an equivalence.

Proposition 5 *[AS92] Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be reflexive interleaving place autobisimulations of a net N. Then $(\mathcal{R}_1 \cup \mathcal{R}_2)^\star$ (transitive closure of $(\mathcal{R}_1 \cup \mathcal{R}_2)$) is an interleaving place autobisimulation.*

Definition 43 *For a net N,*
$\mathcal{R}_i(N) = \bigcup\{\mathcal{R} \mid \mathcal{R} : N \sim_i N,\ \mathcal{R}$ *is reflexive$\}$ is a* canonical interleaving place bisimulation.

Definition 44 *Let for a net N $\mathcal{E} \subseteq P_N \times P_N$ be an equivalence.*

For $p \in P_N$, $[p]_\mathcal{E} = \{q \mid (p,q) \in \mathcal{E}\}$ is an equivalence class *of p w.r.t. $\mathcal{E}$.*

For $M \in \mathcal{M}(P_N)$, $M/_\mathcal{E} = \sum_{p \in P_N} [p]_\mathcal{E}$ is a categorization (partitioning) *of M w.r.t. $\mathcal{E}$.*

$N/_\mathcal{E} = \langle P_N/_\mathcal{E}, T_N, F_N/_\mathcal{E}, l_N, M_N/_\mathcal{E}\rangle$, *where $F_N/_\mathcal{E}$ is constructed as:*

1. $^\bullet t = M$ in $N \Rightarrow {}^\bullet t = M/_\mathcal{E}$ in $N/_\mathcal{E}$;
2. $t^\bullet = M$ in $N \Rightarrow t^\bullet = M/_\mathcal{E}$ in $N/_\mathcal{E}$.

$M \xrightarrow{t} \widetilde{M}$ in N implies $M/_\mathcal{E} \xrightarrow{t} \widetilde{M}/_\mathcal{E}$ in $N/_\mathcal{E}$.

Proposition 6 *[AS92] If $\mathcal{R} : N \sim_i N$ is an equivalence then $[\cdot]_\mathcal{R} : N \sim_i N/_\mathcal{E}$.*

Definition 45 *A* canonical interleaving categorization *of a net N is a net $N/_{\sim_i} = N/_{\mathcal{R}_i(N)}$.*

Definition 46 *For a net N, $\mathcal{R} \subseteq P_N \times P_N$ has a* transfer property*, if $\forall t \in T_N\ \forall p \in {}^\bullet t\ \forall q:\ (p,q) \in \mathcal{R}$ holds:*
$\exists u \in T_N:\ l_N(t) = l_N(u),\ {}^\bullet t - p + q \xrightarrow{u} \widetilde{M}$ and $(t^\bullet, \widetilde{M}) \in \mathcal{R}$.

Theorem 7 *[AS92] If for a net N, $\mathcal{R} \subseteq P_N \times P_N$ is a reflexive and symmetrical relation having transfer property then $\mathcal{R}^\star$ (transitive closure of $\mathcal{R}$) is an interleaving place bisimulation in N.*

Theorem 8 *[AS92] For a net N, the maximal relation $\mathcal{R} \subseteq P_N \times P_N$ having transfer property is $\mathcal{R}_i(N)$.*

An effective algorithm of computing $\mathcal{R}_i(N)$ [AS92]:

1. The initial relation: $\mathcal{R} = P_N \times P_N$.

2. Check all pairs $(p,q) \in \mathcal{R}$ for transfer property.

   (a) If the property is valid for all that pairs then $\mathcal{R} = \mathcal{R}_i(N)$.

   (b) Otherwise, there exists a pair (p,q), for which transfer property is not valid. Then we remove the pairs (p,q) and (q,p) from $\mathcal{R}$ and go to item 2.

If a net is finite then a number of the pairs is finite too.

A complexity: $\mathcal{O}(|P_N|^2 \times |T_N|^2)$, if $\forall t \in T_N\ |{}^\bullet t| + |t^\bullet| \leq d$ (the constant depends on d) [Pfi92].

An implementation: a system CAESAR on LOTOS programming language [Pfi92].

The results on using $\sim_{pom}$ and $\sim_{pr}$ for net reduction

- We cannot use $\sim_{pom}$ for net simplification, since there is an example s.t. for a net N : $N \not\sim_{pom} N/_{\sim_{pom}}$ [AS92].

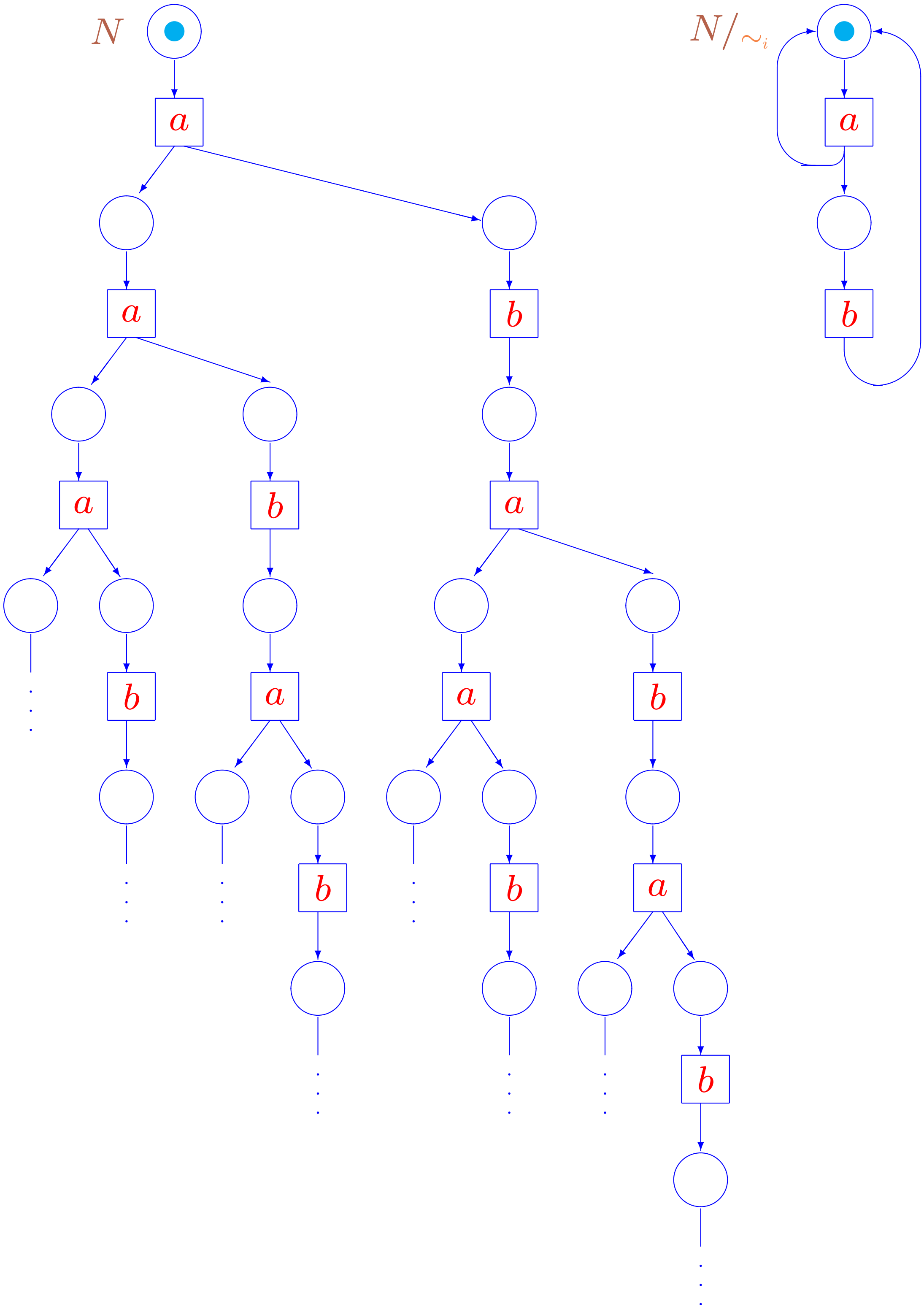
- Since $\sim_{pr} = \approx_i$, we can modify the algorithm for $\mathcal{R}_i$ to obtain $\mathcal{R}_{pr}$: we shall look for bisimulation between transitions in the pairs appearing during check of the transfer property.

  A complexity of the algorithm will be the same. Thus, it is possible to reduce net effectively modulo $\sim_{pr}$.

Important results (due to interrelations of $\sim_{pr}$ with the other equivalences).

1. Since $\sim_{pr}$ implies $\underline{\leftrightarrow}_{prh}$ and $\underline{\leftrightarrow}_{prST}$, a reduced net has the same histories of behavior and *timed traces* [GV87] as the initial one.

2. Since $\underline{\leftrightarrow}_{prh}$ coincide with $=_{PrBFL}$, all the properties that can be specified in logic $PrBFL$ are preserved in the reduced net.



Reduction of the net corresponding to a PBC formula
$\mu X.(a;(X\|(b;X)))$ modulo $\sim_i$

SM-refinements [BDKP91]

Definition 47 *An SM-net is a net $D = \langle P_D, T_D, F_D, l_D, M_D \rangle$ s.t.:*

1. $\forall t \in T_D \; |{}^\bullet t| = |t^\bullet| = 1$, *i.e. each transition has exactly one input and one output place;*

2. $\exists p_{in},\; p_{out} \in P_D$ 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.*

3. $M_D = \{p_{in}\}$, *i.e. at the beginning there is unique token in* p_{in}.

Definition 48 *Let $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ be a 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, $ref(N, a, D)$, is a net $\overline{N} = \langle P_{\overline{N}}, T_{\overline{N}}, F_{\overline{N}}, l_{\overline{N}}, M_{\overline{N}} \rangle$:*

- $P_{\overline{N}} = P_N \cup \{\langle p, u\rangle \mid p \in P_D \setminus \{p_{in}, p_{out}\},\; u \in l_N^{-1}(a)\}$;

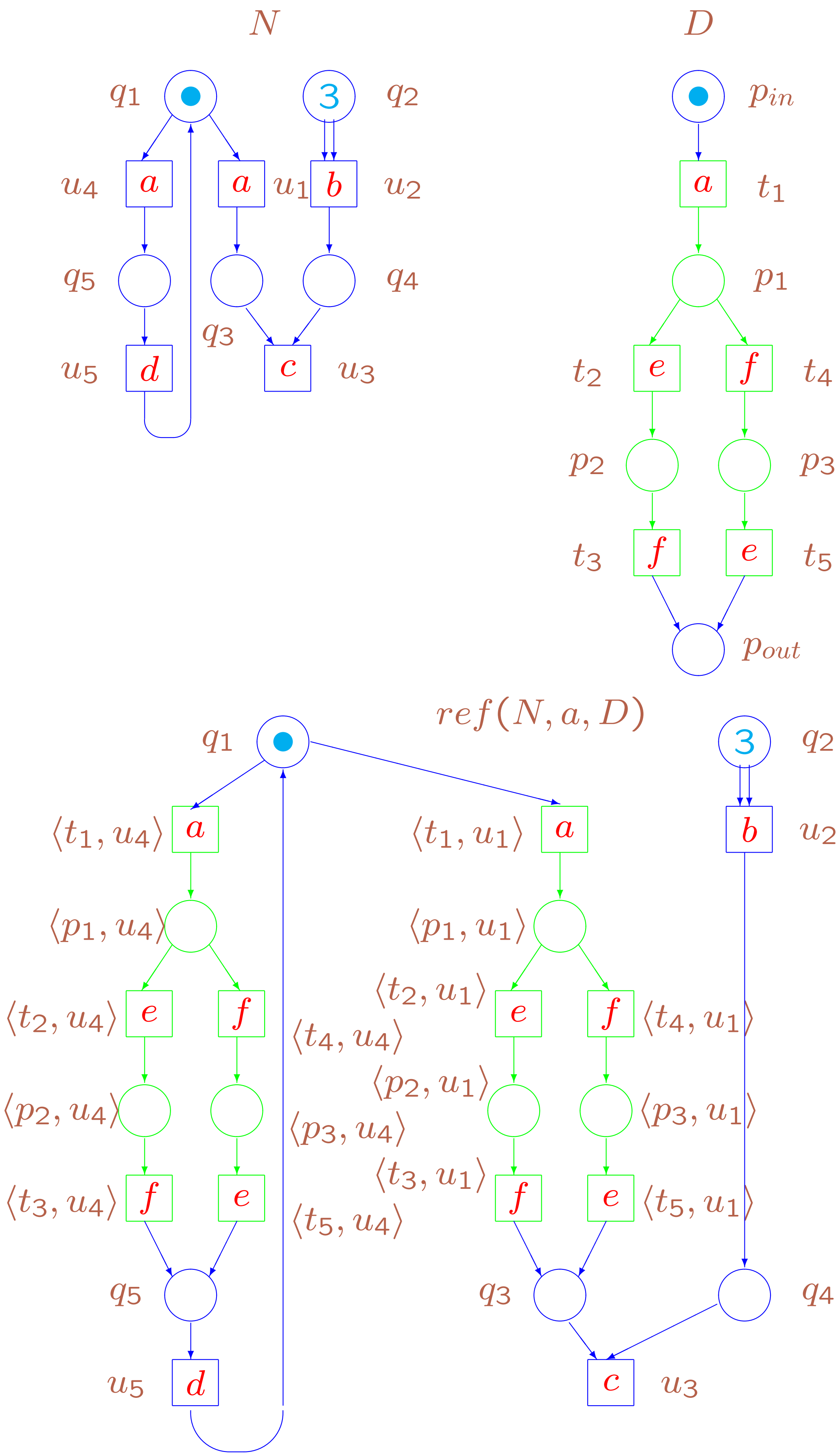
- $T_{\overline{N}} = (T_N \setminus l_N^{-1}(a)) \cup \{\langle t, u\rangle \mid t \in T_D,\; u \in l_N^{-1}(a)\}$;

- $F_{\overline{N}}(\bar{x}, \bar{y}) = \begin{cases} F_N(\bar{x}, \bar{y}), & \bar{x}, \bar{y} \in P_N \cup (T_N \setminus l_N^{-1}(a)); \\ F_D(x, y), & \bar{x} = \langle x, u\rangle,\; \bar{y} = \langle y, u\rangle,\; u \in l_N^{-1}(a); \\ F_N(\bar{x}, u), & \bar{y} = \langle y, u\rangle,\; \bar{x} \in {}^\bullet u,\; u \in l_N^{-1}(a),\; y \in p_{in}^\bullet; \\ F_N(u, \bar{y}), & \bar{x} = \langle x, u\rangle,\; \bar{y} \in {}^\bullet u,\; u \in l_N^{-1}(a),\; x \in {}^\bullet p_{out}; \\ 0, & \textit{otherwise}; \end{cases}$

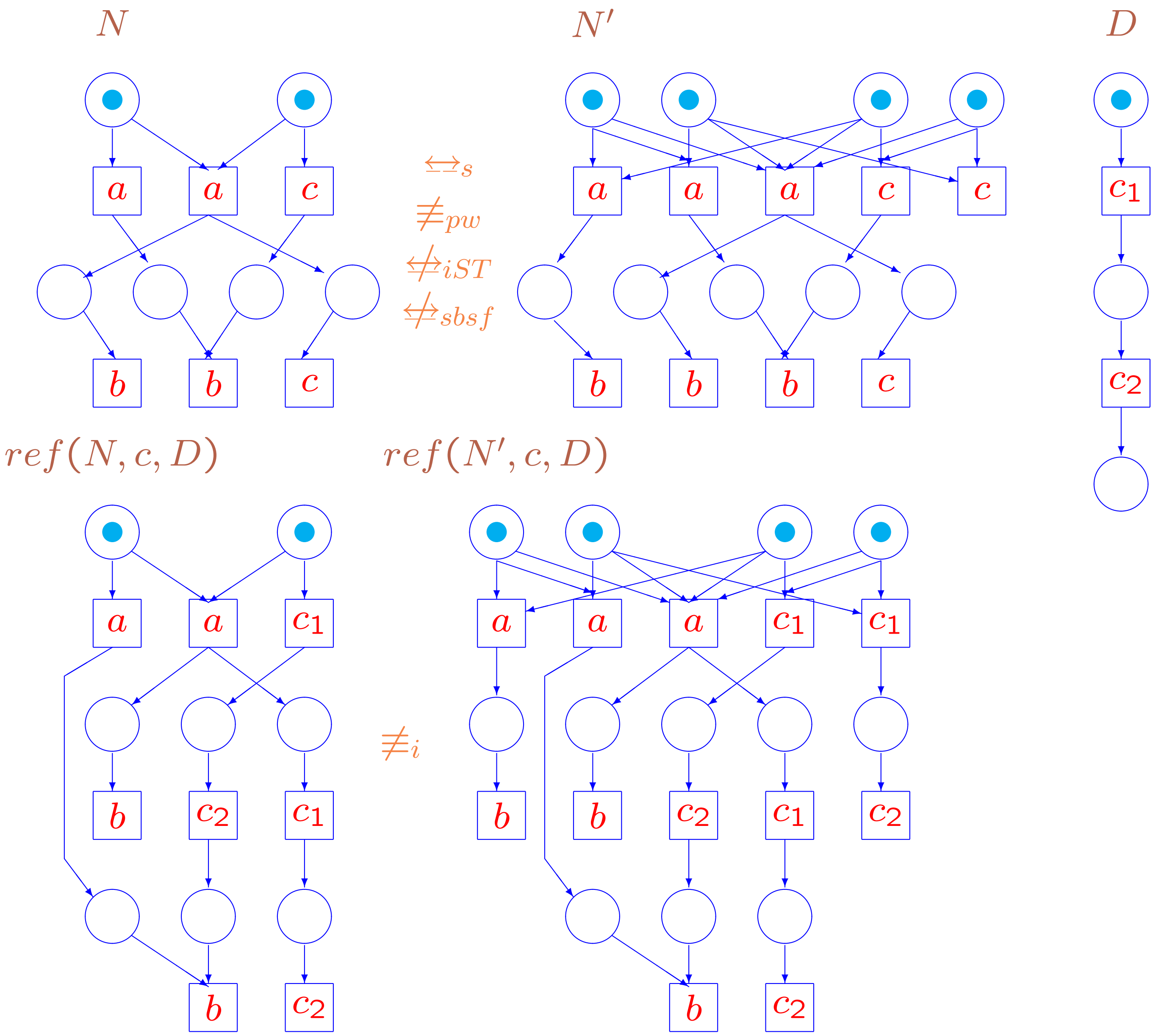
- $l_{\overline{N}}(\bar{u}) = \begin{cases} l_N(\bar{u}), & \bar{u} \in T_N \setminus l_N^{-1}(a); \\ l_D(t), & \bar{u} = \langle t, u\rangle,\; t \in T_D,\; u \in l_N^{-1}(a); \end{cases}$

- $M_{\overline{N}}(p) = \begin{cases} M_N(p), & p \in P_N; \\ 0, & \textit{otherwise}. \end{cases}$

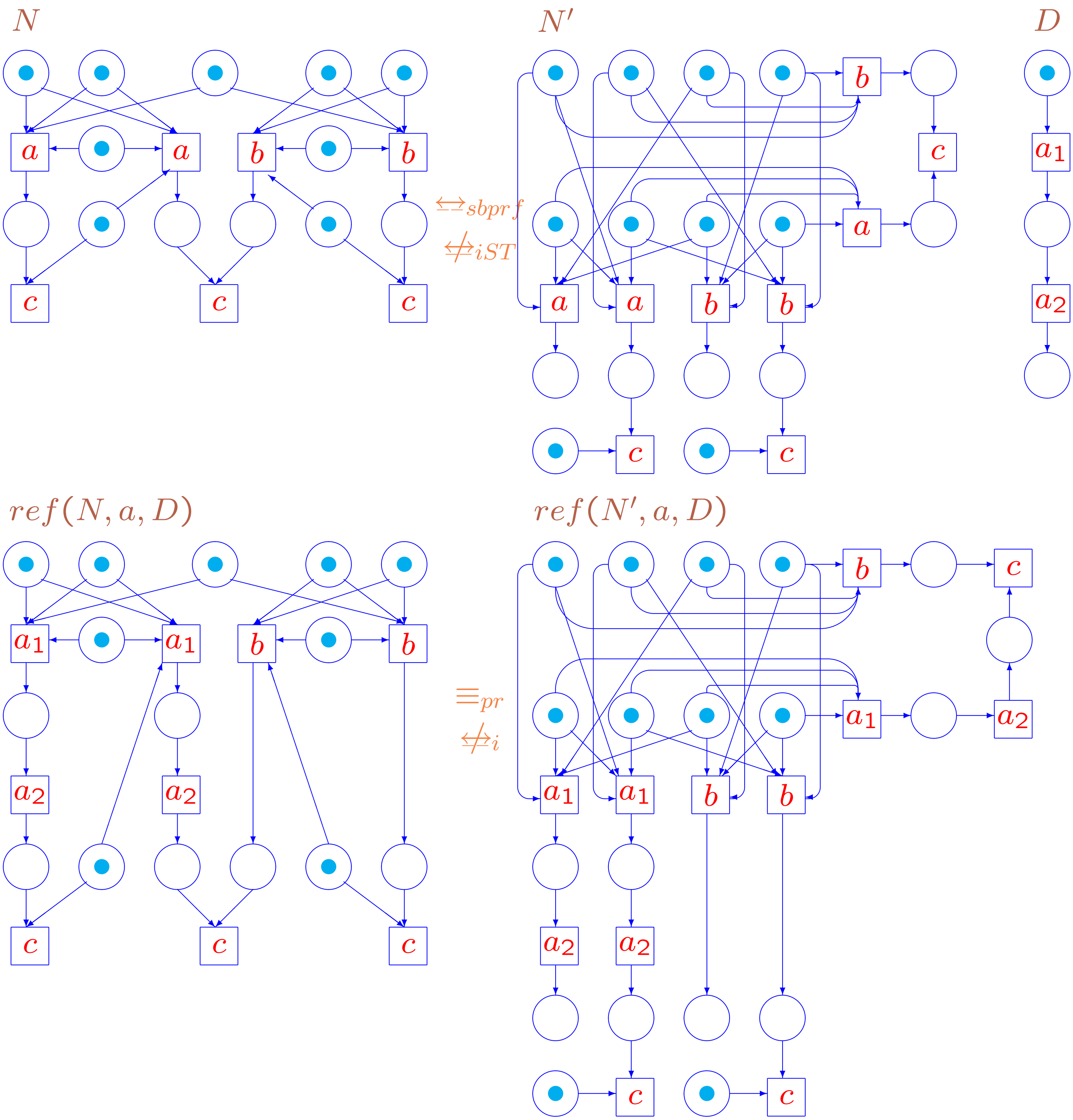
An equivalence is *preserved by refinements*, if equivalent nets remain equivalent after applying any refinement operator to them accordingly.



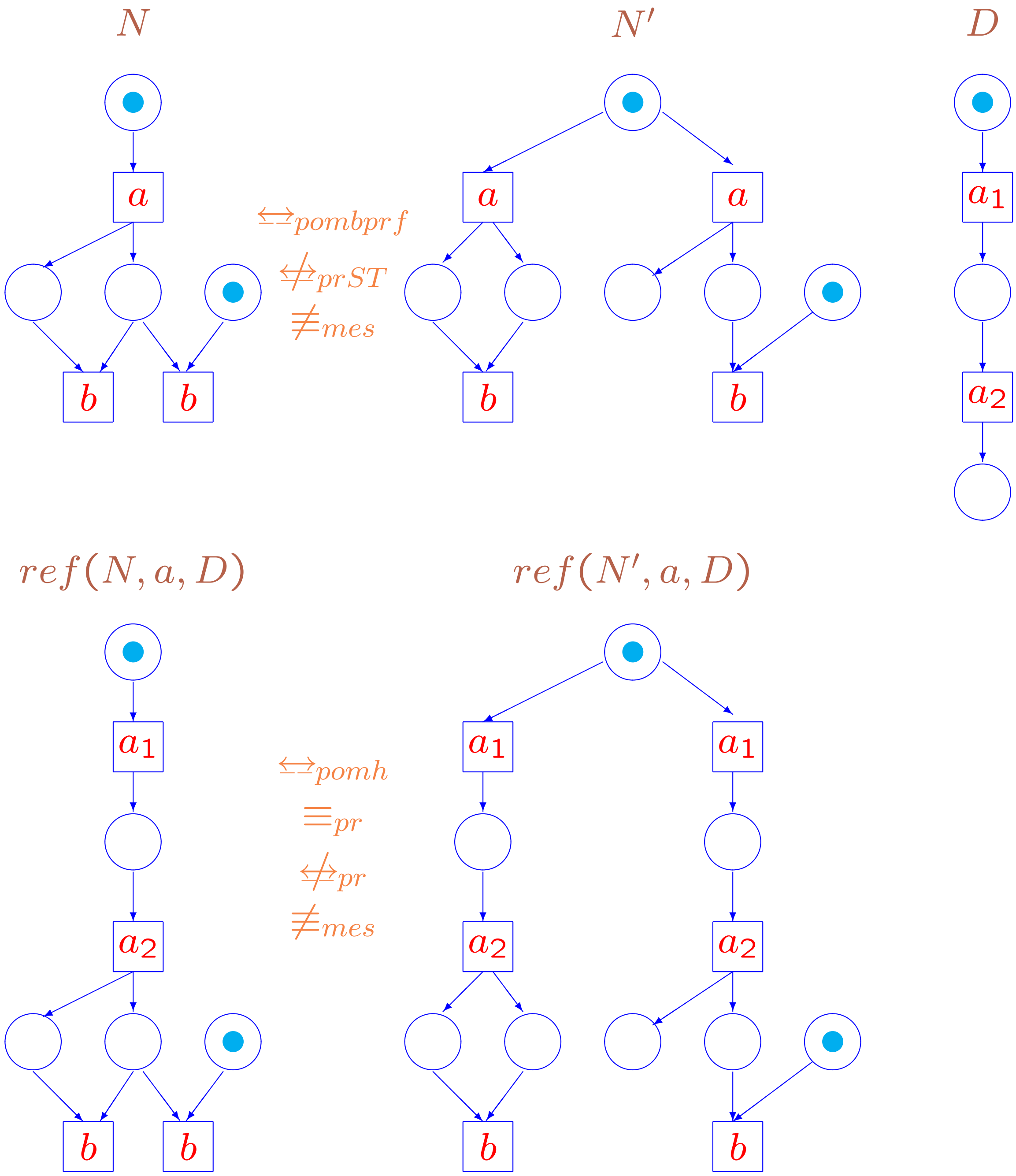
Example of SM-refinement



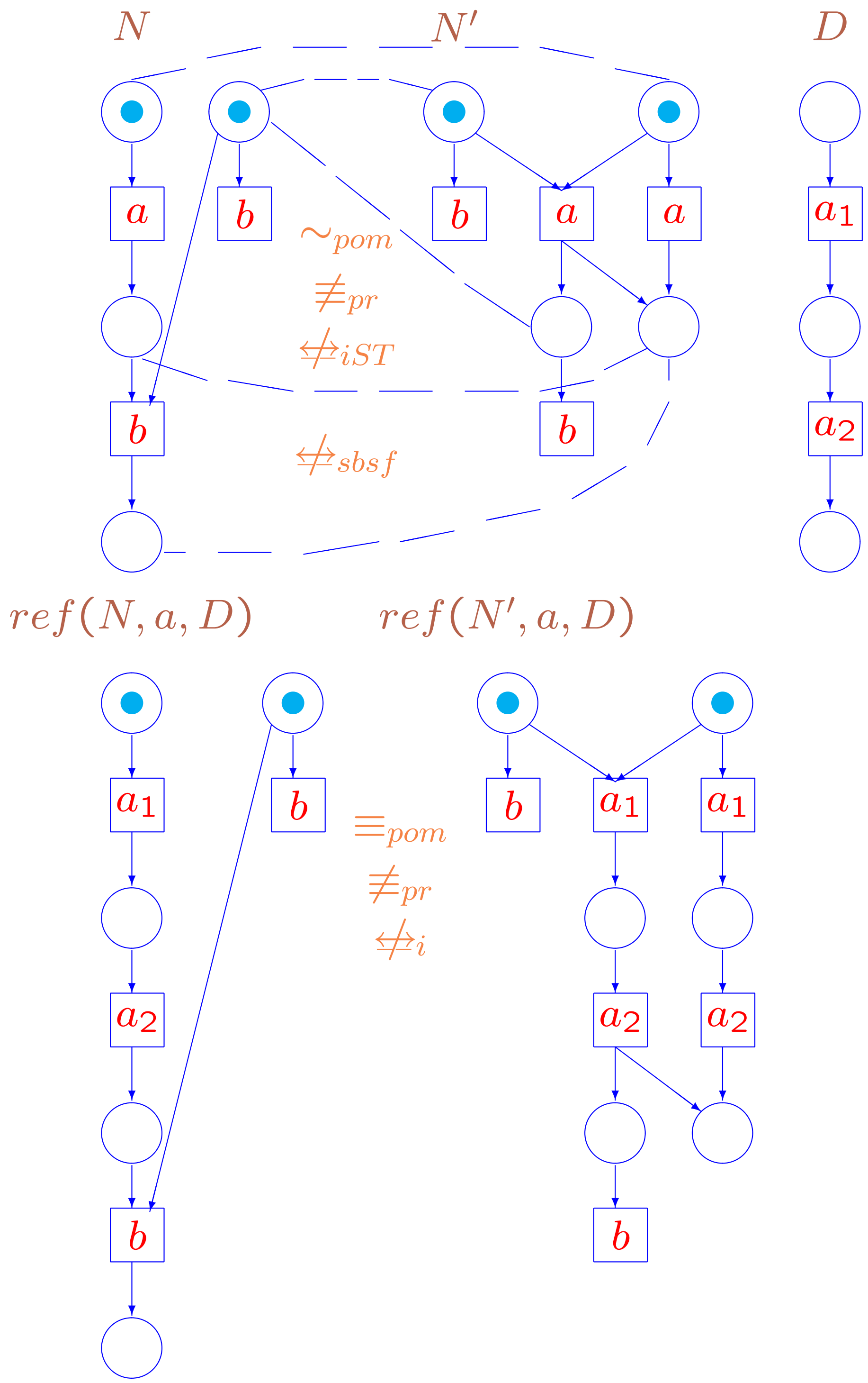
RB: The equivalences between $\equiv_i$ and $\underline{\leftrightarrow}_s$ are not preserved by SM-refinements



RBF: The equivalences between $\underline{\leftrightarrow}_i$ and $\underline{\leftrightarrow}_{sbprf}$ are not preserved by SM-refinements



RBF1: The equivalences between $\leftrightarrow_{pr}$ and $\leftrightarrow_{pombprf}$ are not preserved by SM-refinements



RP: The equivalences between $\leftrightarrow_i$ and $\sim_{pom}$ are not preserved by SM-refinements

- In Figure RB, $N \underline{\leftrightarrow}_s N'$, but $ref(N,c,D) \not\equiv_i ref(N',c,D)$, since only in $ref(N',c,D)$ the sequence of actions c_1abc_2 can happen.

- In Figure RBF, $N \underline{\leftrightarrow}_{sbprf} N'$, but $ref(N,a,D) \not\underline{\leftrightarrow}_i ref(N',a,D)$, since only in the net $ref(N',a,D)$ action a_1 can happen so that immediately after it:

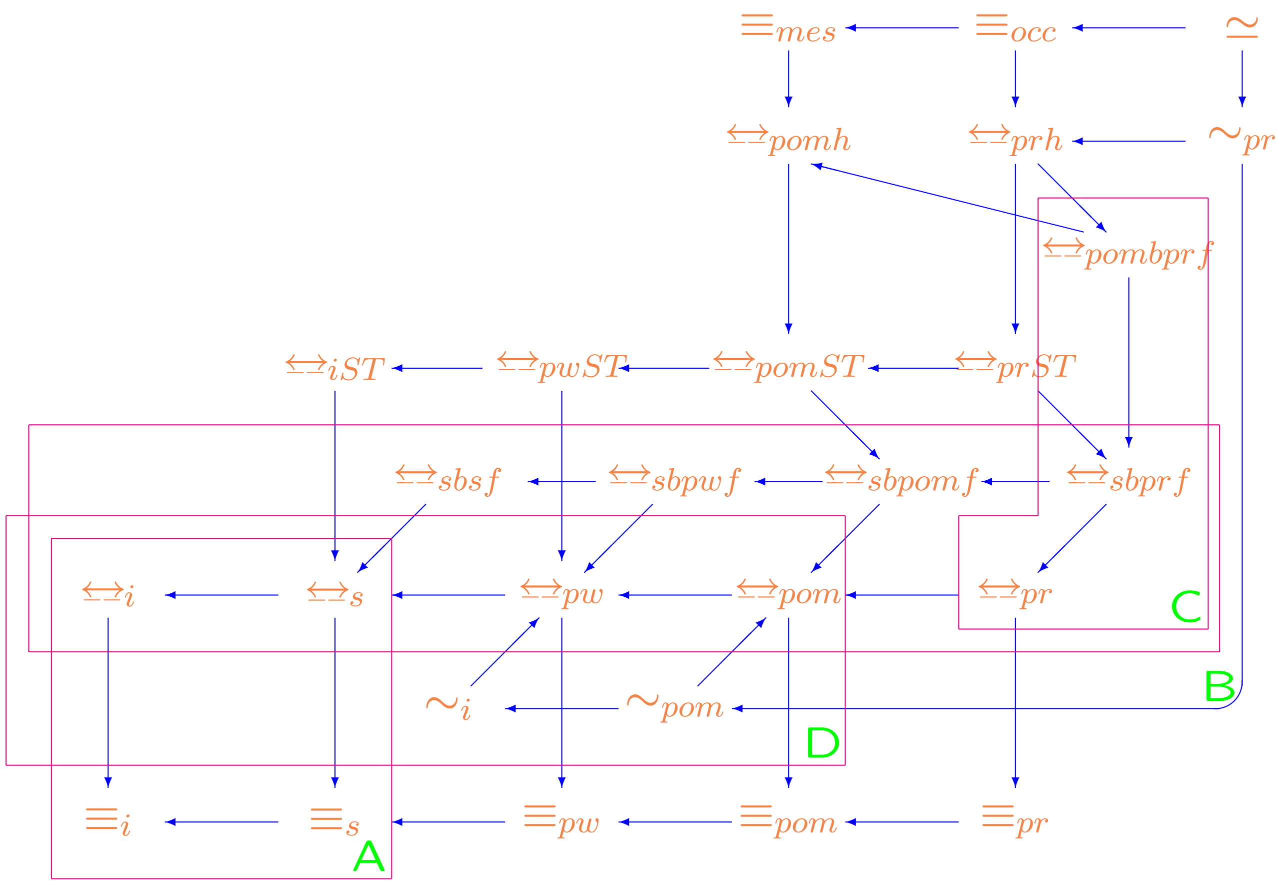
  1. the sequence of actions bc cannot happen, and

  2. the sequence of actions a_2c cannot happen.

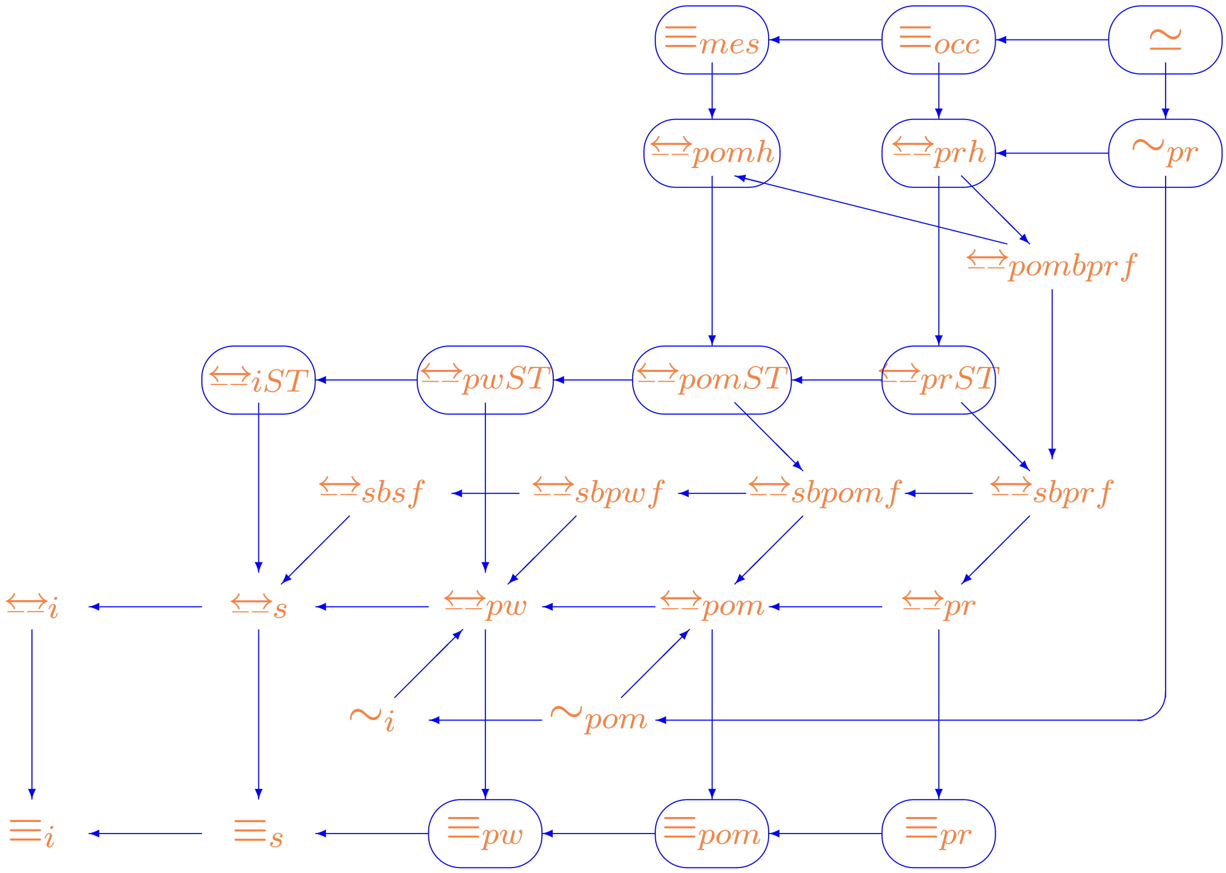
- In Figure RBF1, $N \underline{\leftrightarrow}_{pombprf} N'$, but $ref(N,a,D) \not\underline{\leftrightarrow}_{pr} ref(N',a,D)$, since only in the net $ref(N',a,D)$ action a_1 can happen so that after it the sequence of actions a_2b can happen which has only one corresponding process (the transition labeled by b connects with transition with label a_2 in the only way).

- In Figure RP, $N \sim_{pom} N'$, but $ref(N,a,D) \not\underline{\leftrightarrow}_i ref(N',a,D)$, since only in the net $ref(N',a,D)$ after action a_1 action b cannot happen.

Proposition 7 [BDKP91,Tar97] *Let* $\star \in \{i, s\}$, $\star\star \in \{i, s, pw, pom, pr, sbsf, sbpwf, sbpomf, sbprf, pombprf\}$, $\star\star\star \in \{i, pom\}$. *Then the equivalences* $\equiv_\star$, $\underline{\leftrightarrow}_{\star\star}$, $\sim_{\star\star\star}$ *are not preserved by SM-refinements.*



The equivalences which are not preserved by SM-refinements



Preservation of the equivalences by SM-refinements

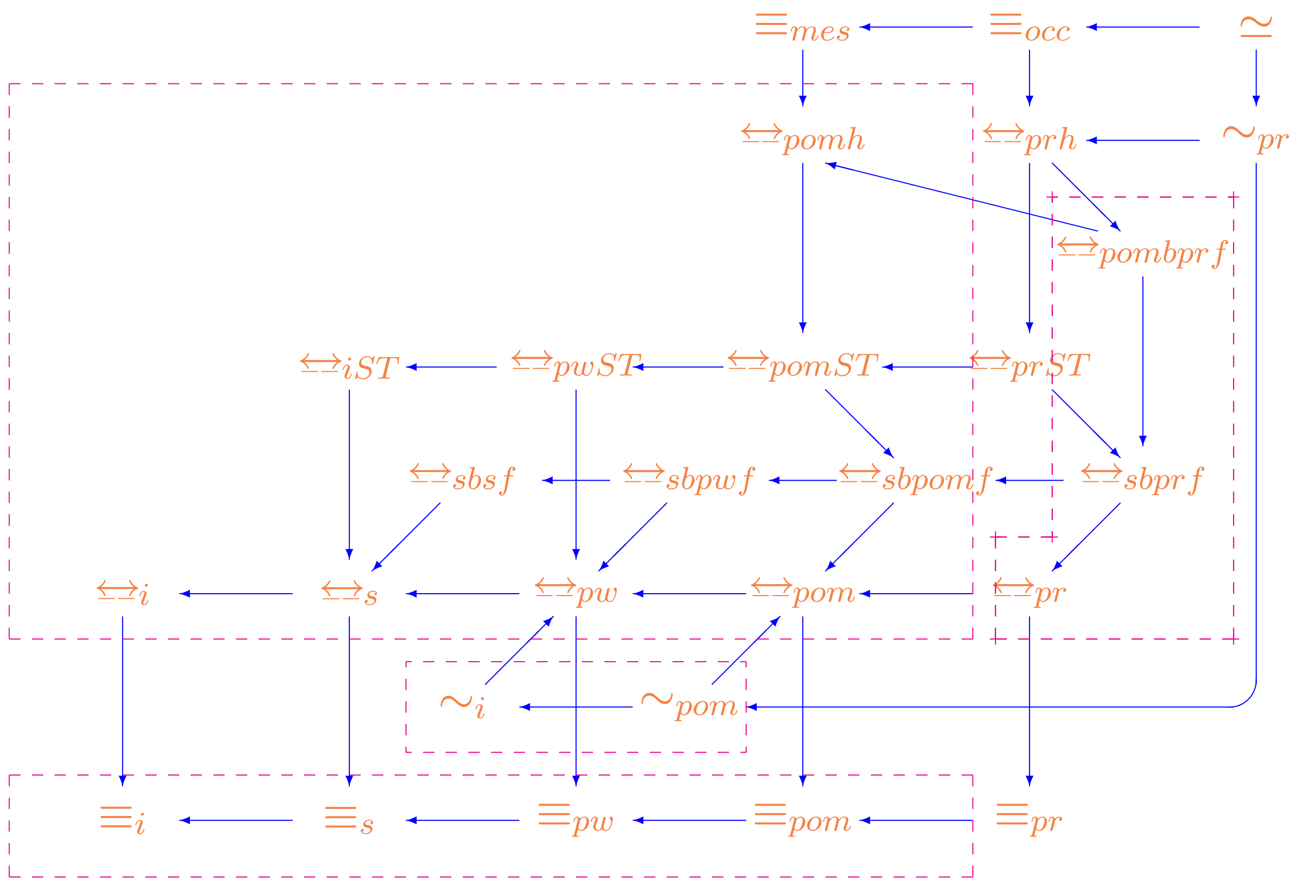
Theorem **9** *Let* $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, \sim, \simeq\}$ *and* $\star \in \{\_, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ, sbsf, sbpwf, sbpomf, sbprf, pombprf\}$*. For nets* N*,* N' *s.t.* $a \in l_N(T_N) \cap l_{N'}(T_{N'})$ *and SM-net* D*:* $N \leftrightarrow_\star N' \Rightarrow ref(N, a, D) \leftrightarrow_\star ref(N', a, D)$ *iff the equivalence* $\leftrightarrow_\star$ *is in oval in the figure above.*

The equivalences on sequential nets

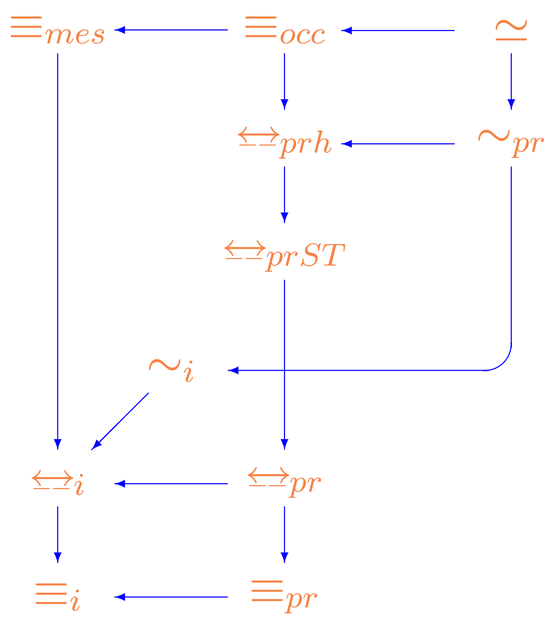
Definition 49 *A net* $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ *is* sequential, *if* $\forall M \in Mark(N)\ \neg\exists t, u \in T_N :\ {}^\bullet t + {}^\bullet u \subseteq M$.

Proposition 8 *For sequential nets* N *and* N'*:*

1. $N \equiv_i N' \ \Leftrightarrow\ N \equiv_{pom} N'$ [Eng85];
2. $N \leftrightarrows_i N' \ \Leftrightarrow\ N \leftrightarrows_{pomh} N'$ [BDKP91];
3. $N \leftrightarrows_{pr} N' \ \Leftrightarrow\ N \leftrightarrows_{pombprf} N'$ [Tar97];
4. $N \sim_i N' \ \Leftrightarrow\ N \sim_{pom} N'$ [Tar97].

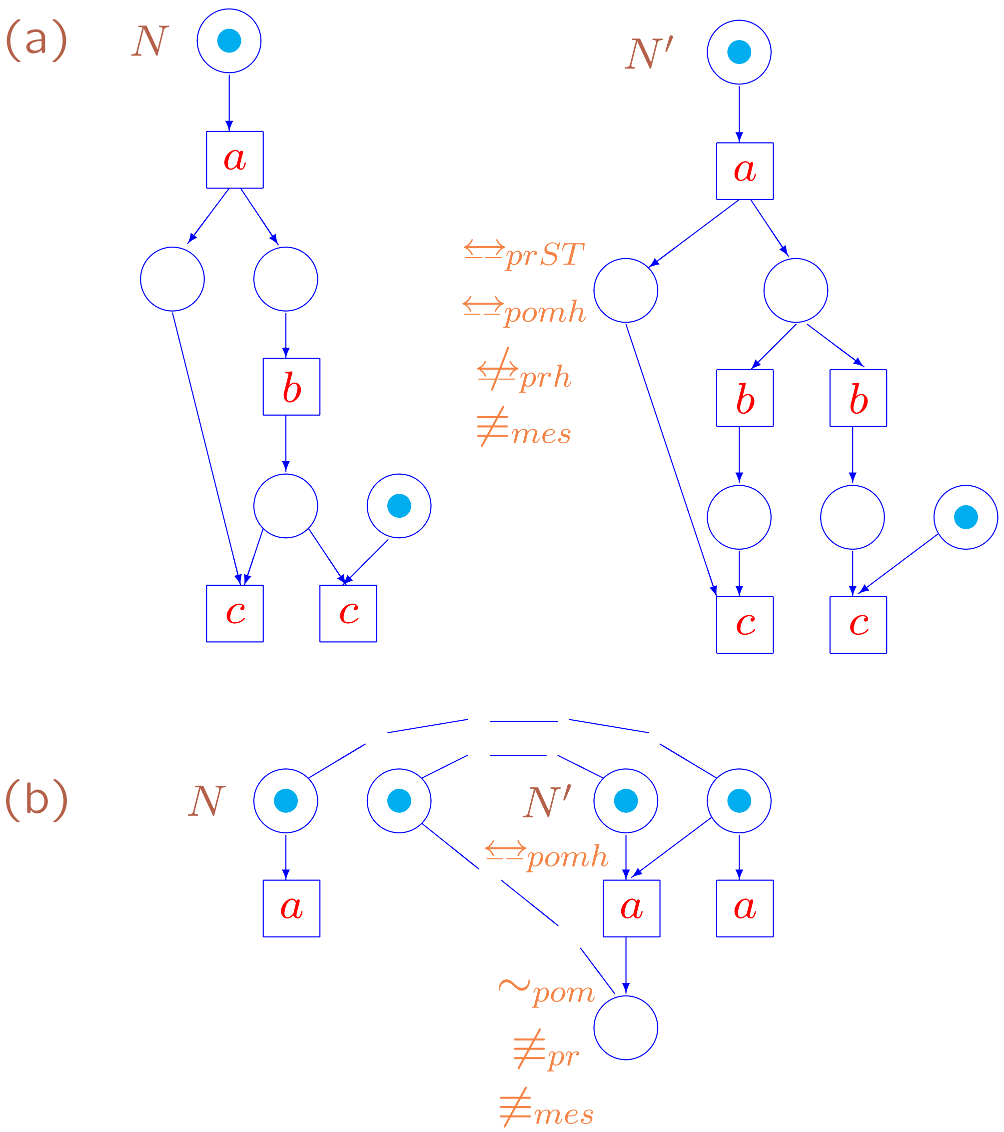


Merging of the equivalences on sequential nets



Interrelations of the equivalences on sequential nets

Theorem 10 *Let* $\leftrightarrow, \twoheadleftarrow\!\!\twoheadrightarrow \in \{\equiv, \underline{\leftrightarrow}, \sim, \simeq\}$, $\star, \star\star \in \{\_, i, pr, prST, prh, mes, occ\}$. *For sequential nets* N *and* N' $N \leftrightarrow_\star N' \Rightarrow N \twoheadleftarrow\!\!\twoheadrightarrow_{\star\star} N'$ *iff in the graph above there exists a directed path from* $\leftrightarrow_\star$ *to* $\twoheadleftarrow\!\!\twoheadrightarrow_{\star\star}$.



SN: Examples of the equivalences on sequential nets

- In Figure B(d), $N \equiv_{mes} N'$, but $N \not\equiv_{pr} N'$.

- In Figure RB(e), $N \equiv_{pr} N'$, but $N \not\underline{\leftrightarrow}_{i} N'$.

- In Figure BF(c), $N \underline{\leftrightarrow}_{pr} N'$, but $N \not\underline{\leftrightarrow}_{prST} N'$.

- In Figure SN(a), $N \underline{\leftrightarrow}_{prST} N'$, but $N \not\underline{\leftrightarrow}_{prh} N'$, since only in the net N' there is process with actions a and b s.t. it can be extended by process with action c in the only way (i.e. so that connection of causal net with action c and a-containing subnet of causal net with actions a and b be unique).

- In Figure B1(c), $N \underline{\leftrightarrow}_{prh} N'$, but $N \not\equiv_{mes} N'$.

- In Figure B1(d), $N \equiv_{occ} N'$, but $N \not\simeq N'$.

- In Figure SN(b), $N \sim_{i} N'$, but $N \not\equiv_{pr} N'$, since only in the net N' the transition with label a has two input places.

- In Figure P(c), $N \equiv_{occ} N'$, but $N \not\sim_{i} N'$.

- In Figure B1(c), $N \sim_{pr} N'$, but $N \not\equiv_{mes} N'$.

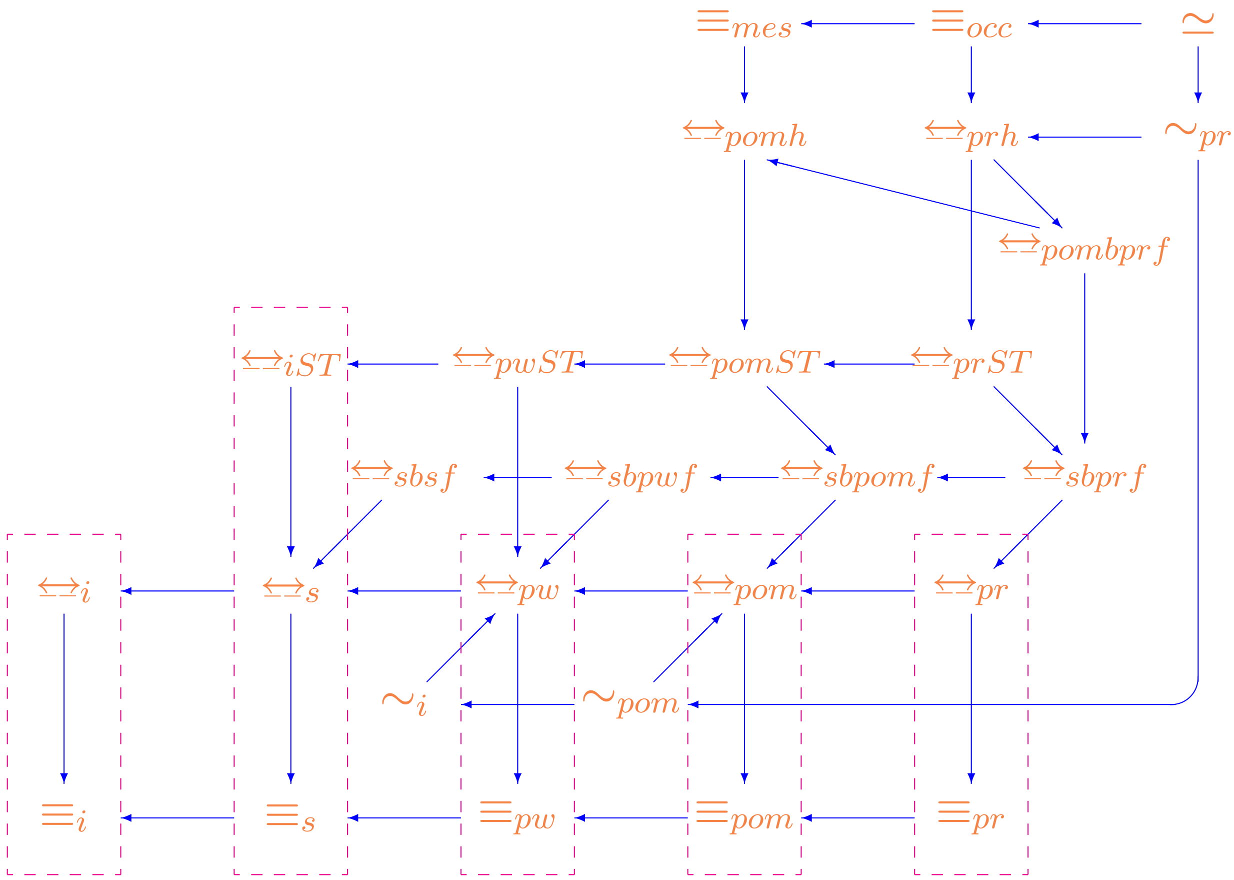
The equivalences on strictly labeled nets

Definition 50 *A net $N = \langle P_N, T_N, F_N, l_N \rangle$ is strictly labeled (unlabeled) if $\forall t, u \in T_N \ l_N(t) \neq l_N(u)$.*

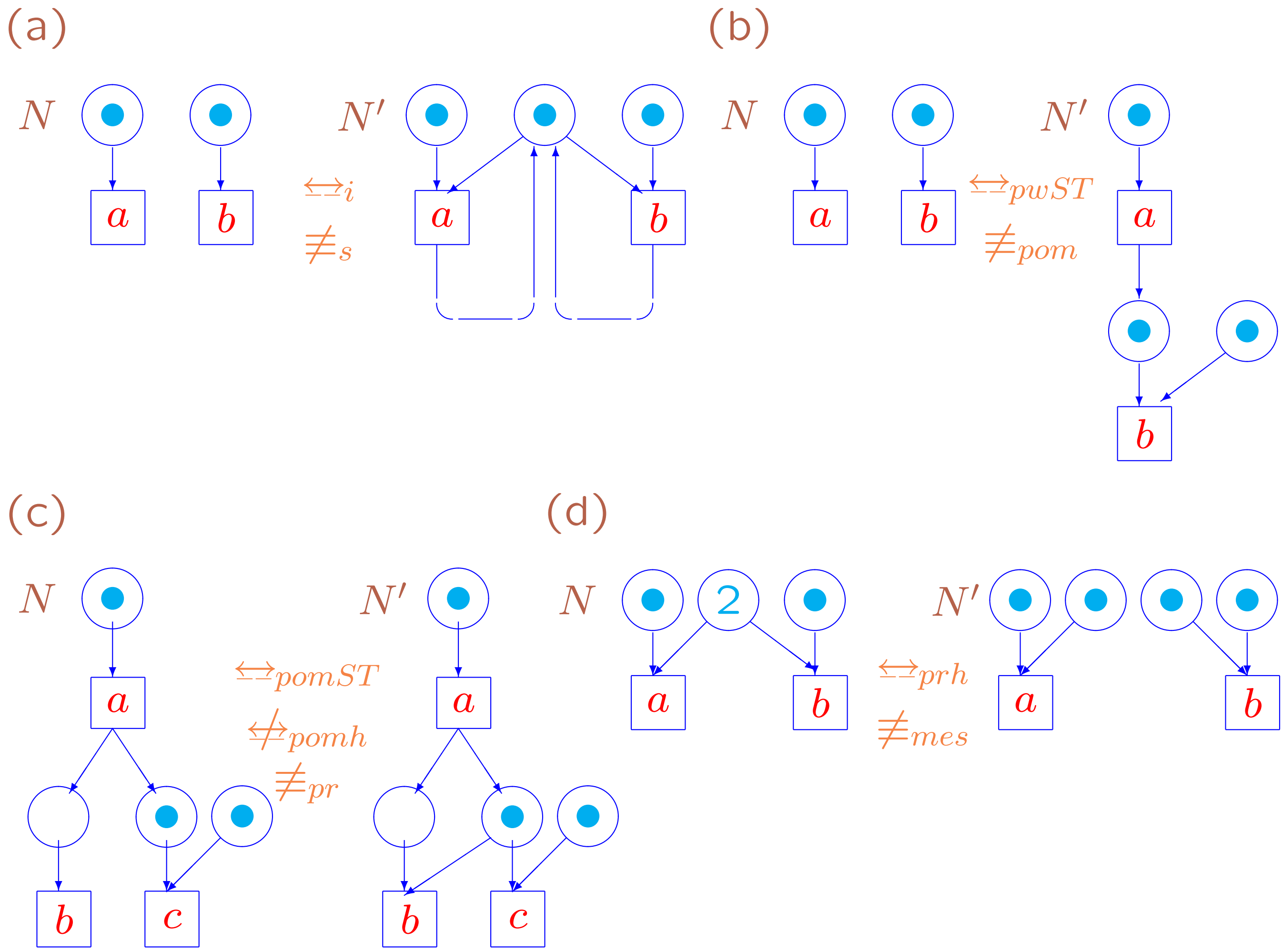
Proposition 9 *Let $\star \in \{i, pw, pom, pr\}$. For strictly labeled nets N and N':*

1. $N \equiv_\star N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_\star N'$ *[PRS92, Tar97];*

2. $N \equiv_s N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{iST} N'$ *[Tar97].*



Merging of the equivalences on strictly labeled nets



UL: Examples of the equivalences on strictly labeled nets

- In Figure UN(a), $N \underline{\leftrightarrow}_i N'$, but $N \not\equiv_s N'$, since only in the net N actions a and b can happen concurrently.

- In Figure UN(b), $N \underline{\leftrightarrow}_{pwh} N'$, but $N \not\equiv_{pom} N'$, since only in the net N' action b can depend on a.

- In Figure B(d), $N \equiv_{mes} N'$, but $N \not\equiv_{pr} N'$.

- In Figure UN(c), $N \underline{\leftrightarrow}_{pomST} N'$, but $N \not\underline{\leftrightarrow}_{pomh} N'$, since only in the net N' a sequence of actions ab can happen so that c must depend on a.

- In Figure UN(d), $N \underline{\leftrightarrow}_{prh} N'$, but $N \not\equiv_{mes} N'$, since only in the unfolding of the net N' transitions with labels a and b have common input place. A MES with conflict actions a and b corresponds to this unfolding.

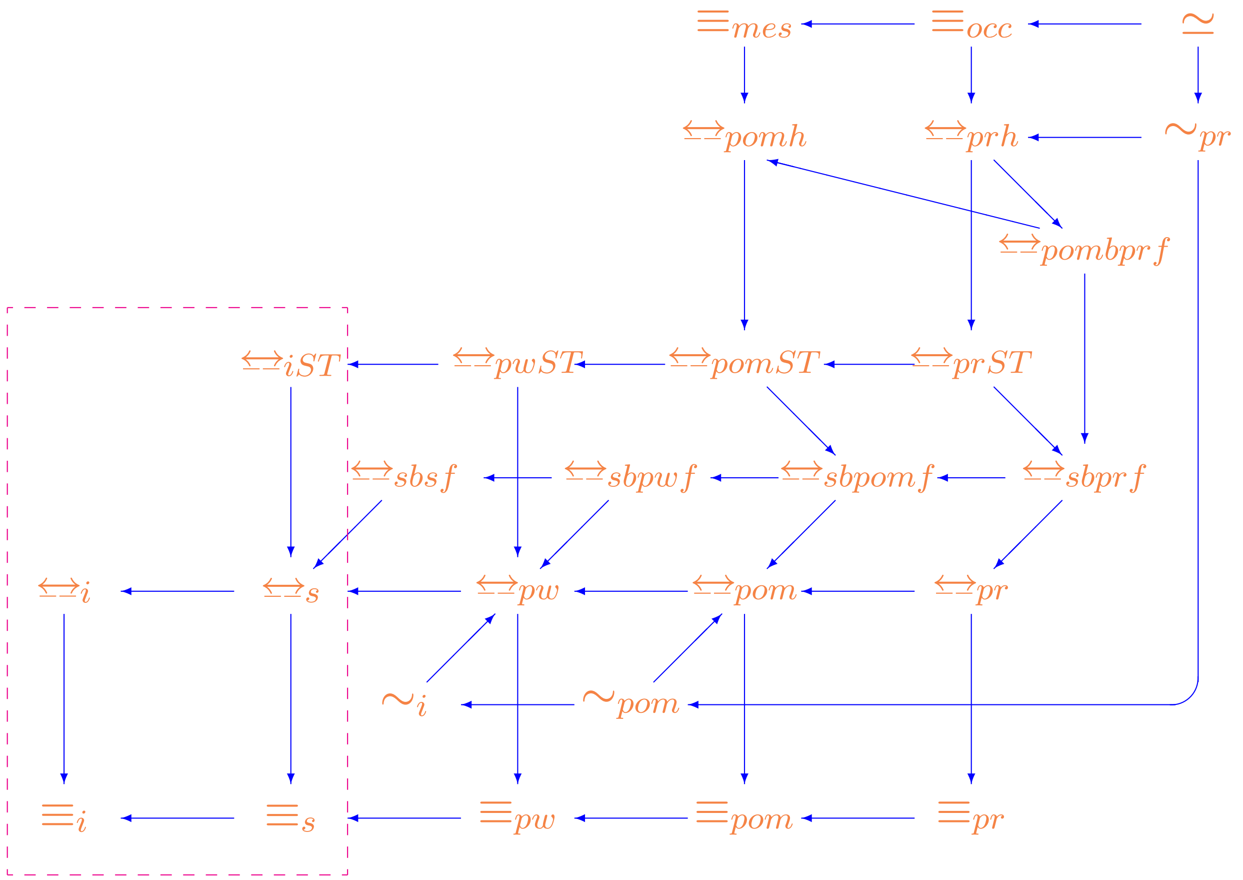
- In Figure B1(d), $N \equiv_{occ} N'$, but $N \not\simeq N'$.

- In Figure P(c), $N \equiv_{occ} N'$, but $N \not\asymp_i N'$.

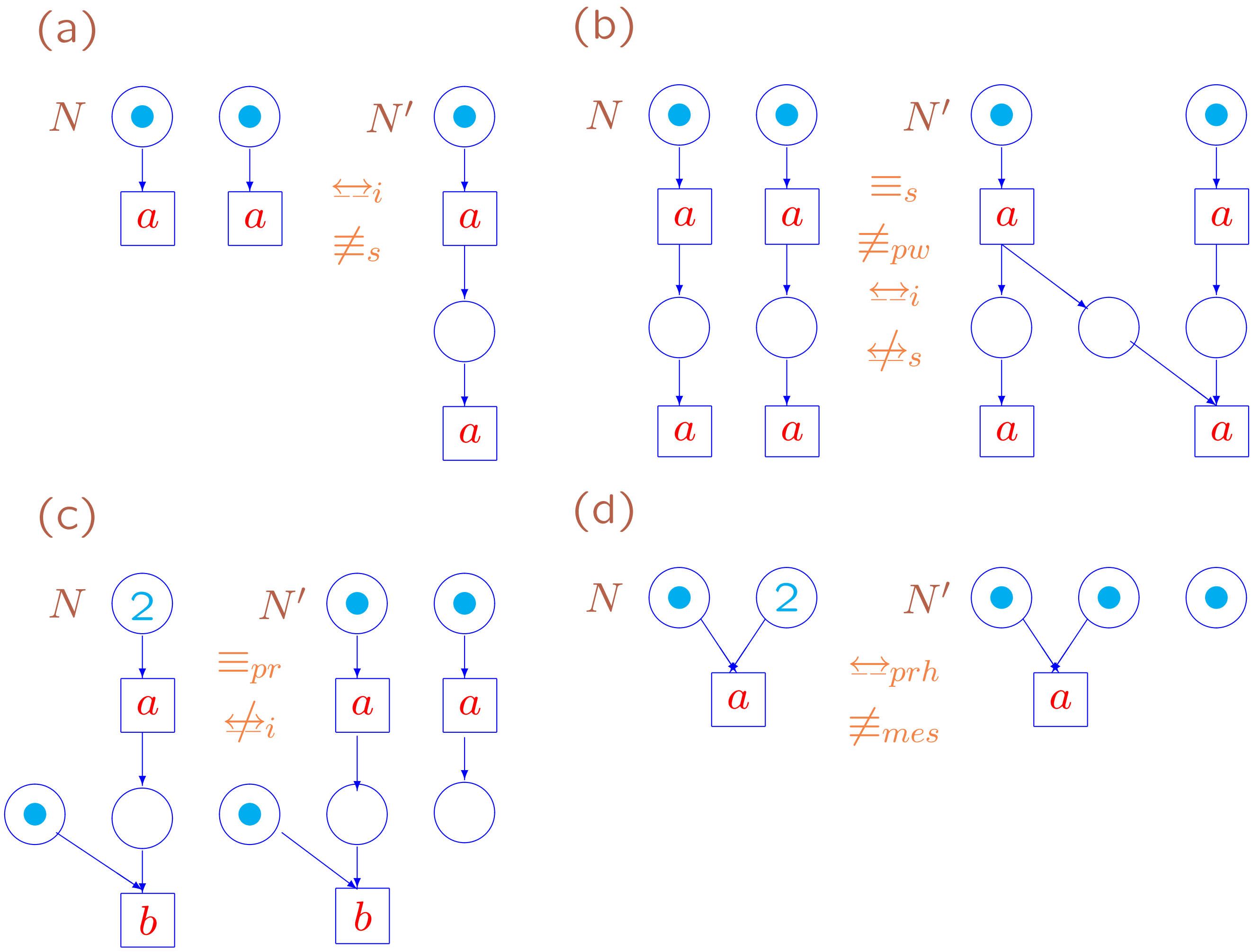
The equivalences on T-nets

Definition **51** *A net* $N = \langle P_N, T_N, F_N, l_N \rangle$ *is a* T-net*, if* $\forall p \in P_N \ |^\bullet p| \leq 1$ *and* $|p^\bullet| \leq 1$.

Proposition **10** *[Tar97] For auto-concurrency free T-nets* N *and* N' $N \equiv_i N' \Leftrightarrow N \underline{\leftrightarrow}_{iST} N'$.



Merging of the equivalences on T-nets



TN: Examples of the equivalences on T-nets

- In Figure TN(a), $N \underline{\leftrightarrow}_i N'$, but $N \not\equiv_s N'$, since only in the net N' an action a cannot happen concurrently with itself (i.e., it is not auto-concurrent).

- In Figure TN(b), $N \equiv_s N'$, but $N \not\equiv_{pw} N'$, since the net N structurally represents a pomset s.t. even less sequential one cannot happen in N'.

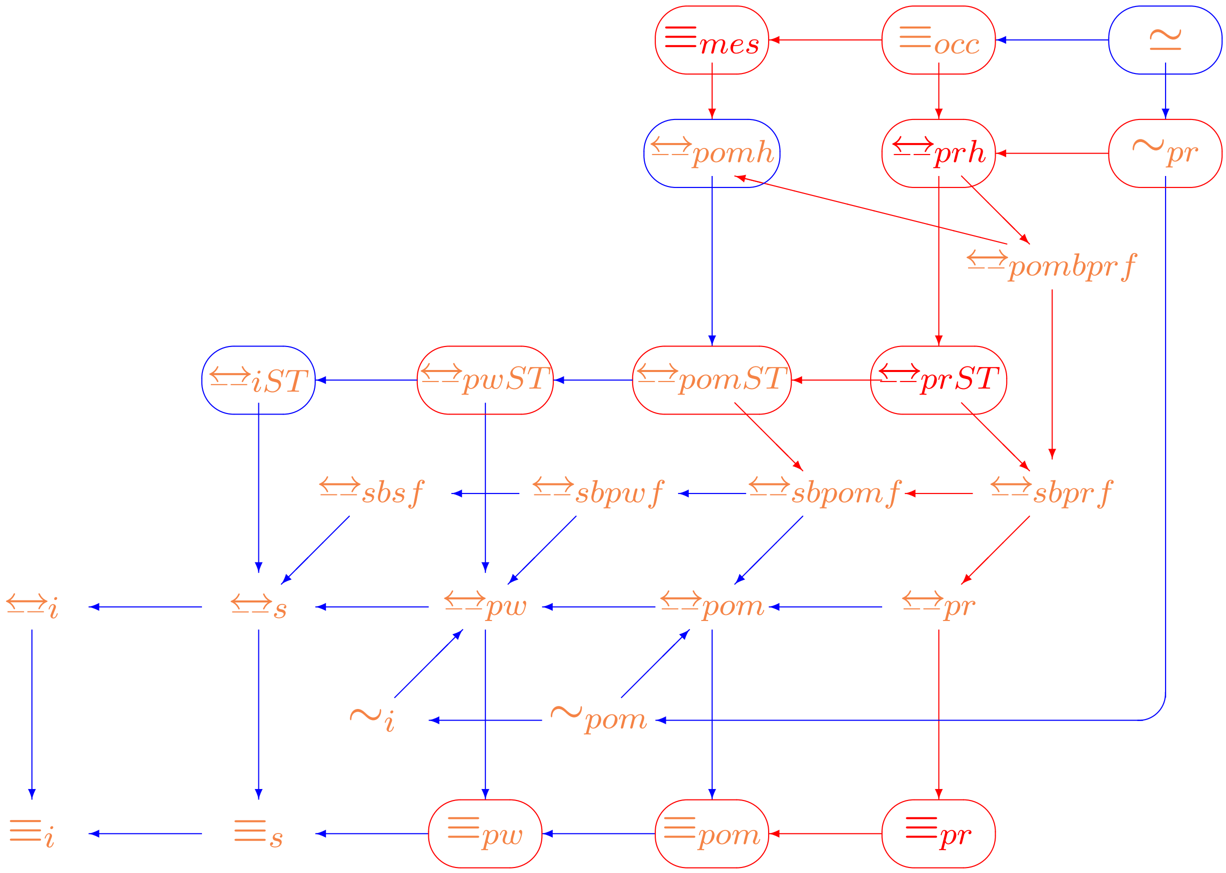
- In Figure UN(b), $N \underline{\leftrightarrow}_{pwST} N'$, but $N \not\equiv_{pom} N'$.

- In Figure B(d), $N \equiv_{mes} N'$, but $N \not\equiv_{pr} N'$.

- In Figure TN(c), $N \equiv_{pr} N'$, but $N \not\underline{\leftrightarrow}_i N'$, since only in the net N' an action a can happen so that no b is possible afterwards.

- In Figure TN(d), $N \underline{\leftrightarrow}_{prh} N'$, but $N \not\equiv_{mes} N'$, since only in the behaviour of N' there is a MES with two conflict actions a.

- In Figure B1(d), $N \equiv_{occ} N'$, but $N \not\simeq N'$.



New results for the equivalences

Decidability results for the equivalences

- $\equiv_i$
  - is decidable for:
    unlabeled (strictly labeled) nets [Jan94];
    finite safe nets (EXPSPACE) [JM96].
  - is undecidable for:
    communication free (BPP) nets [CHM93];
    nets with ≥ 2 unbounded places [Jan94].

- $\equiv_s$
  - is decidable for:
    finite safe nets (EXPSPACE) [JM96].

- $\equiv_{pom}$
  - is decidable for:
    unlabeled (strictly labeled) nets [Jan94];
    finite safe nets (EXPSPACE) [JM96];
    communication free (BPP) nets [CHM93].

- $\underline{\leftrightarrow}_i$
  - is decidable for:
    unlabeled (strictly labeled) nets [Jan94];
    finite safe nets (DEXPTIME) [JM96];
    communication free (BPP) nets [CHM93];
    nets s.t. one of them is deterministic up to bisimilarity [Jan94].
  - is undecidable for:
    nets with ≥ 2 unbounded places [Jan94].

- $\underline{\leftrightarrow}_{s}$
  - is decidable for:
    finite safe nets (DEXPTIME) [JM96].
- $\underline{\leftrightarrow}_{pom}$
  - is decidable for:
    finite safe nets (DEXPTIME / EXPSPACE) [JM96].
- $\underline{\leftrightarrow}_{iST}$
  - is decidable for:
    bounded nets [Dev92];
    finite safe nets (DEXPTIME) [JM96].
- $\underline{\leftrightarrow}_{pomST}$
  - is decidable for:
    finite safe nets (DEXPTIME / EXPSPACE) [JM96].
- $\underline{\leftrightarrow}_{pomh}$
  - is decidable for:
    safe nets (DEXPTIME) [Vog91b].
- $\sim_{i}$
  - is decidable for:
    arbitrary nets (polynomial, $O(|P_N|^2 \times |T_N|^2)$,
    if $\forall t \in T_N \; |^{\bullet}t| + |t^{\bullet}| \leq const$) [AS92].
- $\sim_{pr}$
  - is decidable for:
    arbitrary nets (polynomial, $O(|P_N|^2 \times |T_N|^2)$,
    if $\forall t \in T_N \; |^{\bullet}t| + |t^{\bullet}| \leq const$) [AS92].

Equivalences for Petri Nets with Silent Transitions

Abstract: Behavioural equivalences of concurrent systems modeled by Petri nets with silent transitions are considered.

Known basic τ-equivalences and back-forth τ-bisimulation equivalences are supplemented by new ones.

Their interrelations are examined for the general Petri nets as well as for their subclasses of no silent transitions and sequential nets (no concurrent transitions).

A logical characterization of back-forth τ-equivalences in terms of logics with past modalities is proposed.

A preservation of all the equivalences by refinements is investigated to find out their appropriateness for top-down design.

Keywords: Petri nets with silent transitions, sequential nets, basic τ-equivalences, back-forth τ-bisimulation equivalences, logical characterization, refinement.

Contents

- Back-forth τ-simulation and logics
  - Back-forth τ-bisimulation equivalences
  - Comparing back-forth τ-bisimulation equivalences
  - Comparing back-forth τ-bisimulation equivalences with basic ones
  - Logic BFL
  - Logic $SPBFL$
- Simulation with and without silent actions
  - Interrelations of equivalences with τ-equivalences
- Refinements
  - SM-refinements
- Net subclasses
  - The τ-equivalences on nets without silent transitions
  - The τ-equivalences on sequential nets
- Decidability
  - Decidability results for the τ-equivalences
- Open questions
  - Further research

Previous work

Equivalences which abstract of silent actions are *τ-equivalences* (they are labeled by τ). The following basic τ-equivalences are known:

- *τ-trace equivalences* (respect protocols of behavior):

  interleaving ($\equiv_i^\tau$) [Pom86], step ($\equiv_s^\tau$) [Pom86], partial word ($\equiv_{pw}^\tau$) [Vog91a] and pomset ($\equiv_{pom}^\tau$) [PRS92].

- *Usual τ-bisimulation equivalences* (respect branching structure of behavior):

  interleaving ($\leftrightarrow_i^\tau$) [Mil80], step ($\leftrightarrow_s^\tau$) [Pom86], partial word ($\leftrightarrow_{pw}^\tau$) [Vog91a] and pomset ($\leftrightarrow_{pom}^\tau$) [PRS92].

- *ST-τ-bisimulation equivalences* (respect the duration or maximality of events in behavior):

  interleaving ($\leftrightarrow_{iST}^\tau$) [Vog91a], partial word ($\leftrightarrow_{pwST}^\tau$) [Vog91a] and pomset ($\leftrightarrow_{pomST}^\tau$) [Vog91a].

- *History preserving τ-bisimulation equivalences* (respect the "history" of behavior):

  pomset ($\leftrightarrow_{pomh}^\tau$) [Dev92].

- *History preserving ST-τ-bisimulation equivalences* (respect the "history" and the duration or maximality of events in behavior):

  pomset ($\leftrightarrow_{pomhST}^\tau$) [Dev92].

- *Usual branching τ-bisimulation equivalences* (respect branching structure of behavior with a special care for silent actions):

  interleaving ($\leftrightarrow_{ibr}^\tau$) [Gla93].

- *History preserving branching* τ*-bisimulation equivalences* (respect "history" and branching structure of behavior with a special care for silent actions):

  pomset ($\underline{\leftrightarrow}^{\tau}_{pomhbr}$) [Dev92].

- *Isomorphism* (coincidence up to renaming of components):

  ($\simeq$).

Back-forth bisimulation equivalences: bisimulation relation do not only require simulation in the forward direction but also also when going back in history, i.e. backward. They connected with equivalences of logics with past modalities.

Interleaving *back* interleaving *forth* τ*-bisimulation equivalence* ($\underline{\leftrightarrow}^{\tau}_{ibif} = \underline{\leftrightarrow}^{\tau}_{ibr}$) [NMV90].

Pomset *back* pomset *forth* τ*-bisimulation equivalence* ($\underline{\leftrightarrow}^{\tau}_{pombpomf} = \underline{\leftrightarrow}^{\tau}_{pomhbr}$) [Pin93].

New τ-equivalences

- Basic τ-equivalences:

  interleaving *ST-branching τ-bisimulation* ($\underline{\leftrightarrow}^{\tau}_{iSTbr}$),

  pomset *history preserving ST-branching τ-bisimulation* ($\underline{\leftrightarrow}^{\tau}_{pomhSTbr}$) and

  multi event structure ($\equiv^{\tau}_{mes}$).

- Back-forth τ-bisimulation equivalences:

  interleaving *back* step *forth* ($\underline{\leftrightarrow}^{\tau}_{ibsf}$),

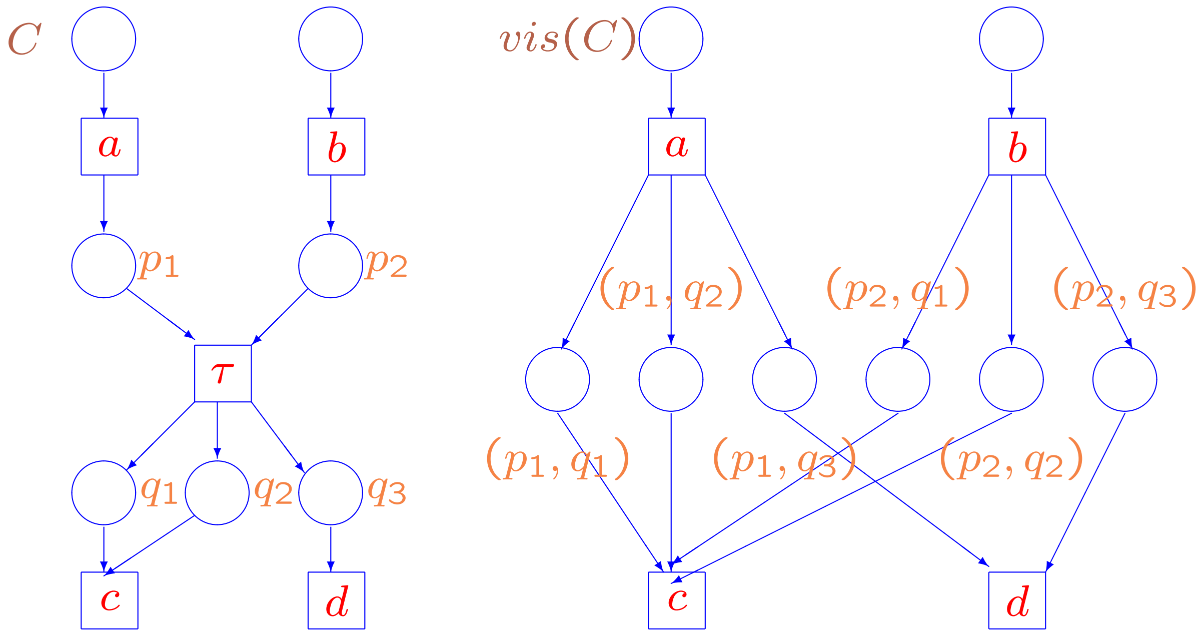
  interleaving *back* partial word *forth* ($\underline{\leftrightarrow}^{\tau}_{ibpwf}$),

  interleaving *back* pomset *forth* ($\underline{\leftrightarrow}^{\tau}_{ibpomf}$),

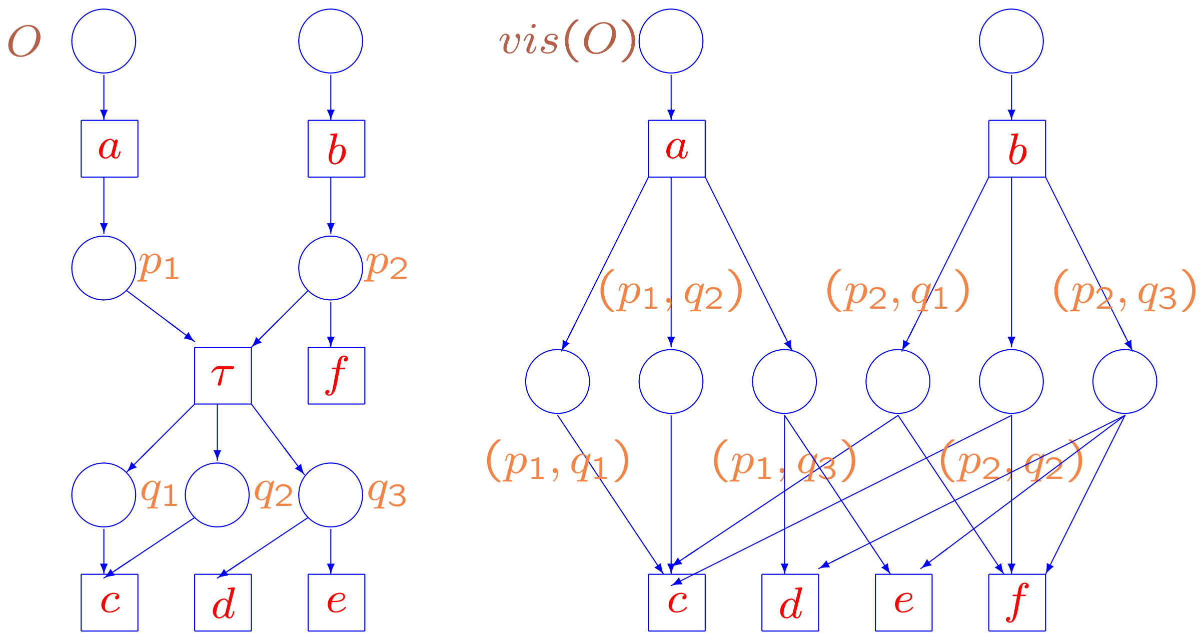
  step *back* step *forth* ($\underline{\leftrightarrow}^{\tau}_{sbsf}$),

  step *back* partial word *forth* ($\underline{\leftrightarrow}^{\tau}_{sbpwf}$) and

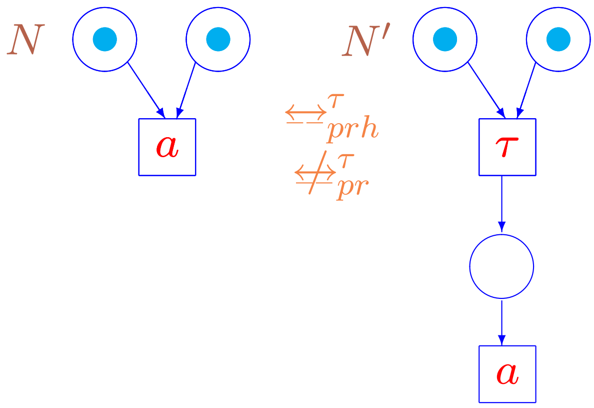
  step *back* pomset *forth* ($\underline{\leftrightarrow}^{\tau}_{sbpomf}$).



An application of the mapping vis to a causal net



An application of the mapping vis to an occurrence net



A crash of interrelations of the process τ-bisimulation equivalences comparing with that of the process bisimulation equivalences

τ-trace equivalences

The empty string is denoted by ε.

Let $\sigma = a_1 \cdots a_n \in Act_\tau^*$ and $a \in Act_\tau$. We define $vis(\sigma)$ as:

1. $vis(\varepsilon) = \varepsilon$;

2. $vis(\sigma a) = \begin{cases} vis(\sigma)a, & a \neq \tau; \\ vis(\sigma), & a = \tau. \end{cases}$

Definition 52 *A visible interleaving trace of a net N is a sequence $vis(a_1 \cdots a_n) \in Act^*$ s.t. $\pi_N \xrightarrow{a_1} \pi_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} \pi_n,\ \pi_i \in \Pi(N)\ (1 \leq i \leq n)$.*

The set of all visible interleaving traces of N is $VisIntTraces(N)$.

N and N' are interleaving τ-trace equivalent, $N \equiv_i^\tau N'$, if $VisIntTraces(N) = VisIntTraces(N')$.

Let $\Sigma = A_1 \cdots A_n \in (\mathcal{M}(Act_\tau))^*$ and $A \in \mathcal{M}(Act_\tau)$. We define $vis(\Sigma)$ as:

1. $vis(\varepsilon) = \varepsilon$;

2. $vis(\Sigma A) = \begin{cases} vis(\Sigma)(A \cap Act), & A \cap Act \neq \emptyset; \\ vis(\Sigma), & \text{otherwise}. \end{cases}$

Definition 53 *A visible step trace of a net N is a sequence $vis(A_1 \cdots A_n) \in (\mathcal{M}(Act))^*$ s.t. $\pi_N \xrightarrow{A_1} \pi_1 \xrightarrow{A_2} \ldots \xrightarrow{A_n} \pi_n,\ \pi_i \in \Pi(N)\ (1 \leq i \leq n)$.*

The set of all visible step traces of N is $VisStepTraces(N)$.

N and N' are step τ-trace equivalent, $N \equiv_s^\tau N'$, if $VisStepTraces(N) = VisStepTraces(N')$.

Let $\rho = \langle X, \prec, l \rangle$ is lposet s.t. $l : X \to Act_\tau$. We denote:

- $vis(X) = \{x \in X \mid l(x) \in Act\}$;
- $vis(\rho) = \rho|_{vis(X)}$.

Definition 54 *A visible pomset trace of a net N is a pomset $vis(\rho)$, an isomorphism class of lposet $vis(\rho_C)$ for $\pi = (C, \varphi) \in \Pi(N)$.*

The set of all visible pomset traces of N is $VisPomsets(N)$.

N and N' are partial word τ-trace equivalent, $N \equiv^\tau_{pw} N'$, if $VisPomsets(N) \sqsubseteq VisPomsets(N')$ and $VisPomsets(N') \sqsubseteq VisPomsets(N)$.

Definition 55 *N and N' are pomset τ-trace equivalent, $N \equiv^\tau_{pom} N'$, if $VisPomsets(N) = VisPomsets(N')$.*

Usual τ-bisimulation equivalences

Let $C = \langle P_C, T_C, F_C, l_C\rangle$ be causal net. We denote:

- $vis(T_C) = \{v \in T_C \mid l_C(v) \in Act\}$;
- $vis(\prec_C) = \prec_C \cap (vis(T_C) \times vis(T_C))$.

Definition 56 $\mathcal{R} \subseteq \Pi(N) \times \Pi(N')$ *is a* $\star$-τ-bisimulation *between nets* N *and* N', $\star \in$\{interleaving, step, partial word, pomset\}, $\mathcal{R} : N \underline{\leftrightarrow}_{\star}^{\tau} N'$, $\star \in \{i, s, pw, pom\}$, *if:*

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

2. $(\pi, \pi') \in \mathcal{R}$, $\pi \xrightarrow{\widehat{\pi}} \tilde{\pi}$,

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

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

   $\Rightarrow \exists \tilde{\pi}' : \pi' \xrightarrow{\widehat{\pi}'} \tilde{\pi}'$, $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$ *and*

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

   (b) $vis(\rho_{\widehat{C}}) \simeq vis(\rho_{\widehat{C}'})$, *if* $\star \in \{i, s, pom\}$.

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

N *and* N' *are* $\star$-τ-bisimulation equivalent, $\star \in$\{interleaving, step, partial word, pomset\}, $N \underline{\leftrightarrow}_{\star}^{\tau} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}_{\star}^{\tau} N'$, $\star \in \{i, s, pw, pom\}$.

ST-τ-bisimulation equivalences

Definition 57 *An* ST-τ-process *of a net* N *is a pair* (π_E, π_P)*:*

1. $\pi_E, \pi_P \in \Pi(N)$, $\pi_P \stackrel{\pi_W}{\to} \pi_E$;
2. $\forall v, w \in T_{C_E}\ (v \prec_{C_E} w) \vee (l_{C_E}(v) = \tau) \Rightarrow v \in T_{C_P}$.

- π_E is a current process;
- π_P is the completed part;
- π_W is the still working part.

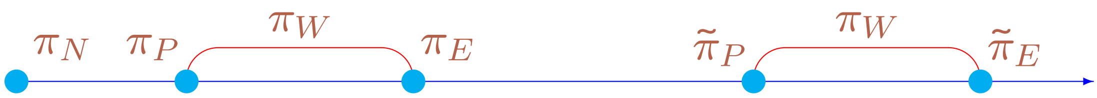
Obviously, $\prec_{C_W} = \emptyset$.

$ST^\tau - \Pi(N)$ is the set of *all ST-τ-processes* of a net N.

(π_N, π_N) is the *initial ST-τ-process* of a net N.

Let (π_E, π_P), $(\tilde{\pi}_E, \tilde{\pi}_P) \in ST^\tau - \Pi(N)$.

We write $(\pi_E, \pi_P) \to (\tilde{\pi}_E, \tilde{\pi}_P)$, if $\pi_E \to \tilde{\pi}_E$ and $\pi_P \to \tilde{\pi}_P$.

$\Pi(N)$ 

ST-τ-processes

Definition 58 $\mathcal{R} \subseteq ST^\tau - \Pi(N) \times ST^\tau - \Pi(N') \times \mathcal{B}$, *where* $\mathcal{B} = \{\beta \mid \beta : vis(T_C) \to vis(T_{C'}),\ \pi = (C, \varphi) \in \Pi(N),$ $\pi' = (C', \varphi') \in \Pi(N')\}$ *is a* $\star$-ST-τ-bisimulation *between nets* N *and* N', $\star \in$ {interleaving, partial word, pomset}, $\mathcal{R} : N \underline{\leftrightarrow}^\tau_{\star ST} N'$, $\star \in \{i, pw, pom\}$, *if:*

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

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

3. $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R},\ (\pi_E, \pi_P) \longrightarrow (\tilde{\pi}_E, \tilde{\pi}_P) \ \Rightarrow$ $\exists \tilde{\beta},\ (\tilde{\pi}'_E, \tilde{\pi}'_P) :\ (\pi'_E, \pi'_P) \to (\tilde{\pi}'_E, \tilde{\pi}'_P),\ \tilde{\beta}|_{vis(T_{C_E})} = \beta,$ $((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$, *and if* $\pi_P \xrightarrow{\pi} \tilde{\pi}_E$, $\pi'_P \xrightarrow{\pi'} \tilde{\pi}'_E,\ \gamma = \tilde{\beta}|_{vis(T_C)}$, *then:*

   (a) $\gamma^{-1} : vis(\rho_{C'}) \sqsubseteq vis(\rho_C)$, *if* $\star = pw$;

   (b) $\gamma : vis(\rho_C) \simeq vis(\rho_{C'})$, *if* $\star = pom$.

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

N *and* N' *are* $\star$-ST-τ-bisimulation equivalent, $\star \in$ {interleaving, partial word, pomset}, $N \underline{\leftrightarrow}^\tau_{\star ST} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}^\tau_{\star ST} N'$, $\star \in \{i, pw, pom\}$.

History preserving τ-bisimulation equivalences

Definition 59 $\mathcal{R} \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$, *where* $\mathcal{B} = \{\beta \mid \beta : vis(T_C) \to vis(T_{C'}),\ \pi = (C, \varphi) \in \Pi(N),$ $\pi' = (C', \varphi') \in \Pi(N')\}$, *is a* pomset history preserving τ-bisimulation *between nets* N *and* N', $\mathcal{R} : N \underline{\leftrightarrow}^{\tau}_{pomh} N'$, *if:*

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

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

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

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

N *and* N' *are* pomset history preserving τ-bisimulation equivalent, $N \underline{\leftrightarrow}^{\tau}_{pomh} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}^{\tau}_{pomh} N'$.

History preserving ST-τ-bisimulation equivalences

Definition 60 $\mathcal{R} \subseteq ST^\tau - \Pi(N) \times ST^\tau - \Pi(N') \times \mathcal{B}$, *where* $\mathcal{B} = \{\beta \mid \beta : vis(T_C) \to vis(T_{C'}),\ \pi = (C, \varphi) \in \Pi(N),$ $\pi' = (C', \varphi') \in \Pi(N')\}$, *is a* pomset history preserving ST-τ-bisimulation *between nets* N *and* N', $\mathcal{R} : N \underline{\leftrightarrow}^\tau_{pomhST} N'$, *if:*

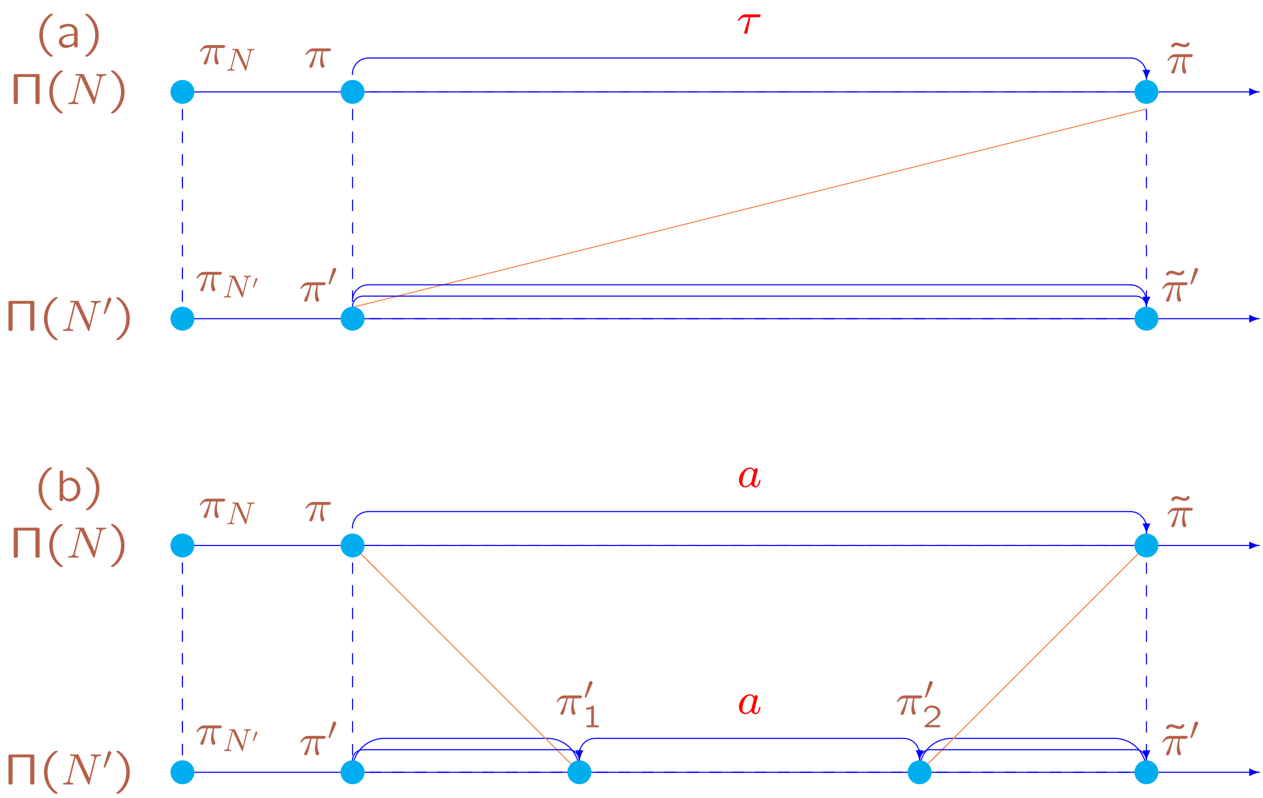
1. $((\pi_N, \pi_N), (\pi_{N'}, \pi_{N'}), \emptyset) \in \mathcal{R}$.

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

3. $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R},\ (\pi_E, \pi_P) \to (\tilde{\pi}_E, \tilde{\pi}_P) \ \Rightarrow$ $\exists \tilde{\beta},\ (\tilde{\pi}'_E, \tilde{\pi}'_P) :\ (\pi'_E, \pi'_P) \to (\tilde{\pi}'_E, \tilde{\pi}'_P),\ \tilde{\beta}|_{vis(T_{C_E})} = \beta,$ $((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$.

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

N *and* N' *are* pomset history preserving ST-τ-bisimulation equivalent, $N \underline{\leftrightarrow}^\tau_{pomhST} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}^\tau_{pomhST} N'$.



A distinguish ability of the usual and the branching τ-bisimulation equivalences

Usual branching τ-bisimulation equivalences

For a net N and $\pi, \tilde{\pi} \in \Pi(N)$ we write $\pi \Rightarrow \tilde{\pi}$ when $\exists \widehat{\pi} = (\widehat{C}, \widehat{\varphi})$ s.t. $\pi \xrightarrow{\widehat{\pi}} \tilde{\pi}$ and $vis(T_{\widehat{C}}) = \emptyset$.

Definition 61 $\mathcal{R} \subseteq \Pi(N) \times \Pi(N')$ *is an* interleaving branching τ-bisimulation *between nets* N *and* N', $\mathcal{R} : N \underline{\leftrightarrow}^{\tau}_{ibr} N'$, *if:*

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

2. $(\pi, \pi') \in \mathcal{R}$, $\pi \xrightarrow{a} \tilde{\pi} \;\Rightarrow$

   (a) $a = \tau$ *and* $(\tilde{\pi}, \pi') \in \mathcal{R}$ *or*

   (b) $a \neq \tau$ *and* $\exists \bar{\pi}', \tilde{\pi}' : \pi' \Rightarrow \bar{\pi}' \xrightarrow{a} \tilde{\pi}'$, $(\pi, \bar{\pi}') \in \mathcal{R}$, $(\tilde{\pi}, \tilde{\pi}') \in \mathcal{R}$.

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

N *and* N' *are* interleaving branching τ-bisimulation equivalent, $N \underline{\leftrightarrow}^{\tau}_{ibr} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}^{\tau}_{ibr} N'$.

History preserving branching τ-bisimulation equivalences

Definition 62 $\mathcal{R} \subseteq \Pi(N) \times \Pi(N') \times \mathcal{B}$, *where* $\mathcal{B} = \{\beta \mid \beta : T_C \to T_{C'},\ \pi = (C, \varphi) \in \Pi(N),$ $\pi' = (C', \varphi') \in \Pi(N')\}$, *is a* pomset history preserving branching τ-bisimulation *between nets* N *and* N', $\mathcal{R} : N \underline{\leftrightarrow}^{\tau}_{pomhbr} N'$, *if:*

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

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

3. $(\pi, \pi', \beta) \in \mathcal{R},\ \pi \to \tilde{\pi} \ \Rightarrow$

   (a) $(\tilde{\pi}, \pi', \beta) \in \mathcal{R}$ *or*

   (b) $\exists \tilde{\beta},\ \bar{\pi}',\ \tilde{\pi}' : \pi' \Rightarrow \bar{\pi}' \to \tilde{\pi}',\ \tilde{\beta}|_{vis(T_C)} = \beta,$
   $(\pi, \bar{\pi}', \beta) \in \mathcal{R},\ (\tilde{\pi}, \tilde{\pi}', \tilde{\beta}) \in \mathcal{R}$.

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

N *and* N' *are* pomset history preserving branching τ-bisimulation equivalent, $N \underline{\leftrightarrow}^{\tau}_{pomhbr} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}^{\tau}_{pomhbr} N'$.

ST-branching τ-bisimulation equivalences

Let $(\pi_E, \pi_P),\ (\tilde{\pi}_E, \tilde{\pi}_P) \in ST^\tau - \Pi(N)$. We write $(\pi_E, \pi_P) \Rightarrow (\tilde{\pi}_E, \tilde{\pi}_P)$, if $\pi_E \Rightarrow \tilde{\pi}_E$ and $\pi_P \Rightarrow \tilde{\pi}_P$.

Definition 63 $\mathcal{R} \subseteq ST^\tau - \Pi(N) \times ST^\tau - \Pi(N') \times \mathcal{B}$, *where* $\mathcal{B} = \{\beta \mid \beta : vis(T_C) \to vis(T_{C'}),\ \pi = (C, \varphi) \in \Pi(N),\ \pi' = (C', \varphi') \in \Pi(N')\}$ *is an* interleaving ST-branching τ-bisimulation *between nets* N *and* N', $\mathcal{R} : N \underline{\leftrightarrow}^\tau_{iSTbr} N'$, *if:*

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

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

3. $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R},\ (\pi_E, \pi_P) \to (\tilde{\pi}_E, \tilde{\pi}_P) \ \Rightarrow$

   (a) $((\tilde{\pi}_E, \tilde{\pi}_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R}$ *or*

   (b) $\exists \tilde{\beta},\ (\bar{\pi}'_E, \bar{\pi}'_P),\ (\tilde{\pi}'_E, \tilde{\pi}'_P) : \ (\pi'_E, \pi'_P) \Rightarrow (\bar{\pi}'_E, \bar{\pi}'_P) \to (\tilde{\pi}'_E, \tilde{\pi}'_P),\ \tilde{\beta}|_{vis(T_{C_E})} = \beta,\ ((\pi_E, \pi_P), (\bar{\pi}'_E, \bar{\pi}'_P), \beta) \in \mathcal{R},\ ((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$.

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

N *and* N' *are* interleaving ST-branching τ-bisimulation equivalent, $N \underline{\leftrightarrow}^\tau_{iSTbr} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}^\tau_{iSTbr} N'$.

History preserving ST-branching τ-bisimulation equivalences

Definition 64 $\mathcal{R} \subseteq ST^\tau - \Pi(N) \times ST^\tau - \Pi(N') \times \mathcal{B}$, *where* $\mathcal{B} = \{\beta \mid \beta : vis(T_C) \to vis(T_{C'}),\ \pi = (C, \varphi) \in \Pi(N),$ $\pi' = (C', \varphi') \in \Pi(N')\}$ *is a* pomset history preserving ST-branching τ-bisimulation *between nets* N *and* N', $\mathcal{R} : N \underline{\leftrightarrow}^\tau_{pomhSTbr} N'$, *if:*

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

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

3. $((\pi_E, \pi_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R},\ (\pi_E, \pi_P) \to (\tilde{\pi}_E, \tilde{\pi}_P) \ \Rightarrow$

   (a) $((\tilde{\pi}_E, \tilde{\pi}_P), (\pi'_E, \pi'_P), \beta) \in \mathcal{R}$ *or*

   (b) $\exists \tilde{\beta},\ (\bar{\pi}'_E, \bar{\pi}'_P),\ (\tilde{\pi}'_E, \tilde{\pi}'_P) : \ (\pi'_E, \pi'_P) \Rightarrow (\bar{\pi}'_E, \bar{\pi}'_P) \to (\tilde{\pi}'_E, \tilde{\pi}'_P),\ \tilde{\beta}|_{vis(T_{C_E})} = \beta,\ ((\pi_E, \pi_P), (\bar{\pi}'_E, \bar{\pi}'_P), \beta) \in \mathcal{R},$ $((\tilde{\pi}_E, \tilde{\pi}_P), (\tilde{\pi}'_E, \tilde{\pi}'_P), \tilde{\beta}) \in \mathcal{R}$.

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

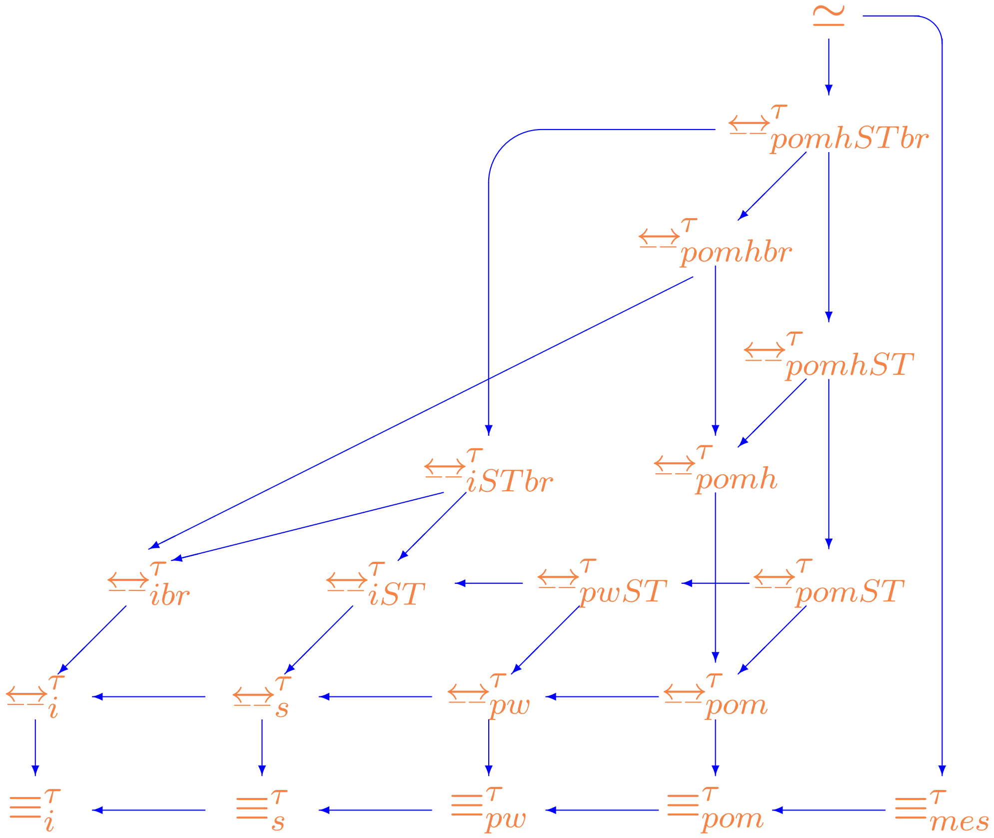
N *and* N' *are* pomset history preserving ST-branching τ-bisimulation equivalent, $N \underline{\leftrightarrow}^\tau_{pomhSTbr} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}^\tau_{pomhSTbr} N'$.

Conflict preserving τ-equivalences

Let $\xi = \langle X, \prec, \#, l \rangle$ be a LES s.t. $l : X \rightarrow Act_\tau$. We denote $vis(X) = \{x \in X \mid l(x) \in Act\}$ and $vis(\xi) = \xi|_{vis(X)}$.

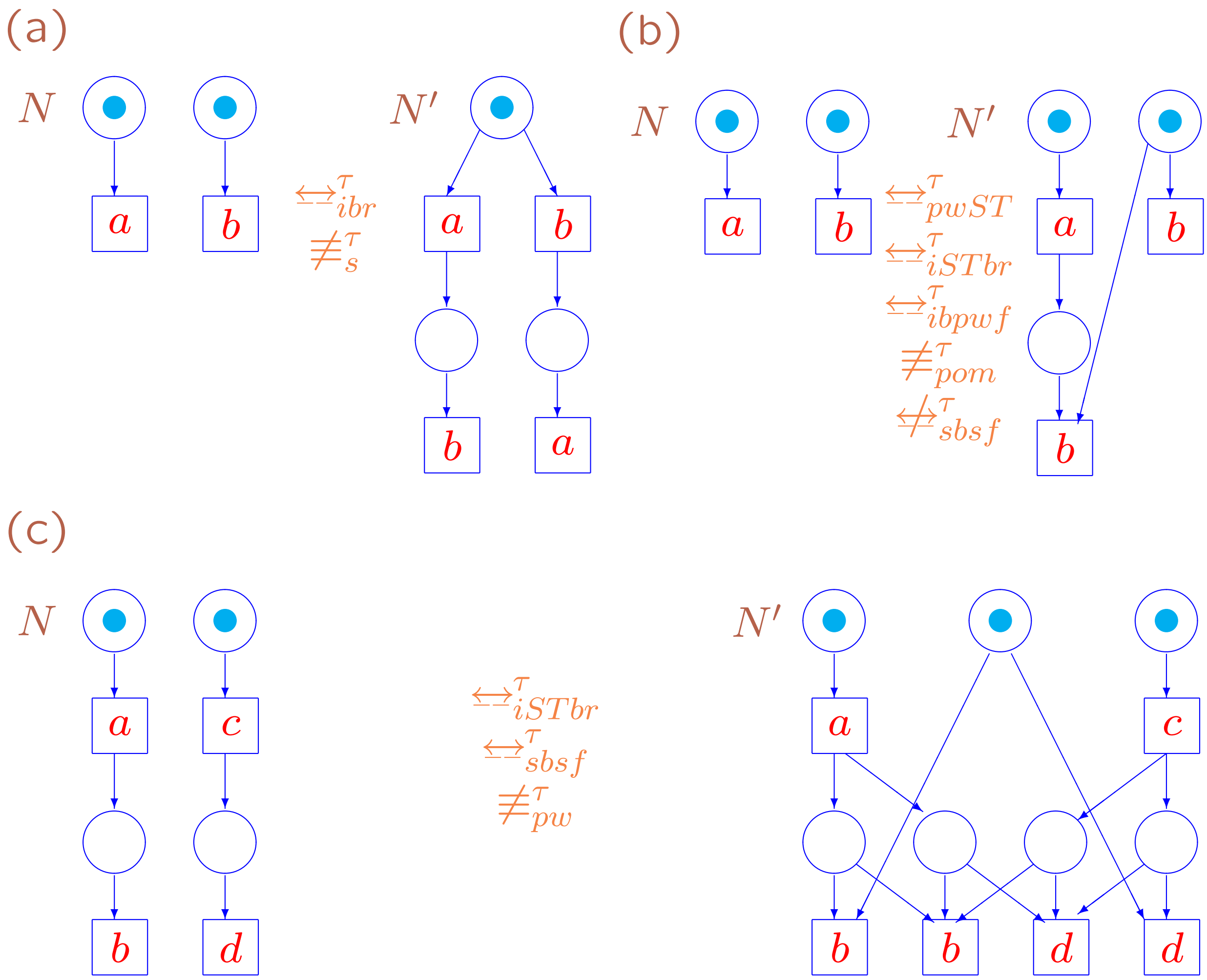
Definition 65 *N and N' are MES-τ-conflict preserving equivalent, $N \equiv^{\tau}_{mes} N'$, if $vis(\mathcal{E}(N)) = vis(\mathcal{E}(N'))$.*

Comparing basic τ-equivalences

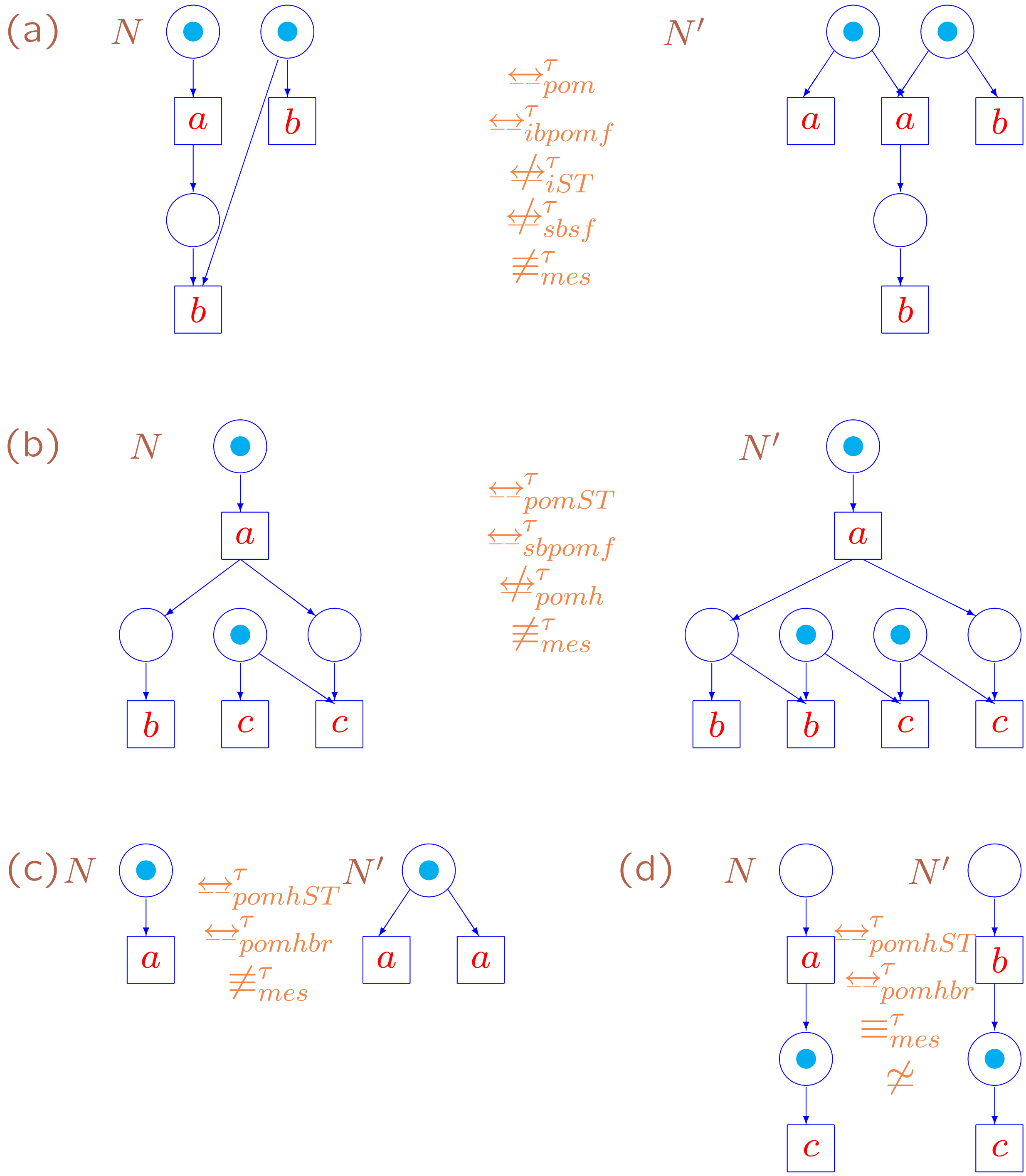


Interrelations of basic τ-equivalences

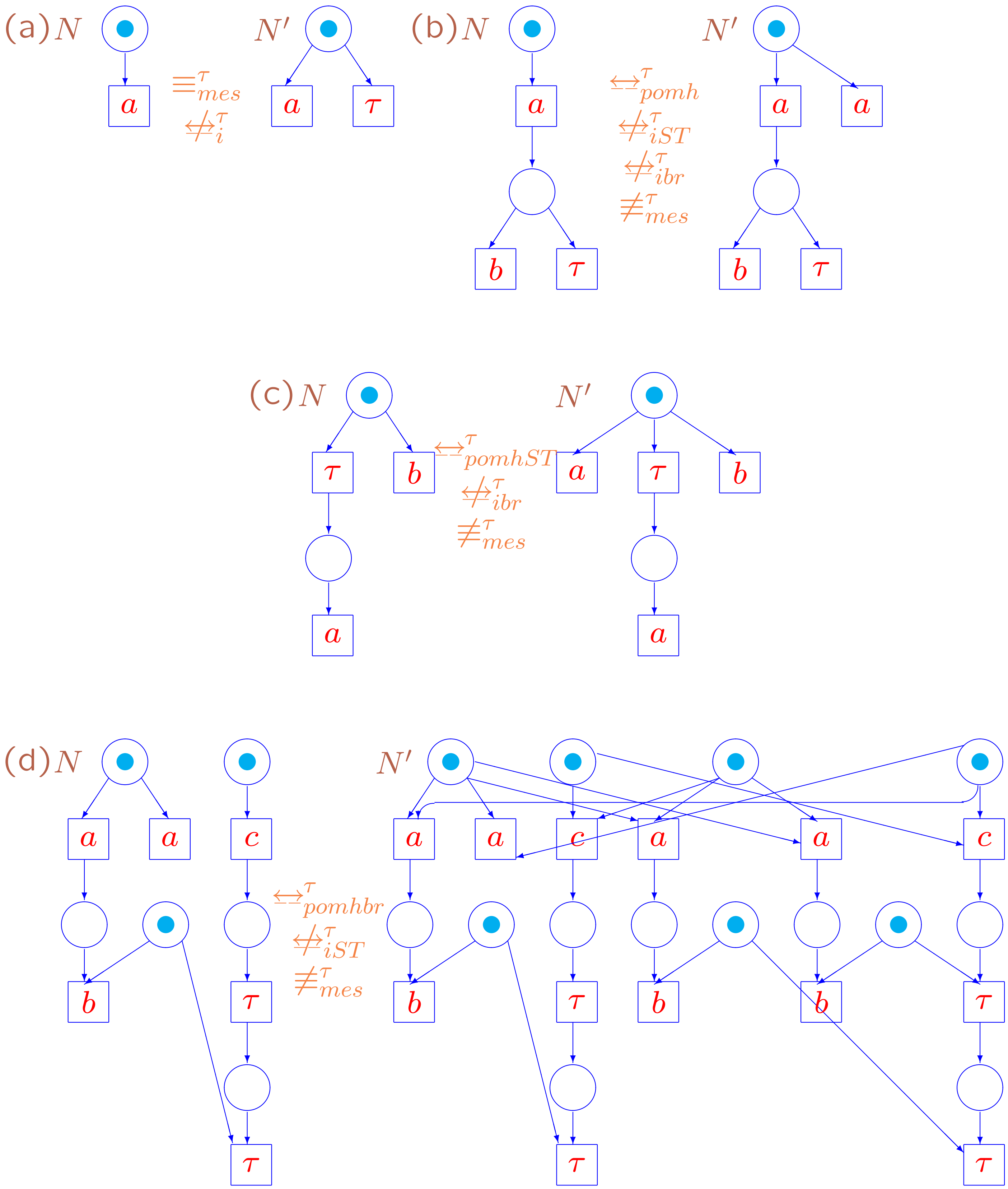
Theorem 11 *Let* $\leftrightarrow, \twoheadleftarrow\!\!\!\twoheadrightarrow \in \{\equiv^\tau, \underline{\leftrightarrow}^\tau, \simeq\}$, $\star, \star\star \in \{\_, i, s, pw, pom, iST, pwST, pomST, pomh, pomhST, ibr, pomhbr, iSTbr, pomhSTbr, mes\}$. *For nets* N *and* N' $N \leftrightarrow_\star N' \Rightarrow N \twoheadleftarrow\!\!\!\twoheadrightarrow_{\star\star} N'$ *iff in the graph above there exists a directed path from* $\leftrightarrow_\star$ *to* $\twoheadleftarrow\!\!\!\twoheadrightarrow_{\star\star}$.



BT: Examples of basic τ-equivalences



BT1: Examples of basic τ-equivalences (continued)



BT2: Examples of basic τ-equivalences (continued 2)

- In Figure BT(a), $N \underline{\leftrightarrow}_{ibr}^{\tau} N'$, but $N \not\equiv_{s}^{\tau} N'$, since only in the net N' actions a and b cannot happen concurrently.

- In Figure BT(c), $N \underline{\leftrightarrow}_{iSTbr}^{\tau} N'$, but $N \not\equiv_{pw}^{\tau} N'$, since for the pomset corresponding to the net N there is no even less sequential pomset in N'.

- In Figure BT(b), $N \underline{\leftrightarrow}_{pwST}^{\tau} N'$, but $N \not\equiv_{pom}^{\tau} N'$, since only in the net N' action b can depend on a.

- In Figure BT2(a), $N \equiv_{mes}^{\tau} N'$, but $N \not\underline{\leftrightarrow}_{i}^{\tau} N'$, since only in the net N' action τ can happen so that in the corresponding initial state of the net N action a cannot happen.

- In Figure BT1(a), $N \underline{\leftrightarrow}_{pom}^{\tau} N'$, but $N \not\underline{\leftrightarrow}_{iST}^{\tau} N'$, since only in the net N' action a can start so that no action b can begin to work until finishing a.

- In Figure BT1(b), $N \underline{\leftrightarrow}_{pomST}^{\tau} N'$, but $N \not\underline{\leftrightarrow}_{pomh}^{\tau} N'$, since only in the net N' after action a action b can happen so that action c must depend on a.

- In Figure BT2(b), $N \underline{\leftrightarrow}_{pomh}^{\tau} N'$, but $N \not\underline{\leftrightarrow}_{iST}^{\tau} N'$, since only in the net N' action a can start so that the action b can never occur.

- In Figure BT2(c), $N \underline{\leftrightarrow}_{pomhST}^{\tau} N'$, but $N \not\underline{\leftrightarrow}_{ibr}^{\tau} N'$, since in the net N' an action a can happen so that it will be simulated by sequence of actions τa in N. Then the state of the net N reached after τ must be related with the initial state of a net N, but in such a case the occurrence of action b from the initial state of N' cannot be imitated from the corresponding state of N.

- In Figure BT2(d), $N \underline{\leftrightarrow}^{\tau}_{pomhbr} N'$, but $N \not\underline{\leftrightarrow}^{\tau}_{iST} N'$, since in the net N' an action c may start so that during work of the corresponding action c in the net N an action a may happen in such a way that the action b never occur.

- In Figure BT1(c), $N \underline{\leftrightarrow}^{\tau}_{pomhSTbr} N'$, but $N \not\equiv^{\tau}_{mes} N'$, since only the MES corresponding to the net N' has two conflict actions a.

- In Figure BT1(d), $N \equiv^{\tau}_{mes} N'$, but $N \not\simeq N'$, since unfireable transitions of the nets N and N' are labeled by different actions (a and b).

Back-forth τ-bisimulation equivalences

Definition 66 $\mathcal{R} \subseteq Runs(N) \times Runs(N')$ *is a* $\star$-back $\star\star$-forth τ-bisimulation *between nets* N *and* N', $\star, \star\star \in$ {interleaving, step, partial word, pomset}, $\mathcal{R} : N \underline{\leftrightarrow}^{\tau}_{\star b \star\star f} N'$, $\star, \star\star \in \{i, s, pw, pom\}$, *if:*

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

2. $((\pi, \sigma), (\pi', \sigma')) \in \mathcal{R}$

   - *(back)* $(\tilde{\pi}, \tilde{\sigma}) \xrightarrow{\widehat{\pi}} (\pi, \sigma)$,

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

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

     $\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') : (\tilde{\pi}', \tilde{\sigma}') \xrightarrow{\widehat{\pi}'} (\pi', \sigma')$,
     $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ *and*

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

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

   - *(forth)* $(\pi, \sigma) \xrightarrow{\widehat{\pi}} (\tilde{\pi}, \tilde{\sigma})$,

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

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

     $\Rightarrow \exists (\tilde{\pi}', \tilde{\sigma}') : (\pi', \sigma') \xrightarrow{\widehat{\pi}'} (\tilde{\pi}', \tilde{\sigma}')$,
     $((\tilde{\pi}, \tilde{\sigma}), (\tilde{\pi}', \tilde{\sigma}')) \in \mathcal{R}$ *and*

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

     (b) $vis(\rho_{\widehat{C}}) \simeq vis(\rho_{\widehat{C}'})$, *if* $\star\star \in \{i, s, pom\}$.

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

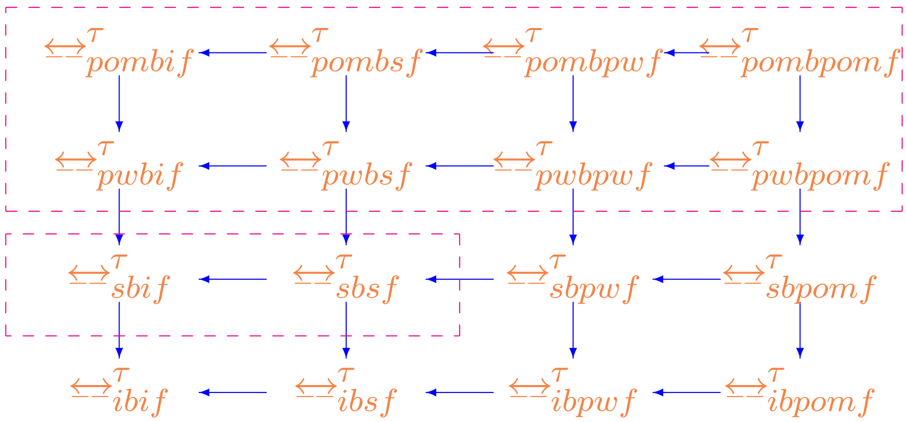
N *and* N' *are* $\star$-back $\star\star$-forth τ-bisimulation equivalent, $\star, \star\star \in$ {interleaving, step, partial word, pomset}, $N \underline{\leftrightarrow}^{\tau}_{\star b \star\star f} N'$, *if* $\exists \mathcal{R} : N \underline{\leftrightarrow}^{\tau}_{\star b \star\star f} N'$, $\star, \star\star \in \{i, s, pw, pom\}$.

Comparing back-forth τ-bisimulation equivalences

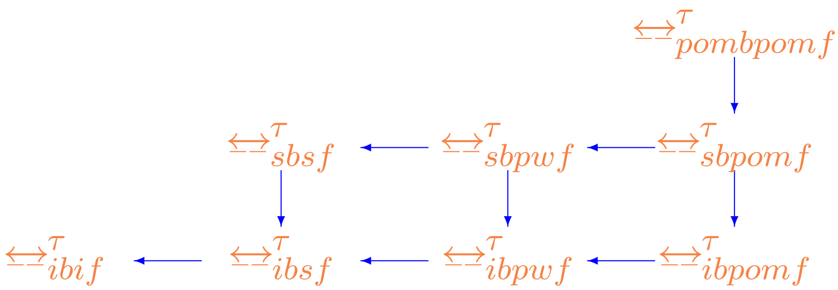
Proposition 11 [Pin93, Tar97] *Let $\star \in \{i, s, pw, pom\}$. For nets N and N':*

1. $N \underline{\leftrightarrow}^{\tau}_{pwb\star f} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}^{\tau}_{pomb\star f} N'$;

2. $N \underline{\leftrightarrow}^{\tau}_{\star bif} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}^{\tau}_{\star b\star f} N'$.



Merging of back-forth τ-bisimulation equivalences



Interrelations of back-forth τ-bisimulation equivalences

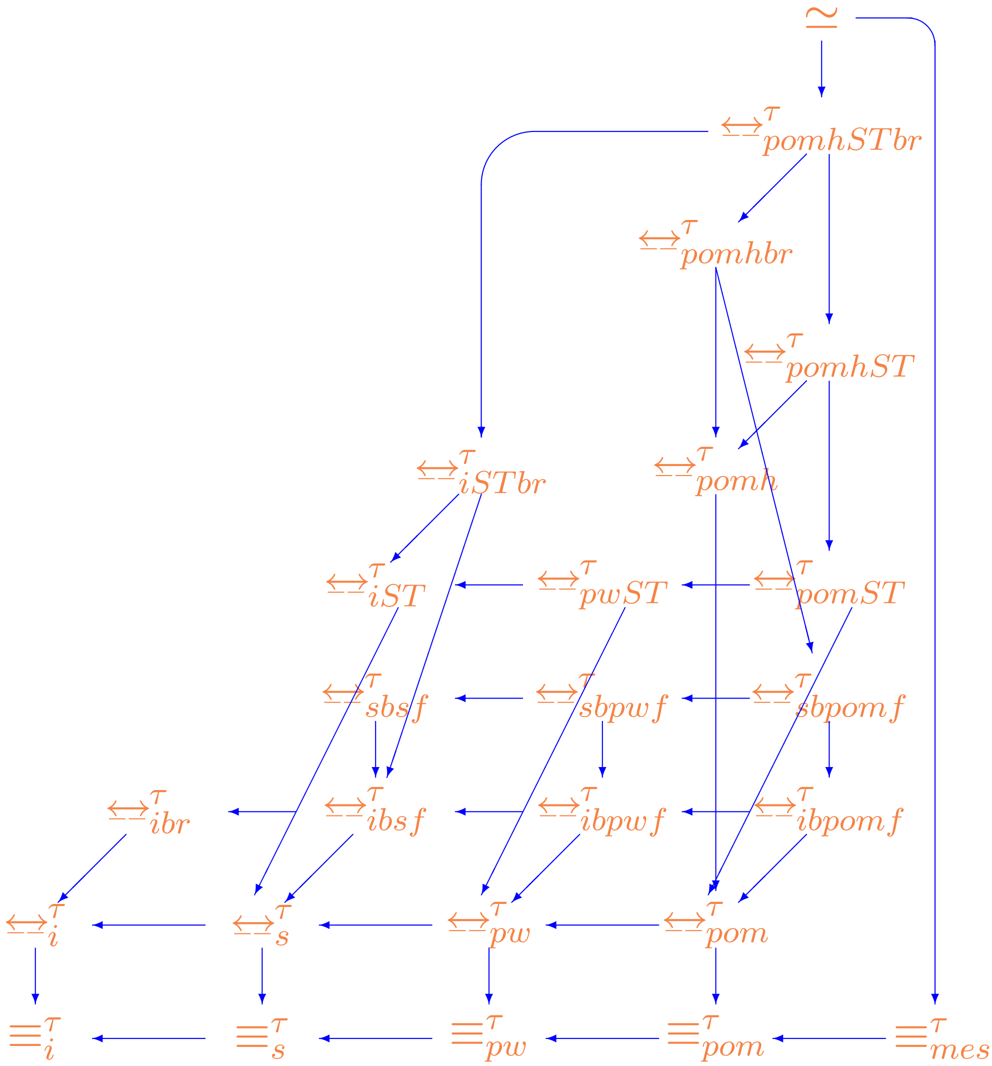
Comparing back-forth τ-bisimulation equivalences with basic ones

Proposition **12** *For nets N and N':*

1. $N \underline{\leftrightarrow}^{\tau}_{ibif} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}^{\tau}_{ibr} N'$ [Gla93];

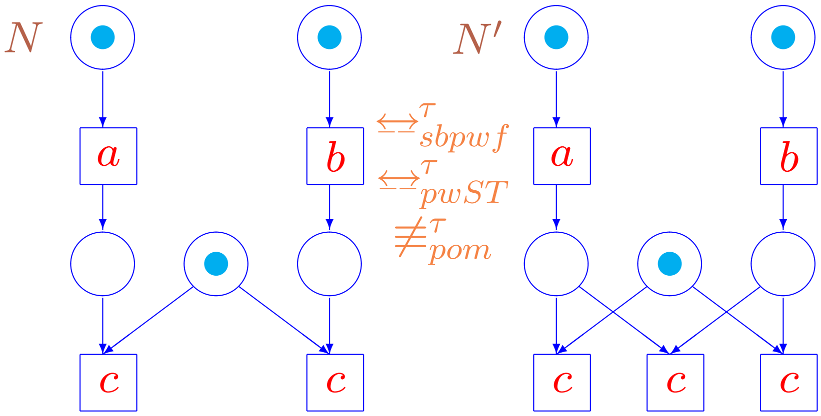
2. $N \underline{\leftrightarrow}^{\tau}_{pombpomf} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}^{\tau}_{pomhbr} N'$ [Pin93];

3. $N \underline{\leftrightarrow}^{\tau}_{iSTbr} N' \;\Rightarrow\; N \underline{\leftrightarrow}^{\tau}_{ibsf} N'$ [Tar97].



Interrelations of back-forth τ-bisimulation equivalences with basic ones

Theorem **12** *Let* $\leftrightarrow, \leftrightarrow\!\!\!\!\leftrightarrow \in \{\equiv^{\tau}, \underline{\leftrightarrow}^{\tau}, \simeq\}$ *and* $\star, \star\star \in \{\_, i, s, pw, pom, iST, pwST, pomST, pomh, pomhST, ibr, iSTbr, pomhSTbr, pomhbr, mes, ibsf, ibpwf, ibpomf, sbsf, sbpwf, sbpomf\}$. *For nets N and N'* $N \leftrightarrow_{\star} N' \;\Rightarrow\; N \leftrightarrow\!\!\!\!\leftrightarrow_{\star\star} N'$ *iff in the graph above there exists a directed path from* $\leftrightarrow_{\star}$ *to* $\leftrightarrow\!\!\!\!\leftrightarrow_{\star\star}$.



BFT: Example of back-forth τ-bisimulation equivalences

- In Figure BT(c), $N \underline{\leftrightarrow}^{\tau}_{sbsf} N'$, but $N \not\equiv^{\tau}_{pw} N'$.
- In Figure BFT, $N \underline{\leftrightarrow}^{\tau}_{sbpwf} N'$, but $N \not\equiv^{\tau}_{pom} N'$.
- In Figure BT1(a), $N \underline{\leftrightarrow}^{\tau}_{ibpomf} N'$, but $N \underline{\not\leftrightarrow}^{\tau}_{sbsf} N'$.
- In Figure BT(b), $N \underline{\leftrightarrow}^{\tau}_{iSTbr} N'$, but $N \underline{\not\leftrightarrow}^{\tau}_{sbsf} N'$.

Logic BFL [NMV90]

Definition 67 $\top$ *denotes the truth,* $a \in Act$.

A formula *of* BFL*:*

$$\Phi ::= \top \mid \neg\Phi \mid \Phi\wedge\Psi \mid \langle\leftarrow a\rangle\Phi \mid \langle a\rangle\Phi$$

BFL is the *set of all formulas* of BFL.

Definition 68 *Let* N *be a net and* $(\pi,\sigma) \in Runs(N)$. *The* satisfaction relation $\models_N \in Runs(N) \times \mathbf{BFL}$*:*

1. $(\pi,\sigma) \models_N \top$ *— always;*
2. $(\pi,\sigma) \models_N \neg\Phi$*, if* $(\pi,\sigma) \not\models_N \Phi$*;*
3. $(\pi,\sigma) \models_N \Phi\wedge\Psi$*, if* $(\pi,\sigma) \models_N \Phi$ *and* $(\pi,\sigma) \models_N \Psi$*;*
4. $(\pi,\sigma) \models_N \langle\leftarrow a\rangle\Phi$*, if* $\exists(\tilde{\pi},\tilde{\sigma}) \in Runs(N)$ $(\tilde{\pi},\tilde{\sigma}) \xrightarrow{\widehat{\pi}} (\pi,\sigma)$*, where* $\widehat{\pi} = (\widehat{C},\widehat{\varphi})$, $vis(l_{\widehat{C}}(T_{\widehat{C}})) = a$ *and* $(\tilde{\pi},\tilde{\sigma}) \models_N \Phi$*;*
5. $(\pi,\sigma) \models_N \langle a\rangle\Phi$*, if* $\exists(\tilde{\pi},\tilde{\sigma}) \in Runs(N)$ $(\pi,\sigma) \xrightarrow{\widehat{\pi}} (\tilde{\pi},\tilde{\sigma})$*, where* $\widehat{\pi} = (\widehat{C},\widehat{\varphi})$, $vis(l_{\widehat{C}}(T_{\widehat{C}})) = a$ *and* $(\tilde{\pi},\tilde{\sigma}) \models_N \Phi$.

$[a]\Phi = \neg\langle a\rangle\neg\Phi$, $[\leftarrow a]\Phi = \neg\langle\leftarrow a\rangle\neg\Phi$.

$N \models_N \Phi$, if $(\pi_N,\varepsilon) \models_N \Phi$.

Definition 69 N *and* N' *are* logical equivalent *in* BFL, $N{=}_{BFL}N'$, *if* $\forall\Phi \in \mathbf{BFL}$ $N \models_N \Phi \;\Leftrightarrow\; N' \models_{N'} \Phi$.

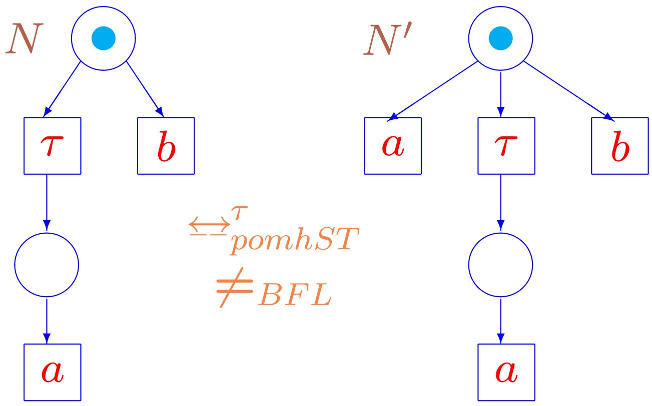
Let N be a net and $\pi \in \Pi(N),\ a \in Act$.

The set of *visible extensions* of a process π by action a *(image set)* is $VisImage(\pi, a) = \{\tilde{\pi} \mid \pi \xrightarrow{\hat{\pi}} \tilde{\pi},$
$\hat{\pi} = (\widehat{C}, \hat{\varphi}),\ vis(l_{\widehat{C}}(T_{\widehat{C}})) = a\}$.

A net N is a *finite-image* one, if $\forall \pi \in \Pi(N)\ \forall a \in Act$ $|VisImage(\pi, a)| < \infty$.

Theorem 13 *For image-finite nets N and N'*
$N \underline{\leftrightarrow}^{\tau}_{ibr} N' \ \Leftrightarrow\ N \underline{\leftrightarrow}^{\tau}_{ibif} N' \ \Leftrightarrow\ N =_{BFL} N'$.

Example on logical equivalence of BFL



Differentiating power of $=_{BFL}$

$N \underline{\leftrightarrow}^{\tau}_{pomhST} N'$, but $N \neq_{BFL} N'$, because for $\Phi = \langle a \rangle [\leftarrow a] \langle b \rangle \top$, $N \not\models_N \Phi$, but $N' \models_{N'} \Phi$, since in the net N' an action a can happen so that it will be simulated by sequence τa in N.

Then the state of the net N reached after τ must be related with the initial state of a net N, but in such a case the occurrence of action b from the initial state of N' cannot be imitated from the corresponding state of N.

Logic $SPBFL$ [Pin93]

Definition **70** *$\top$ denotes the truth, ρ is a pomset with labeling into Act.*

A formula *of $SPBFL$:*

$$\Phi ::= \ \top \mid \neg\Phi \mid \Phi\wedge\Psi \mid \langle\leftarrow \rho\rangle\Phi \mid \langle a\rangle\Phi$$

$\mathbf{SPBFL}$ is the *set of all formulas* of $SPBFL$.

Definition **71** *Let N be a net and $(\pi,\sigma) \in Runs(N)$. The* satisfaction relation *$\models_N \in Runs(N) \times \mathbf{SPBFL}$:*

1. $(\pi,\sigma) \models_N \top$ — *always;*
2. $(\pi,\sigma) \models_N \neg\Phi$, *if* $(\pi,\sigma) \not\models_N \Phi$;
3. $(\pi,\sigma) \models_N \Phi\wedge\Psi$, *if* $(\pi,\sigma) \models_N \Phi$ *and* $(\pi,\sigma) \models_N \Psi$;
4. $(\pi,\sigma) \models_N \langle\leftarrow \rho\rangle\Phi$, *if* $\exists(\tilde{\pi},\tilde{\sigma}) \in Runs(N)$ $(\tilde{\pi},\tilde{\sigma}) \stackrel{\widehat{\pi}}{\to} (\pi,\sigma)$, *where* $\widehat{\pi} = (\widehat{C},\widehat{\varphi})$, $vis(\rho_{\widehat{C}}) \in \rho$ *and* $(\tilde{\pi},\tilde{\sigma}) \models_N \Phi$;
5. $(\pi,\sigma) \models_N \langle a\rangle\Phi$, *if* $\exists(\tilde{\pi},\tilde{\sigma}) \in Runs(N)$ $(\pi,\sigma) \stackrel{\widehat{\pi}}{\to} (\tilde{\pi},\tilde{\sigma})$, *where* $\widehat{\pi} = (\widehat{C},\widehat{\varphi})$, $vis(l_{\widehat{C}}(T_{\widehat{C}})) = a$ *and* $(\tilde{\pi},\tilde{\sigma}) \models_N \Phi$.

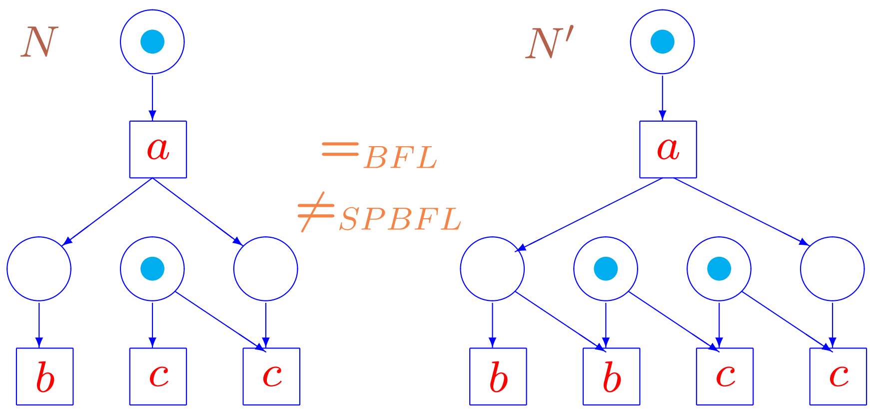
$[a]\Phi = \neg\langle a\rangle\neg\Phi$, $[\leftarrow \rho]\Phi = \neg\langle\leftarrow \rho\rangle\neg\Phi$.

$N \models_N \Phi$, if $(\pi_N,\varepsilon) \models_N \Phi$.

Definition **72** *N and N' are* logical equivalent *in $SPBFL$, $N =_{SPBFL} N'$, if $\forall\Phi \in \mathbf{SPBFL}$ $N \models_N \Phi \ \Leftrightarrow \ N' \models_{N'} \Phi$.*

Theorem **14** *For image-finite nets N and N' $N \underline{\leftrightarrow}^{\tau}_{pomhbr} N' \ \Leftrightarrow \ N \underline{\leftrightarrow}^{\tau}_{pombpomf} N' \ \Leftrightarrow \ N =_{SPBFL} N'$.*

Example on logical equivalence of $SPBFL$



Differentiating power of $=_{SPBFL}$

$N =_{BFL} N'$, but $N \neq_{SPBFL} N'$, because for $\Phi = [a][b]\langle c\rangle\langle\leftarrow (a;b)\|c\rangle\top$, $N \models_N \Phi$, but $N' \not\models_{N'} \Phi$ since only in N' after a action b can happen so that c must depend on a.

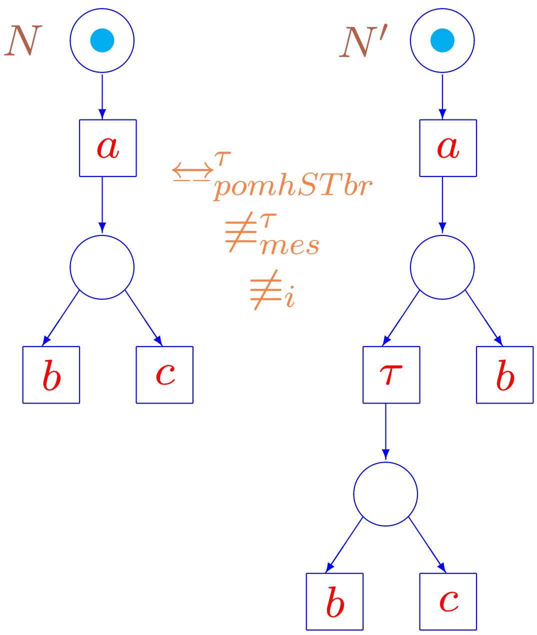
Here $(a;b)\|c$ denotes the pomset where b depends on a, and a, b are independent with c.

Interrelations of equivalences with τ-equivalences

Theorem 15 *Let* $\leftrightarrow\ \in \{\equiv, \underline{\leftrightarrow}\}$, $\star \in \{i, s, pw, pom, iST, pwST, pomST, mes, sbsf, sbpwf, sbpomf\}$, $\star\star \in \{s, pw, pom\}$. *For nets* N *and* N'*:*

1. $N \leftrightarrow_{\star} N' \ \Rightarrow\ N \leftrightarrow^{\tau}_{\star} N'$;
2. $N \underline{\leftrightarrow}_{i} N' \ \Rightarrow\ N \underline{\leftrightarrow}^{\tau}_{ibr} N'$;
3. $N \underline{\leftrightarrow}_{iST} N' \ \Rightarrow\ N \underline{\leftrightarrow}^{\tau}_{iSTbr} N'$;
4. $N \underline{\leftrightarrow}_{pomh} N' \ \Rightarrow\ N \underline{\leftrightarrow}^{\tau}_{pomhSTbr} N'$;
5. $N \underline{\leftrightarrow}_{\star\star} N' \ \Rightarrow\ N \underline{\leftrightarrow}^{\tau}_{ib\star\star f} N'$.

and all the implications are strict.

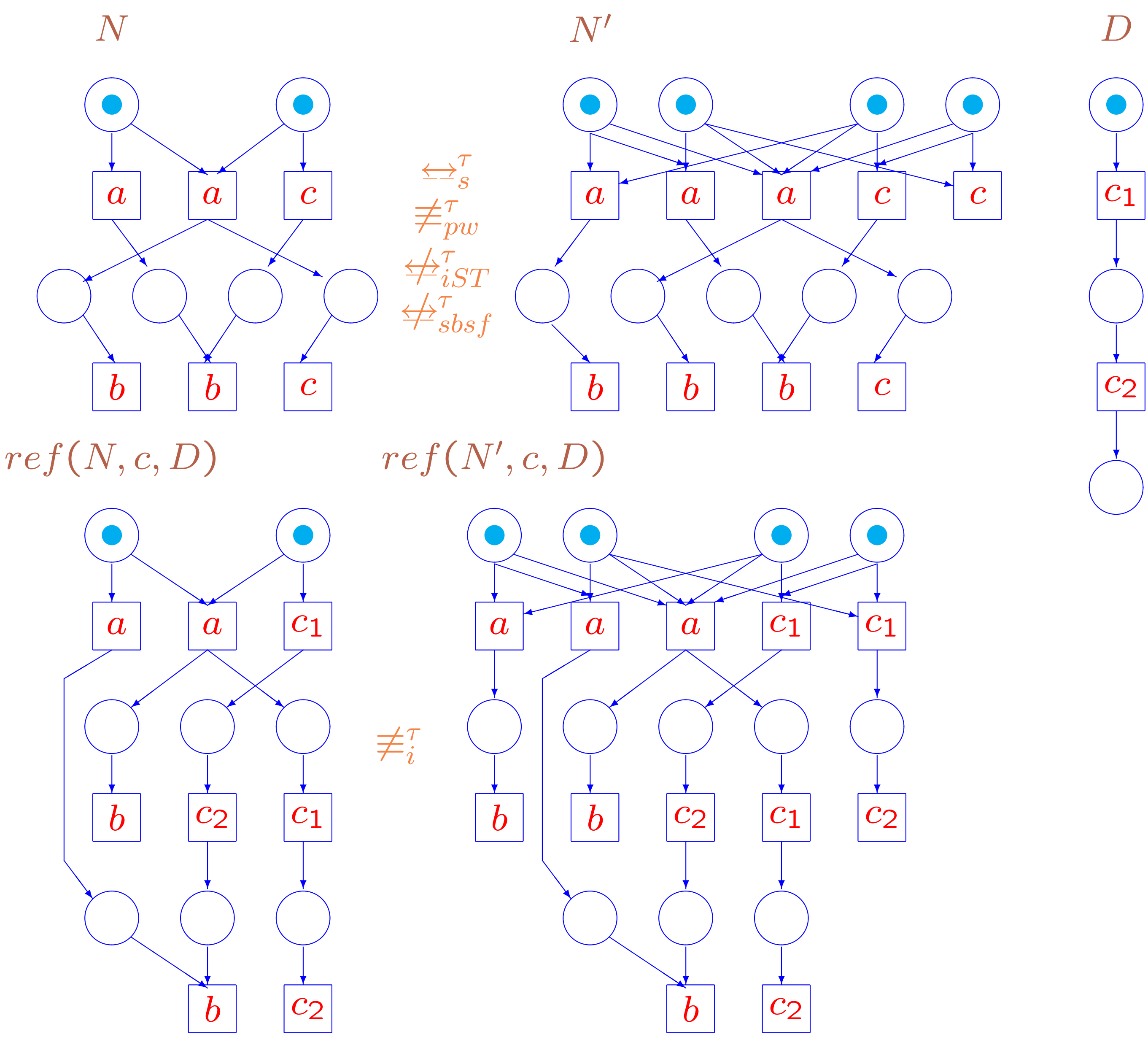


ETE: Example of interrelations of equivalences and τ-equivalences

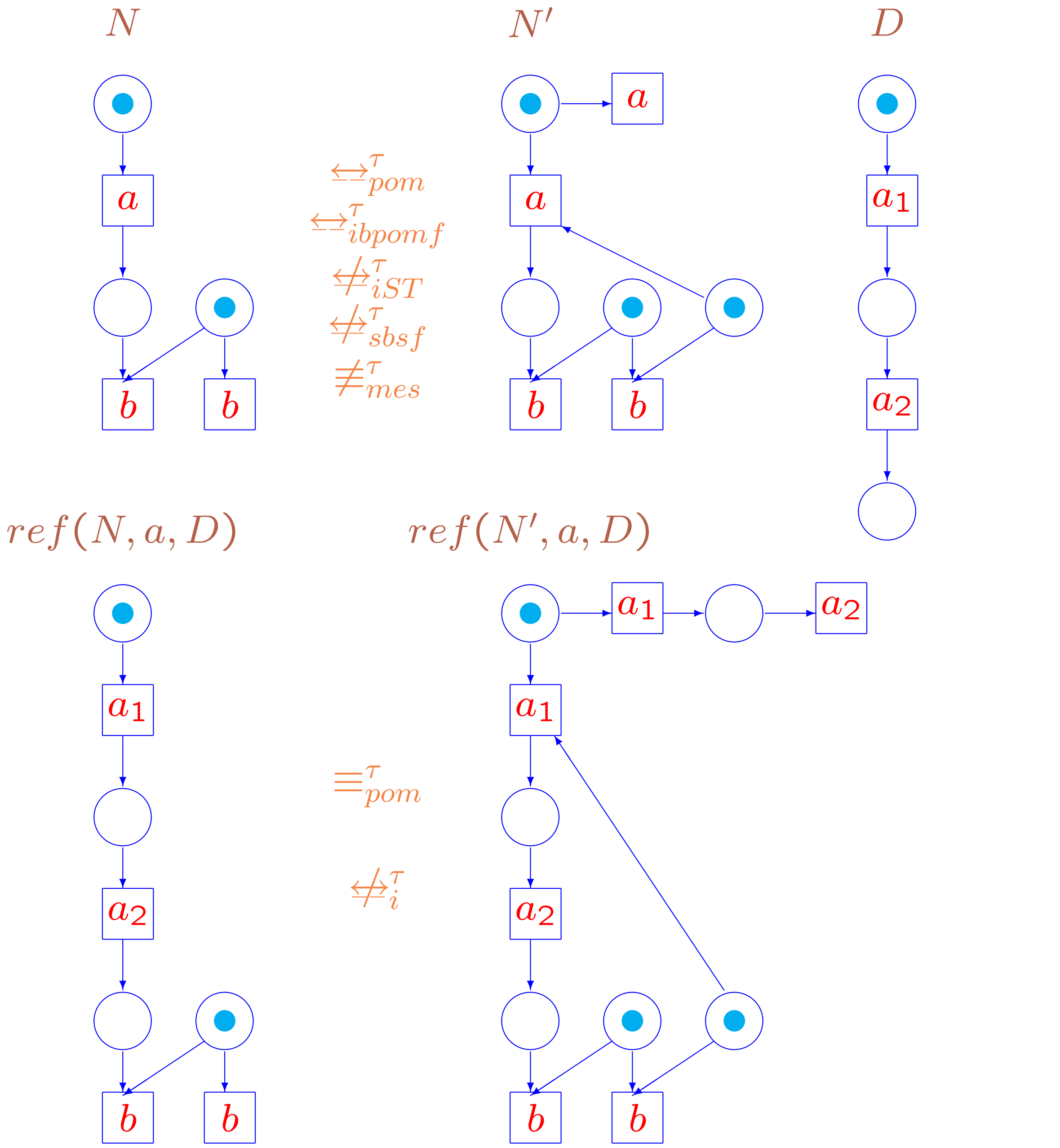
- In Figure ETE, $N \underline{\leftrightarrow}^{\tau}_{pomhSTbr} N'$, but $N \not\equiv_i N'$, since only in the net N' an action a can happen in the initial state.

- In Figure BT2(a), $N \equiv^{\tau}_{mes} N'$, but $N \not\equiv_i N'$, since only in the net N' an action τ can happen in the initial state.

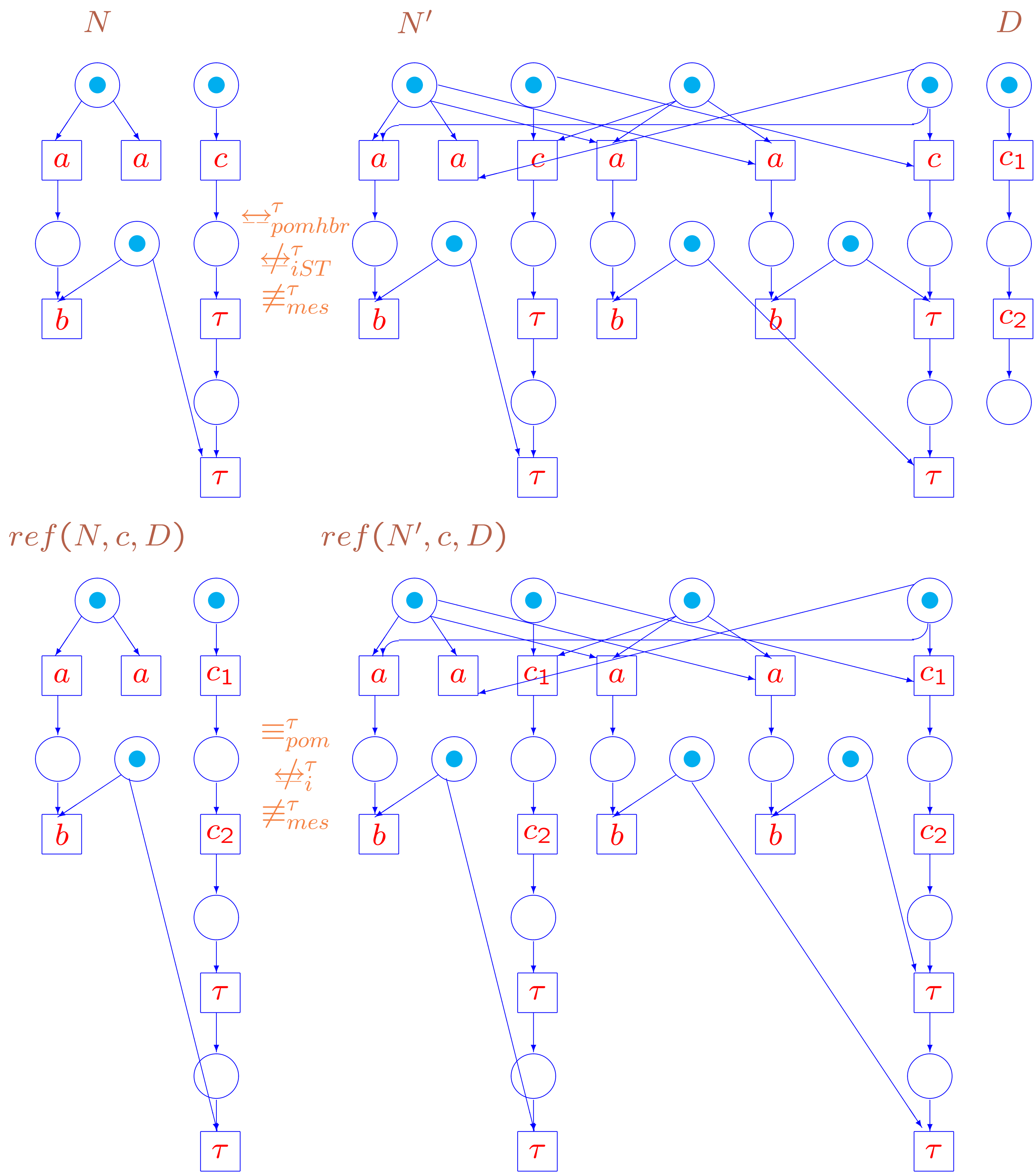
SM-refinements



RBT: The τ-equivalences between $\equiv_i^\tau$ and $\underline{\leftrightarrow}_s^\tau$ are not preserved by SM-refinements



RBT1: The τ-equivalences between $\underline{\leftrightarrow}^{\tau}_{i}$ and $\underline{\leftrightarrow}^{\tau}_{pom}$ are not preserved by SM-refinements



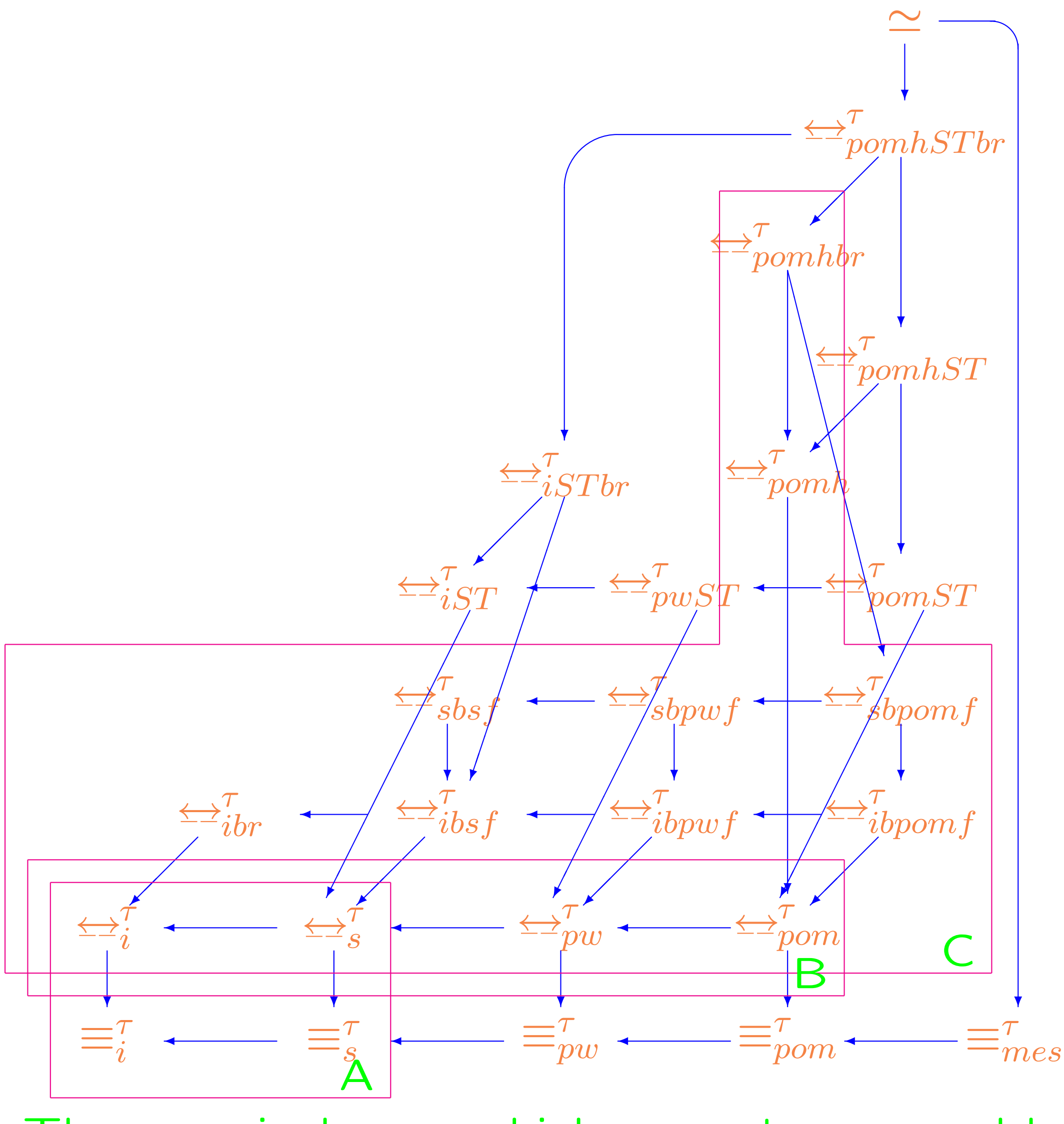
RBT2: The τ-equivalences between $\underline{\leftrightarrow}^{\tau}_{i}$ and $\underline{\leftrightarrow}^{\tau}_{pomhbr}$ are not preserved by SM-refinements

- In Figure RBT, $N \underline{\leftrightarrow}_s^\tau N'$, but $ref(N,c,D) \not\equiv_i^\tau ref(N',c,D)$, since only in $ref(N',c,D)$ the sequence of actions c_1abc_2 can happen.

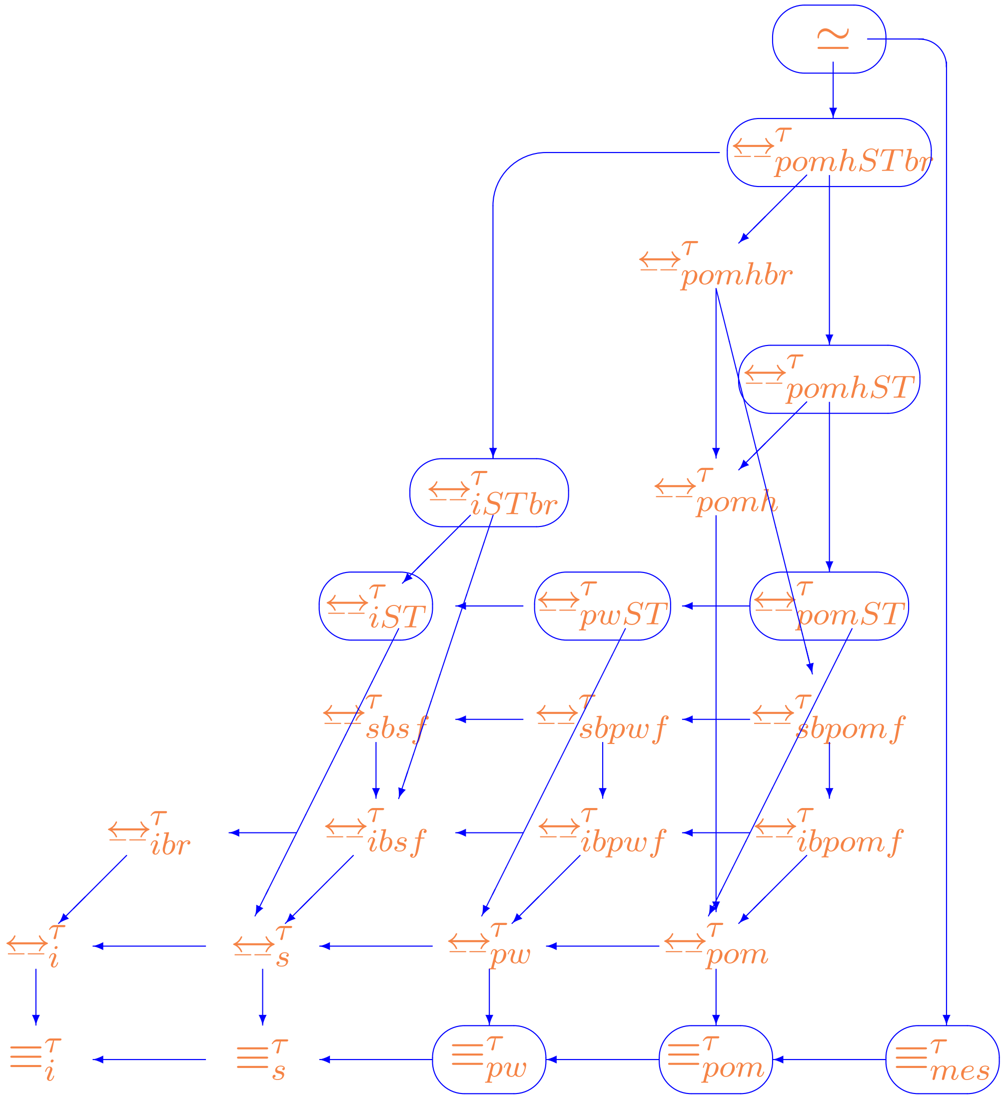
- In Figure RBT1, $N \underline{\leftrightarrow}_{pom}^\tau N'$, but $ref(N,a,D) \not\underline{\leftrightarrow}_i^\tau ref(N',a,D)$, since only in $ref(N',a,D)$ after occurrence of action a_1 action b can not happen.

- In Figure RBT2, $N \underline{\leftrightarrow}_{pomhbr}^\tau N'$, but $ref(N,a,D) \not\underline{\leftrightarrow}_i^\tau ref(N',a,D)$, since only in $ref(N',a,D)$ an action c_1 may happen so that after the corresponding action c_1 in the net N an action a may happen in such a way that the action b never occur.

Proposition 13 *[BDKP91,Dev92,Tar97]* *Let* $\star \in \{i, s\}$, $\star\star \in \{i, s, pw, pom, pomh, ibr, pomhbr, ibsf, ibpwf, ibpomf, sbsf, sbpwf, sbpomf\}$. *Then the* τ*-equivalences* $\equiv^{\tau}_{\star}$, $\underline{\leftrightarrow}^{\tau}_{\star\star}$ *are not preserved by SM-refinements.*



The τ-equivalences which are not preserved by SM-refinements



Preservation of the τ-equivalences by SM-refinements

Theorem 16 *Let* $\leftrightarrow \in \{\equiv^{\tau}, \underline{\leftrightarrow}^{\tau}, \simeq\}$ *and* $\star \in \{\_, i, s, pw, pom, iST, pwST, pomST, pomh, pomhST, ibr, pomhbr, iSTbr, pomhSTbr, mes, ibsf, ibpwf, ibpomf, sbsf, sbpwf, sbpomf\}$. *For nets* N, N' *s.t.* $a \in l_N(T_N) \cap l_{N'}(T_{N'}) \cap Act$ *and SM-net* D: $N \leftrightarrow_{\star} N' \Rightarrow ref(N, a, D) \leftrightarrow_{\star} ref(N', a, D)$ *iff the equivalence* $\leftrightarrow_{\star}$ *is in oval in the figure above.*

The τ-equivalences on nets without silent transitions

Proposition 14 *Let* $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}\}$, $\star \in \{i, s, pw, pom, iST, pwST, pomST, mes, sbsf, sbpwf, sbpomf\}$, $\star\star \in \{s, pw, pom\}$. *For nets without silent transitions* N *and* N'*:*

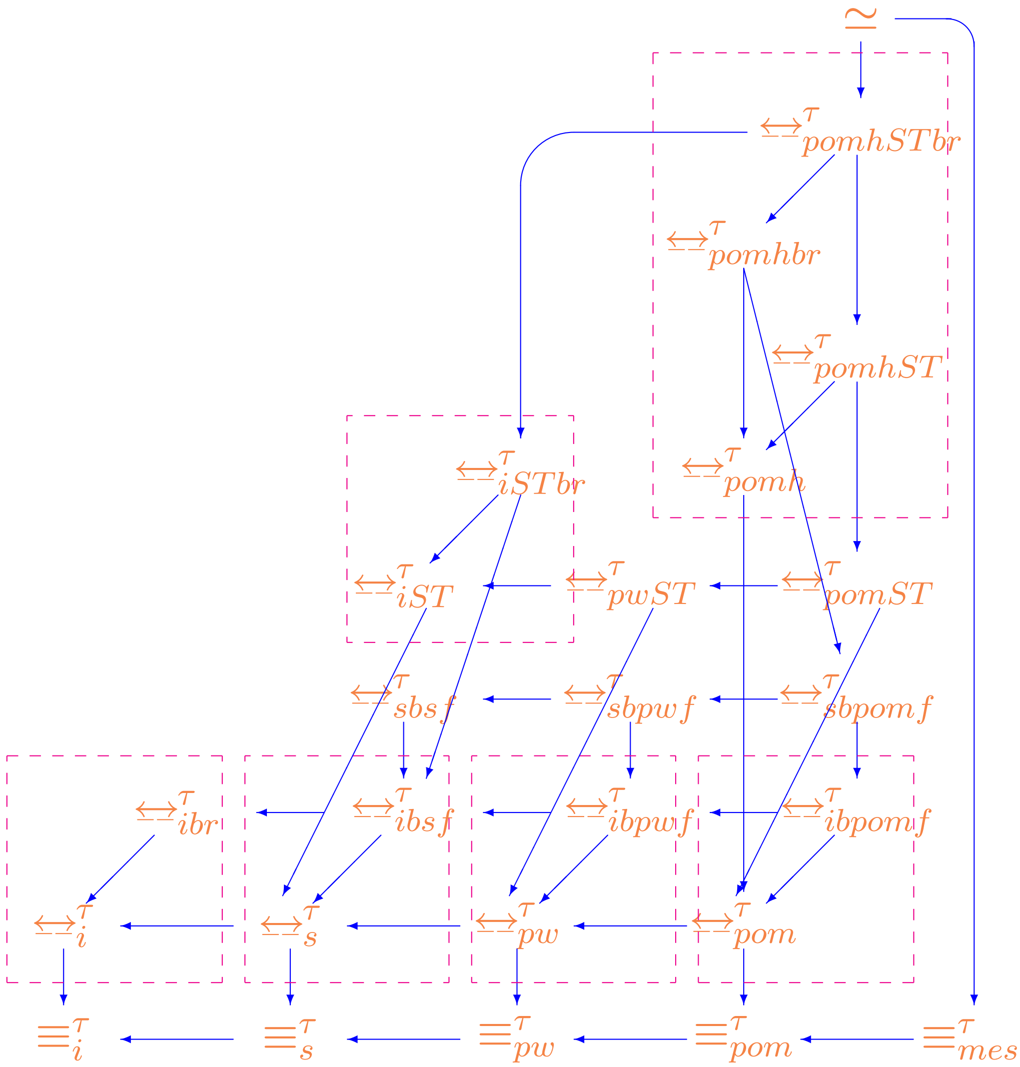
1. $N \leftrightarrow_\star N' \Leftrightarrow N \leftrightarrow_\star^\tau N'$ *[Gla93, Tar97];*

2. $N \underline{\leftrightarrow}_i N' \Leftrightarrow N \underline{\leftrightarrow}_{ibr}^\tau N'$ *[Gla93];*

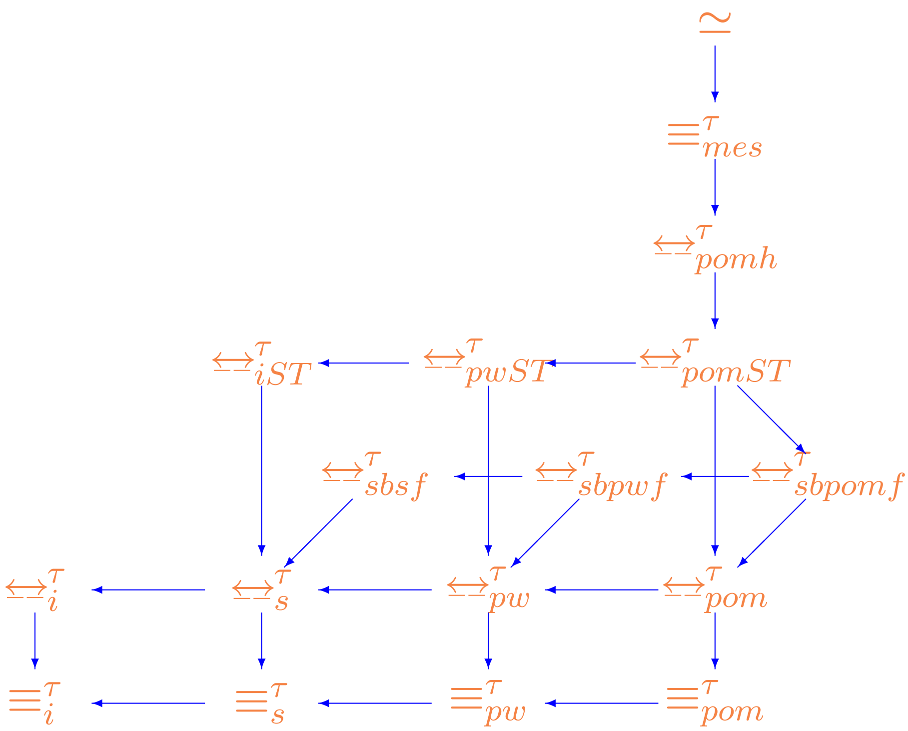
3. $N \underline{\leftrightarrow}_{iST} N' \Leftrightarrow N \underline{\leftrightarrow}_{iSTbr}^\tau N'$ *[Tar97];*

4. $N \underline{\leftrightarrow}_{pomh} N' \Leftrightarrow N \underline{\leftrightarrow}_{pomhSTbr}^\tau N'$ *[Tar97];*

5. $N \underline{\leftrightarrow}_{\star\star} N' \Leftrightarrow N \underline{\leftrightarrow}_{ib\star\star f}^\tau N'$ *[Tar97].*



Merging of the τ-equivalences on nets without silent transitions



Interrelations of the τ-equivalences on nets without silent transitions

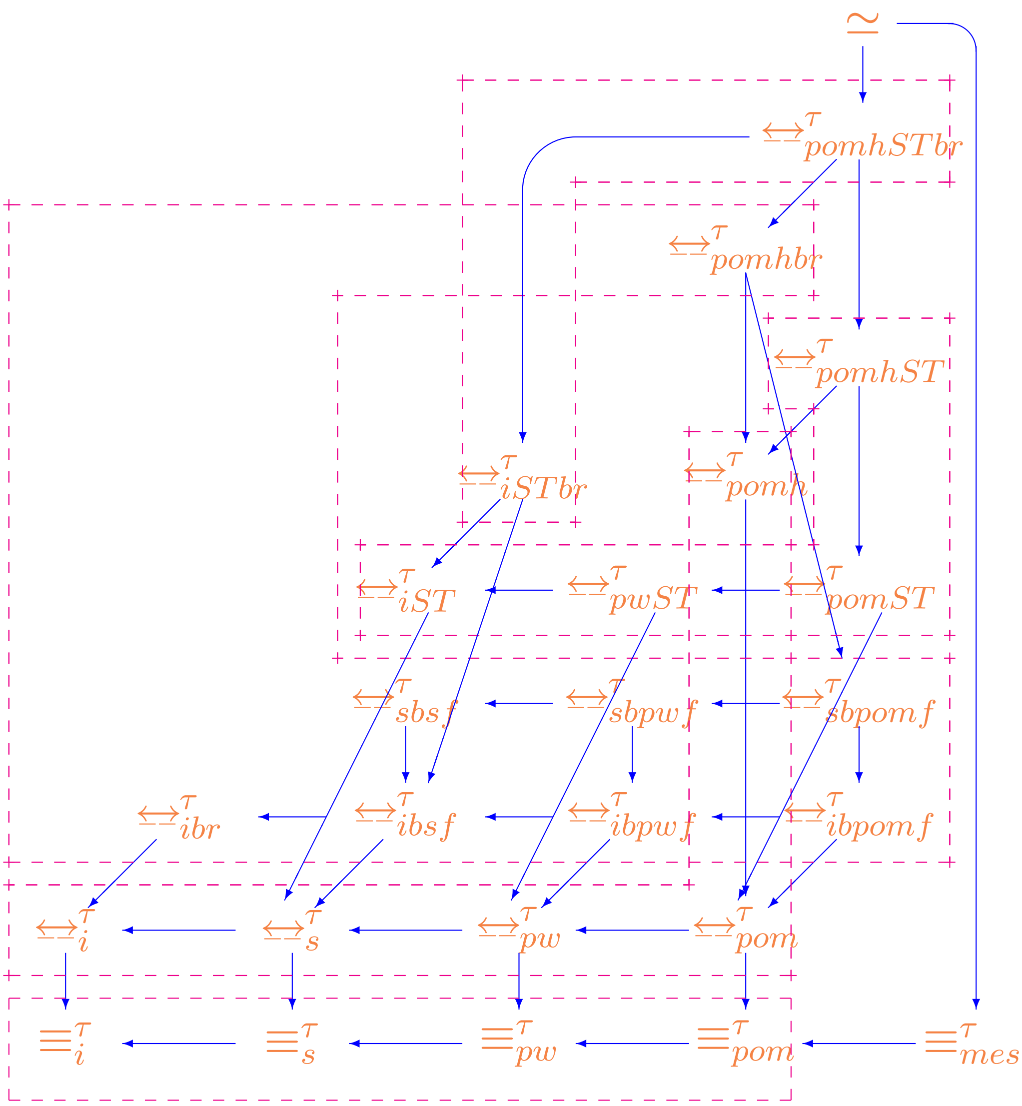
Theorem 17 *Let* $\leftrightarrow, \twoheadleftarrow\!\!\twoheadrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$, $\star, \star\star \in \{\_, i, s, pw, pom, iST, pwST, pomST, pomh, ibr, mes, sbsf, sbpwf, sbpomf\}$. *For nets without silent transitions* N *and* N' $N \leftrightarrow_{\star} N' \Rightarrow N \twoheadleftarrow\!\!\twoheadrightarrow_{\star\star} N'$ *iff in the graph above there exists a directed path from* $\leftrightarrow_{\star}$ *to* $\twoheadleftarrow\!\!\twoheadrightarrow_{\star\star}$.

The τ-equivalences on sequential nets

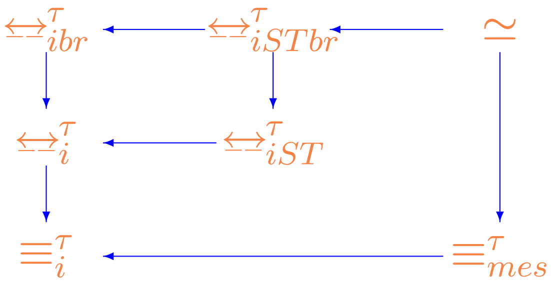
Definition **73** *A net* $N = \langle P_N, T_N, F_N, l_N, M_N\rangle$ *is* sequential*, if* $\forall M \in Mark(N)\ \neg\exists t, u \in T_N :\ {}^\bullet t + {}^\bullet u \subseteq M$.

Proposition **15** *For sequential nets* N *and* N'*:*

1. $N \equiv_i^\tau N' \ \Leftrightarrow\ N \equiv_{pom}^\tau N'$ [Eng85];
2. $N \leftrightarrow_i^\tau N' \ \Leftrightarrow\ N \leftrightarrow_{pomh}^\tau N'$ [BDKP91];
3. $N \leftrightarrow_{iST}^\tau N' \ \Leftrightarrow\ N \leftrightarrow_{pomhST}^\tau N'$ [Tar98a];
4. $N \leftrightarrow_{ibr}^\tau N' \ \Leftrightarrow\ N \leftrightarrow_{pomhbr}^\tau N'$ [Tar98a];
5. $N \leftrightarrow_{iSTbr}^\tau N' \ \Leftrightarrow\ N \leftrightarrow_{pomhSTbr}^\tau N'$ [Tar98a].



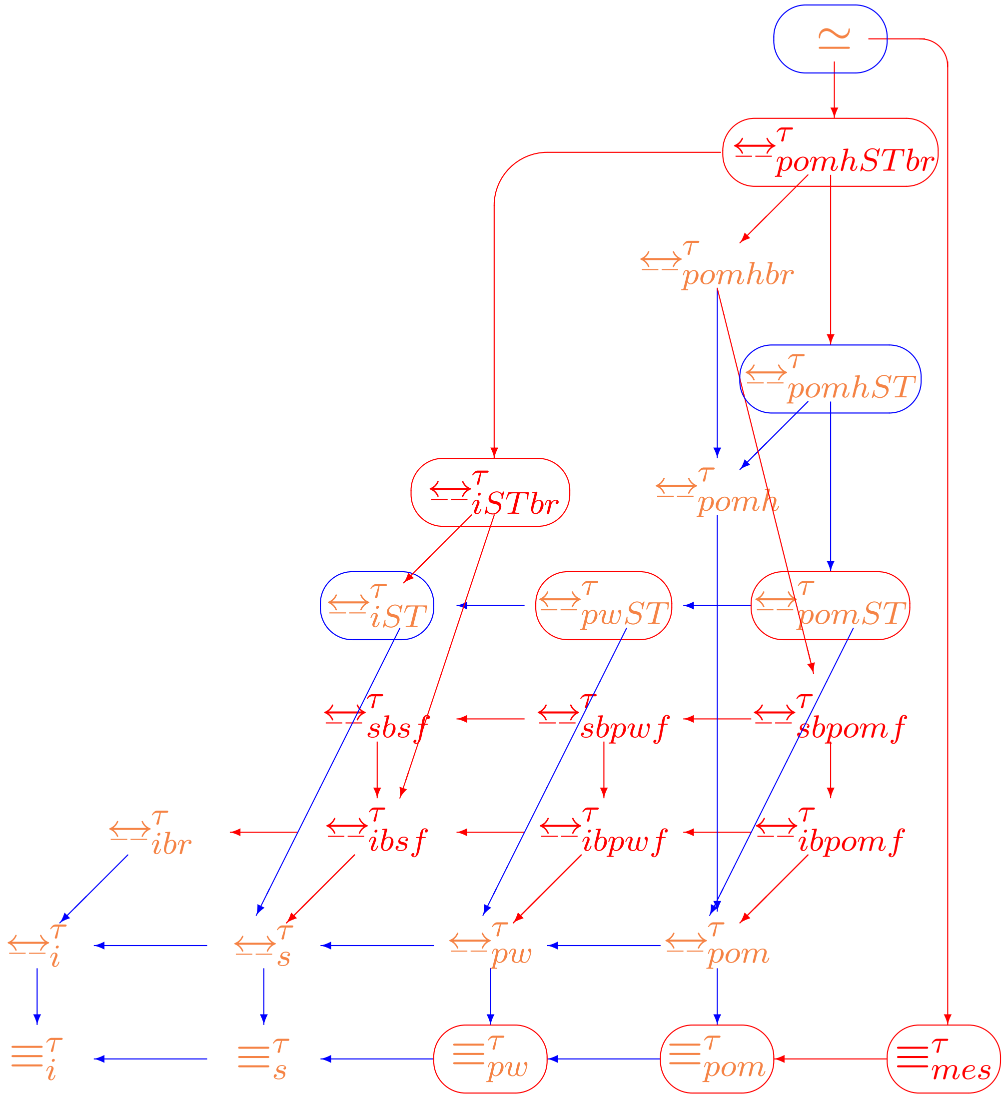
Merging of the τ-equivalences on sequential nets



Interrelations of the τ-equivalences on sequential nets

Theorem 18 *Let* $\leftrightarrow, \twoheadleftarrow\!\!\twoheadrightarrow \in \{\equiv^\tau, \underline{\leftrightarrow}^\tau, \simeq\}$, $\star, \star\star \in \{\_, i, iST, ibr, iSTbr, mes\}$. *For sequential nets* N *and* N' $N \leftrightarrow_\star N' \Rightarrow N \twoheadleftarrow\!\!\twoheadrightarrow_{\star\star} N'$ *iff in the graph above there exists a directed path from* $\leftrightarrow_\star$ *to* $\twoheadleftarrow\!\!\twoheadrightarrow_{\star\star}$.

- In Figure BT2(a), $N \equiv^\tau_{mes} N'$, but $N \not\underline{\leftrightarrow}^\tau_{i} N'$.
- In Figure BT2(c), $N \underline{\leftrightarrow}^\tau_{i} N'$, but $N \not\underline{\leftrightarrow}^\tau_{ibr} N'$.
- In Figure BT2(b), $N \underline{\leftrightarrow}^\tau_{i} N'$, but $N \not\underline{\leftrightarrow}^\tau_{iST} N'$.
- In Figure BT1(c), $N \underline{\leftrightarrow}^\tau_{iSTbr} N'$, but $N \not\equiv^\tau_{mes} N'$.



New results for the τ-equivalences

Decidability results for the τ-equivalences

- $\equiv_i^\tau$
  - is decidable for:
    finite safe nets (EXPSPACE) [JM96].
  - is undecidable for:
    labeled nets [Jan95].

- $\equiv_s^\tau$
  - is decidable for:
    finite safe nets (EXPSPACE) [JM96].

- $\equiv_{pom}^\tau$
  - is decidable for:
    finite safe nets (EXPSPACE) [JM96].

- $\underline{\leftrightarrow}_i^\tau$
  - is decidable for:
    finite safe nets (DEXPTIME) [JM96].
  - is undecidable for:
    labeled nets [Jan95].

- $\underline{\leftrightarrow}_s^\tau$
  - is decidable for:
    finite safe nets (DEXPTIME) [JM96].

- $\underline{\leftrightarrow}_{pom}^\tau$
  - is decidable for:
    finite safe nets (DEXPTIME / EXPSPACE)[JM96].

- $\underline{\leftrightarrow}^{\tau}_{iST}$
  - is decidable for:
    bounded nets [Dev92];
    finite safe nets (DEXPTIME) [JM96].
- $\underline{\leftrightarrow}^{\tau}_{pomST}$
  - is decidable for:
    finite safe nets (DEXPTIME / EXPSPACE) [JM96].
- $\underline{\leftrightarrow}^{\tau}_{pomh}$
  - is decidable for:
    finite safe nets (DEXPTIME) [Vog91b,JM96].
- $\underline{\leftrightarrow}^{\tau}_{pomhST}$
  - is decidable for:
    finite safe nets (DEXPTIME) [Vog91b,JM96].
- $\underline{\leftrightarrow}^{\tau}_{ibr}$
  - is decidable for:
    finite safe nets (DEXPTIME) [JM96].

Further research

τ-variants of place bisimulation equivalences.

- New equivalences.

  Interleaving *place τ-bisimulation equivalence* ($\sim_i^\tau$).
  Behavior preserving reduction of Petri nets with silent transitions [Aut93,APS94].

- Interleaving *branching place τ-bisimulation equivalence* ($\sim_{ibr}^\tau$).

- Non-interleaving variants of *place τ-bisimulations* ($\sim_s^\tau, \sim_{pw}^\tau$ and $\sim_{pom}^\tau$).

- Interrelations of the place τ-bisimulations.

  Whether any two of $\sim_i^\tau, \sim_s^\tau$ and $\sim_{pw}^\tau$ coincide?

  We have only counterexamples showing that $\sim_{ibr}^\tau$ and $\sim_{pom}^\tau$ do not imply each other and do not merge with any of three mentioned τ-equivalences.

- Interrelations of the place τ-bisimulations with the other τ-equivalences we proposed.

  We compared place equivalences with other ones on Petri nets without silent transitions [Tar98b].

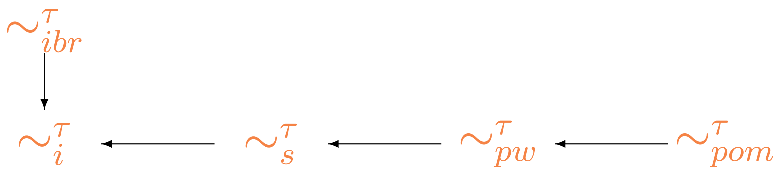
- Preservation of place τ-bisimulations by SM-refinements.

  We can show that no place τ-bisimulation relation is preserved by SM-refinements [Tar98b].

- Interrelations of place τ-bisimulations on net subclasses.

  On nets without silent transitions place τ-equivalences coincide with the corresponding relations that do not abstract of silent actions. In particular, $\sim^{\tau}_{ibr}$ merges with $\sim_i$.

  On sequential nets, all non-interleaving place relations coincide with interleaving ones: only $\sim^{\tau}_{i}$ and $\sim^{\tau}_{ibr}$ are remained.



Interrelations of place τ-bisimulation equivalences

Equivalences for process algebras: calculus AFP_2

Abstract: A process algebra AFP_2 was proposed by L.A. Cherkasova in 1989. It has a semantics of posets with non-actions and deadlocked actions to respect non-determinism.

Via formulas of AFP_2, one can analyze behavior of A-nets (Acyclic nets). The considered Petri net equivalences are investigated on this net subclass.

Semantic equivalences of formulas AFP_2 (algebraic equivalences) are transferred into A-nets, and their interrelations with the net equivalences are investigated.

A term rewrite system RWS_2 is produced from axiom system Θ_2 for semantic equivalences. Its confluence (in the case of termination) is proved.

A method of automatic check for algebraic equivalences based on RWS_2 was implemented as a program $CANON$ in C programming language.

Keywords: Process algebras, syntax, semantics, semantic (algebraic) equivalences, axiomatization, A-nets, net equivalences, term rewrite systems, implementation.

Contents

Process algebras: semantics of concurrency

In process calculi, a process is specified by an algebraic formula.

A verification of its properties is accomplished by means of equivalences, axioms and inference rules.

The calculi below construct a process from atomic *actions* with *precedence, parallelism, non-determinism* and some auxilary operations.

1. Interleaving semantics.

   CCS [Mil80], CSP [Hoa80], $TCSP$ [Hoa85,Old87a], BPA [BK89].

   Concurrency is interpreted as sequential non-determinism.

2. Step semantics.

   $SCCS$ [Mil83], ACP [BK84], $CCSP$ [Old87b], PBC [BDH92].

   A special operator for simultaneous occurrence of actions.

3. Pomset semantics.

   Algebra of event structures [BCa87].

   A causal dependence relation over actions imposes partial ordering. Two actions are parallel if they are causally independent.

Interleaving calculi are more suitable in technical staff.

Algebras based on step and pomset semantics have more natural specification of concurrency.

Process algebras: specification and analysis

1. *Descriptive* calculi.

   They provide a description of structural properties of systems: specification.

   An example is AFP_0 [Ch89].

2. *Analytical* calculi.

   They combine mechanisms as for specification of processes as for investigation of their behavioral properties: analysis, verification.

   An example is AFP_2 [Ch89].

Algebra of finite processes AFP_2

AFP_2 has semantics of posets with non-actions and deadlocked actions (to respect non-determinism).

A synchronization is by action name. The only event corresponds to equally named actions.

Syntax

The symbol alphabets.

- $\alpha = \{a, b, \ldots\}$ is an alphabet of *actions*.
- $\bar{\alpha} = \{\bar{a}, \bar{b}, \ldots\}$ is an alphabet of *non-actions*.
- $\Delta_\alpha = \{\delta_a, \delta_b, \ldots\}$ is an alphabet of *deadlocked actions*.

$\widehat{\alpha} = \alpha \cup \bar{\alpha} \cup \Delta_\alpha$.

Symbols of $\widehat{\alpha}$ are combined into formulas by operations $;$ (*precedence*), $\triangledown$ (*exclusive or, alternative*), $\|$ (*concurrency*), $\vee$ (*disjunction, union*), $\rceil\!\rceil$ (*"not occur"*), $\tilde{\rceil\!\rceil}$ (*"mistaken not occur"*).

Definition 74 *A* formula *of* AFP_2 *is:*

$$P ::= \ a \mid \bar{a} \mid \delta_a \mid \rceil\!\rceil a \mid \tilde{\rceil\!\rceil} P \mid P;Q \mid P\|Q \mid P\triangledown Q \mid P\vee Q.$$

Here $a \in \alpha,\ \bar{a} \in \bar{\alpha},\ \delta_a \in \Delta_\alpha$ are *elementary formulas*. $\mathbf{AFP_2}$ is the set of *all formulas* of AFP_2.

Definition 75 *Formulas of* AFP_2 P *and* P' *are* isomorphic, $P \simeq P'$, *if they coincide up to associativity rules w.r.t.* $;, \|, \vee, \triangledown$ *and commutativity rules w.r.t.* $\|, \vee, \triangledown$.

For example, $(a\|b\|\bar{c}) \vee (c\|\bar{a}\|\bar{b}) \simeq (\bar{a}\|\bar{b}\|c) \vee (b\|a\|\bar{c})$.

Denotational semantics

Let $X \subseteq \widehat{\alpha}$. We propose the following notations.

- $X^+ = X \cap \alpha$ is the subset of *actions* of X;
- $X^- = X \cap \bar{\alpha}$ is the subset of *non-actions* of X;
- $\Delta_X = X \cap \Delta_\alpha$ is the subset of *deadlocked actions* of X.

We consider only posets $\rho = \langle X, \prec \rangle$ over $\widehat{\alpha}$ with the following restrictions.

1. a, $\bar{a}$ and δ_a do not occur in X together;
2. $\prec$ is irreflexive;
3. $\forall x, y \in X^- \cup \Delta_X \ (x \not\prec y) \wedge (y \not\prec x)$, i.e. all elements of $X^- \cup \Delta_X$ are incomparable;
4. $\forall x \in X^+ \ \forall y \in X^- \cup \Delta_X \ (x \not\prec y) \wedge (y \not\prec x)$, i.e. all elements of X^+ and $X^- \cup \Delta_X$ are incomparable.

The *modified union* of posets absorbs equal computations and ones which can be continued in another behaviour of nondeterministic process.

$$\rho \tilde{\cup} \rho' = \begin{cases} \rho, & \rho' \trianglelefteq \rho; \\ \rho', & \rho \trianglelefteq \rho'; \\ \{\rho, \rho'\}, & \text{otherwise.} \end{cases}$$

The operations over posets are introduced: ; (*precedence*), $\|$ (*concurrency*), $\triangledown$ (*alternative*), $\rceil\!\rceil$ (*not occur*) and $\tilde{\rceil\!\rceil}$ (*mistaken not occur*).

If a constructed poset ρ does not satisfy the conditions 1-4, we “correct” it with *regularization* operation $[\rho]$.

It singles out the maximal prefix of ρ “before” some contradictions arise. All the actions occuring “after” that contradictions are announced as the deadlocked ones.

- $D_1 = \{\delta_a \mid (a \in X) \wedge (a \prec a)\} \cup \{\delta_a \mid (a \in X) \wedge (\bar{a} \in X)\} \cup \{\delta_a \mid (a \in X) \wedge (\delta_a \in X)\} \cup \{\delta_a \mid (\bar{a} \in X) \wedge (\delta_a \in X)\} \cup \Delta_X$;

- $D_2 = \{\delta_a \mid (a \in X) \wedge (\delta_b \in D_1) \wedge (\delta_b \prec a)\}$;

- $D_3 = \{\delta_a \mid \bar{a} \in X\}$.

$$D = \begin{cases} \emptyset, & D_1 = \emptyset; \\ D_1 \cup D_2 \cup D_3, & \text{otherwise.} \end{cases}$$

Then $[\rho] = \langle D, \emptyset \rangle \cup \langle Y, \prec \cap (Y \times Y) \rangle$, where $Y = X \setminus \widehat{\alpha}(D)$. If ρ satisfies the conditions 1-4, then $[\rho] = \rho$.

Let $\rho = \langle X, \prec \rangle$, $\rho' = \langle X, \prec' \rangle$. We define poset operations.

Not occur $\urcorner\!\urcorner \rho = \langle \bar{\alpha}(X), \emptyset \rangle$.

Mistaken not occur $\tilde{\urcorner\!\urcorner} \rho = \langle \Delta_\alpha(X), \emptyset \rangle$.

Precedence $\rho ; \rho' = [\langle X \cup X', \prec \cup \prec' \cup (X^+ \times (X')^+) \cup (\Delta_X \times (X')^+) \rangle]$.

Concurrency $\rho \| \rho' = [\langle X \cup X', (\prec \cup \prec')^\star \rangle]$, where $(\prec \cup \prec')^\star$ is a transitive closure of $\prec \cup \prec'$.

Alternative $\rho \bigtriangledown \rho' = [\langle X \cup \bar{\alpha}(X'), \prec, l \cup l' \rangle] \tilde{\cup} [\langle \bar{\alpha}(X) \cup X', \prec' \rangle]$ (note that $\rho \bigtriangledown \rho'$ is not a poset, but a set of two posets describing alternative behaviours).

We extend the operations above to sets of posets. Let $\mathcal{P} = \cup_{i=1}^n \rho_i$ and $\mathcal{P}' = \cup_{j=1}^m \rho'_j$.

Then $\neg \mathcal{P} = \tilde{\cup}_{i=1}^n \neg \rho_i$, where $\neg \in \{\urcorner\!\urcorner, \tilde{\urcorner\!\urcorner}\}$ and $\mathcal{P} \circ \mathcal{P}' = \tilde{\cup}_{i=1}^n (\tilde{\cup}_{j=1}^m \rho_i \circ \rho'_j)$, where $\circ \in \{;, \|, \bigtriangledown\}$.

Definition **76** *A* denotational semantics *of* AFP_2 *is a mapping* $\mathcal{D}_2$ *from* $\mathbf{AFP_2}$ *into set of posets.*

1. $\mathcal{D}_2[a] = \langle\{a\},\emptyset\rangle,\ \mathcal{D}_2[\bar{a}] = \langle\{\bar{a}\},\emptyset\rangle,\ \mathcal{D}_2[\delta_a] = \langle\{\delta_a\},\emptyset\rangle$;
2. $\mathcal{D}_2[\neg P] = \neg\mathcal{D}_2[P],\ \neg \in \{\rceil\!\rceil, \tilde{\rceil\!\rceil}\}$;
3. $\mathcal{D}_2[P \circ Q] = \mathcal{D}_2[P] \circ \mathcal{D}_2[Q],\ \circ \in \{;, \|, \bigtriangledown\}$;
4. $\mathcal{D}_2[P \vee Q] = \mathcal{D}_2[P] \tilde{\cup} \mathcal{D}_2[Q]$.

Definition **77** *Formulas of* AFP_2 P *and* P' *are* equivalent w.r.t. denotational semantics, $P {=}_{\mathcal{D}_2} P'$, *iff* $\mathcal{D}_2[P] = \mathcal{D}_2[P']$.

If $\rho = \langle X, \prec\rangle$ is a poset, then $\rho^+ = \langle X^+, \prec\rangle$ is the *"observable"* part of ρ over α.

For any formula P of AFP_2, $\mathcal{D}_2[P] = \cup_{i=1}^{n}\rho_i$ is a set of posets, which characterize a nondeterministic process specified by P.

An *"observable"* part of this set is defined as: $\mathcal{D}_2^+[P] = \cup_{i=1}^{n}\rho_i^+$.

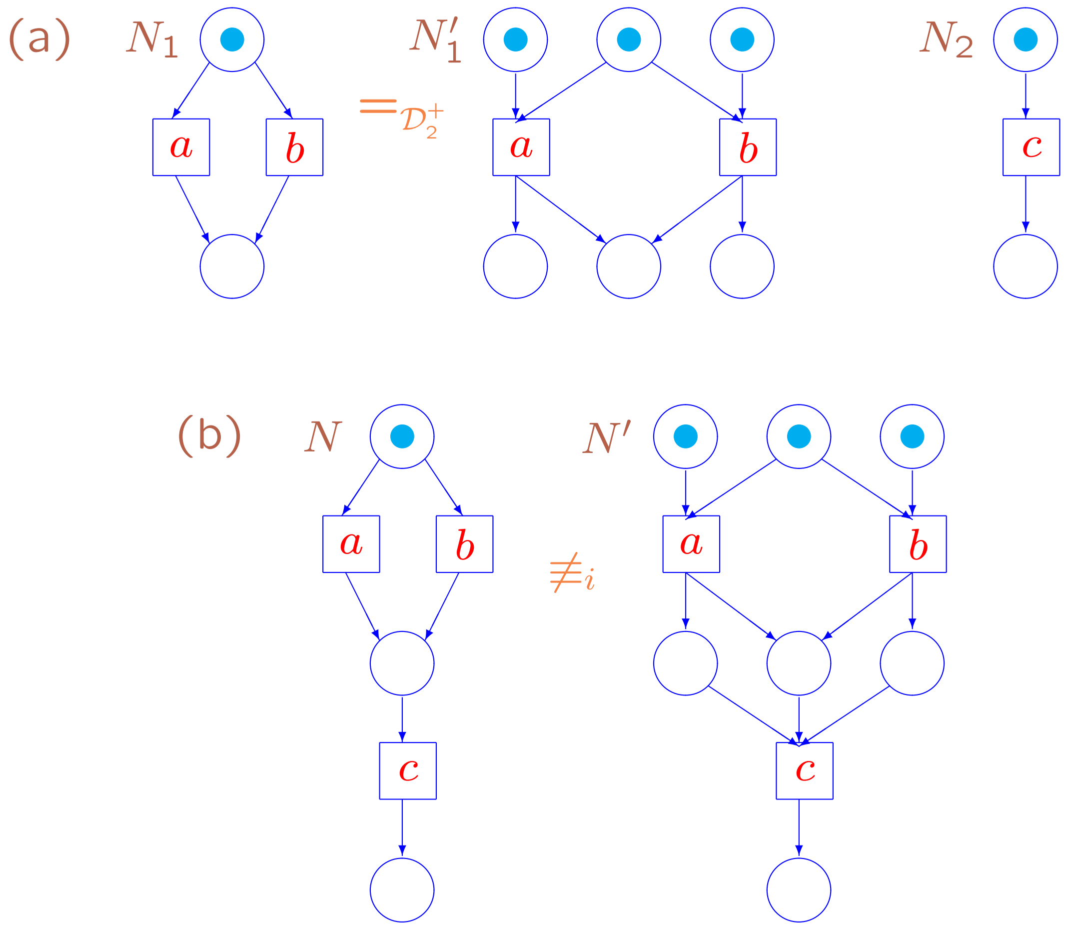
Definition **78** *Formulas* P *and* P' *are* observation equivalent w.r.t. denotational semantics, $P {=}_{\mathcal{D}_2^+} P'$, *iff* $\mathcal{D}_2^+[P] = \mathcal{D}_2^+[P']$.

A *context* $\mathcal{C}$ is a formula of AFP_2 with zero or more subformulas replaced by "holes" to be filled by other formulas.

$\mathcal{C}[P]$ means putting of the formula P in each such "hole".

Proposition **16** *[Ch89] For any two formulas* P *and* P' *of* AFP_2 $P {=}_{\mathcal{D}_2} P' \;\Leftrightarrow\; \forall\mathcal{C}\ \mathcal{C}[P] {=}_{\mathcal{D}_2} \mathcal{C}[P']$.

Example of semantic equivalence of AFP_2



A-nets from example on congruence

Thus, $=_{\mathcal{D}_2}$ is a congruence w.r.t. operations of AFP_2.

But $=_{\mathcal{D}_2^+}$ is not a congruence.

Let $P_1 = a \bigtriangledown b$, $P_1' = (a \bigtriangledown b)\|a\|b$ and $P_2 = c$.

Then $\mathcal{D}_2^+[P_1] = \mathcal{D}_2^+[P_1'] = \{\langle\{a\},\emptyset\rangle,\langle\{b\},\emptyset\rangle$ and $P_1 =_{\mathcal{D}_2^+} P_1'$.

But $\mathcal{D}_2^+[P_1;P_2] = \{\langle\{a,b\},\prec_1\rangle,\langle\{b,c\},\prec_2\rangle\}$, whereas $\mathcal{D}_2^+[P_1';P_2] = \{\langle\{a\},\emptyset\rangle,\langle\{b\},\emptyset\rangle\}$, and $P_1;P_2 \neq_{\mathcal{D}_2^+} P_1';P_2$.

Axiomatization

An axiom system Θ_2 is in accordance to the equivalence $=_{\mathcal{D}_2}$. Here $P,\ Q,\ R \in \mathbf{AFP_2}$, $a \in \alpha,\ \bar{a} \in \bar{\alpha},\ \delta_a \in \mathbf{\Delta}_\alpha$.

1. Associativity
   - 1.1 $P \| (Q \| R) = (P \| Q) \| R$
   - 1.2 $P \bigtriangledown (Q \bigtriangledown R) = (P \bigtriangledown Q) \bigtriangledown R$
   - 1.3 $P \vee (Q \vee R) = (P \vee Q) \vee R$
   - 1.4 $P; (Q; R) = (P; Q); R$
2. Commutativity
   - 2.1 $P \| Q = Q \| P$
   - 2.2 $P \bigtriangledown Q = Q \bigtriangledown P$
   - 2.3 $P \vee Q = Q \vee P$
3. Distributivity
   - 3.1 $(P \| Q); R = (P; R) \| (Q; R)$
   - 3.2 $P; (Q \| R) = (P; Q) \| (P; R)$
   - 3.3 $(P \vee Q); R = (P; R) \vee (Q; R)$
   - 3.4 $P; (Q \vee R) = (P; Q) \vee (P; R)$
   - 3.5 $(P \vee Q) \| R = (P \| R) \vee (Q \| R)$
   - 3.6 $P \bigtriangledown (Q \| R) = (P \bigtriangledown Q) \| (P \bigtriangledown R)$

4. Axioms for $\bigtriangledown$ and $\rceil\!\rceil$

4.1 $P \bigtriangledown Q = (P \| (\rceil\!\rceil Q)) \vee ((\rceil\!\rceil P) \| Q)$

4.2 $\rceil\!\rceil (P \| Q) = (\rceil\!\rceil P) \| (\rceil\!\rceil Q)$

4.3 $\rceil\!\rceil (P \vee Q) = (\rceil\!\rceil P) \vee (\rceil\!\rceil Q)$

4.4 $\rceil\!\rceil (P; Q) = (\rceil\!\rceil P) \| (\rceil\!\rceil Q)$

4.5 $\rceil\!\rceil a = \bar{a}$

4.6 $\rceil\!\rceil \bar{a} = \bar{a}$

4.7 $\rceil\!\rceil \delta_a = \bar{a}$

5. Structural properties

5.1 $\bar{a}; P = \bar{a} \| P$

5.2 $P; \bar{a} = P \| \bar{a}$

5.3 $P \| (P; Q) = (P; Q)$

5.4 $Q \| (P; Q) = (P; Q)$

5.5 $P; Q; R = (P; Q) \| (Q; R)$

5.6 $(P; Q) \| (Q; R) = (P; Q) \| (Q; R) \| (P; R)$

5.7 $P \| P = P$

5.8 $P \vee P = P$

5.9 $P \vee Q = P$ or $Q \lhd P$

6. Axioms for deadlocked actions and $\tilde{\prod}$

6.1 $a \| \bar{a} = \delta_a$

6.2 $a; a = \delta_a$

6.3 $a \| \delta_a = \delta_a$

6.4 $\delta_a; P = \delta_a \| (\tilde{\prod} P)$

6.5 $P; \delta_a = P \| \delta_a$

6.6 $\delta_a \| (\prod P) = \delta_a \| (\tilde{\prod} P)$

6.7 $\tilde{\prod} a = \delta_a$

6.8 $\tilde{\prod} \bar{a} = \delta_a$

6.9 $\tilde{\prod} \delta_a = \delta_a$

6.10 $\tilde{\prod}(P \| Q) = (\tilde{\prod} P) \| (\tilde{\prod} Q)$

6.11 $\tilde{\prod}(P; Q) = (\tilde{\prod} P) \| (\tilde{\prod} Q)$

6.12 $\tilde{\prod}(P \vee Q) = (\tilde{\prod} P) \vee (\tilde{\prod} Q)$

The axiom system Θ_2 is *sound* for $=_{\mathcal{D}_2}$, i.e. if $P = P'$ is an axiom of Θ_2, then $P =_{\mathcal{D}_2} P'$.

In order to prove that Θ_2 is *complete* for $=_{\mathcal{D}_2}$, we introduce a canonical form of formulas.

Canonical form of formulas

A canonical form of formulas of AFP_2 is a disjunctive normal form.

Elementary members: symbols from $\hat{\alpha}$ or elementary precedences (of two actions). Conjunction: $\|$, disjunction: $\vee$.

Let P be a formula of AFP_2. An *alphabet* of P, denoted by $\alpha(P)$, is defined as follows.

1. $\alpha(a) = \alpha(\bar{a}) = \alpha(\delta_a) = a$;
2. $\alpha(\neg P) = \alpha(P)$, $\neg \in \{\rceil\!\rceil, \tilde{\rceil\!\rceil}\}$;
3. $\alpha(P \circ Q) = \alpha(P) \cup \alpha(Q)$, $\circ \in \{;, \|, \triangledown, \vee\}$.

- $\bar{\alpha}(P) = \{\bar{a} \mid a \in \alpha(P)\}$;
- $\Delta_\alpha(P) = \{\delta_a \mid a \in \alpha(P)\}$;
- $\hat{\alpha}(P) = \alpha(P) \cup \bar{\alpha}(P) \cup \Delta_\alpha(P)$.

A *contents* of P, denoted by $cont(P)$, is defined as follows.

1. $cont(a) = a$, $cont(\bar{a}) = \bar{a}$, $cont(\delta_a) = \delta_a$;
2. $cont(\neg P) = cont(P)$, $\neg \in \{\rceil\!\rceil, \tilde{\rceil\!\rceil}\}$;
3. $cont(P \circ Q) = cont(P) \cup cont(Q)$, $\circ \in \{;, \|, \triangledown, \vee\}$.

- $cont^+(P) = cont(P) \cap \alpha$ is a set of *actions* of P;
- $cont^-(P) = cont(P) \cap \bar{\alpha}$ is a set of *non-actions* of P;
- $\Delta_{cont}(P) = cont(P) \cap \Delta_\alpha$ is a set of *deadlocked actions* of P.

A *precedence* is a formula $P_1; \dots ; P_n = ;_{i=1}^{n} P_i$, where $P_i \in \hat{\alpha}$ $(1 \le i \le n)$;

A *conjunction* is a formula $P_1 \| \dots \| P_n = \|_{i=1}^{n} P_i$, where P_i are precedences $(1 \le i \le n)$.

A *disjunction* is a formula $P = P_1 \vee \dots \vee P_n = \vee_{i=1}^{n} P_i$, where P_i $(1 \le i \le n)$ are conjunctions.

A *normal conjunction* is a conjunction $P = \|_{i=1}^{n} P_i$ with the following properties.

1. Every formula P_i $(1 \leq i \leq n)$ has one of the forms:
   (a) elementary formula a $(a \in \alpha)$, $\bar{a}$ $(\bar{a} \in \bar{\alpha})$, δ_a $(\delta_a \in \Delta_\alpha)$;
   (b) *elementary precedence* $(a; b)$, where $a, b \in \alpha$ and $a \neq b$;
2. If there is a formula P_i $(1 \leq i \leq n)$ s.t. $P_i = \delta_a$ $(\delta_a \in \Delta_\alpha)$, then there is no another one P_j $(1 \leq j \leq n)$ s.t. $P_j = \bar{b}$ $(\bar{b} \in \bar{\alpha})$;
3. For any formulas P_i and P_j $(1 \leq i \neq j \leq n)$ s.t. $\alpha(P_i) \cap \alpha(P_j) \neq \emptyset$, P_i and P_j have a form of different elementary precedences;
4. For any pair $P_i = (a; b)$ and $P_j = (b; c)$ $(1 \leq i \neq j \leq n)$ there exists a formula $P_k = (a; c)$ $(1 \leq k \leq n)$ describing the transitive closure of the precedence relation for actions a, b and c.

1 (2,3,4)-conjunction is a conjunction that satisfy the condition 1 (2,3,4) from the definition above.

For example, 1,2,3,4-conjunction is a normal one.

Let P and Q be normal conjunctions. A formula P is a *strict prefix* of Q, $P \triangleleft Q$, if the following holds.

1. $cont^{+}(P) \subset cont^{+}(Q)$;
2. elementary precedence $(a; b)$ is a conjunctive member of Q and $b \in cont^{+}(P)$ iff $(a; b)$ is a conjunctive member of P.

A formula P is a *prefix* of Q, $P \trianglelefteq Q$, if $P \triangleleft Q$ or $P \simeq Q$.

For example, in the formula $(a\|c\|\bar{b}\|\bar{d}\|\bar{e}) \vee (c\|\delta_a\|\delta_b\|\delta_d\|\delta_e) \vee (a\|\delta_b\|\delta_c\|\delta_d\|\delta_e) \vee ((b;d)\|(b;e)\|\bar{a}\|\bar{c})$, the second and third conjunctions are strict prefixes of the first one.

Definition 79 *A formula P is in canonical form if it is a disjunction $P = \vee_{i=1}^{n} P_i$ with the following properties.*

1. *P_i $(1 \le i \le n)$ is a normal conjunction;*

2. *for any P_i and P_j $(1 \le i \ne j \le n)$ $P_i \not\simeq P_j$;*

3. *for any P_i and P_j $(1 \le i \ne j \le n)$ $\neg(P_i \lhd P_j \vee P_j \lhd P_i)$.*

As for conjunction, we define *1* (*2*,*3*)*-disjunction*. For example, 1,2,3-disjunction is a canonical form.

Each disjunctive member of canonical form characterizes one of alternative behaviours of the nondeterministic process specified by the formula.

It has a form practically coinciding with a poset corresponding to this behaviour.

For example, the formula $(a\|c\|\bar{b}\|\bar{d}\|\bar{e}) \vee ((b;d)\|(b;e)\|\bar{a}\|\bar{c})$ is in canonical form.

A conjunction (disjunction) is *maximal* if there is no longer one containing it as a conjunctive (disjunctive) member.

Theorem 19 *[Ch89] Any formula of AFP_2 can be reduced to the unique (up to isomorphism) canonical form.*

The *set of all canonical forms* of a formula P is $canon(P)$.

Definition 80 *$P {=_{\ominus_2}} P'$ means that the equality of P and P' can be proved using $\ominus_2$.*

Theorem 20 *[Ch89] For any formulas P and P' of AFP_2: $P {=_{\mathcal{D}_2}} P' \Leftrightarrow P {=_{\ominus_2}} P'$.*

To check equivalence of formulas P and P' of AFP_2, one can reduce them to canonical forms Q and Q' and compare the latter up to isomorphism.

Equivalences on A-nets

A descriptive algebra AFP_0 with semantics based on finite A-nets [KCh85].

Any finite A-net can be specified by a formula of the algebra using "regularization" algorithm [Kot78].

A mapping Ψ from the set of all formulas of AFP_0 into that of AFP_2 s.t. the set of posets of the net specified by a formula P of AFP_0, coincide with the set of posets of nondeterministic process specified by the formula $\Psi(P)$ of AFP_2 [Ch89].

Hence, given the A-net specified by a formula P of AFP_0, one can analyze its behavior by means of the same formula P of AFP_2.

Definition 81 *An* A-net (Acyclic net) *is an acyclic ordinary strictly labeled net* $N = \langle P_N, T_N, F_N, l_N, M_N \rangle$ *with the following properties.*

1. $\forall p \in P_N \; ({}^\bullet p \neq \emptyset) \vee (p^\bullet \neq \emptyset)$, *i.e., there are no isolated places;*
2. $\forall p, q \in P_N \; ({}^\bullet p = {}^\bullet q) \wedge (p^\bullet = q^\bullet) \;\Rightarrow\; p = q$, *i.e., there are no "superfluous" places;*
3. $\forall t \in T_N \; ({}^\bullet t \neq \emptyset) \wedge (t^\bullet \neq \emptyset)$, *i.e., all transitions have input and output places;*
4. $\forall x \in P_N \cup T_N \; |\downarrow x| < \infty$, *i.e., the set of causes is finite;*
5. $\forall p \in P_N \; \forall t, u \in T_N \; t, u \in {}^\bullet p \;\Rightarrow\; t \; \mathbf{al} \; u$, *i.e., transitions with common output place are alternative;*
6. $M_N = {}^\circ N$, *i.e., an initial marking is a set of input places of the net.*

The *alternative* relation, denoted by $\mathbf{al}$, is defined as follows. Let $t, u \in T_N$ for A-net N. $t \ \mathbf{al} \ u$ if the following is valid.

1. $(t \not\prec_N u) \wedge (u \not\prec_N t)$;

2. $({}^\bullet t \cap {}^\bullet u \neq \emptyset) \vee (\exists p \in {}^\bullet t \ \forall t' \in {}^\bullet p \ t' \ \mathbf{al} \ u) \vee$
$(\exists q \in {}^\bullet u \ \forall u' \in {}^\bullet q \ t \ \mathbf{al} \ u') \vee (t = u)$.

Since we consider nets only with finite processes, item 4 may be ignored.

Items 5 and 6 guarantee a safeness of A-nets.

A mapping $\Psi : \mathbf{AFP_0} \to \mathbf{AFP_2}$ is defined as follows.

1. $\Psi(a) = a$,

2. $\Psi(P;_0 Q) = P;_2 Q$,

3. $\Psi(P\|_0 Q) = P\|_2 Q$,

4. $\Psi(P \triangledown_0 Q) = P \triangledown_2 Q$.

The number 0 (2) marks the operations of AFP_0 (AFP_2).

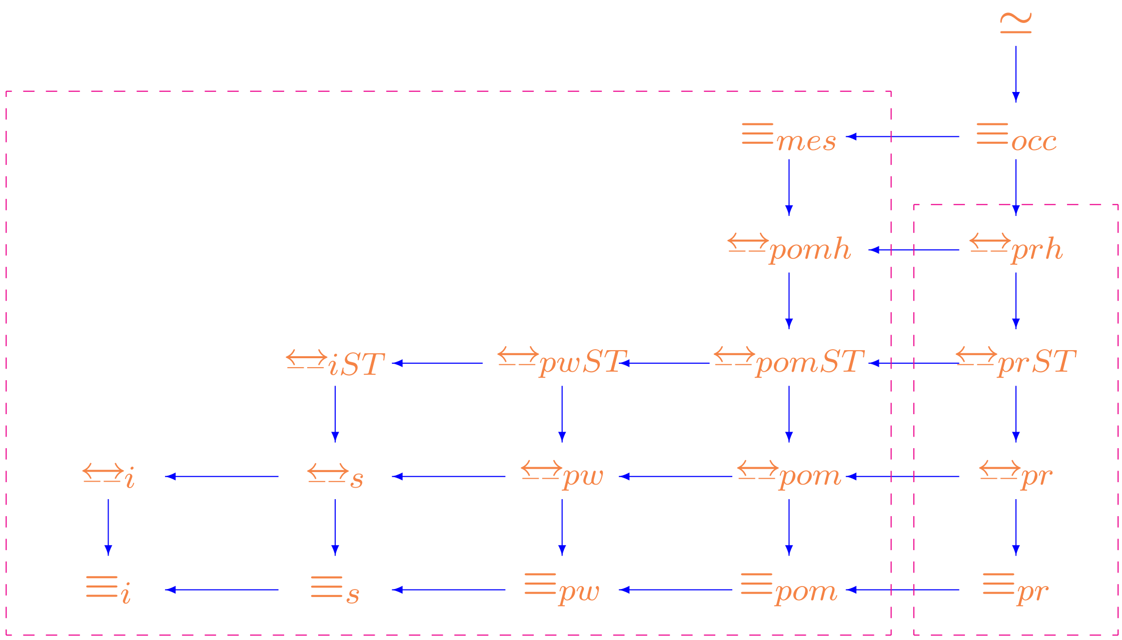
Denotational semantics of AFP_0 is a mapping $\mathcal{D}_0$, which associates with every formula P a set of maximal C-subnets of finite A-net N, specified by the formula.

Theorem 21 *[Ch89] Let P be a formula of AFP_0 and Q be a formula of AFP_2 s.t. $Q = \Psi(P)$. Then* $\{\rho_C \mid C \in \mathcal{D}_0[P]\} = \mathcal{D}_2^+[Q]$.

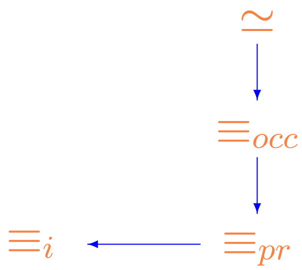
Proposition 17 [Tar97] *For A-nets N and N':*

1. $N \equiv_i N' \;\Leftrightarrow\; N \equiv_{mes} N'$;

2. $N \equiv_{pr} N' \;\Leftrightarrow\; N \underline{\leftrightarrow}_{prh} N'$.

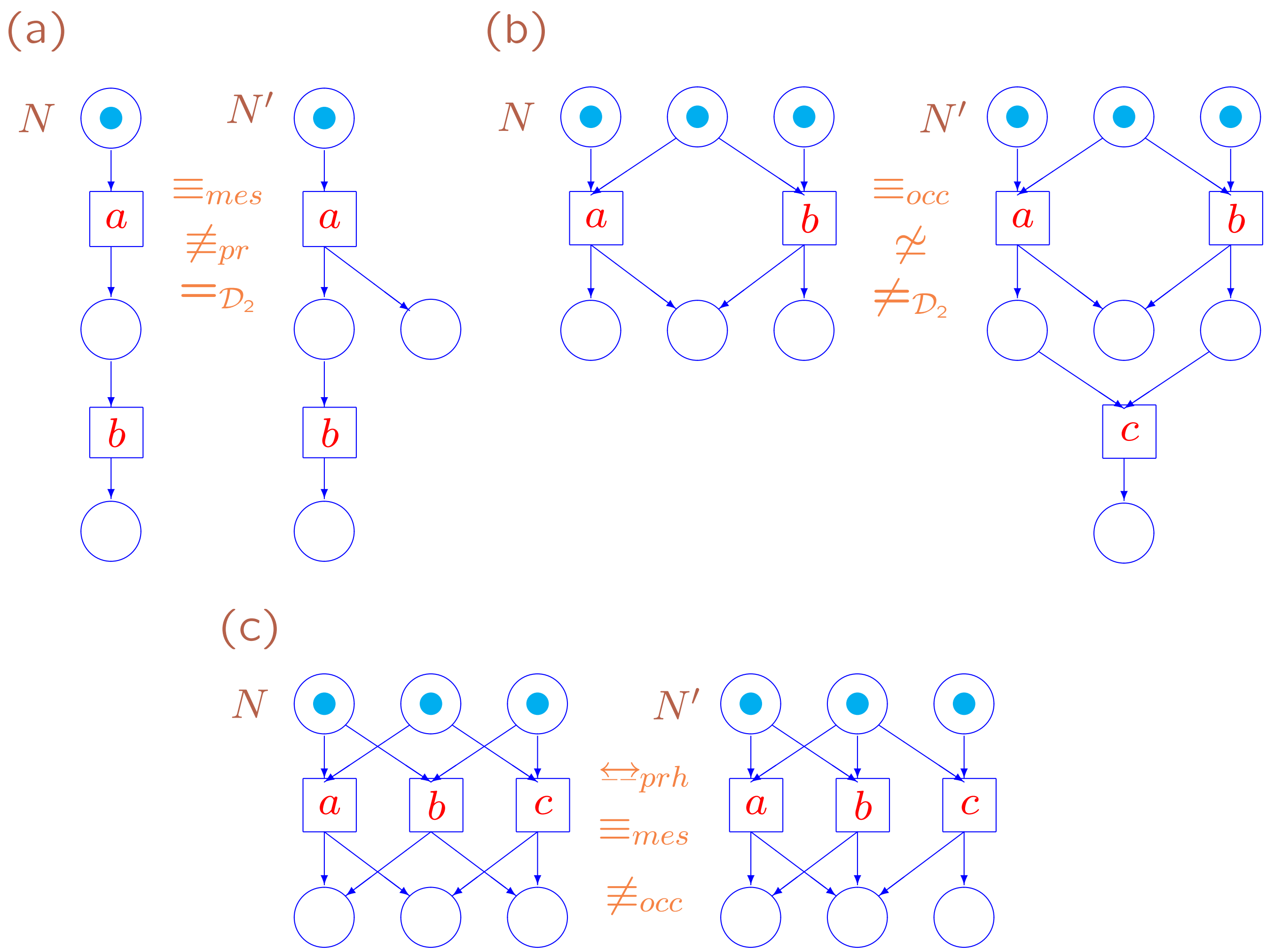


Merging of the basic equivalences on A-nets



Interrelations of the basic equivalences on A-nets

Theorem 22 *Let* $\leftrightarrow, \twoheadleftarrow\!\!\twoheadrightarrow \in \{\equiv, \simeq\}$, $\star, \star\star \in \{\_, i, pr, occ\}$. *For A-nets* N *and* N' $N \leftrightarrow_{\star} N' \;\Rightarrow\; N \twoheadleftarrow\!\!\twoheadrightarrow_{\star\star} N'$ *iff there exists a directed path from* $\leftrightarrow_{\star}$ *to* $\twoheadleftarrow\!\!\twoheadrightarrow_{\star\star}$ *in the graph above.*



AN: Examples of the basic equivalences on A-nets

- In Figure AN(a), $N \equiv_i N'$, but $N \not\equiv_{pr} N'$, since a causal net of process of N' with action a not isomorphic to any causal net of process of N.

  $P = a; b, \; P' = (a; b) \| a.$

- In Figure AN(c), $N \equiv_{pr} N'$, but $N \not\equiv_{occ} N'$, since only in the unfolding of N' there is a place which is an input one for three transitions.

  $P = (a \bigtriangledown b) \| (b \bigtriangledown c) \| (a \bigtriangledown c), \; P' = (a \bigtriangledown b \bigtriangledown c) \| (a \bigtriangledown b) \| c.$

- In Figure AN(b), $N \equiv_{occ} N'$, but $N \not\simeq N'$, since only in the net N' there is a transition labeled by c (which never fires).

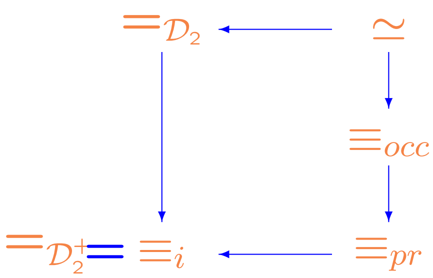
  $P = (a \bigtriangledown b) \| a \| b, \; P' = (a \bigtriangledown b) \| (a; c) \| (b; c).$

Comparing the net and algebraic equivalences

Definition 82 *Let $\leftrightarrow$ be a formula equivalence of AFP_2, and the formulas P and P' correspond to the finite A-nets N and N' (as described before).*

Two nets N and N' are equivalent (w.r.t. $\leftrightarrow$), notation $N \leftrightarrow N'$, iff the formulas corresponding them are also equivalent, i.e. $P \leftrightarrow P'$.

Proposition 18 [Tar97] *For A-nets N and N' $N \equiv_i N' \Leftrightarrow N =_{\mathcal{D}_2^+} N'$.*

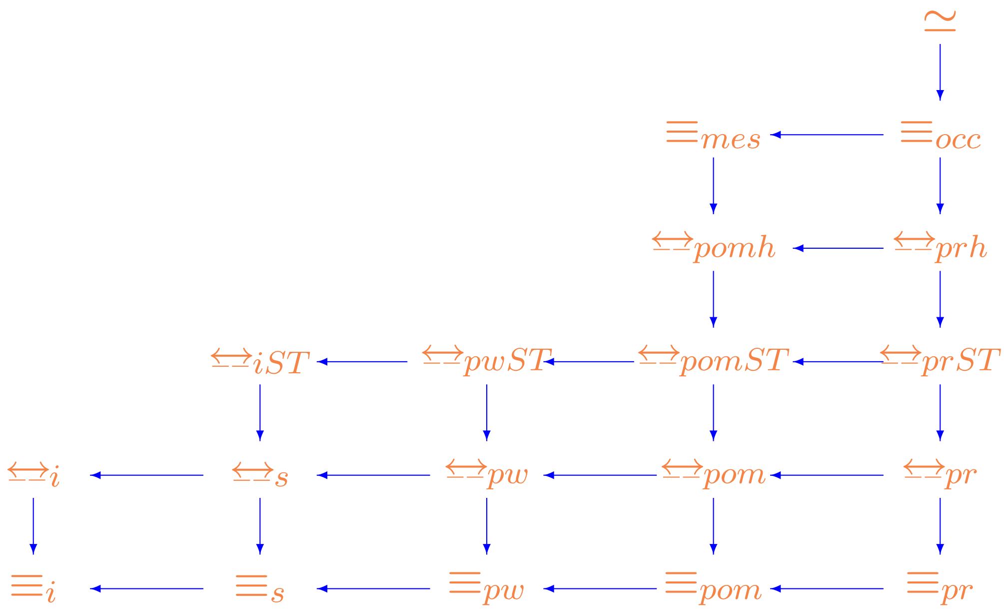


Interrelations of the basic net and algebraic equivalences

Theorem 23 *Let $\leftrightarrow, \leftrightarrow\!\!\!\!\!\!\twoheadleftarrow\!\!\twoheadrightarrow \in \{\equiv, \simeq, =\}$, $\star, \star\star \in \{\_, i, pr, occ, \mathcal{D}_2^+, \mathcal{D}_2\}$. For A-nets N and N' $N \leftrightarrow_\star N' \Rightarrow N \leftrightarrow_{\star\star} N'$ iff there exists a directed path from $\leftrightarrow_\star$ to $\twoheadleftarrow\!\!\twoheadrightarrow_{\star\star}$ in the graph above.*

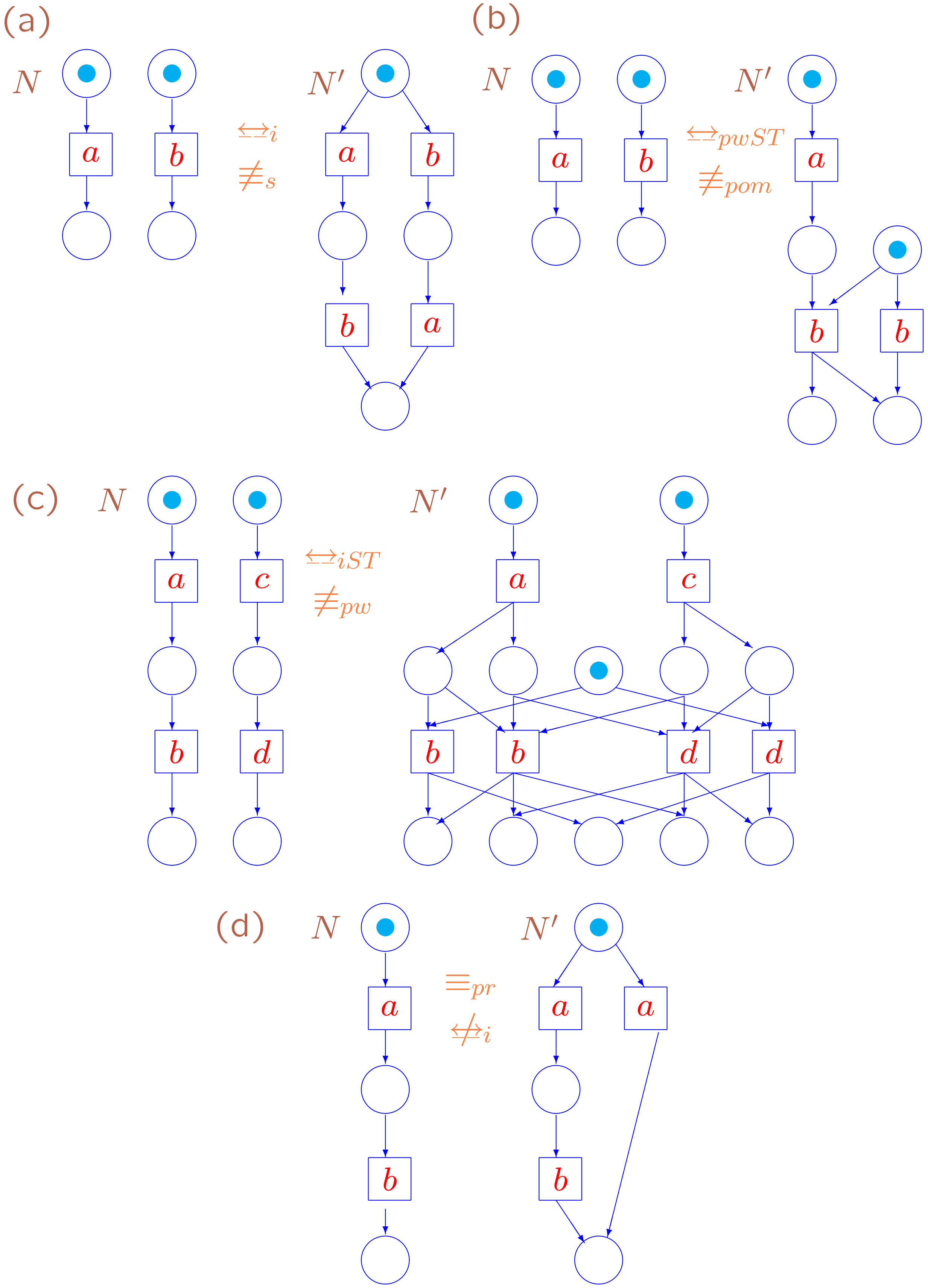
Equivalences on weakly labeled A-nets

Definition 83 *A* weakly labeled A-net *is an net with all properties of A-net with exception of strict labeling.*



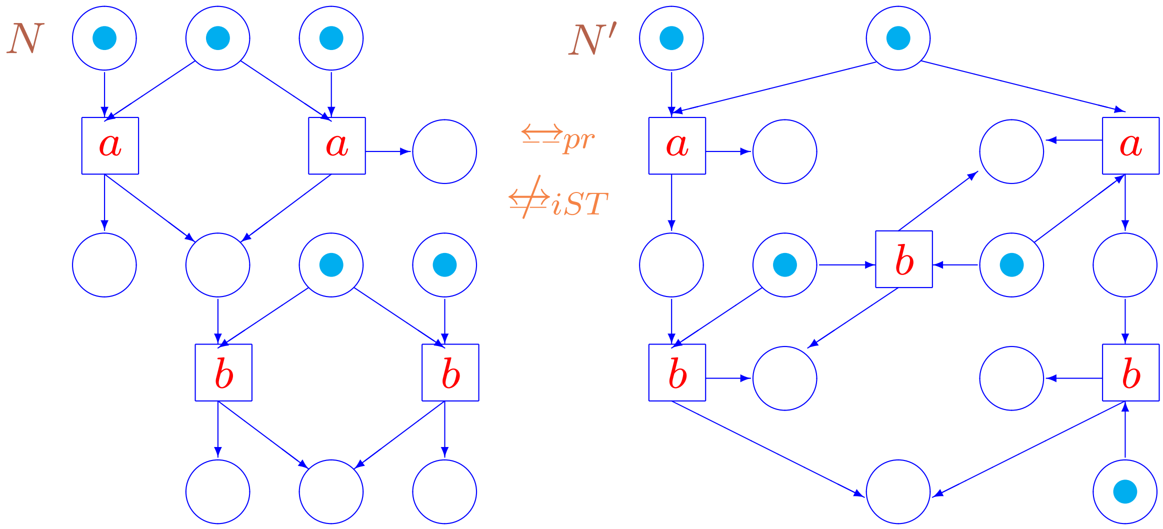
Interrelations of the basic equivalences on weakly labeled A-nets

Theorem 24 *Let* $\leftrightarrow, \twoheadleftarrow\!\!\twoheadrightarrow \in \{\equiv, \underline{\leftrightarrow}, \simeq\}$, $\star, \star\star \in \{\_, i, s, pw, pom, pr, iST, pwST, pomST, prST, pomh, prh, mes, occ\}$*. For weakly labeled A-nets* N *and* N' $N \leftrightarrow_{\star} N' \Rightarrow N \twoheadleftarrow\!\!\twoheadrightarrow_{\star\star} N'$ *iff there exists a directed path from* $\leftrightarrow_{\star}$ *to* $\twoheadleftarrow\!\!\twoheadrightarrow_{\star\star}$ *in the graph above.*

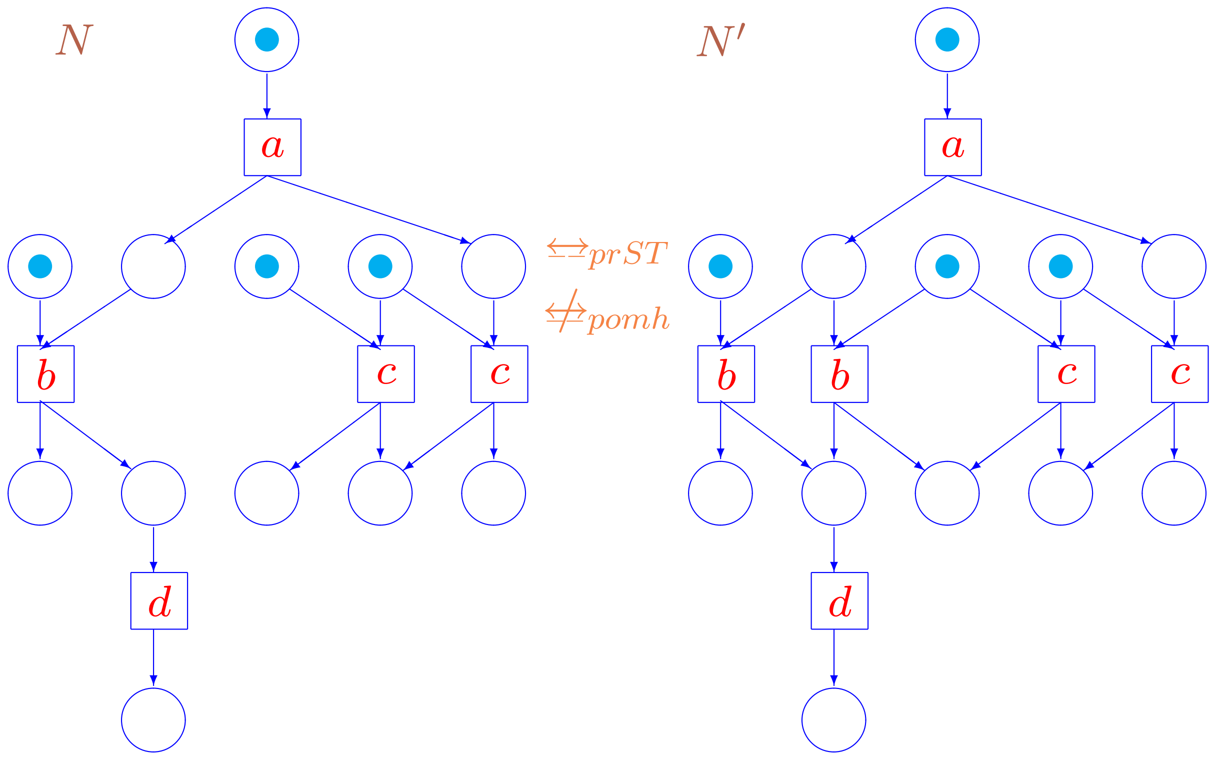


LAN: Examples of weakly labeled A-nets

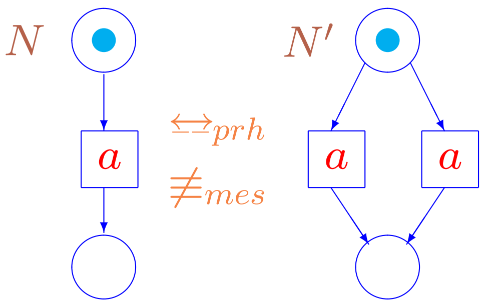
(a)



(b)



(c)



LAN1: Examples of weakly labeled A-nets (continued)

In the following examples, $E,\ E'$ are formulas of $AFLP_2$ [Tar96] and $B,\ B'$ are that of PBC [BDH92].

- In Figure LAN(a), $N \underline{\leftrightarrow}_i N'$, but $N \not\equiv_s N'$, since only in N' actions a and b can be executed concurrently.

  $E = e \| f,\ \ E' = (e_1; f_1) \bigtriangledown (e_2; f_2)$.

  $B = a \| b,\ \ B' = (a; b) [\!] (b; a)$.

- In Figure LAN(b), $N \underline{\leftrightarrow}_{pwST} N'$, but $N \not\equiv_{pom} N'$, since only in N' action b can depend on a.

  $E = e \| f,\ \ E' = (e; f_1) \| (f_1 \bigtriangledown f_2)$.

  $B = a \| b,\ \ B' = x : ((a; \{b, x\}) \| (b [\!] \widehat{x}))$.

- In Figure AN(a), $N \equiv_{mes} N'$, but $N \not\equiv_{pr} N'$.

  $E = e; f,\ \ E' = (e; f) \| e$.

  $B = a; b,\ \ B' = x : ((\{a, x\}; b) \| \widehat{x})$.

- In Figure LAN(d), $N \equiv_{pr} N'$, but $N \not\underline{\leftrightarrow}_i N'$, since only in N' action a can happen so that b cannot happen after it.

  $E = e; f,\ \ E' = (e_1; f) \bigtriangledown e_2$.

  $B = a; b,\ \ B' = (a; b) [\!] a$.

- In Figure LAN1(a), $N \underline{\leftrightarrow}_{pr} N'$, but $N \not\underline{\leftrightarrow}_{iST} N'$, since only in N' action a can begin working so that no b can start unless a finishes.

  $E = ((e_1 \triangledown e_2); f_1) \| (f_1 \triangledown f_2) \| e_1 \| e_2 \| f_2,$
  $E' = ((e_1; f_1) \triangledown (e_2; f_3)) \| (f_1 \triangledown f_2) \| (e_2 \triangledown f_2) \| e_1 \| f_3.$

  $B = \{x_1, x_2, y_1, y_2\} : (((\{a, x_1\} [] \{a, x_2\}); \{b, y_1\}) \|$
  $(\widehat{y}_1 [] \{b, y_2\}) \| \widehat{x}_1 \| \widehat{x}_2 \| \widehat{y}_2),$
  $B' = \{x_1, x_2, y_1, y_2, y_3\} : ((\{a, x_1\}; \{b, y_1\}) [] (\{a, x_2\};$
  $\{b, y_3\})) \| (\widehat{y}_1 [] \{b, y_2\}) \| (\widehat{x}_2 [] \widehat{y}_2) \| \widehat{x}_1 \| \widehat{y}_3).$

- In Figure LAN1(b), $N \underline{\leftrightarrow}_{prST} N'$, but $N \not\underline{\leftrightarrow}_{pomh} N'$, since only in N' actions a and b can happen so that the next action, c, must depend on a.

  $E = (e; f; h) \| (e; g_2) \| (g_1 \triangledown g_2) \| f \| g_1,$
  $E' = (e; (f_1 \triangledown f_2); h) \| (e; g_2) \| (f_2 \triangledown g_1) \| (g_1 \triangledown g_2) \| f_1.$

  $B = \{x, y, z_1, z_2\} : ((\{a, x\}; \{b, y\}; d) \| (\widehat{x}; \{c, z_2\}) \|$
  $(\{c, z_1\} [] \widehat{z}_2) \| \widehat{y} \| \widehat{z}_1),$
  $B' = \{x, y_1, y_2, z_1, z_2\} : (\{a, x\}; (\{b, y_1\} [] \{b, y_2\}); d) \|$
  $(\widehat{x}; \{c, z_2\}) \| (\widehat{y}_2 [] \{c, z_1\}) \| (\widehat{z}_1 [] \widehat{z}_2) \| \widehat{y}_1).$

- In Figure LAN1(c), $N \underline{\leftrightarrow}_{prh} N'$, but $N \not\equiv_{mes} N'$, since only MES that corresponding to N' has two conflict actions a.

  $E = e, \;\; E' = e_1 \triangledown e_2.$

  $B = a, \;\; B' = a [] a.$

- In Figure AN(b), $N \equiv_{occ} N'$, but $N \not\simeq N'$.

  $E = (e \triangledown f) \| e \| f, \;\; E' = (e \triangledown f) \| (e; g) \| (f; g).$

  $B = \{x, y\} : ((\{a, x\} [] \{b, y\}) \| \widehat{x} \| \widehat{y}),$
  $B' = \{x, y, z\} : ((\{a, x\} [] \{b, y\}) \| (\widehat{x}; \{c, z\}) \| (\widehat{y}; \widehat{z})).$

Term rewrite system RWS_2

A *substitution* of subformula P_i of a formula P by another subformula Q, denoted by $[P]^{P_i}_Q$, is the formula $P_1 \circ \ldots P_{i-1} \circ Q \circ P_{i+1} \circ \ldots \circ P_n$, where $P = P_1 \circ \ldots \circ P_{i-1} \circ P_i \circ P_{i+1} \circ \ldots \circ P_n, \ \circ \in \{;, \|, \triangledown, \vee\}$.

In the following rules of RWS_2, P, Q, R denote formulas of AFP_2 and $a, b, c \in \alpha, \ \bar{a}, \bar{b} \in \bar{\alpha}, \ \delta_a, \delta_b \in \Delta_\alpha$. The numbers in parentheses are the that of equalities of Θ_2 used to produce the corresponding transition rules.

1.1 $\circ \in \{;, \|, \vee\} \ \Rightarrow$

$P \circ (Q \circ R) \to (P \circ Q) \circ R$

(1.1, 1.3, 1.4)

2.1 $(\bullet, \circ) \in \{(\|, ;), (\vee, ;), (\vee, \|)\} \ \Rightarrow$

$(P \circ Q) \bullet R \to (P \bullet R) \circ (Q \bullet R)$

(3.1, 3.3, 3.5)

2.2 $(\bullet, \circ) \in \{(\|, ;), (\vee, ;), (\vee, \|)\} \ \Rightarrow$

$P \bullet (Q \circ R) \to (P \bullet Q) \circ (P \bullet R)$

(2.1, 3.2, 3.4, 3.5)

3.1 $P \triangledown Q \to (P \| (\urcorner\urcorner Q)) \vee ((\urcorner\urcorner P) \| Q)$

(4.1)

4.1 $\circ \in \{\|, ;\}, \neg \in \{\urcorner\urcorner, \tilde{\urcorner\urcorner}\} \ \Rightarrow$

$\neg(P \circ Q) \to (\neg P) \| (\neg Q)$

(4.2, 4.4, 6.10, 6.11)

4.2 $\neg \in \{\urcorner\urcorner, \tilde{\urcorner\urcorner}\} \ \Rightarrow$

$\neg(P \vee Q) \to (\neg P) \vee (\neg Q)$

(4.3, 6.12)

4.3 $P = a$ or $P = \bar{a}$ or $P = \delta_a \;\Rightarrow$

$\prod P \rightarrow \bar{a}$

(4.5, 4.6, 4.7)

4.4 $P = a$ or $P = \bar{a}$ or $P = \delta_a \;\Rightarrow$

$\tilde{\prod} P \rightarrow \delta_a$

(6.7, 6.8, 6.9)

5.1 $P, Q, R \in \hat{\alpha} \;\Rightarrow$

$(P; Q); R \rightarrow ((P; Q) \| (Q; R)) \| (P; R)$

(5.5, 5.6)

5.2 $Q \in \hat{\alpha} \;\Rightarrow$

$\bar{a}; Q \rightarrow \bar{a} \| Q$

(5.1)

5.3 $P \in \hat{\alpha} \;\Rightarrow$

$P; \bar{a} \rightarrow P \| \bar{a}$

(5.2)

5.4 $a; a \rightarrow \delta_a$

(6.2)

5.5 $Q = b$ or $Q = \bar{b}$ or $Q = \delta_b \;\Rightarrow$

$\delta_a; Q \rightarrow \delta_a \| \delta_b$

(6.4, 6.7, 6.8, 6.9)

5.6 $P \in \hat{\alpha} \;\Rightarrow$

$P; \delta_a \rightarrow P \| \delta_a$

(6.5)

6.1 P is 1-conjunction, $P' = \delta_a$ is a conjunctive member of P $\Rightarrow$

$P \| \overline{b} \to P \| \delta_b$

(1.1, 2.1, 4.5, 6.6, 6.7)

6.2 P is 1-conjunction, $P' = \overline{b}$ is a conjunctive member of P $\Rightarrow$

$P \| \delta_a \to [P]^{P'}_{\delta_b} \| \delta_a$

(1.1, 2.1, 4.5, 6.6, 6.7)

7.1 P is 1,2-conjunction, P' is a conjunctive member of P, $P' = a$ or $P' = b$ $\Rightarrow$

$P \| (a; b) \to [P]^{P'}_{(a;b)}$

(1.1, 2.1, 5.3, 5.4)

7.2 P is 1,2-conjunction, P' is a conjunctive member of P, $P' = (a; b)$ or $P' = (b; a)$ $\Rightarrow$

$P \| a \to P$

(1.1, 2.1, 5.3, 5.4)

7.3 P is 1,2-conjunction, $P' = a$ is a conjunctive member of P, $\Diamond \in \{^-, \delta\}$ $\Rightarrow$

$P \| \Diamond a \to [P]^{P'}_{\delta_a}$

(1.1, 2.1, 6.1, 6.3)

7.4 P is 1,2-conjunction, P' is a conjunctive member of P, $P' = \bar{a}$ or $P' = \delta_a$ $\Rightarrow$

$P \| a \to [P]^{P'}_{\delta_a}$

(1.1, 2.1, 6.1, 6.3)

7.5 P is 1,2-conjunction, $P' = (a; b)$ is a conjunctive member of P, $\Diamond \in \{^-, \delta\}$ $\Rightarrow$

$P \| \Diamond a \to [P]^{P'}_{\delta_b} \| \delta_a$

(1.1, 1.4, 2.1, 5.1, 6.1, 6.3, 6.4, 6.7)

7.6 P is 1,2-conjunction, $P' = (b;a)$ is a conjunctive member of P, $\diamondsuit \in \{^-, \delta\}$ $\Rightarrow$

$P \| \diamondsuit a \rightarrow [P]_b^{P'} \| \delta_a$

(1.1, 2.1, 5.2, 6.1, 6.3, 6.5)

7.7 P is 1,2-conjunction, P' is a conjunctive member of P, $P' = \bar{a}$ or $P' = \delta_a$ $\Rightarrow$

$P \| (a;b) \rightarrow [P]_{\delta_a}^{P'} \| \delta_b$

(1.1, 1.4, 2.1, 5.1, 6.1, 6.3, 6.4, 6.7)

7.8 P is 1,2-conjunction, P' is a conjunctive member of P, $P' = \bar{a}$ or $P' = \delta_a$ $\Rightarrow$

$P \| (b;a) \rightarrow [P]_{\delta_a}^{P'} \| b$

(1.1, 2.1, 5.2, 6.1, 6.3, 6.5)

7.9 P is 1,2-conjunction, $P' = Q$ is a conjunctive member of P $\Rightarrow$

$P \| Q \rightarrow P$

(1.1, 2.1, 5.7)

8.1 P is 1,2,3-conjunction, $P' = (a;b)$ is a conjunctive member of P, in the maximal 1,2,3-conjunction containing P as a conjunctive member, there is no conjunctive member $P'' = (a;c)$ $\Rightarrow$

$P \| (b;c) \rightarrow (P \| (b;c)) \| (a;c)$

(1.1, 2.1, 5.6)

8.2 P is 1,2,3-conjunction, $P' = (c;a)$ is a conjunctive member of P, in the maximal 1,2,3-conjunction containing P as a conjunctive member there is no conjunctive member $P'' = (b;a)$ $\Rightarrow$

$P \| (b;c) \rightarrow (P \| (b;c)) \| (b;a)$

(1.1, 2.1, 5.6)

9.1 P is 1-disjunction, P' is a disjunctive member of P, $P' \simeq Q \Rightarrow$

$P \vee Q \rightarrow P$

(1.1, 1.3, 2.1, 2.3, 5.8)

10.1 P is 1,2-disjunction, Q is a normal conjunction, P' is a disjunctive member of P, $Q \lhd P' \Rightarrow$

$P \vee Q \rightarrow P$

(1.3, 2.3, 5.9)

10.2 P is 1,2-disjunction, Q is a normal conjunction, P' is a disjunctive member of P, $P' \lhd Q \Rightarrow$

$P \vee Q \rightarrow [P]_{Q}^{P'}$

(1.3, 2.3, 5.9)

Notices on RWS_2

- Rule 1.1 (left associativity): to avoid infinite chains $P\circ(Q\circ R) \to (P\circ Q)\circ R \to P\circ(Q\circ R) \to \cdots,\ \circ \in \{;\, ,\|,\vee\}$.

  No commutativity rules: to avoid infinite chains $P \circ Q \to Q \circ P \to P \circ Q \to \cdots,\ \circ \in \{\|,\vee\}$.

  Symmetrical rules are required.

- Rules 2.1-2.2 (symmetrical distributivity): to obtain disjunction of conjunctions with precedences or elementary formulas as conjunctive members.

- Rule 3.1: to remove $\triangledown$.

- Rules 4.1–4.4: to remove $\rceil\!\rceil$ and $\tilde{\rceil\!\rceil}$.

- Rules 5.1–5.6: to transform precedences into elementary ones (property 1 of normal conjunction).

  Conjunctive (disjunctive) members we want to transform in a pair are not always adjacent: search in conjunction (disjunction) is required.

- Rules 6.1–6.2: to avoid conjunction of non-actions and deadlocked actions (property 2 of normal conjunction).

- Rules 7.1–7.9: to avoid common alphabet symbols in conjunctive members, with exception of that in two different elementary precedences (property 3 of normal conjunction).

- Rules 8.1–8.2: to add a "transitive closure" elementary precedence to the pair of ones with common action (property 4 of normal conjunction).

  Search in a maximal conjunction: to avoid infinite chains $(a;b)\|(b;c) \to ((a;b)\|(b;c))\|(a;c) \to (((a;b)\|(b;c))\|(a;c))\|(a;c) \to \cdots$.

- Rule 9.1: to remove isomorphic disjunctive members (property 1 of normal disjunction).

- Rules 10.1–10.2: to remove prefixed disjunctive members (property 2 of normal disjunction).

Rules 6.1–6.2 and 7.5–7.8 are based on the following derived axioms. Numbers over equality signs are that of axioms of Θ_2. Symbol $*$ marks reverse axiom application. Numbers in parentheses are that of previous derived axioms.

1. $\bar{a}\|(a;b) \overset{5.1*}{=} \bar{a};(a;b) \overset{1.4}{=} (\bar{a};a);b \overset{5.1}{=} (\bar{a}\|a);b \overset{2.1}{=} (a\|\bar{a});b \overset{6.1}{=} \delta_a;b \overset{6.4}{=} \delta_a\|(\tilde{\prod}b) \overset{6.7}{=} \delta_a\|\delta_b$;

2. $\delta_a\|(a;b) \overset{6.1*}{=} (a\|\bar{a})\|(a;b) \overset{1.1*}{=} a\|(\bar{a}\|(a;b)) \overset{(1)}{=} a\|(\delta_a\|\delta_b) \overset{1.1}{=} (a\|\delta_a)\|\delta_b \overset{6.3}{=} \delta_a\|\delta_b$;

3. $\bar{a}\|(b;a) \overset{2.1}{=} (b;a)\|\bar{a} \overset{5.2*}{=} (b;a);\bar{a} \overset{1.1*}{=} b;(a;\bar{a}) \overset{5.2}{=} b;(a\|\bar{a}) \overset{6.1}{=} b;\delta_a \overset{6.5}{=} b\|\delta_a \overset{2.1}{=} \delta_a\|b$;

4. $\delta_a\|(b;a) \overset{6.1*}{=} (a\|\bar{a})\|(b;a) \overset{1.1*}{=} a\|(\bar{a}\|(b;a)) \overset{(3)}{=} a\|(\delta_a\|b) \overset{1.1}{=} (a\|\delta_a)\|b \overset{6.3}{=} \delta_a\|b$;

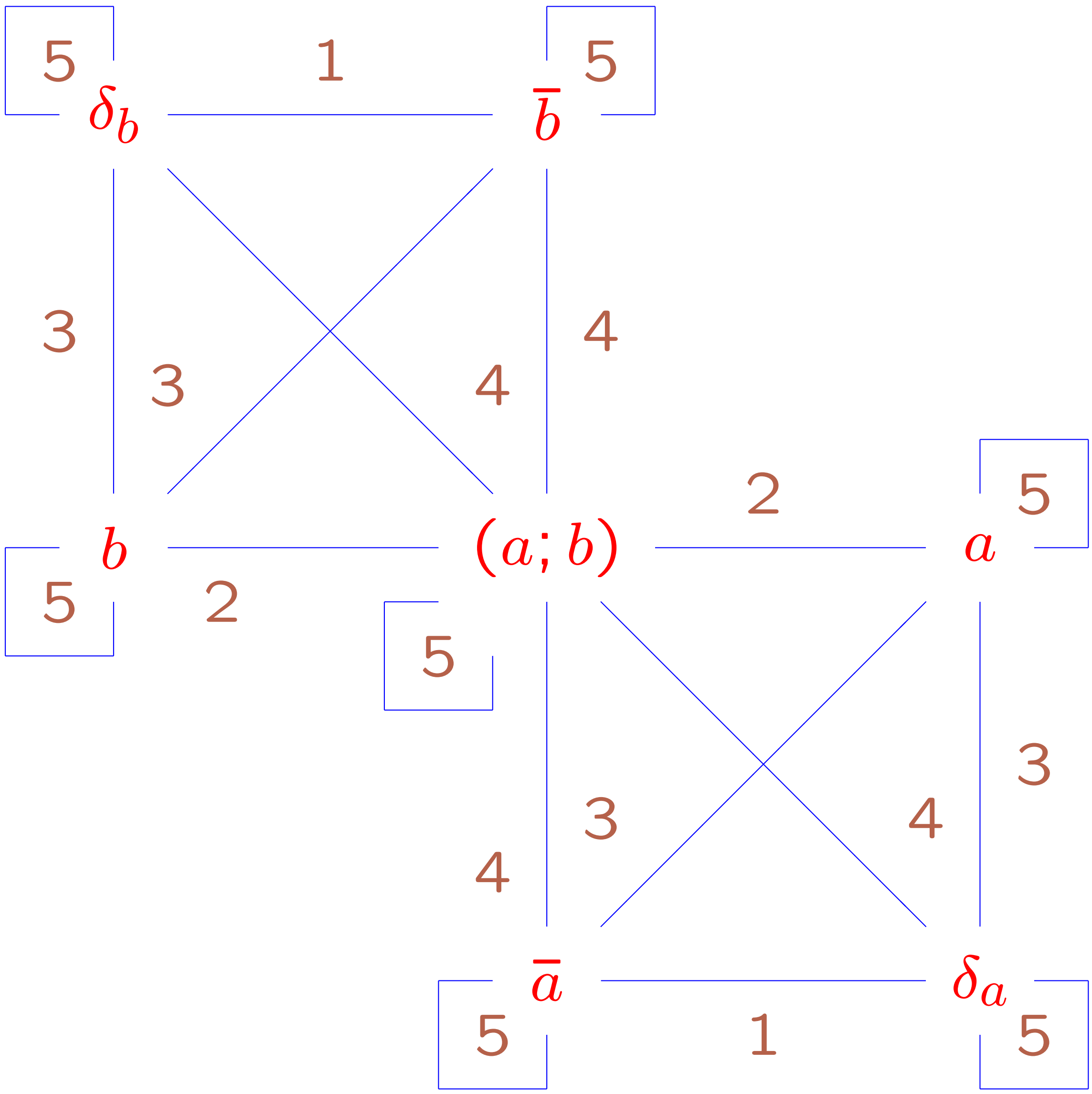
5. $\delta_a\|\bar{b} \overset{4.5*}{=} \delta_a\|(\prod b) \overset{6.6}{=} \delta_a\|(\tilde{\prod}b) \overset{6.7}{=} \delta_a\|\delta_b$.

Confluence of RWS_2

Proposition 19 [Tar97] *No rule of the groups 1–5 can be applied to a formula P of AFP_2 iff it is a disjunction of 1-conjunctions.*

Proposition 20 [Tar97] *No rule of the groups 1–6 can be applied to a formula P of AFP_2 iff it is a disjunction of 1,2-conjunctions.*

Proposition 21 [Tar97] *No rule of the groups 1–7 can be applied to a formula P of AFP_2 iff it is a disjunction of 1,2,3-conjunctions.*



Conjunctive members with intersecting alphabets

Proposition **22** [Tar97] *No rule of the groups 1–8 can be applied to a formula P of AFP_2 iff it is a 1-disjunction.*

Proposition **23** [Tar97] *No rule of the groups 1–9 can be applied to a formula P of AFP_2 iff it is a 1,2-disjunction.*

Theorem **25** [Tar97] *No rule of a RWS_2 can be applied to a formula P of AFP_2 iff it is in a canonical form.*

Hence, to check semantic equivalence of two formulas of AFP_2, it is enough to transform them to canonical forms with the use of RWS_2 and then compare these canonical forms by isomorphism.

Program $CANON$

A program $CANON$ in C programming language (about 3000 lines) based on the previous results. It transforms any formula of AFP_2 into canonical form.

A structure of function main.

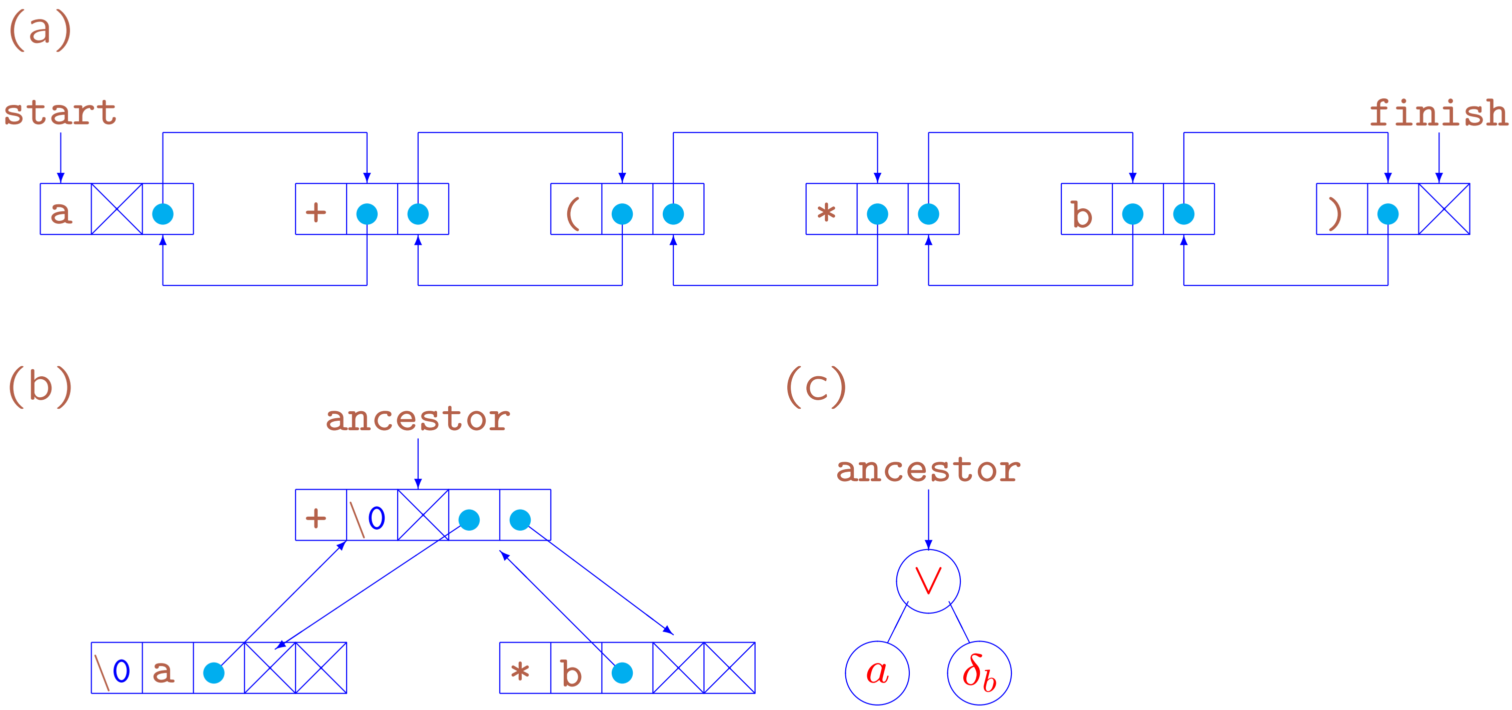
```
Print information about program
and format of input formula;

Print "Formula has been read";

Transform list into tree;

Dispose list;

Print formula;

step=1; /*step number*/

do
{
 Print step;

 nar=0; /*the initial number of rule
          applications at the present step*/

 Apply rules;

 Print nar;

 step++; /*next step*/
}
while(nar!=0);

Print canonical form.
```

Initial symbol	$\overline{\ }$	δ	Π	$\tilde{\Pi}$	;	‖	$\triangledown$	$\vee$
Symbol constant	NOT	DLT	NOC	MNO	PRC	CNC	ALT	DSJ
ASCII-symbol	-	*	‘	~	;	|	#	+

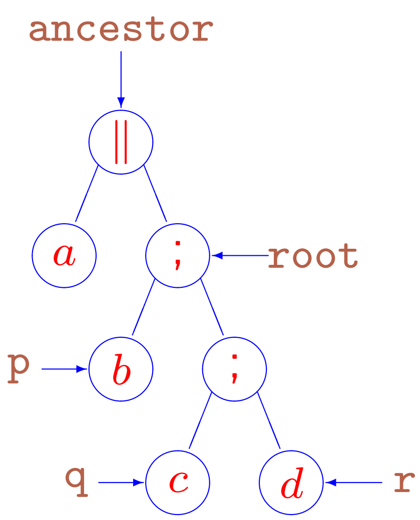
Special symbol representation in *CANON*

A structure of formulas.

1. a;
2. `-a , *a ;`
3. `‘a , ~a ;`
4. `‘(P) , ~(P) ;`
5. `a#b , a+b , a|b , a;b ;`
6. `a#(P) , a+(P) , a|(P) , a;(P) ;`
7. `(P)#a , (P)+a , (P)|a , (P);a ;`
8. `(P)#(Q) , (P)+(Q) , (P)|(Q) , (P);(Q) .`



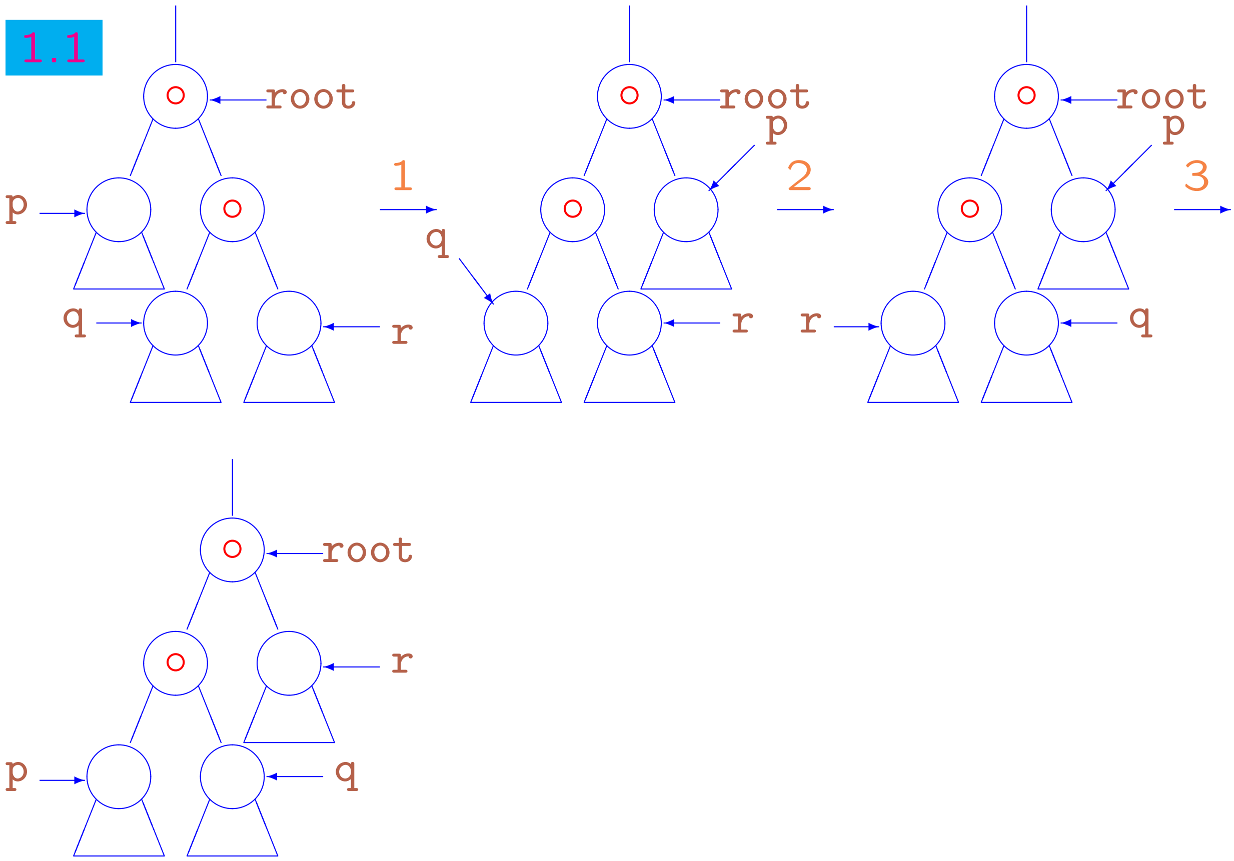
List and tree representations of the formula $a \vee \delta_b$

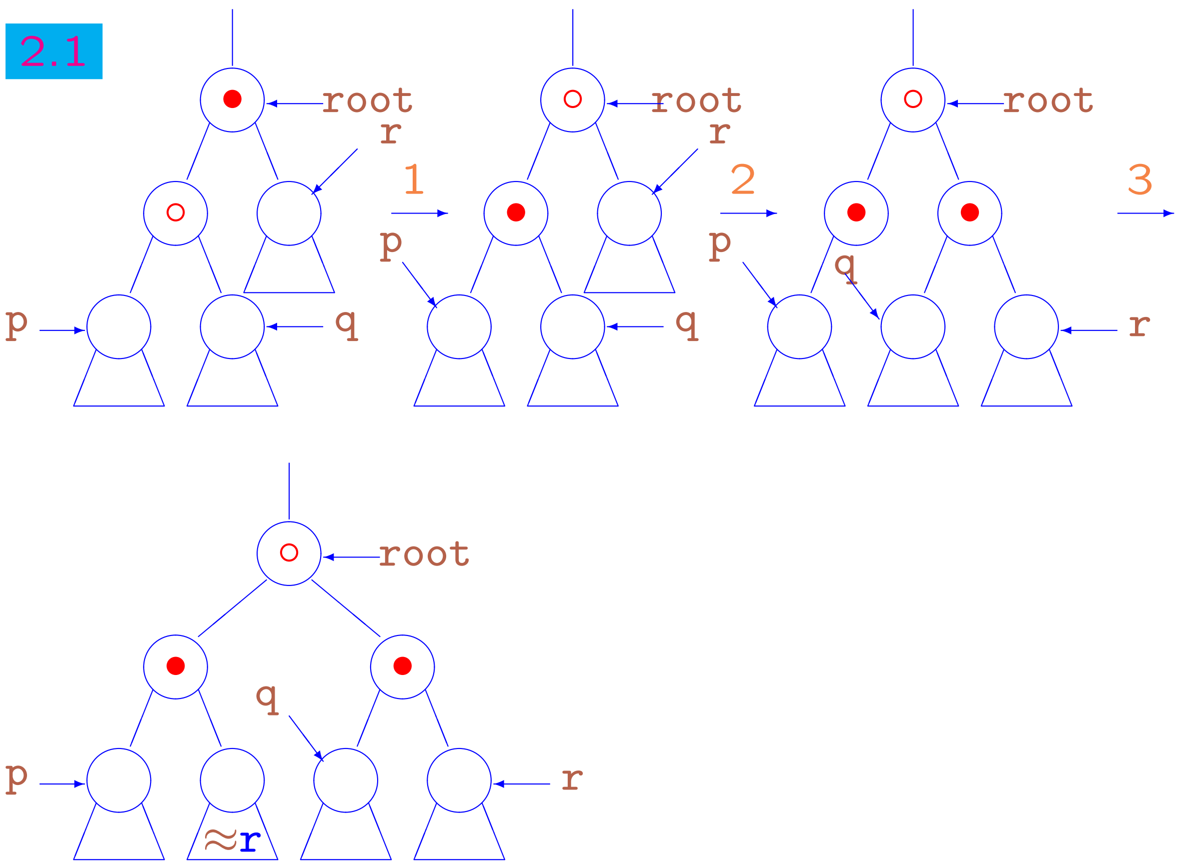


A tree to which the rule 1.1 can be applied

A structure of rules.

```
if(root!=NULL)
{
 if(the rule is directly applicable
    to the tree with pointer root)
 {
  Set pointers to subtrees corresponding
  to subformulas in the rule;

  Print rule number and subformulas;

  Transform tree in accordance to the rule;

 (*addrnar)++; /*increase counter of rules
                applied at the the present step*/

 Print new formula;
 }
 else
 {
  Apply rule to the left subtree;

  Apply rule to the right subtree;
 }
}
```





Tree transformations with rules 1.1 and 2.1

Examples of formula transformation with *CANON*

The initial formula: $(a \bigtriangledown (b;e))\|(d \bigtriangledown (c;e))$.

```
The author of this program is I.V. Tarasyuk
Program CANON transforms formulas of algebras AFP_2, AFLP_2
into canonical form

Input formula should be in one of the following forms:

1. a
2. -a  *a
3. ‘a  ~a
4. ‘(P)  ~(P)
5. a;b  a|b  a#b  a+b
6. a;(P)  a|(P)  a#(P)  a+(P)
7. (P);a  (P)|a  (P)#a  (P)+a
8. (P);(Q)  (P)|(Q)  (P)#(Q)  (P)+(Q)

where a and b are symbols of elementary actions,
P and Q are formulas types 2-8

Input formula
Sign of end is EOF

Formula has been read

Your formula is:
(a#(b;e))|(d#(c;e))

Step 1

Rule 3.1 is applied
P=a
Q=(b;e)
New formula is:
((a|(‘(b;e)))+((‘a)|(b;e)))|(d#(c;e))

Rule 3.1 is applied
P=d
Q=(c;e)
New formula is:
((a|(‘(b;e)))+((‘a)|(b;e)))|((d|(‘(c;e)))+((‘d)|(c;e)))

Rule 4.1 is applied
P=b
Q=e
New formula is:
((a|((‘b)|(‘e)))+((‘a)|(b;e)))|((d|(‘(c;e)))+((‘d)|(c;e)))
```

```
Rule 4.1 is applied
P=c
Q=e
New formula is:
((a|((‘b)|(‘e))+((‘a)|(b;e)))|((d|((‘c)|(‘e))+((‘d)|(c;e)))

Rule 4.3 is applied
P=b
New formula is:
((a|((-b)|(‘e))+((‘a)|(b;e)))|((d|((‘c)|(‘e))+((‘d)|(c;e)))

Rule 4.3 is applied
P=e
New formula is:
((a|((-b)|(-e))+((‘a)|(b;e)))|((d|((‘c)|(‘e))+((‘d)|(c;e)))

Rule 4.3 is applied
P=a
New formula is:
((a|((-b)|(-e))+((-a)|(b;e)))|((d|((‘c)|(‘e))+((‘d)|(c;e)))

Rule 4.3 is applied
P=c
New formula is:
((a|((-b)|(-e))+((-a)|(b;e)))|((d|((-c)|(‘e))+((‘d)|(c;e)))

Rule 4.3 is applied
P=e
New formula is:
((a|((-b)|(-e))+((-a)|(b;e)))|((d|((-c)|(-e))+((‘d)|(c;e)))

Rule 4.3 is applied
P=d
New formula is:
((a|((-b)|(-e))+((-a)|(b;e)))|((d|((-c)|(-e))+((-d)|(c;e)))

Number of applied rules in step 1 is 10

Step 2

Rule 1.1 is applied
P=a
Q=(-b)
R=(-e)
New formula is:
(((a|(-b))|(-e))+((-a)|(b;e)))|((d|((-c)|(-e))+((-d)|(c;e)))
```

```
Rule 1.1 is applied
P=d
Q=(-c)
R=(-e)
New formula is:
(((a|(-b))|(-e))+((-a)|(b;e)))|(((d|(-c))|(-e))+((-d)|(c;e)))

Rule 2.1 is applied
P=((a|(-b))|(-e))
Q=((-a)|(b;e))
R=(((d|(-c))|(-e))+((-d)|(c;e)))
New formula is:
(((a|(-b))|(-e))|(((d|(-c))|(-e))+((-d)|(c;e))))+
(((-a)|(b;e))|(((d|(-c))|(-e))+((-d)|(c;e))))

Rule 2.2 is applied
P=((a|(-b))|(-e))
Q=((d|(-c))|(-e))
R=((-d)|(c;e))
New formula is:
((((a|(-b))|(-e))|((d|(-c))|(-e)))+(((a|(-b))|(-e))|((-d)|(c;e))))+
(((-a)|(b;e))|(((d|(-c))|(-e))+((-d)|(c;e))))

Rule 2.2 is applied
P=((-a)|(b;e))
Q=((d|(-c))|(-e))
R=((-d)|(c;e))
New formula is:
((((a|(-b))|(-e))|((d|(-c))|(-e)))+(((a|(-b))|(-e))|((-d)|(c;e))))+
((((-a)|(b;e))|((d|(-c))|(-e)))+(((-a)|(b;e))|((-d)|(c;e))))

Number of applied rules in step 2 is 5

Step 3

Rule 1.1 is applied
P=((((a|(-b))|(-e))|((d|(-c))|(-e)))+(((a|(-b))|(-e))|((-d)|(c;e))))
Q=(((-a)|(b;e))|((d|(-c))|(-e)))
R=(((-a)|(b;e))|((-d)|(c;e)))
New formula is:
(((((a|(-b))|(-e))|((d|(-c))|(-e)))+(((a|(-b))|(-e))|((-d)|(c;e))))+
(((-a)|(b;e))|((d|(-c))|(-e))))+(((-a)|(b;e))|((-d)|(c;e)))

Number of applied rules in step 3 is 1
```

Step 4

Rule 1.1 is applied
P=((a|(-b))|(-e))
Q=(d|(-c))
R=(-e)
New formula is:
((((((a|(-b))|(-e))|(d|(-c)))|(-e))+(((a|(-b))|(-e))|((-d)|(c;e)))+
(((-a)|(b;e))|((d|(-c))|(-e))))+(((-a)|(b;e))|((-d)|(c;e)))

Rule 1.1 is applied
P=((a|(-b))|(-e))
Q=(-d)
R=(c;e)
New formula is:
((((((a|(-b))|(-e))|(d|(-c)))|(-e))+((((a|(-b))|(-e))|(-d))|(c;e)))+
(((-a)|(b;e))|((d|(-c))|(-e))))+(((-a)|(b;e))|((-d)|(c;e)))

Rule 1.1 is applied
P=((-a)|(b;e))
Q=(d|(-c))
R=(-e)
New formula is:
((((((a|(-b))|(-e))|(d|(-c)))|(-e))+((((a|(-b))|(-e))|(-d))|(c;e)))+
((((-a)|(b;e))|(d|(-c)))|(-e)))+(((-a)|(b;e))|((-d)|(c;e)))

Rule 1.1 is applied
P=((-a)|(b;e))
Q=(-d)
R=(c;e)
New formula is:
((((((a|(-b))|(-e))|(d|(-c)))|(-e))+((((a|(-b))|(-e))|(-d))|(c;e)))+
((((-a)|(b;e))|(d|(-c)))|(-e)))+((((-a)|(b;e))|(-d))|(c;e))

Number of applied rules in step 4 is 4

Step 5

Rule 1.1 is applied
P=((a|(-b))|(-e))
Q=d
R=(-c)
New formula is:
(((((((a|(-b))|(-e))|d)|(-c))|(-e))+((((a|(-b))|(-e))|(-d))|(c;e)))+
((((-a)|(b;e))|(d|(-c)))|(-e)))+((((-a)|(b;e))|(-d))|(c;e))

```
Rule 1.1 is applied
P=((-a)|(b;e))
Q=d
R=(-c)
New formula is:
(((((((a|(-b))|(-e))|d)|(-c))|(-e))+((((a|(-b))|(-e))|(-d))|(c;e)))+
(((((-a)|(b;e))|d)|(-c))|(-e)))+((((-a)|(b;e))|(-d))|(c;e))

Number of applied rules in step 5 is 2

Step 6

Rule 7.6 is applied
P=((((-a)|(b;e))|d)|(-c))
P'=(b;e)
Q=(-e)
New formula is:
(((((((a|(-b))|(-e))|d)|(-c))|(-e))+((((a|(-b))|(-e))|(-d))|(c;e)))+
(((((-a)|b)|d)|(-c))|(*e)))+((((-a)|(b;e))|(-d))|(c;e))

Rule 7.8 is applied
P=(((a|(-b))|(-e))|(-d))
P'=(-e)
Q=(c;e)
New formula is:
(((((((a|(-b))|(-e))|d)|(-c))|(-e))+((((a|(-b))|(*e))|(-d))|c))+
(((((-a)|b)|d)|(-c))|(*e)))+((((-a)|(b;e))|(-d))|(c;e))

Rule 7.9 is applied
P=((((a|(-b))|(-e))|d)|(-c))
P'=(-e)
Q=(-e)
New formula is:
((((((a|(-b))|(-e))|d)|(-c))+((((a|(-b))|(*e))|(-d))|c))+
(((((-a)|b)|d)|(-c))|(*e)))+((((-a)|(b;e))|(-d))|(c;e))

Number of applied rules in step 6 is 3

Step 7

Rule 6.1 is applied
P=((a|(-b))|(*e))
P'=(*e)
Q=(-d)
New formula is:
((((((a|(-b))|(-e))|d)|(-c))+((((a|(-b))|(*e))|(*d))|c))+
(((((-a)|b)|d)|(-c))|(*e)))+((((-a)|(b;e))|(-d))|(c;e))
```

```
Rule 6.2 is applied
P=((a|(-b))|(*e))
P'=(-b)
Q=(*d)
New formula is:
(((((a|(-b))|(-e))|d)|(-c))+((((a|(*b))|(*e))|(*d))|c))+
(((((-a)|b)|d)|(-c))|(*e)))+((((-a)|(b;e))|(-d))|(c;e))

Rule 6.2 is applied
P=((((-a)|b)|d)|(-c))
P'=(-c)
Q=(*e)
New formula is:
(((((a|(-b))|(-e))|d)|(-c))+((((a|(*b))|(*e))|(*d))|c))+
(((((-a)|b)|d)|(*c))|(*e)))+((((-a)|(b;e))|(-d))|(c;e))

Number of applied rules in step 7 is 3

Step 8

Rule 6.2 is applied
P=((((-a)|b)|d)|(*c))
P'=(-a)
Q=(*e)
New formula is:
(((((a|(-b))|(-e))|d)|(-c))+((((a|(*b))|(*e))|(*d))|c))+
(((((*a)|b)|d)|(*c))|(*e)))+((((-a)|(b;e))|(-d))|(c;e))

Number of applied rules in step 8 is 1

Step 9

Number of applied rules in step 9 is 0

Canonical form is:
(((((a|(-b))|(-e))|d)|(-c))+((((a|(*b))|(*e))|(*d))|c))+
(((((*a)|b)|d)|(*c))|(*e)))+((((-a)|(b;e))|(-d))|(c;e))
```

Canonical form is: $(a\|d\|\bar{b}\|\bar{c}\|\bar{e}) \vee (a\|c\|\delta_b\|\delta_d\|\delta_e) \vee (b\|d\|\delta_a\|\delta_c\|\delta_e) \vee ((b;e)\|(c;e)\|\bar{a}\|\bar{d})$.

References

[ABS91] C. Autant, Z. Belmesk, Ph. Schnoebelen. *Strong bisimularity on nets revisited. Extended abstract.* *LNCS* **506**, pages 295–312, June 1991.

[APS94] C. Autant, W. Pfister, Ph. Schnoebelen. *Place bisimulations for the reduction of labelled Petri nets with silent moves.* *Proceedings of International Conference on Computing and Information*, 1994.

[AS92] C. Autant, Ph. Schnoebelen. *Place bisimulations in Petri nets.* *LNCS* **616**, pages 45–61, June 1992.

[Aut93] C. Autant. *Petri nets for the semantics and the implementation of parallel processes.* *Ph.D. thesis*, Institut National Polytechnique de Grenoble, May 1993 (in French).

[BCa87] G. Boudol, I. Castellani. *On the semantics of concurrency: partial orders and transition systems.* *LNCS* **249**, pages 123–137, 1987.

[BDH92] E. Best, R. Devillers, J.G. Hall. *The box calculus: a new causal algebra with multi-label communication.* *LNCS* **609**, pages 21–69, 1992.

[BDKP91] E. Best, R. Devillers, A. Kiehn, L. Pomello. *Concurrent bisimulations in Petri nets.* *Acta Informatica* **28**, pages 231–264, 1991.

[BK84] J.A. Bergstra , J.W. Klop. *Process algebra for synchronous communication.* *Information and Control* **60**, pages 109–137, 1984.

[BK89] J.A. Bergstra, J.W. Klop. *Process theory based on bisimulation semantics.* *LNCS* **354**, pages 50–122, 1989.

[Ch89] L.A. Cherkasova *Posets with non-actions: a model for concurrent nondeterministic processes.* *Arbeitspapiere der GMD* **403**, 68 pages, Germany, July 1989.

[Che92a] F. Cherief. *Back and forth bisimulations on prime event structures.* *LNCS* **605**, pages 843–858, June 1992.

[Che92b] F. Cherief. *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).

[Che92c] F. Cherief. *Investigations of back and forth bisimulations on prime event structures.* *Computers and Artificial Intelligence* **11**(5), pages 481–496, 1992.

[CHM93] S. Christensen, Y. Hirshfeld, F. Moller. *Bisimulation equivalence is decidable for basic parallel processes.* *LNCS* **715**, pages 143–157, 1993.

[CLP92] F. Cherief, F. Laroussinie, S. Pinchinat. *Modal logics with past for true concurrency.* *Proceedings of CONCUR'92*, February 1992.

[Dev92] R. Devillers. *Maximality preservation and the ST-idea for action refinements.* *LNCS* **609**, pages 108–151, 1992.

[Eng85] J. Engelfriet. *Determinacy $\rightarrow$ (observation equivalence $=$ trace equivalence).* *TCS* **36**, pages 21–25, 1985.

[Eng91] J. Engelfriet. *Branching processes of Petri nets.* *Acta Informatica* **28**(6), pages 575–591, 1991.

[Gla90] R.J. van Glabbeek. *The linear time – branching time spectrum. Extended abstract.* *LNCS* **458**, pages 278–297, 1990.

[Gla93] R.J. van Glabbeek. *The linear time – branching time spectrum II: the semantics of sequential systems with silent moves. Extended abstract.* *LNCS* **715**, pages 66–81, 1993.

[Gra81] J. Grabowski. *On partial languages.* *Fundamenta Informaticae* **IV(2)**, pages 428–498, 1981.

[GV87] R.J. van Glabbeek, F.W. Vaandrager. *Petri net models for algebraic theories of concurrency.* *LNCS* **259**, pages 224–242, 1987.

[HM85] M. Hennessy, R.A.J. Milner. *Algebraic laws for nondeterminism and concurrency.* *Journal of the ACM* **32**(1), pages 137–161, January 1985.

[Hoa80] C.A.R. Hoare. *Communicating sequential processes, on the construction of programs.* (McKeag R.M., Macnaghten A.M. eds.) Cambridge University Press, pages 229–254, 1980.

[Hoa85] C.A.R. Hoare. *Communicating sequential processes.* Prentice-Hall, London, 1985.

[Jan94] P. Jančar. *Decidability questions for bisimularity of Petri nets and some related problems.* *LNCS* **775**, pages 581–594, 1994.

[Jan95] P. Jančar. *High decidability of weak bisimularity for Petri nets.* *LNCS* **915**, pages 349–363, 1995.

[JM96] L. Jategaonkar, A.R. Meyer. *Deciding true concurrency equivalences on safe, finite nets.* *TCS* **154**, pages 107–143, 1996.

[KCh85] V.E. Kotov, L.A. Cherkasova. *On structural properties of generalized processes.* *LNCS* **188**, pages 288–306, 1985.

[Kot78] V.E. Kotov. *An algebra for parallelism based on Petri nets.* *LNCS* **64**, pages 39–55, 1978.

[Mil80] R.A.J. Milner. *A calculus of communicating systems.* *LNCS* **92**, pages 172–180, 1980.

[Mil83] R.A.J. Milner. *Calculi for synchrony and asynchrony.* *TCS* **25**, pages 267–310, 1983.

[NMV90] R. De Nicola, U. Montanari, F.W. Vaandrager. *Back and forth bisimulations.* *LNCS* **458**, pages 152–165, 1990.

[NPW81] M. Nielsen, G. Plotkin, G. Winskel. *Petri nets, event structures and domains.* *TCS* **13**, pages 85–108, 1981.

[NT84] M. Nielsen, P.S. Thiagarajan. *Degrees of non-determinism and concurrency: A Petri net view.* *LNCS* **181**, pages 89–117, December 1984.

[Old87a] E.-R. Olderog. *TCSP: theory of communicating sequential processes.* *LNCS* **255**, pages 441–465, 1987.

[Old87b] E.-R. Olderog. *Operational Petri net semantics for CCSP.* *LNCS* **266**, pages 196–223, 1987.

[Old89b] E.-R. Olderog. *Strong bisimularity on nets: a new concept for comparing net semantics.* *LNCS* **354**, pages 549–573, 1989.

[Old91] E.-R. Olderog. *Nets, terms and formulas.* *Cambridge Tracts in Theoretical Computer Science* **23**, Cambridge University Press, 1991.

[Par81] D.M.R. Park. *Concurrency and automata on infinite sequences.* *LNCS* **104**, pages 167–183, March 1981.

[Pfi92] W. Pfister. *Simplification sémantique des réseaux de Petri par la bisimulation de places.* *Technical Report of DEA*, University of Grenoble, France, June 1992 (in French).

[Pin93] S. Pinchinat. *Bisimulations for the semantics of reactive systems.* *Ph.D. thesis*, Institut National Politechnique de Grenoble, January 1993 (in French).

[Pom86] L. Pomello. *Some equivalence notions for concurrent systems. An overview.* *LNCS* **222**, pages 381–400, 1986.

[Pra86] V.R. Pratt. *The pomset model of parallel processes: unifying the temporal and the spatial.* *LNCS* **197**, pages 180–196, 1986.

[PRS92] L. Pomello, G. Rozenberg, C. Simone. *A survey of equivalence notions for net based systems.* *LNCS* **609**, pages 410–472, 1992.

[RT88] A. Rabinovitch, B.A. Trakhtenbrot. *Behaviour structures and nets.* *Fundamenta Informaticae* **XI**, pages 357–404, 1988.

[Tar96] I.V. Tarasyuk. *Algebra $AFLP_2$: a calculus of labelled nondeterministic processes.* *Hildesheimer Informatik-Berichte* **4/96**, part 2, 18 pages, Institut für Informatik, Universität Hildesheim, Germany, January 1996.

[Tar97] I.V. Tarasyuk. *Equivalence notions for models of concurrent and distributed systems.* *Ph.D. thesis*, 191 pages, Institute of Informatics Systems, Novosibirsk, 1997 (in Russian).

[Tar98a] I.V. Tarasyuk. *τ-equivalences and refinement.* Proceedings of International Refinement Workshop and Formal Methods Pacific - 98 (IRW/FMP'98), Work-in-Progress Papers, Canberra, Australia, September 29 – October 2, 1998, Grundy, Jim; Schwenke, Martin and Vickers, Trevor, eds., *Joint Computer Science Technical Report Series* **TR-CS-98-09**, The Australian National University, pages 110–128, September 1998.

[Tar98b] I.V. Tarasyuk. *Place bisimulation equivalences for design of concurrent and sequential systems.* Proceedings of MFCS'98 Workshop on Concurrency, Brno, Czech Republic, August 27–29, 1998, *Electronic Notes in Theoretical Computer Science* **18**, 16 pages, 1998 (`http://www.elsevier.nl/locate/entcs/volume18.html`).

[Vog91a] W. Vogler. *Bisimulation and action refinement.* *LNCS* **480**, pages 309–321, 1991.

[Vog91b] W. Vogler. *Deciding history preserving bisimilarity.* *LNCS* **510**, pages 495–505, 1991.

[Vog92] W. Vogler. *Modular construction and partial order semantics of Petri nets.* *LNCS* **625**, 252 pages, 1992.