Reducing k-safe Petri Nets to Pomset-equivalent 1-safe Petri Nets

Eike Best[1] and Harro Wimmel[2]

[1] Carl von Ossietzky Universität Oldenburg, D-26111 Oldenburg
[2] Universität Koblenz-Landau, D-56016 Koblenz

Abstract. It is a well-known fact that for every k-safe Petri net, i.e. a Petri net in which no place contains more than $k \in \mathbb{N}$ tokens under any reachable marking, there is a 1-safe Petri net with the same interleaving behaviour. Indeed these types of Petri nets generate regular languages. In this paper, we show that this equivalence of k-safe and 1-safe Petri nets holds also for their pomset languages, a true-concurrency semantics.

Keywords: Causality / partial order theory of concurrency, Petri nets, Pomsets.

1 Introduction

The well-known straightforward constructions of turning a k-safe Petri net into an 'equivalent' 1-safe net preserve its interleaving behaviour, but not its partial order behaviour [Bes84]. Indeed, it has been an open question whether or not the pomsets [Pra84] of a k-safe Petri net are also the pomsets of a 1-safe Petri net. The present paper answers the question affirmatively. It does so via coloured Petri nets [Jen92, Rei98, Smi98]. A k-safe Petri net is first translated into a coloured Petri net whose unfolding is 1-safe and back into an uncoloured net, and it is shown that all nets involved in this construction have the same pomsets.

This result is presented here in a self-contained fashion, but is related strongly to [PW97], [PW98] where the pomset languages of general Petri nets are investigated.

2 Nets and their pomset languages

Let Σ denote some fixed alphabet of actions. We frequently use a, b, ... as names for actions. We may or may not use a silent action $\tau \notin \Sigma$; it will not have any influence on the result of this paper.

For a relation R on some set A, $R \subseteq A \times A$, let $R^+ = R \cup R \circ R \cup R \circ R \circ R \ldots$ denote the transitive closure of R. We call R a partial order on A if it is irreflexive and transitive. Given a partial order R on A, we call $b, c \in A$ *concurrent*, b co_R c, if neither bRc nor cRb. A set $C \subseteq A$ is called a *chain* if bRc or cRb for all $b, c \in C$, and it is a *line* if additionally all $a \in A \backslash C$ are concurrent with some $c \in C$; i.e., a line is a maximal chain. Further, we call $C \subseteq A$ an *antichain* if all $b, c \in C$ are concurrent and a *cut* if additionally for all $a \in A \backslash C$ there is some $c \in C$ with aRc or cRa; i.e., a cut is a maximal antichain. $C \subseteq A$ is a B-cut for some $B \subseteq A$ if C is a cut in A and $C \subseteq B$.

Any mapping $h: A \to B$ with arbitrary sets A and B may be canonically extended to a mapping on multisets $h': \mathbb{N}^A \to \mathbb{N}^B$ by defining

$$h'(f)(b) = \sum_{a \in h^{-1}(b)} f(a)$$

for any $f: A \to \mathbb{N}$ and $b \in B$. We will not distinguish between h and h'. If in doubt, a mapping should be interpreted as multiset mapping. Moreover, we can perform the following operations on multisets $X, Y \in \mathbb{N}^U$:

- add them: $(X{+}Y)(a) = X(a){+}Y(a)$ for all $a{\in}U$,
- subtract them: $(X{-}Y)(a) = \max\{0, X(a){-}Y(a)\}$,
- multiply them by a scalar $k{\in}\mathbb{N}$: $(k{\cdot}X)(a) = k{\cdot}X(a)$,
- compare them: $(X{\geq}Y) \iff X(a){\geq}Y(a)$ for all $a{\in}U$,
- form their union: $(X{\cup}Y)(a) = \max\{X(a), Y(a)\}$,
- their intersection: $(X{\cap}Y)(a) = \min\{X(a), Y(a)\}$,
- or their difference: $(X{\setminus}Y)(a) =$ **if** $Y(a){=}0$ **then** $X(a)$ **else** 0 **fi**.

Definition 2.1. *Posets, causality structures, pomsets*

A *partially ordered set (poset)* is a pair (E, R) of a finite set E and a partial order $R \subseteq E{\times}E$. We call e_1 a *predecessor* of e_2 (and e_2 a *successor* of e_1) if $e_1 R e_2$ holds. Further, e_1 is a *direct predecessor* of e_2 (and e_2 is the *direct successor* of e_1) if additionally there is no e with $e_1 R e$ and $e R e_2$.

A *causality structure κ over Σ* is a triple $\kappa = (E_\kappa, R_\kappa, \ell_\kappa)$ consisting of a poset (E_κ, R_κ) and a labelling function $\ell_\kappa: E_\kappa \to \Sigma$. Two causality structures κ_1, κ_2 are *isomorphic* if there is a bijection $\beta: E_{\kappa_1} \to E_{\kappa_2}$ with $\beta \circ R_{\kappa_2} \circ \beta^{-1} = R_{\kappa_1}$ and $\beta \circ \ell_{\kappa_2} = \ell_{\kappa_1}$.[1] A *pomset (partially ordered multiset)* φ is an isomorphism class of causality structures. By $[\kappa]$ we denote the pomset containing the causality structure κ as a representative. □ 2.1

Graphically, a pomset $[(E, R, \ell)]$ will be presented as a directed graph whose nodes are labelled by elements of the multiset $\ell(E)$ and whose directed arcs, $\longrightarrow$, represent the direct predecessor relation. For an example see figure 1.

$\varphi = [(E, R, \ell)]$ with

$E = \{1, 2, 3, 4, 5, 6, 7, 8\}$

$R = \{(1,4), (2,4), (1,5), (4,5)$
$\quad (5,7), (4,6), (3,6), (6,8)\}^+,$

$\ell \;:\; 1,7 \mapsto a;\; 2,4 \mapsto b;\; 5,6 \mapsto c;$
$\quad 3 \mapsto d;\; 8 \mapsto e.$

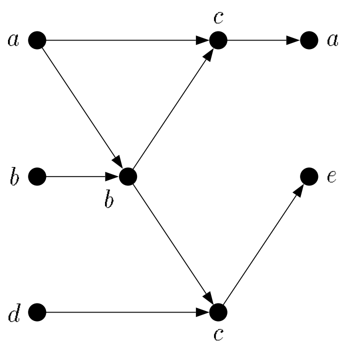

Figure 1. Formal and graphical view of a pomset φ.

We use pomsets as a generalisation of finite words, and thus we do not consider pomsets with an infinite number of events.

[1] We interpret $\circ$ 'relationally', i.e. $(x, y){\in}R{\circ}Q \iff \exists z: (x, z){\in}R \wedge (z, y){\in}Q$.

In this paper we will use (finite, labelled) coloured Petri nets [Jen92, Rei98, Smi98], which will simply be called nets below. Our definitions are parametrised with respect to arbitrary but fixed, mutually disjoint, sets Σ (of actions), VAL (of values) and MOD (of modes). The first set is the target of transition labellings, and values and modes play the role of place colours and transition colours, respectively.

Definition 2.2. *Petri net*

An (unmarked, unlabelled, finite) *net* is a triple (S, T, ι) where S and T are finite sets with $S \cap T = \emptyset$ and ι is a function

$$\iota: S \cup (S \times T) \cup (T \times S) \cup T \to 2^{VAL} \cup \mathbb{N}^{VAL \times MOD} \cup \mathbb{N}^{MOD \times VAL} \cup 2^{MOD}$$

such that:

- $\forall s \in S: \iota(s) \subseteq VAL$ and $\forall t \in T: \iota(t) \subseteq MOD$.
- $\forall (x, y) \in (S \times T) \cup (T \times S): \iota((x, y)) \in \mathbb{N}^{\iota(x) \times \iota(y)}$.

A net is called *simple* if all $\iota((x, y))$ are subsets of $\iota(x) \times \iota(y)$ (rather than true multisets). A (transition) *labelling* is a function $\lambda: T \to \Sigma \cup \{\tau\}$. A *marking* is an element $M \in \mathbb{N}^{S \times VAL}$. Sometimes a net is *marked* with an initial marking M_0. Sometimes, also, a finite set of final markings $\mathcal{M}_f \subseteq \mathbb{N}^{S \times VAL}$ is provided. As the need may be, we enlarge the definition by such parameters; thus, (S, T, ι, λ) is a labelled unmarked net, while $(S, T, \iota, \lambda, M_0, \mathcal{M}_f)$ is a marked labelled net with an additional set of final markings. □ 2.2

Definition 2.2 contains the class of (arc-weighted) *place/transition nets* (*PT-nets*, for short) as a special case: consider $\bullet \in VAL$ and $\blacksquare \in MOD$ and $\iota(s) = \{\bullet\}$ for every $s \in S$ and $\iota(t) = \{\blacksquare\}$ for every $t \in T$. By the fact that ι on arcs may be a multiset (rather than just a set), arbitrary non-negative integer arc weights (including 0, for no arc) can be modelled. We denote the class of finite, labelled, not necessarily simple, PT-nets with an initial and a finite set of final markings by $\mathcal{N}$.

The transition rule for nets is defined as follows. A marking M *activates* a transition t in mode $m \in \iota(t)$ (in symbols: $M[(t, m)\rangle$) iff $M \geq \iota(., t)(., m)$. Moreover, M leads to a marking M' through the *occurrence* of t in mode m (in symbols: $M[(t, m)\rangle M'$) iff $M[(t, m)\rangle$ and $M' = M - \iota(., t)(., m) + \iota(t, .)(m, .)$. A marking M is *reachable* from M_0 iff there exists a series of occurrences from M_0 to M.[2] A marked net (S, T, ι, M_0) is *k-safe* (for $k > 0$) iff in every marking M reachable from M_0 all places $s \in S$ contain at most k same tokens (i.e., M is a multiset with multiplicities at most k). Let $\mathcal{N}_k \subseteq \mathcal{N}$ denote the class of all k-safe PT-nets in $\mathcal{N}$.

We assume all our nets to be *T-restricted*, that is, every transition has at least one preplace and at least one postplace, formally: $\forall t \in T: \iota(., t) \neq \emptyset \neq \iota(t, .)$.[3]

Our formalisation of the pomset behaviour of a net is very similar to that of Pomello et al. [PRS92]. Pomsets are seen as an abstraction of processes, which in turn are based on occurrence nets. A process reflects an actual simulation of the token game of a net in a 'truly concurrent' way.

[2] Such sequences are the *interleaving behaviour*.

[3] This requirement greatly simplifies the definition of concurrent behaviour, and it is moreover not a severe restriction in the case of k-safe nets: transitions without any preplaces and postplaces are uninteresting; transitions without preplaces but with postplaces will not occur since the net is k-safe; and transitions with preplaces but without postplaces can be made T-restricted without changing concurrent behaviour, by adding a postplace artificially.

Definition 2.3. *Occurrence net*

A (finite) *occurrence net*, *O-net* for short, is a simple net $O = (B, E, F)$ satisfying the following conditions:

(1) For all $x \in B \cup E$, the set $F(x)$ is a singleton.
(2) The relation F (which, by simpleness and property (1), may as well be viewed as a relation on $(B \times E) \cup (E \times B)$) is acyclic (i.e. F^+ is irreflexive).
(3) $\forall b \in B\colon |F(.,b)| \leq 1 \geq |F(b,.)|$.

Elements of B and E are called *conditions* and *events*, respectively. Further, let $\min O = \{b \in B \mid F(.,b) = \emptyset\}$ and $\max O = \{b \in B \mid F(b,.) = \emptyset\}$. □ 2.3

The set of B-cuts of an O-net (that is, the B-cuts of the partial order $(B \cup E, F^+)$) can itself be partially ordered (yielding, in fact, a finite lattice) by stipulating that $C_1 \sqsubseteq C_2$ iff $\forall b_1 \in C_1, b_2 \in C_2\colon (b_1, b_2) \in F^+ \vee (b_1, b_2) \in co_{F^+}$.

Definition 2.4. *Process*

A *process* π of a net $N = (S, T, \iota, M_0, \mathcal{M}_f)$ is a tuple (B, E, F, r), where $r\colon (B \cup E) \to (S \cup T)$ and:

- $O = (B, E, F)$ is an O-net.
- $r(B) \subseteq S$ and $r(E) \subseteq T$, i.e. r 'folds' conditions onto places and event onto transitions.
- r satisfies an initiality condition: $r(\min O) = M_0$; a 'progress' condition:

  $$\forall e \in E\colon r(F(.,e)) = \iota(.,r(e)) \wedge r(F(e,.)) = \iota(r(e),.),$$

  and a finality condition: $r(\max O) \in \mathcal{M}_f$. □ 2.4

The *length* of a process π is given as $|\pi| = |E|$.[4] Let $Proc(N)$ denote the class of all processes of N. We call two processes *isomorphic* if one of them results from the other by renaming conditions and events.

The upper part of figure 2 shows an example. We have given a PT-net (shown here in full as a coloured net) as an example, since the major result of this paper holds for such nets. The function r is shown as inscription inside conditions and events of π. The sets below the places/conditions and transitions/events x are the $\iota(x)$, and the sets next to arcs (x, y) are the $\iota((x, y))$.

It is a well-known fact (see e.g. [GV83, Jen92]) that a net $N = (S, T, \iota, M_0)$ is k-safe iff for all B-cuts C of processes $\pi = (B, E, F, r)$ of N and all places $s \in S$ and values $v \in \iota(s)$ the following holds: $|\{b \in B \mid r(b) = s \wedge F(b) = \{v\}\} \cap C| \leq k$. This means that any value may 'occur' in any B-cut at most k times. For PT-nets, this specialises to the fact that N is k-safe iff for all π and C and for all places s: $|r^{-1}(s) \cap C| \leq k$.

Possible observable behaviour is specified by an additional labelling λ. As a pomset is supposed to describe such a behaviour, we consider only the observable events and their relations in a process.

[4] Recall that we consider only finite O-nets.

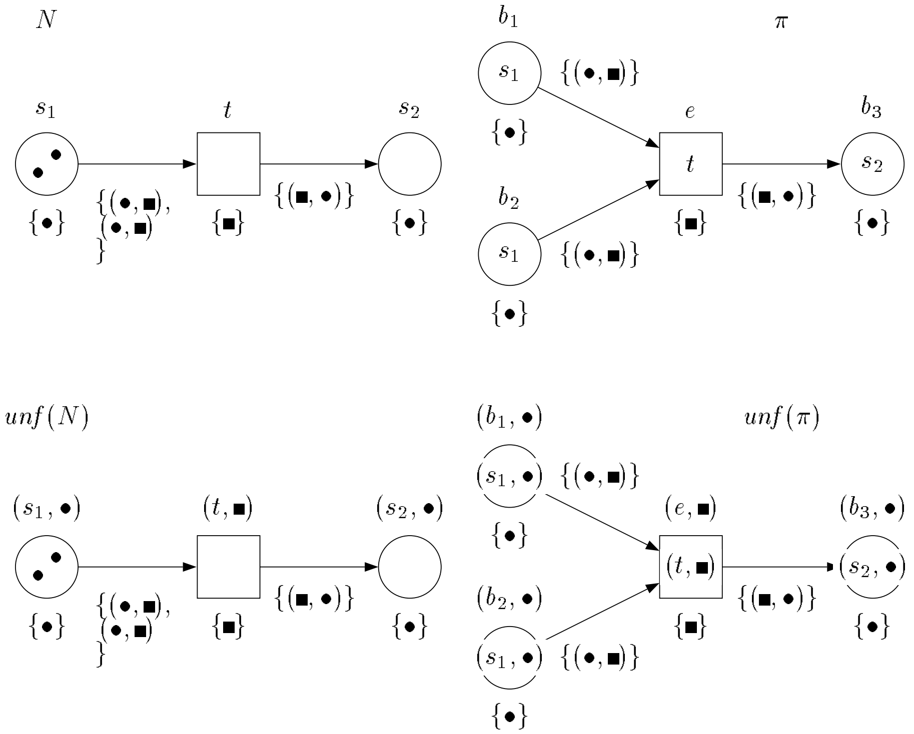

Figure 2. A PT-net N, a process π of N, and their unfoldings.

Definition 2.5. *Pomset of a process*

Let $N = (S, T, \iota, \lambda, M_0, \mathcal{M}_f)$ and let $\pi = (B, E, F, r)$ be a process of N. The pomset $\Phi(\pi)$ is defined as $[(E_\varphi, R_\varphi, \ell_\varphi)]$ with $E_\varphi = \{e{\in}E \mid \lambda(r(e)){\neq}\tau\}$, $R_\varphi = F^+|_{E_\varphi \times E_\varphi}$, and $\ell_\varphi = r|_{E_\varphi} \circ \lambda$. □ 2.5

Thus, the pomset $\Phi(\pi)$ is an abstraction from the process π. We allow for auto-concurrency here, i.e., multiple instances of the same action may occur in a pomset $\Phi(\pi)$ without any causal ordering between them.

Definition 2.6. *Pomset language*

We define the set of all pomsets, or *pomset language*, of a given net N by $\mathcal{P}(N) = \{\Phi(\pi) \mid \pi \text{ is a process of } N\}$. Moreover, $\mathcal{P}_k = \{\mathcal{P}(N) \mid N \in \mathcal{N}_k\}$ denotes the class of pomset languages of k-safe PT-nets for $k{>}0$. □ 2.6

Some examples of PT-nets with typical pomsets from their respective languages can be seen in figure 3. Note that while N_1 is 2-safe and N_3 and N_4 are 8-safe, the net N_2 is not k-safe for any $k{>}0$. The net N_1 is the same as the net N of figure 2, written in more traditional PT-net notation.

3 Balanced coloured nets

In this section, we will consider specific values *VAL* and, related to them, specific modes *MOD*. Let $\mathcal{F}$ be a finite set. At the present time, $\mathcal{F}$ is arbitrary, but we will later specialise it in order to prove our main result. Put $\mathit{VAL} = 2^{\mathcal{F}}$, i.e., every value is a subset of $\mathcal{F}$. Let $N = (S, T, \iota)$ be a simple net such that $\iota(s) = \mathit{VAL} = 2^{\mathcal{F}}$ for every place s in S. Let t be a transition in T and let m be a mode in $\iota(t)$. We will call m *balanced* w.r.t. t iff the following hold:

(A) $\forall s{\in}\iota(.,t)\,\exists v{\in}\iota(s)\colon \iota(s,t)(v,m){>}0$ and $\forall s{\in}\iota(t,.)\,\exists v{\in}\iota(s)\colon \iota(t,s)(m,v){>}0$.
(B) $\sum_{s\in\iota(.,t),(s,v)\in\iota(.,t)(.,m)} v$ and $\sum_{s\in\iota(t,.),(s,v)\in\iota(t,.)(m,.)} v$ are sets.
(C) $\sum_{s\in\iota(.,t),(s,v)\in\iota(.,t)(.,m)} v \;=\; \sum_{s\in\iota(t,.),(s,v)\in\iota(t,.)(m,.)} v$.

Condition (A) means that m 'covers' all input and output places of t in the sense that it removes a value from every input place and puts a value on every output place. Condition (B) means that any two values m takes from input places of t must be disjoint,[5] and similarly for two values m puts on output places of t must be disjoint. Condition (C) means that m 'redistributes' the values it gets from input places of t onto the output places of t, in the sense that the union of the former is split into the latter; hence the term 'balanced'.

The balancedness conditions tell us how the elements of values are distributed through the events of a process. Such elements cannot get lost or created on the way; each element, as it turns out, must follow some line through the process.

A marking $M \in \mathbb{N}^{S\times 2^{\mathcal{F}}}$ of N is called *spread* if all its tokens are mutually disjoint, that is, if $\sum_{s\in S,(s,v)\in M} v$ is a set. Clearly, a spread marking is 1-safe. What is more:

[5] Remember that values are sets.

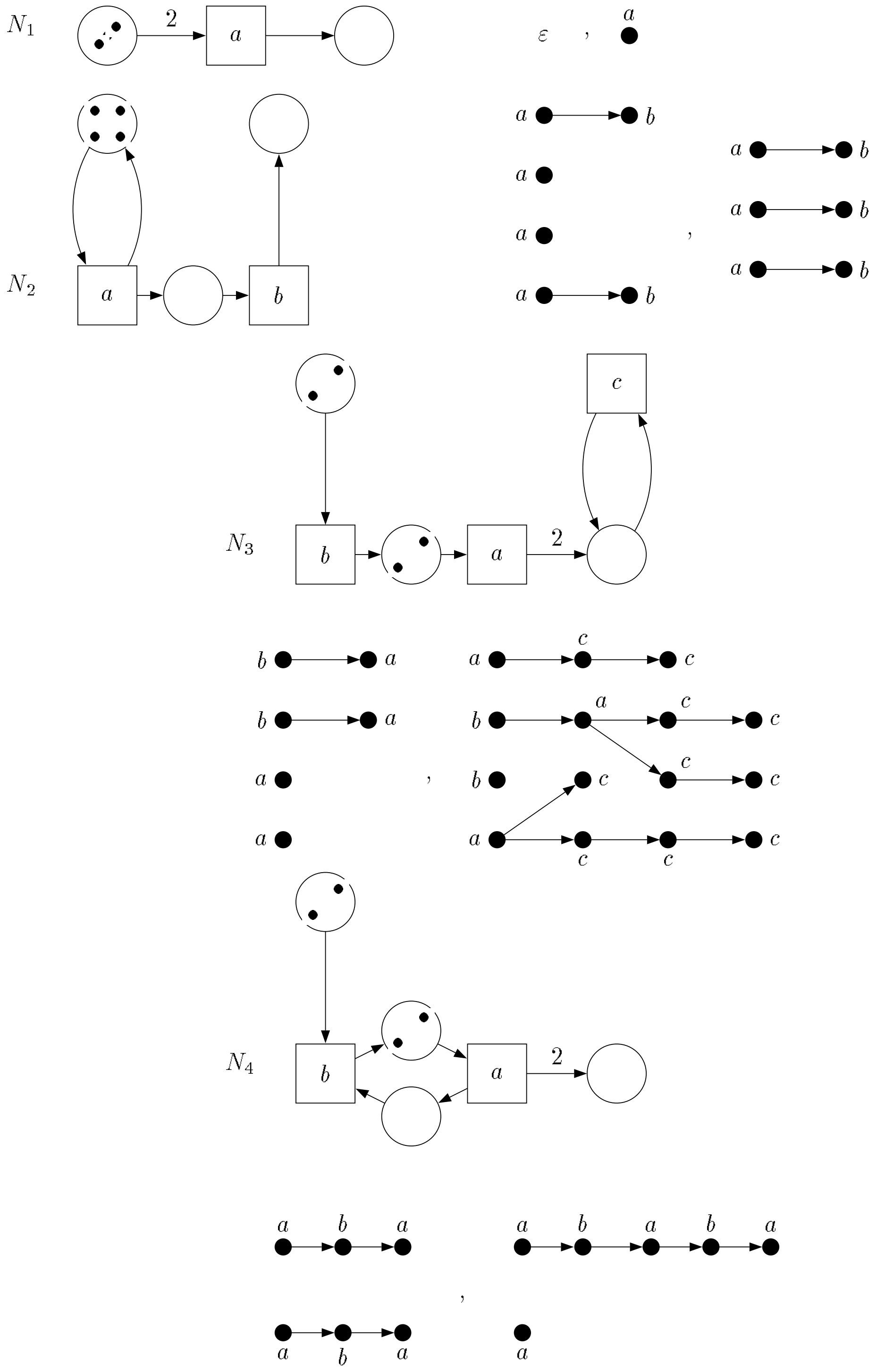

Figure 3. Some PT-nets with some pomsets (without considering final markings).

Lemma 3.1.

Let $N = (S, T, \iota, M_0)$ be a simple net in which all modes are balanced and let M_0 be spread. Then N is 1-safe.

Proof: By induction. First, M_0 is clearly spread and 1-safe. Then, suppose $M[(t,m)\rangle M'$ such that M is spread. Since m is balanced, it can redistribute only already existing values by (B) and (C). Thus it cannot create any set that has an element in common with other (created or untouched) sets, whence M' is also spread and thus 1-safe. ■ 3.1

Definition 3.2. *Net unfolding*

Let $N = (S, T, \iota, \lambda, M_0)$ be a marked labelled net. Its *unfolding*, $unf(N) = (S', T', \iota', \lambda', M'_0)$, is defined as follows: its places S' are all the tuples (s, v) with $s \in S$ and $v \in \iota(s)$, and $\iota'((s, v)) = \{v\}$; its transitions T' are all the tuples (t, m) with $t \in T$ and $m \in \iota(t)$, and $\iota'((t, m)) = \{m\}$; its arcs are $\iota'(((s, v), (t, m))) = \iota(s, t)(v, m) \cdot \{(v, m)\}$ and $\iota'(((t, m), (s, v))) = \iota(t, s)(m, v) \cdot \{(m, v)\}$ for any $(s, v) \in S'$ and $(t, m) \in T'$; its labelling is $\lambda'((t, m)) = \lambda(t)$; and the marking M'_0 are all the tokens $((s, v), v)$ with (s, v) being a token of M_0. □ 3.2

The lower part of figure 2 shows an example. The operation of unfolding just serves to 'flatten out' the net (introducing new names for places and transitions), but it leaves its structure, i.e. the connectivity of pairs (s, v) and (t, m), intact. More precisely, we have the following lemma which states that N and $unf(N)$ have exactly the same (concurrent) behaviour. First, we define the unfolding of a process $\pi = (B, E, F, r)$ of N as $unf(\pi) = (B', E', F', r)$, where $(B', E', F') = unf((B, E, F))$ and $r'((b, \bullet)) = (r(b), \bullet)$ and $r'((e, \blacksquare)) = (r(e), \blacksquare)$.

Lemma 3.3.

π is a process of the coloured net N if and only if $unf(\pi)$ is a process of $unf(N)$, and what is more, π and $unf(\pi)$ are isomorphic.

Proof: The proof follows standard theory [GV83, Jen92, Rei98, Smi98]. First, we note that if all value sets $\iota(s)$ and all mode sets $\iota(t)$ of N are singletons, then $unf(N)$ is isomorphic to N (i.e., the same as N, up to renaming of places and transitions). This follows directly from the definition of unfolding, and it has as a consequence that the unfolding of any PT-net is isomorphic to that PT-net, as well as that the unfolding of any O-net or process is isomorphic to the original O-net, resp. process. (It also has the consequence that $unf(N)$ is isomorphic to $unf(unf(N))$.)

Now let π be a process of N. Since the definition of r depends only on (the initial and final markings and) the connectivity of tuples (s, v) with tuples (t, m), and because this connectivity is not changed by unfolding, it is not hard to check that $unf(\pi)$ is indeed a process of $unf(N)$, and vice versa. ■ 3.3

We will need the following two corollaries:

Corollary 3.4.

For any marked labelled PT-net $N \in \mathcal{N}$, $\mathcal{P}(N) = \mathcal{P}(unf(N))$. ■ 3.4

Corollary 3.5.

For any coloured net N, N is k-safe iff $unf(N)$ is k-safe. In particular, for a simple N with only balanced modes and a spread initial marking M_0, $unf(N)$ is 1-safe.

Proof: The second claim follows with lemma 3.1. ■ 3.5

There is also a reverse relationship between coloured nets N whose value and mode sets are singletons (such as unfoldings) and PT-nets; let us call nets whose value and mode sets are singletons *basic*. Since all types and mode sets of basic nets are singletons, their exact identity does not matter as long as we keep the place and transition sets intact. Thus, we may translate N into a PT-net, $PT(N)$, with the same behaviour, by replacing all value sets by the standard singleton set $\{\bullet\}$, all mode sets by $\{\blacksquare\}$, and modifying arc multisets accordingly (but keeping places, transitions and arcs unchanged otherwise). We may also apply this operation PT to non-basic nets, but there will generally be a change in the behaviour: the resulting net behaves as if every mode was allowed for every transition.

Lemma 3.6.

Let N be basic. Then π is a process of N if and only if $PT(\pi)$ is a process of $PT(N)$, and they are isomorphic.

Proof: Similar to the one above. ■ 3.6

Again there are two corollaries:

Corollary 3.7.

Let N be basic. Then $\mathcal{P}(N) = \mathcal{P}(PT(N))$. ■ 3.7

Corollary 3.8.

Let N be basic. If N is 1-safe, then so is $PT(N)$. ■ 3.8

These considerations will allow us to convert a k-safe PT-net into a simple, balanced coloured net with a spread marking, and apply the $unf(.)$ and $PT(.)$ operations to the latter to get a 1-safe net with the same pomsets.

Let S be any set of places and let M be any function in $\mathbb{N}^{S\times 2^{\mathcal{F}}}$. We define from M the function $discolour(M)$ in $\mathbb{N}^{S\times\{\bullet\}}$ by

$$\forall s{\in}S\colon\ discolour(M)(s,\bullet) = \sum_{v\in 2^{\mathcal{F}}} M(s,v).$$

That is, the marking $discolour(M)$ has as many tokens on each place as there have been values in M on that place. In the next definition, c is to be understood informally as colouring the initial marking M_0 of an uncoloured net and discolouring (through c^{-1}) the set of final markings of the resulting coloured net.

Definition 3.9. *Colouring a net*

Let $N = (S, T, \iota, \lambda, M_0, \mathcal{M}_f)$ be a PT-net. Any relation $c \subseteq (\{M_0\} \cup \mathcal{M}_f) \times (\mathbb{N}^{S \times 2^{\mathcal{F}}})$ is valid *colouring* of N if

(a) $c(M_0)$ is spread and $M_0 = discolour(c(M_0))$.
(b) $\forall M \in \mathcal{M}_f : c(M) = \{M' \in \mathbb{N}^{S \times 2^{\mathcal{F}}} \mid M'$ is spread and $discolour(M') = M\}$.

For a valid colouring c, the coloured net $colour(N, c) = (S, T, \iota', \lambda, c(M_0), c(\mathcal{M}_f))$ is defined as a coloured version of N, if for every $s \in S$, $\iota'(s) = 2^{\mathcal{F}}$; for every $t \in T$, the set of $\iota'(t)$ consists of *all* modes that are balanced w.r.t. t; and the arcs satisfy $\iota'((x, y)) \neq \emptyset \iff \iota((x, y)) \neq \emptyset$ (and are otherwise defined uniquely by the balanced modes). □ 3.9

Thus, the coloured version of a net contains the maximal set of balanced modes for every transition. It is easy to see that these mode sets are still finite, since every mode can be determined by the transition it belongs to and by the way it combines and distributes the pre-values and the post-values (and all sets involved are finite, due to the finiteness of $\mathcal{F}$). Note also that condition (b) ensures that c is uniquely determined on $\mathcal{M}_f$, and thus in order to define a valid colouring, it is only necessary to specify $c(M_0)$ and verify (a).

Lemma 3.10.

For all PT-nets $N \in \mathcal{N}$ and colourings c, $\mathcal{P}(colour(N, c)) \subseteq \mathcal{P}(N)$.

Proof: Take any pomset $\varphi \in \mathcal{P}(colour(N, c))$. There is a process $\pi = (O, r)$ of $colour(N, c)$ with $\Phi(\pi) = \varphi$. Now consider $PT(colour(N, c))$. Clearly, this is the original PT-net N and $PT(\pi)$ is a process of N. π is easily seen to be a coloured version of $PT(\pi)$, and as a colouring does not influence the way pomsets are constructed from processes, both processes give rise to the same pomset. Overall, $\mathcal{P}(colour(N, c)) \subseteq \mathcal{P}(N)$. ■ 3.10

The reverse inclusion also holds with a carefully chosen c, provided that N is k-safe, but it is not done as easily. Take a situation in the coloured net $colour(N, c)$ where a transition t with, say, two places s_1 and s_2 in its preset and three places in its postset is supposed to occur, and assume there are tokens $\{1\}$ in s_1 and $\{2\}$ in s_2. Clearly, t cannot occur, as it cannot redistribute two singleton sets into three mutually disjoint nonempty sets (remember that $colour(N, c)$ is balanced). In the 'uncoloured' net N though, there are just two 'black tokens' $\bullet$ in the preset, and the transition t can occur.

At this point, our undetermined set $\mathcal{F}$ from the beginning of this section comes to the rescue. We will choose $\mathcal{F}$ large enough (but still, importantly, finite) so that it is always possible to redistribute the tokens in a way such that the transition t we have in mind can eventually occur (in the coloured net, of course).

4 A pumping lemma for processes

For a single process π we can easily determine the set $\mathcal{F}$ needed to enable all the necessary transitions at the correct moments: Choose some possible linear order in which the transitions can occur to generate this process π and proceed through the transitions. If a transition can occur, just let it occur and redistribute the token sets arbitrarily. If there are not sufficiently many elements of $\mathcal{F}$ in order for the transition to occur, just add new elements to tokens of the initial marking so that they can be sent through the process so far and get to lie in the preset of our hitherto unfireable transition. Repeat if necessary, until there are enough elements in all presets. When this is done all through the sequence, we get a colouring for the initial marking under which the process π can be simulated. It is clearly visible in this algorithm that the (new) elements follow some lines through our process. These lines become important now.

Definition 4.1. *Line system*

Let $\pi = (B, E, F, r)$ be a process of a N. A *line system* L for π is a minimal set of lines of F^+ that covers all conditions $b \in B$ of π. Let $\mathcal{L}(\pi)$ denote the set of all line systems for π. □ 4.1

It is known that every line and every cut of π have exactly one element in common. The basic idea is to get a line system L for π, generate one element of $\mathcal{F}$ for each line in L, and put it onto the condition on the beginning of its line. Since more than one line may lie on a condition, this may yield non-singleton sets, but since every condition lies on some line, we always get nonempty sets. The minimal conditions exactly represent the initial marking, and thus each token of the initial marking gets the elements we have put onto the corresponding condition, and since each line can lie on exactly one condition, this marking will be spread. When a transition occurs, each element that is processed through this transition just follows 'its' line, and thus the balancedness conditions are satisfied as well.

The algorithm remains easy still if we have a finite number of processes. We just apply it to each process and ensure that no element appears for two different processes. A simple union then guarantees that all processes can be simulated.[6]

A problem arises clearly when we consider an infinite number of processes. In fact, we have two problems: first, will the number of elements that we must assign to a single (finite, but arbitrarily large) process be bounded by some number, and second, is there a way we can use an assignment of elements for a whole (arbitrarily large) class of processes, and will the number of classes be finite? If both questions can be answered with yes, we can indeed find the set $\mathcal{F}$ and a colouring for the initial marking that we are looking for.

The answer to the first question is simply that a process, no matter how long, of a k-safe PT-net $N = (S, T, \iota, \lambda, M_0, \mathcal{M}_f)$ has a line system with at most $k \cdot |S|$ lines. To prove this, we use a theorem from graph theory, compare [EHK96, Tve67].

Theorem 4.2. Dilworth 1950

In a partial order, the maximal cardinality of a cut is equal to the minimal number of chains into which the partial order can be partitioned. ■ 4.2

Lemma 4.3. *Maximal Number of Lines*

Let $\pi = (B, E, F, r)$ be a process of a k-safe PT-net $N = (S, T, \iota, \lambda, M_0, \mathcal{M}_f) \in \mathcal{N}$. Then, there is a line system L for π with $|L| \leq k \cdot |S|$.

[6] We do not need to consider isomorphic processes here, as it is not necessary to handle more than one representative from each isomorphism class in this way.

Proof: N is k-safe, so we know that no marking can have more than $k \cdot |S|$ tokens and, accordingly, no B-cut can contain more than $k \cdot |S|$ elements. By Dilworth's theorem then, we know that the minimal number of chains into which the partial order $(B, F^+|_{B\times B})$ can be partitioned into is $k \cdot |S|$ or less. We enlarge each of these chains to a line arbitrarily and get a set of lines covering all conditions in B. Clearly, this set of lines contains a line system for π with at most $k \cdot |S|$ lines. ■ 4.3

In fact, the set of lines constructed in this proof is already a line system, i.e. no line may be removed. For suppose we could remove one line. This means that all conditions b in this line are also element in some other line. We remove this obsolete line. We convert our set of lines back to a set of chains by removing conditions appearing in more than one line in all but one of them. Clearly, we get a set of chains partitioning $(B, F^+|_{B\times B})$ with one chain less than the minimal number, which is a contradiction.

Dilworth's theorem thus provides us with an upper bound on the number of lines covering an arbitrary process, such that this bound is independent of the size of that process, and thus with the answer 'yes' to the first question above.

For the next step we introduce the notion of equivalence of cuts in a process, relative to a given line system of that process. The rationale behind this definition is to provide a means by which a process can be 'cut' into pieces, such that the relevant pieces are bounded in their size.

Definition 4.4. *Equivalence of cuts*

Let $N = (S, T, \iota, \lambda, M_0, \mathcal{M}_f)$ be a PT-net, $\pi = (O, r)$ a process of N and $L \in \mathcal{L}(\pi)$ a line system for π. Let C_1 and C_2 be B-cuts of π. We call the cuts C_1 and C_2 equivalent with respect to L, $C_1 \equiv_L C_2$ iff the relation $\rho \subseteq C_1 \times C_2$, defined by

$$(b_1, b_2) \in \rho \iff \exists l \in L : l \cap C_1 = \{b_1\} \wedge l \cap C_2 = \{b_2\}$$

is an r-preserving bijection. □ 4.4

It is easy to check that $\equiv_L$ is indeed an equivalence relation. Clearly, if $\min \pi \equiv_L \max \pi$ for some process π with line system L, then π can be concatenated (denoted by $\cdot$) with itself (by identifying conditions via $\equiv_L$) to yield processes $\pi \cdot \pi$, $\pi \cdot \pi \cdot \pi$ etc., which are larger than π (unless π has no events, in which case $\pi = \pi \cdot \pi$). Such a π is called *repeatable*. Also, if some process π with line system L contains two B-cuts C_1 and C_2 with $C_1 \sqsubseteq C_2$ and $C_1 \equiv_L C_2$, then π can be split into an initial part π_1 (up to and including C_1), a middle part π_2 (from C_1 up to C_2, inclusively) and a final part π_3 (from C_2, inclusively, onwards), such that π_1 and π_3 can be concatenated to yield a process $\pi_1 \cdot \pi_3$ which is smaller than π (unless C_1 and C_2 are the same cut, in which case $\pi = \pi_1 \cdot \pi_3$).

The following lemma shows that there is a bound on the number of non-equivalent cuts in a process, which does not depend on the size of the process.

Lemma 4.5. *Number of non-equivalent cuts*

Let $N = (S, T, \iota, \lambda, M_0, \mathcal{M}_f)$ be a k-safe PT-net, $\pi = (B, E, F, r)$ a process of N, and $L \in \mathcal{L}(\pi)$ a line system for π with $|L| \leq k \cdot |S|$. Any set of cuts of π that are pairwise non-equivalent with respect to L contains no more than $(k \cdot |S|^2)^{k \cdot |S|} \cdot k \cdot |S|$ elements.

Proof: Let $\mu(|C|,|L|)$ denote the maximal number of different ways in which the lines of the line system L can cross the fixed cut C.[7] We have the following recurrence for $\mu(|C|,|L|)$:

$$\begin{aligned} |C|=1 &: \mu(|C|,|L|) &&= 1 \\ |C|>1 &: \mu(|C|,|L|{+}1) &&= |C|\cdot\mu(|C|,|L|). \end{aligned}$$

The first line comes from the fact that all lines must touch the single element of the cut. The second line comes from the fact that the $|L|{+}1$'st line has $|C|$ possibilities, and that each time all other lines have all possibilities. Evidently, $\mu(|C|,|L|)=|C|^{|L|}$ solves the recurrence.

This implies that any set of non-equivalent B-cuts, each of cardinality m (where m varies between 1 and $k\cdot|S|$), can contain at most $|S|^m\cdot m^{k\cdot|S|}$ elements, where the first factor bounds the number of possible labellings of conditions in such a cut, and the second number bounds the number of different ways the line system can cross the cut (since $|L|$ is bounded by $k\cdot|S|$). Since two cuts of different cardinality can never be equivalent, it follows that the number of possible different cuts is bounded by the sum

$$|S|^1\cdot 1^{k\cdot|S|}+|S|^2\cdot 2^{k\cdot|S|}+\ldots+|S|^{k\cdot|S|}\cdot(k\cdot|S|)^{k\cdot|S|}.$$

Bounding each summand by the last one yields

$$(|S|^{k\cdot|S|}\cdot(k\cdot|S|)^{k\cdot|S|})\cdot k\cdot|S| = (k\cdot|S|^2)^{k\cdot|S|}\cdot k\cdot|S|$$

as an upper bound for the overall number of non-equivalent cuts, as claimed.

■ 4.5

We may now formulate our pumping lemma.

Lemma 4.6. *Pumping Lemma*

Let $k>0$ and let $N\in\mathcal{N}_k$. Then there exists a number $n_0\in\mathbb{N}$ such that for all processes $\pi\in \mathit{Proc}(N)$ with $|\pi|>n_0$ there are a line system $L\in\mathcal{L}(\pi)$ and processes x,y,z such that:

- $\pi=x\cdot y\cdot z$.
- $x\cdot z$ is defined and $|x\cdot z|\le n_0$.
- y is repeatable and $\forall n\in\mathbb{N}$: $x\cdot y^n\cdot z\in \mathit{Proc}(N)$.

Proof: For $N=(S,T,\iota,\lambda,M_0,\mathcal{M}_f)$ define $n_0=(k\cdot|S|^2)^{k\cdot|S|}\cdot k\cdot|S|$. Let now $\pi=(B,E,F,r)$ be a process of N with $|\pi|>n_0$ and let R be any total ordering of E with $F^+|_{E\times E}\subseteq R$, i.e. a linearisation of the events of π. We denote this ordering by calling the events $e_1,\ldots,e_m$. We construct then a sequence of B-cuts $C_0,C_1,\ldots,C_m$ with $C_0=\min\pi$ and $C_{i+1}=(C_i\backslash F(.,e_{i+1}))\cup F(e_{i+1},.)$ for $1\le i<m$. Clearly, $C_i\sqsubseteq C_{i+1}$ for $0\le i<m$. By lemma 4.3 there is a line system $L\in\mathcal{L}(\pi)$ with at most $k\cdot|S|$ lines, and by lemma 4.5 there are at most n_0 different equivalent cuts in π. Therefore, there must be two cuts C_i and C_j with $i<j$, $C_i\sqsubseteq C_j$ and $C_i\equiv_L C_j$. Choose i and j such that $j{-}i$ becomes maximal with this property, and then split π at the cuts C_i and C_j obtaining x, y, and z with $\pi=x\cdot y\cdot z$. The equivalence of C_i and C_j guarantees that $x\cdot y^n\cdot z$ is well defined (also for

[7] We may assume $|C|\ge 1$ (and thus $|S|\ge 1$), which is a consequence of T-restrictedness.

$n{=}0$). Due to the maximality of $j{-}i$ there are no equivalent cuts in the sequence $C_0, \ldots, C_i, C_{j+1}, \ldots, C_m$ of cuts remaining from the original sequence in $x \cdot z$, thus $|x{\cdot}z| \leq n_0$ by lemma 4.5 (and the fact that a process contains at least as many cuts as events). ■ 4.6

Our pumping lemma guarantees that each process can be 'cut down' to a process (of the form $x{\cdot}z$) below a certain length in just one step, i.e. the pumping lemma will need to be applied only once. Note that it is a true generalisation of the ordinary pumping lemma for regular languages, as it applies, in particular, to a 1-safe net each of whose transitions has exactly one pre-place and one post-place.

5 The equivalence of k-safe and 1-safe nets

Using the pumping lemma for processes we may now show that each k-safe PT-net can be transformed into a simple coloured net with a spread initial marking and balanced modes (and hence, a 1-safe PT-net) with the same pomset language.

Lemma 5.1.

For all k-safe PT-nets N, there exists a colouring c such that the net $colour(N, c)$ is well defined and, moreover, $\mathcal{P}(N) \subseteq \mathcal{P}(colour(N, c))$.

Proof: In the first part of the proof, we show how such a coloured net can be constructed. We need only to specify the colouring function c.

Let $N = (S, T, \iota, \lambda, M_0, \mathcal{M}_f)$ be a k-safe PT-net, $n_0{\in}\mathbb{N}$ the constant of the pumping lemma for N and $\Pi = \{\pi_1, \ldots, \pi_m\} \subseteq \{\pi{\in}Proc(N) \mid |\pi|{\leq}n_0\}$ a finite set of non-pumpable processes of N containing one representative from each isomorphism class of processes of $\{\pi{\in}Proc(N) \mid |\pi|{\leq}n_0\}$. We assume that the conditions of these processes are renamed such that $\min \pi_1 = \min \pi_2 = \ldots = \min \pi_m = B_{\min}$ holds; let r be their common labelling function on the set $B_{\min}$.

For $1{\leq}i{\leq}m$ and $1{\leq}j{\leq}k{\cdot}|S|$ let $\Psi_{i,j}$ be pairwise disjoint sets with $|\Psi_{i,j}| = k{\cdot}|S|$. Let

$$\mathcal{F} = \bigcup_{1\leq i\leq m, 1\leq j\leq k\cdot|S|} \Psi_{i,j}.$$

For each $\pi_i{\in}\Pi$ determine a line system $L_{\pi_i}{\in}\mathcal{L}(\pi_i)$ with $|L_{\pi_i}| \leq k{\cdot}|S|$ (as in lemma 4.3), and enumerate the lines by a bijection $\alpha_{\pi_i}\colon L_{\pi_i} \to \{1, \ldots, |L_{\pi_i}|\}$.

Now define

$$\begin{array}{ll} \beta\colon B_{\min} \to 2^{\mathcal{F}} & \text{by } \beta(b) = \bigcup_{1\leq i\leq m, \ell\in L_{\pi_i}, b\in\ell} \Psi_{i,\alpha_{\pi_i}(\ell)} \\ c \subseteq (\{M_0\}{\cup}\mathcal{M}_f) \times (\mathbb{N}^{S\times 2^{\mathcal{F}}}) & \text{by } c(M_0)(s, v) = |\{b \in B_{\min} \mid r(b) = s \wedge \beta(b) = v\}|. \end{array}$$

The marking $c(M_0)$ is spread as a consequence of the requirement that the sets $\Psi_{i,j}$ are mutually disjoint, and $M_0 = discolour(c(M_0))$ by the initiality condition of definition 2.4. Recall (from definition 3.9) that is only necessary to specify $c(M_0)$ and verify 3.9(a). Hence $colour(N, c)$ is well defined.

The mapping β defines a front end for mode distributions, by which we can obtain whichever process we like. In this way, we show in the second part of the proof that $\mathcal{P}(N) \subseteq \mathcal{P}(colour(N, c))$ with the above c.

Let $\varphi \in \mathcal{P}(N)$ a pomset, and $\pi = (O_\pi, r_\pi) \in Proc(N)$ a process of N with $\Phi(\pi) = \varphi$.

Case 1: $|\pi| \leq n_0$. Clearly, there is a process $\pi_i = (B, E, F, r) \in \Pi$ with $\Phi(\pi_i) = \Phi(\pi)$. We will construct a labelling of π_i such that the resulting object is a process of $colour(N, c)$. To this end, we extend the mapping β, which associates a value (i.e., a subset of $\mathcal{F}$) with every condition in $B_{\min}$ to the set of all conditions of π_i. Initially, every condition in $B_{\min}$ is provided with a value that is a union of certain sets $\Psi_{k,j}$ (including $k{=}i$). We will not split the sets $\Psi_{k,j}$ themselves, but only their distribution in these unions. In doing so, we make sure that for each line $\ell \in L_{\pi_i}$ and each condition $b \in B$ we keep the property $b \in \ell \iff \Psi_{i,\alpha_{\pi_i}(\ell)} \subseteq \beta(b)$ (which is trivially true for $\min \pi_i$ by definition of β). More precisely, if e is an event, ℓ is a line of L_{π_i} and b is an input condition of e whose value β is already known, then by construction $\Psi_{i,\alpha_{\pi_i}(\ell)} \subseteq \beta(b)$. Now let b' be the (unique, if any) output condition of e which also lies on l; then the set $\Psi_{i,\alpha_{\pi_i}(\ell)}$ becomes a subset of $\beta(b')$. We also get that $\beta(b) \neq \emptyset$ for all $b \in B$, since the line system L_{π_i} covers all conditions in B. For each $b \in \min \pi_i$ and each remaining set $\Psi_{k,j}$ of $\beta(b)$ with $k{\neq}i$ we select some line ℓ' in π_i and make sure that for all $b' \in B$ we get $b' \in \ell' \iff \Psi_{k,j} \subseteq \beta(b')$, thus rendering $\bigcup_{b \in F(\cdot,e)} \beta(b) = \bigcup_{b \in F(e,\cdot)} \beta(b)$ valid. We then associate the (singleton) value set $\{\beta(b)\}$ with every b, and define the (singleton) set of modes for every event e in the way it was just constructed. By construction, all modes are balanced, and therefore, each event corresponds to a mode in the net $colour(N, c)$ and the so labelled π_i satisfies initiality and progress properties of a process of $colour(N, c)$. It is easily seen that the finality property is also satisfied. Overall, $\varphi = \Phi(\pi_i) \in \mathcal{P}(colour(N, c))$.

We see additionally that for each condition $b \in B$, $\beta(b)$ has at least $k{\cdot}|S|$ elements – namely those of at least one of the $\Psi_{i,j}$.[8]

Case 2: $|\pi| > n_0$. We can split $\pi = x \cdot y \cdot z$ by the pumping lemma and apply the proof above to $\pi' = x \cdot z$, again with the result that $\beta(b)$ contains at least $k{\cdot}|S|$ elements for each $b \in B$. This holds especially for those b of the splitting cut between x and z. We fit y back into the process and define β on y in the obvious way, choosing the line system from the pumping lemma for y. It is clear that we can attach at least one element to each line, even if all lines lead through the same condition: there are at most $k{\cdot}|S|$ lines and each condition of $\min y$ has $k{\cdot}|S|$ elements attached to it by β. The pumping lemma guarantees now that every line ends in $\max y$ at exactly the same condition it starts in at $\min y$, therefore the distribution of the elements in $\min y$ is exactly as in $\max y$, so β is well-defined at the connecting cut between y and z. Overall, we get $\varphi \in \mathcal{P}(colour(N, c))$. ■ 5.1

The construction defined in the proof of lemma 5.1 may be illustrated on figure 4. There are four (non-isomorphic) processes, and the number $k{\cdot}|S|$ is 4 (since the net is 2-safe and has two places). Hence we define 16 sets $\Psi_{i,j}$ of four elements each. In all, $\mathcal{F}$ has a total of 64 elements. The colouring c is in this case:

$$\begin{array}{ll} c(M_0)(s, \Psi_{1,1} \cup \Psi_{2,1} \cup \Psi_{3,1} \cup \Psi_{4,1}) & = 1 \text{ (because of } b_1) \\ c(M_0)(s, \Psi_{1,2} \cup \Psi_{2,2} \cup \Psi_{3,2} \cup \Psi_{4,2}) & = 1 \text{ (because of } b_2) \\ c(M_0)(s, v) & = 0 \text{ for any other } v. \end{array}$$

For instance, in order to execute π_4 in the so coloured net, the eight sets $\Psi_{1,1}$–$\Psi_{4,2}$ will be combined (by forming their union) at condition b_3.

Finally, combining the previous considerations, we obtain:

[8] This property will be used in 'Case 2' of the proof; if only 'Case 1' would have had to be considered, we could have restricted all $\Psi_{i,j}$ to be singleton sets.

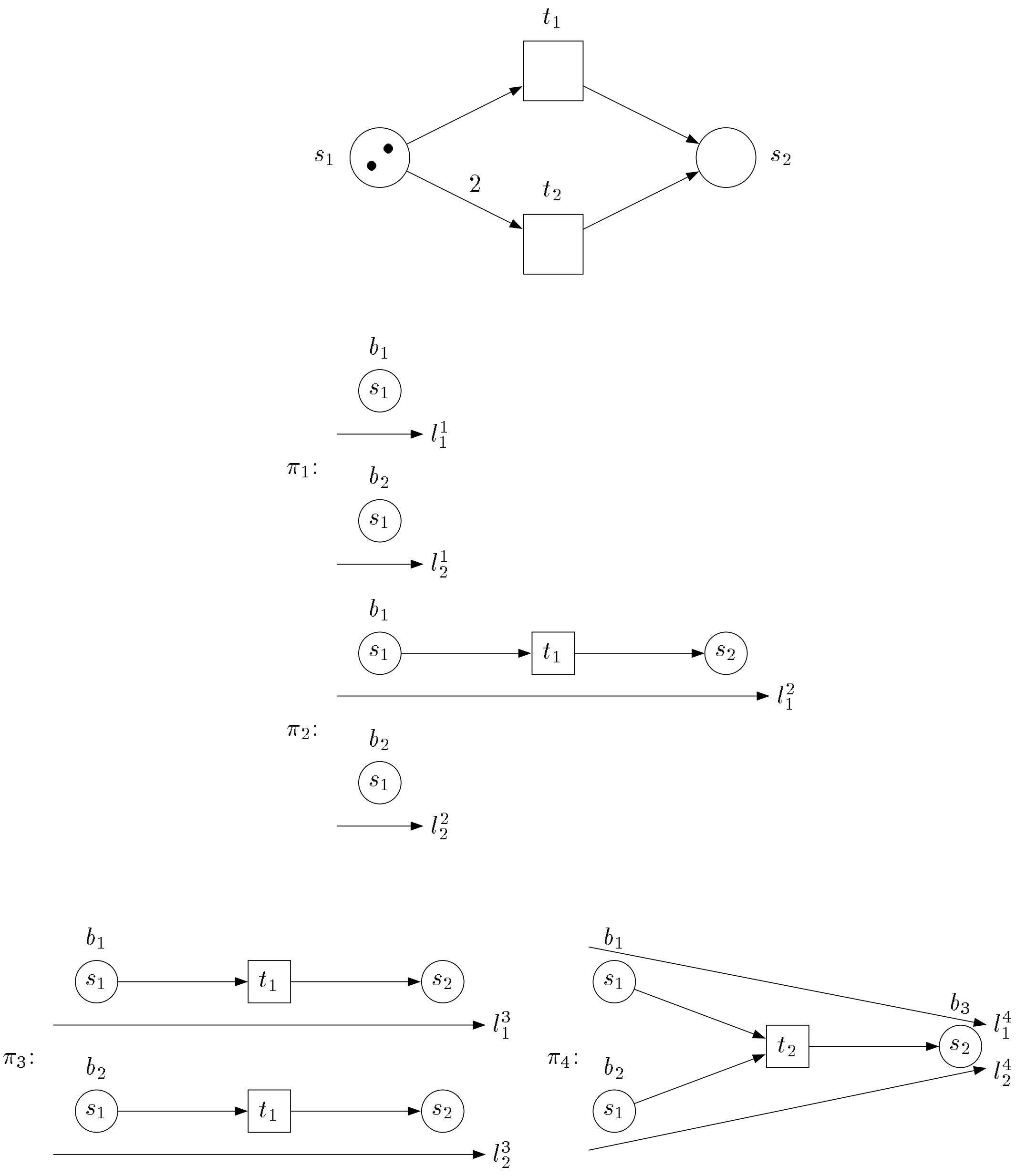

Figure 4. A PT-net, its processes, and a line system for each of them.

Theorem 5.2. Main result

For every k-safe PT-net N there is a 1-safe PT-net N' with $\mathcal{P}(N) = \mathcal{P}(N')$.

Proof: Let c be a colouring for the initial marking of N as in the previous lemma. Define $N' = PT(unf(colour(N, c)))$. Then,

$$\begin{aligned}\mathcal{P}(N) &= \text{(lemma 3.10 and lemma 5.1)}\\ &\quad \mathcal{P}(colour(N, c))\\ &= \text{(corollary 3.4)}\\ &\quad \mathcal{P}(unf(colour(N, c)))\\ &= \text{(corollary 3.7)}\\ &\quad \mathcal{P}(PT(unf(colour(N, c)))).\end{aligned}$$

Thus, $\mathcal{P}(N) = \mathcal{P}(N')$, and moreover, N' is 1-safe by corollaries 3.5 and 3.8.

■ 5.2

Corollary 3.5 applies because $colour(N, c)$ is by definition simple and has only balanced modes and a spread initial marking. Corollaries 3.7 and 3.8 apply because $unf(colour(N, c))$ is basic by definition.

6 Related Work

An earlier result of [Pe92] can be recovered by restricting attention to initial and final markings. Conversely, the transformations described in [Pe92] cannot apparently be generalised to our present problem, since they work inductively.

References

Bes84. E. Best: In Quest of a Morphism. Petri Net Newsletters Vol.18, pp.14–18, 1984.

EHK96. T. Emden-Weinert, S. Hougardy, B. Kreuter, H.J. Prömel, A. Steger: *Einführung in Graphen und Algorithmen.* Skriptum der Humboldt-Universität Berlin, 1996.

GV83. U. Goltz and U. Vogt: Processes of Relation Nets. Petri Net Newsletters Vol.14, pp.10–19, 1983.

Jen92. K. Jensen: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1992.

Pe92. E. Pelz: Normalization of Place/Transition Systems Preserves Net Behaviour. Informatique Théorique et Applications 26/1, pp.19–44, 1992.

PRS92. L. Pomello, G. Rozenberg, C. Simone: A Survey of Equivalence Notions for Net Based Systems. Lecture Notes in Computer Science 609, pp.410–472, 1992.

Pra84. V. Pratt: Modeling Concurrency with Partial Orders. Int. J. of Parallel Processing, no.15, pp.33–71, 1986.

PW97. L. Priese, H. Wimmel: Algebraic Characterization of Petri Net Pomset Languages. Proceedings of the CONCUR '97, Lecture Notes in Computer Science 1243, pp.406–420, 1997.

PW98. L. Priese, H. Wimmel: A Uniform Approach to True-Concurrency and Interleaving Semantics for Petri Nets. Theoretical Computer Science 206, pp.219–256, 1998.

Rei98. W. Reisig: Elements of Distributed Algorithms. Springer-Verlag, 1998.

Smi98. E. Smith: Principles of High-Level Net Theory. in: Lectures on Petri Nets I, Basic Models, Lecture Notes in Computer Science 1491, pp.174–210, 1998.

Tve67. H. Tverberg: On Dilworth's Decomposition Theorem for Partially Ordered Sets. J. Combin. Theory 3, pp.305–306, 1967.