Universität Augsburg

Determinate STG Decomposition of Marked Graphs

Mark Schäfer and Walter Vogler

Report 2000-12 Juli 2004

Institut für Informatik
D-86135 Augsburg

Abstract

STGs give a formalism for the description of asynchronous circuits based on Petri nets. To overcome the state explosion problem one may encounter during circuit synthesis, a nondeterministic algorithm for decomposing STGs was suggested by Chu and improved by one of the present authors. To find the best possible result the algorithm might produce, it would be important to know to what extent nondeterminism influences the result, i.e. to what extent the algorithm is determinate.

The result of the algorithm clearly depends on the partition of output signals that has to be chosen initially. In general, it also depends on the order of computation steps. We prove that for live and bounded marked graphs — a subclass of Petri nets of definite practical importance in the area of circuit design — the decomposition result depends only on the signal partition. In the proof, we also characterize redundant places in these marked graphs as shortcut places; this easy graph-theoretic characterization is of independent interest.

1 Introduction

Signal Transition Graphs (STG) are a formalism for the description of asynchronous circuits. An STG is a labelled Petri net where the labels denote signal changes between logical high and logical low. The synthesis of circuits from STGs is supported by several tools, e.g. PETRIFY [CKK$^+$97], and it often involves the generation of the reachability graph, which may have a size exponential in the size of the STG (state explosion). To cope with this problem, Chu suggested a nondeterministic method for decomposing an STG into several smaller ones [Chu87]. While there are strong restrictions on the structure and labelling of STGs in [Chu87], the improved decomposition algorithm given in [VW02] works under – comparatively moderate – restrictions on the labelling only. To find a decomposition into components with small reachability graphs, some insight into the space of decompositions the algorithm might produce would clearly be desirable; in other words, one would like to know to what extent the algorithm is determinate.

Roughly, the decomposition algorithm works as follows. Initially, a partition of the output signals has to be chosen, and for each set in this partition a component producing the respective output signals will be constructed. The result clearly depends on this partition, so we will only consider the case that it has been fixed, and we will concentrate on the construction of one component. For each component, one finds a set of signals that (at least initially) can be regarded as irrelevant for the output signals under consideration; then, one takes a copy of the original STG and turns each transition corresponding to an irrelevant signal into an internal (λ-labelled) transition; finally, one tries to remove all internal transitions by so-called secure transition contractions and deletions of (structurally) redundant places.

In general, one might find during this process that additional signals are relevant; then, one has to start anew from a suitably modified copy of the original STG – which eventually gives a correct component as proven in [VW02]. Even in simple cases, the order of operations may influence for which signals this backtracking is performed, resulting in different components as shown in [VW02, Fig. 7]. Since this does not give much hope for reasonable determinacy-results, we will not consider backtracking in this paper; we will mostly concentrate on the subclass of live and bounded marked graphs, for which backtracking is never needed as already noted in [VW02, p. 178]. This class of STGs is particularly prominent in benchmark examples studied in the asynchronous circuit community.

As a result of the above considerations, we can abstract from all signals or signal changes, and study the problem under which circumstances the following algorithm is determinate: given an unlabelled Petri net where some transitions are marked as internal, apply secure transition contractions and redundant place deletions as long as possible.

The paper is organized as follows. In the next section, Petri nets and their basic notions are introduced, as well as redundant places and secure transition contractions. In Section 3, as the first main result, we characterize redundant places in marked graphs as so-called shortcut places. This easy graph-theoretic characterization is of independent interest, but it is also a main ingredient to prove our second main result in Section 4: the algorithm we study is determinate for live and bounded marked graphs, i.e. it produces a unique result (up to isomorphism). As a technical result for general Petri nets, we also show that secure transition contractions satisfy a weak diamond property. We conclude with Section 5. Many proofs and some explanations have been moved to the appendix.

2 Basic Definitions

Definition 1. A *Petri net* is a 4-tuple $N = (P, T, W, M_N)$ with

- P the finite set of *places*, T the finite set of *transitions* with $P \cap T = \emptyset$,
- $W : P \times T \cup T \times P \to \mathbb{N}_0$ the *weight function*,
- M_N the *initial marking*, where a *marking* is a function $P \to \mathbb{N}_0$

A Petri net can be considered as a bipartite graph with weighted and directed edges between its nodes. A marking is a function which assigns a number of *tokens* to each place; markings are extended to sets as usual. A *node* is a place or a transition. □

Definition 2. Let N be a Petri net. The *preset* of a node x is denoted as ${}^\bullet x$ and defined by ${}^\bullet x = \{y \in P \cup T \mid W(y,x) > 0\}$, the *postset* of a node x is denoted as $x^\bullet$ and defined by $x^\bullet = \{y \in P \cup T \mid W(x,y) > 0\}$. We say that there is an *arc* from each $y \in {}^\bullet x$ to x. We write ${}^\bullet x^\bullet$ as shorthand for ${}^\bullet x \cup x^\bullet$. All these notions are extended to sets as usual. □

Whenever a Petri net N, N', N_1, etc. is introduced, the corresponding tuples $(P, T, W, M_N), (P', T', W', M_{N'})$, (P_1, T_1, W_1, M_{N_1}) etc. are introduced implicitly. In a graphical representation of a Petri net places are drawn as circles, transitions as rectangles, the weight function as directed arcs xy (labelled with $W(x,y)$ if $W(x,y) > 1$) and a marking of a place as a number or as a set of small dots drawn in the interior of the corresponding circle.

Definition 3. Let N be a Petri net. A *path* w is a sequence $x_0x_1 \dots x_n,\ n \geq 0$ of different nodes such that $W(x_i, x_{i+1}) > 0\ \forall i = 0, \dots, n-1$. A *cycle* c is a sequence $x_0x_1 \dots x_nx_0,\ n \geq 1$ with $x_0 \dots x_n$ is a path and $W(x_n, x_0) > 0$. Frequently, we will treat paths and cycles like sets consisting of the respective nodes. □

Definition 4. Let N be a Petri net. A transition t is *enabled under a marking* M if $M(p) \geq W(p,t)\ \forall p \in {}^\bullet t$, which is denoted by $M[t\rangle$. An enabled transition can *fire* or *occur* yielding a new marking M', which is written as $M[t\rangle M'$ if $M[t\rangle$ and $M'(p) = M(p) - W(p,t) + W(t,p)\ \forall p \in P$.

A transition sequence $v = t_0t_1 \dots t_n$ is *enabled under a marking* M if $M[t_0\rangle M_0[t_1\rangle M_1 \dots M_{n-1}[t_n\rangle M_n$, and we write $M[v\rangle$, $M[v\rangle M_n$ resp., v is called *firing sequnece* if $M_N[v\rangle$. The empty transition sequence is written as λ and enabled under every marking.

M' is called *reachable from* M if a transition sequence v with $M[v\rangle M'$ exists. The set of all markings reachable from M is denoted by $[M\rangle$. For $[M_N\rangle$ we just write *reachable markings* (of N).

A Petri net is called *live* if for every transition t and every reachable marking M a marking $M' \in [M\rangle$ exists which enables t. □

Definition 5. A place p of a Petri net N is *bounded* if for some $k \in \mathbb{N}$, $M(p) \leq k$ holds for every reachable marking M. N is *bounded* if every place is bounded.

A marking M is a *home marking* or a *home state* of N if it is reachable from every reachable marking. N is called *reversible* if M_N is a home marking. □

Definition 6. A Petri net N is a *marked graph* (MG) (or *T-system*) if:

(1) $\forall p \in P.\ |{}^\bullet p| = 1 = |p^\bullet|$
(2) $\forall x, y \in P \cup T. W(x,y) \leq 1$ □

Due to this, we often identify ${}^\bullet p$ and t if ${}^\bullet p = \{t\}$, and analogously for $p^\bullet$.

Definition 7. [Ber87] A place p of a Petri net N is *structurally redundant* if there is a set of places Q – called *reference set* – with $p \notin Q$, a valuation $V : Q \cup \{p\} \rightarrow \mathbb{N}$ and some $d \in \mathbb{N}_0$ which satisfy the following properties for all transitions t:

(1) $V(p)M_N(p) - \sum_{q \in Q} V(q)M_N(q) = d$
(2) $V(p)(W(t,p) - W(p,t)) - \sum_{q \in Q} V(q)(W(t,q) - W(q,t)) \geq 0$
(3) $V(p)W(p,t) - \sum_{q \in Q} V(q)W(q,t) \leq d$ □

The first two items ensure that p is something like a linear combination of the places in Q with factors $V(q)/V(p)$. Indeed, for the case $d = 0$, the first item says that p is such a combination initially; the second item, in the case of equality, says that this relationship is preserved when firing any transition. The proof that p is indeed semantically redundant argues that the valuated token number of p is at least c larger than the valuated token sum on Q for all reachable markings, while the third item says that each transition needs at most d 'valuated tokens' more from p than from the places in Q; this shows that for the enabling of a transition the presence or absence of p does not matter. Therefore, the deletion of a redundant place in N turns each reachable marking of N into one of the transformed Petri net that enables the same transitions and all reachable markings of the latter net can be obtained this way.

Throughout this paper, if a place p (p', p_1, ...) is considered to be redundant, a corresponding reference set Q (Q', Q_1, ...) and valuation function V (V', V_1, ...) are implicitly given. If only some valuation function V is given, the reference set is implicitly determined as its support by $Q = \{p \in P \mid V(p) > 0\}$.

Furthermore, it is useful to distinguish between different types of redundant places as introduced in the following definition.

Definition 8. Let p be a place of a Petri net N.

- p is an *extended duplicate* of place $p' \in P$ if $\forall t \in T.\ W(p,t) = W(p',t) \wedge W(t,p) = W(t,p')$ and $M_N(p) \geq M_N(p')$.
- p is a *loop-only place* place if $\forall t \in T.\ M_N(p) \geq W(p,t) \leq W(t,p)$.
- If N is a marked graph, p is a *shortcut place* if a path $w = {}^\bullet p \ldots p^\bullet$ exists with $p \notin w$ and $M_N(p) \geq M_N(w \cap P)$. A loop-only place can be considered to be a shortcut place with the path $w = {}^\bullet p$. □

Although a loop-only place is a special form of a shortcut place, it will often be useful to treat them separately in our further considerations for marked graphs.

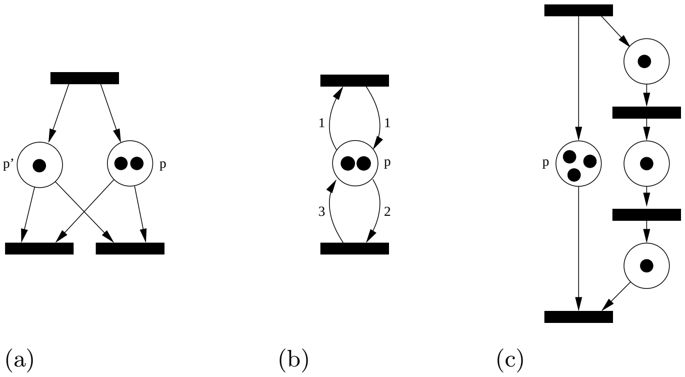


Fig. 1. Examples for redundant places. The redundant place is always labelled p.
(a) Extended duplicate. Observe that p is an extended duplicate of p' but not vice versa. (b) Loop-only place. (c) Shortcut place. Observe that in (b) and (c) $M_N(p)$ cannot be decreased.

Proposition 9.

(1) Extended duplicates, loop-only places and shortcut places are redundant.
(2) If p is a redundant place of a Petri net N, it is a loop-only place iff some reference set Q is empty.

Proof. (1) For an extended duplicate p of place p' set $Q = \{p'\}$, $V(p) = V(p') = 1$. For a loop-only place p set $Q = \emptyset$, $V(p) = 1$. For a shortcut place p with corresponding path w, set $Q = w \cap P$, $V(p) = 1$ and $V(q) = 1$ for $q \in Q$.

(2) The first direction follows from the proof of part (1). Therefore assume the reference set Q to be empty. Since p is redundant we get immediately $\forall t \in T$:

$$\begin{aligned} &V(p)M_N(p) = d \\ &V(p)(W(t,p) - W(p,t)) \geq 0 \\ &V(p)W(p,t) \leq d \end{aligned}$$

Dividing by $V(p)$ and combining the first and the last equation yields: $\forall t \in T.M_N(p) \geq W(p,t),\ W(t,p) \geq W(p,t)$, which is equivalent to the definition of a loop-only place. □

Definition 10. Let N be a Petri net and $t \in T$. If t is not incident to an arc with weight greater 1 and ${}^\bullet t \cap t^\bullet = \emptyset$, we define the *t-contraction of* N, denoted by $\overline{N}^t$ or just $\overline{N}$, as follows:

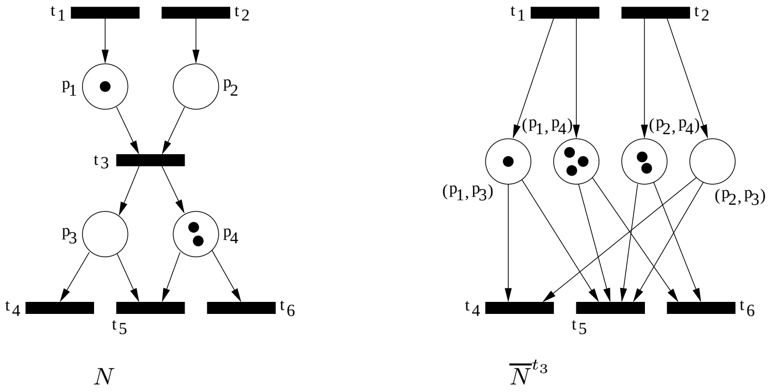
$$\begin{aligned} &\overline{T} = T - \{t\} \qquad \overline{P} = \{(p,\star) | p \notin {}^\bullet t \cup t^\bullet\} \cup \{(p_1,p_2) | p_1 \in {}^\bullet t, p_2 \in t^\bullet\} \\ &\overline{W}((p_1,p_2),t') = W(p_1,t') + W(p_2,t') \\ &\overline{W}(t',(p_1,p_2)) = W(t',p_1) + W(t',p_2) \\ &\overline{M}((p_1,p_2)) = M(p_1) + M(p_2) \end{aligned}$$

In this definition $\star \notin P \cup T$ is a dummy element used to make all places of $\overline{N}$ to be pairs; we assume $M(\star)$, $W(\star,t')$ and $W(t',\star)$ to be 0.

If more than one contraction is applied to a net N, e.g. $\overline{\overline{N}^{t_1}}^{t_2}$, this is denoted by $\overline{N}^{t_1,t_2}$ and analogously for more than 2 transitions.

A t-contraction is called *secure* iff $({}^\bullet t)^\bullet \subseteq \{t\}$ or ${}^\bullet(t^\bullet) = \{t\}$. □

The rationale for secure transition contractions is explained in [VW02]; for this paper it is only important that for marked graphs, all contractions are secure.



N $\overline{N}^{t_3}$

Fig. 2. Example of a transition contraction.

3 Redundant Places in Marked Graphs

This section deals with redundant places in marked graphs. The main result will be that every redundant place in a marked graph is a shortcut place. We start with some well-known statements about marked graphs.

Lemma 11. *(e.g. [DE95]) Let N be a marked graph.*

(1) N is live iff for the initial marking each cycle contains at least one token.
(2) If N is live, it is reversible.

Throughout the rest of this section, *all* Petri nets and in particular marked graphs are assumed to be *live* and *bounded.*

The following definition introduces essential notions for this section. Since the main idea of the central proof is to decrease the size of the reference set Q of a redundant place, it is useful to distinguish different types of subsets of a given reference set.

Definition 12. Let p be a redundant place of a Petri net N.

(1) A path (cycle, set) w is a *Q-path (Q-cycle, Q-set)* if $w \subseteq Q \cup {}^\bullet Q^\bullet$, $q \in w \cap Q$ implies ${}^\bullet q^\bullet \in w$.
(2) A Q-path $(t_0 q_0 \ldots)$ is called *open-origin* if ${}^\bullet t_0 \cap Q = \emptyset$ and $p \notin t_0{}^\bullet$.
(3) A Q-path $(\ldots q_n t_{n+1})$ is called *open-end* if $t_{n+1}{}^\bullet \cap Q = \emptyset$ and $p \notin {}^\bullet t_{n+1}$.
(4) A Q-set Q' is called *isolated* if there is no element of $Q \setminus Q' \cup \{p\}$ which is adjacent to a transition of Q'. □

The last requirement for a set to be a Q-set is needed to exclude useless transitions from it, i.e. transitions which occurrence will not affect the marking of the Q-set. Unfortunately, loop-only places considered as shortcut places are not covered by this definition, in the sense that the corresponding path cannot be a Q-path because $Q = \emptyset$. As mentioned before, this leads to a separate treatment of loop-only places.

Definition 13. Let p be a redundant place of a Petri net N. V is called *balanced* if $V(p)(W(t,p) - W(p,t)) - \sum_{q \in Q} V(q)(W(t,q) - W(q,t)) = 0 \; \forall t \in T$. □

Lemma 14. *Let p be a redundant place of a Petri net N with at least one home marking. Then V is balanced and there exists no open-origin or open-end Q-path.*

Proof. Let M_H be a home marking of N, Using part 2 of Definition 7, it can be shown that $\forall t \in T. M_1[t\rangle M_2 \Rightarrow V(p)M_1(p) - \sum_{q\in Q} V(q)M_1(q) \leq V(p)M_2(p) - \sum_{q\in Q} V(q)M_2(q)$ $(*)$.

Let $M_H[v_1\rangle M[v_2\rangle M_H$, such that v_1 contains every transition $t \in T$ at least once. Such a sequence v_1 exists because N is live, v_2 exists because M_H is a home marking. Together with $(*)$ we get:

$$\begin{aligned} & V(p)M_H(p) - \sum_{q\in Q} V(q)M_H(q) \\ \leq & V(p)M(p) - \sum_{q\in Q} V(q)M(q) \\ \leq & V(p)M_H(p) - \sum_{q\in Q} V(q)M_H(q) \end{aligned}$$

Since N is live, there exists a marking $M_1 \in [M_H\rangle$ for each transition t with $M_1[t\rangle M_2$ and

$$V(p)M_1(p) - \sum_{q\in Q} V(q)M_1(q) = V(p)M_2(p) - \sum_{q\in Q} V(q)M_2(q)$$

Together with $M_2(s) = M_1(s) - W(s,t) + W(t,s)$ $\forall s \in P$ this leads to:

$$\begin{aligned} & V(p)M_1(p) - \sum_{q\in Q} V(q)M_1(q) \\ = & V(p)(M_1(p) - W(p,t) + W(t,p)) - \sum_{q\in Q} V(q)(M_1(q) - W(q,t) + W(t,q)) \\ = & V(p)M_1(p) - (\sum_{q\in Q} V(q)M_1(q)) + V(p)(W(t,p) - W(p,t)) - \sum_{q\in Q} V(q)(W(t,q) - W(q,t)) \\ \Rightarrow & V(p)(W(t,p) - W(p,t)) - \sum_{q\in Q} V(q)(W(t,q) - W(q,t)) = 0 \end{aligned}$$

This implies directly that V is balanced.

We show the second statement by contradiction. Let $w = t_0 q_0 \ldots$ be an open-origin-path and V be a valuation function for p. Lemma 14 implies with Definition 7 (2) that the set $({}^\bullet t_0 \cap Q) \cup (t_0{}^\bullet \cap \{p\})$ is not empty or $V(q_0) = 0$. In the first case w is not open-origin in the latter case w is not a Q-path. □

Lemma 15. *Let p be a redundant place of a marked graph N. Let $c = q_1 t_1 \ldots q_n t_n q_1$ be a Q-cycle with places $q_1, \ldots, q_n$. Then p is redundant due to the valuation V' and its according reference set Q' with:*

$$V'(q) = \begin{cases} V(q) - min_c & \textit{if } q \in c \cap Q \\ V(q) & \textit{else} \end{cases} \quad \textit{and } min_c = \min_{q \in c\cap Q} V(q)$$

Proof. First observe that $p \notin c \cap Q$, since c is a Q-cycle. Obviously part 1 of Definition 7 is fulfilled because:

$$d' = V'(p)M_N(p) - \sum_{q \in Q'} V'(q)M_N(q) \geq V(p)M_N(p) - \sum_{q \in Q} V(q)M_N(q) = d \geq 0$$

Additionally, Lemma 11 implies that at least one token lies on c and therefore $d' - d \geq min_c$.

By Lemma 11 and 14 V is balanced and V' is balanced again since the valuation of the pre- and postset of every transition is decreased by the same amount and part 2 holds, too.

Let us proceed to part 3. This inequality is automatically fulfilled for every transition $t' \neq p^\bullet$ since $V(p)W(p,t') = 0$ in this case. Therefore let $p^\bullet = \{t'\}$. If $t' \notin c \cap T$ we are done. If not, let q' be the only element of $c \cap {}^\bullet t'$. Then

$$\begin{aligned}&(V'(p)W(p,t') - \sum_{q \in Q'} V'(q)W(q,t')) - (V(p)W(p,t') - \sum_{q \in Q} V(q)W(q,t'))\\ &= V(q') - V'(q') = min_c \leq d' - d\end{aligned}$$

From this, it follows immediately: $V'(p)W(p,t') - \sum_{q \in Q'} V'(q)W(q,t') \leq d'$.

□

Lemma 15 is essential for this section. It allows us to delete superfluous places from a reference set in order to simplify its structure. As it will be shown, this can be done until it becomes clear that p is actually a shortcut place. To achieve this, we might have to change $V(p)$, and for this we need a variant of Lemma 15, which we will prove now.

In part 2 of Definition 7, the redundant place p is treated like an element of Q except for the sign of $V(p)$. In order to unify the handling of places, one can change Definition 7 and demand that $V(p)$ be negative or – what is done in this paper – treat p as if it were both, an element of the preset of ${}^\bullet p$ and an element of the postset of $p^\bullet$, i.e. virtually changing the direction of the arcs incident to p, see Figure 3.

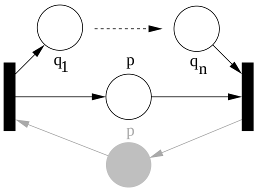


Fig. 3. Virtual cycles.

From this point of view, every Q-path of the form $({}^\bullet p)q_1 \dots q_n(p^\bullet)$ is part of something like a cycle containing p. This is formalised in the following definition.

Definition 16. Let p be a redundant place of a marked graph. If $w = ({}^\bullet p)q_1 \dots$ $\dots q_n(p^\bullet)$ is a Q-path, the sequence pwp is called a *virtual cycle.* □

Analogous to Lemma 15, virtual cycles can be removed from Q under certain circumstances. Different from Lemma 15, we need to show that $V(p)$ remains greater 0.

Lemma 17. *Let p be a redundant place of a marked graph N with Q not containing any Q-cycles. Let $c = p({}^\bullet p)q_1 \dots q_n(p^\bullet)p = pwp$ be a virtual cycle.*

If $M_N(w \cap Q) > M_N(p)$, p is redundant for the valuation V' and the according reference set Q' with $V'(p) \geq 1$.

$$V'(q) = \begin{cases} V(q) - min_c & \textit{if } q \in c \cap P \\ V(q) & \textit{else} \end{cases} \qquad min_c := \min_{q \in c \cap P} V(q)$$

Proof. Condition 1 of the redundancy definition is fulfilled because:

$$\begin{aligned} d' &= V'(p)M_N(p) - \sum_{q \in Q'} V'(q)M_N(q) \\ &= (V(p) - min_c)M_N(p) - (\sum_{q \in w \cap Q} (V(q) - min_c)M_N(q) + \sum_{q \in Q \setminus w} V(q)M_N(q)) \\ &= V(p)M_N(p) - \sum_{q \in Q} V(q)M_N(q) + min_c \cdot ((\sum_{q \in w \cap Q} M_N(q)) - M_N(p)) \\ &= d + min_c \cdot ((\sum_{q \in w \cap Q} M_N(q)) - M_N(p)) > d \geq 0 \end{aligned}$$

Obviously, condition 2 is fulfilled, since — as in the proof of Lemma 15 — the valuation of the preset and the postset of each transition is decreased by the same amount.

For the proof of condition 3, it is sufficient to examine the transition $t' = p^\bullet$, because for all other transitions t the term $V'(p)W(p,t) - \sum_{q \in Q'} V'(q)W(q,t)$ is ≤ 0. Therefore let q' be the only element of ${}^\bullet t' \cap w$. We get:

$$\begin{aligned} &V'(p)W(p,t') - V'(q')W(q',t') \\ &= (V(p) - min_c)W(p,t') - (V(q') - min_c)W(q',t') \\ &= V(p)W(p,t') - V(q')W(q',t') + min_c(W(q',t') - W(p,t')) \\ &\leq d \leq d' \quad \text{(since } W(q',t') - W(p,t') = 0) \end{aligned}$$

Additionally, $V'(p)M_N(p) - \sum_{q \in Q'} V'(q)M_N(q) = d' > d \geq 0$ implies $V'(p) \geq 1$. □

Lemma 18. *Let p be a redundant place of a marked graph N. If p is no loop-only place, ${}^\bullet p$ and $p^\bullet$ are connected by a Q-path.*

Proof. We can assume that Q is not empty (Proposition 9 (2)) and does not have Q-cycles, since each application of Lemma 15 decreases their number.

Furthermore, we know from Lemma 14 that at least one Q-path starts at ${}^\bullet p$. Lemma 14 also implies that no Q-path is open-end and therefore, for every Q-path $w = ({}^\bullet p q_0 \ldots q_n t_n)$ starting at ${}^\bullet p$, we get that $p \in {}^\bullet t_n$ or $t_n^\bullet \cap Q \neq \emptyset$. In the latter case, we can extend w by $q_{n+1} \in t_n^\bullet$ and $q_{n+1}^\bullet$, which is not on w by absence of Q-cycles.

Since N is finite, such Q-paths can only be extended finitely often and at least one path ends at $p^\bullet$. □

We are now ready to prove the main theorem of this section.

Theorem 19. *Every redundant place in a live and bounded marked graph is a shortcut place.*

Proof. Let p be a redundant place. If p is a loop-only place we are done. Therefore let us exclude this case and assume that $Q \neq \emptyset$.

Lemma 18 implies that a Q-path between ${}^\bullet p$ and $p^\bullet$ exists, but we cannot make assumptions about the markings of this Q-path. In the following Q is reduced in a way that eventually only Q-paths between ${}^\bullet p$ and $p^\bullet$ remain.

As a first step, all Q-cycles are removed from Q by repeated application of Lemma 15. This automatically removes all isolated Q-sets.

Then, by repeated application of Lemma 17 all virtual cycles are removed. Let Q' and V' denote the result. At least one Q'-path $w = ({}^\bullet p) \ldots (p^\bullet)$ exists and because the lemma is applicable no longer, we know that $M_N(p) \geq M_N(w \cap Q)$ and p is a shortcut place. □

A weaker version of this theorem could be proved using Theorem 2.25 from [DE95] – and we thank Javier Esparza, who pointed this theorem out to us. Assume p is a redundant place of a live and bounded marked graph N (or more generally: free-choice net N); then the removal of p results again in a live and bounded marked graph N', which is (roughly speaking) strongly connected by [DE95, Theorem 2.25]; in particular the transitions ${}^\bullet p$ and $p^\bullet$ are connected by a path in N'. This result is close to the above theorem, but it is in fact not useful for the purpose of the present paper, since it does not make any statements about the marking of such a path; the pure existence of a path is not sufficient for a place to be redundant. In Figure 4 an example for such a place is given.

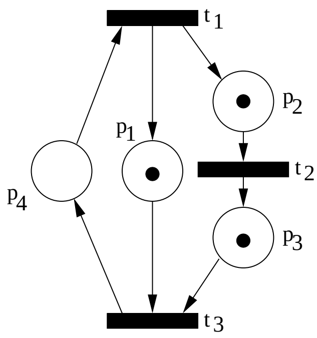


Fig. 4. Example for a non-redundant place with shortcut path

The set of firing sequences of the given net does not contain $t_2t_3t_3$, but this a firing sequence of the net obtained by deleting p_1. Therefore p_1 is not redundant, although a 'shortcut path' $t_1p_2t_2p_3t_3$ exists.

To determine whether a place is structurally redundant, one can set up an instance of linear programming [STC98]. Our theorem leads to a more efficient algorithm for live and bounded marked graphs: to check whether place p is structurally redundant, regard each place p_1 as an edge from ${}^\bullet p_1$ to $p_1^\bullet$, weighted according to the initial marking. Remove the edge corresponding to p and determine the shortest path from ${}^\bullet p$ to $p^\bullet$; if its length (i.e. its cumulated weight) is at most $M_N(p)$, p is redundant. With the basic version of Dijkstra's algorithm, this takes time $O(n^2)$, where n is the number of transitions. Dijkstra's algorithm determines all distances from ${}^\bullet p$ in increasing order; hence, one cannot only stop when the distance for $p^\bullet$ has been found, one can also stop with a negative answer if all transitions with distance at most $M_N(p)$ have been found and $p^\bullet$ is not among them. If $M_N(p) = 0$, one can delete all edges corresponding to initially marked places, and simply check for a path from ${}^\bullet p$ to $p^\bullet$ in the remainder e.g. with depth first search in time linear in the number of transitions and places.

4 Determinacy of Petri Net Operations

In this section the determinacy of the decomposition method — with its operations of secure transition contraction and redundant place deletion — is studied. For this, we view these Petri net operations as a terminating reduction system, such that determinacy is related to confluence and local confluence.

The notion 'reduction system' comes from the field of term rewriting. The following definition and lemma are taken from [BN98], where a detailed introduction can be found.

Definition 20. Let A be a nonempty set with $a, a', \ldots \in A$.

(1) A reduction system is a pair $(A, \rightarrow)$ with $\rightarrow \subseteq A \times A$. The relation $\rightarrow$ is called *reduction* or *reduction rule*; $\rightarrow^*$ denotes the reflexive and transitive closure of $\rightarrow$, and $\rightarrow^=$ the reflexive closure.

(2) A reduction $\rightarrow$

(a) is *terminating* if there exists no infinite chain $a_0 \rightarrow a_1 \rightarrow a_2 \ldots$

(b) is *confluent* if $a \rightarrow^* a_1, a \rightarrow^* a_2$ implies $a_1 \rightarrow^* a', a_2 \rightarrow^* a'$

(c) is *locally confluent* if $a \rightarrow a_1, a \rightarrow a_2$ implies $a_1 \rightarrow^* a', a_2 \rightarrow^* a'$

(d) has the *diamond property* if $a \rightarrow a_1, a \rightarrow a_2$ implies $a_1 \rightarrow a', a_2 \rightarrow a'$

(3) An element a is

(a) in *normal form* if $\neg\exists a'.\ a \rightarrow a'$

(b) a *normal form of* a' if $a' \rightarrow^* a$ and a is in normal form.

Lemma 21.

(1) A terminating relation is confluent iff it is locally confluent.

(2) If $\rightarrow$ is terminating and confluent, every element has a unique normal form.

Next we model the behaviour of the decomposition algorithm as a reduction system. As explained in the introduction, we can restrict ourselves to the processing of one net, where repeatedly structurally redundant places are removed and transitions from a distinguished set are securely contracted. Also, we concentrate on live and bounded marked graphs, although the reduction rules below are actually defined for general nets; Theorem 26 gives a result for general Petri nets.

Definition 22. Let $MGR := \{(N, \Lambda) | N$ is a live and bounded marked graph, $\Lambda \subseteq T\}$, where Λ denotes the set of internal transitions to be contracted. We define the following reduction rules on MGR.

(1) $(N, \Lambda) \rightarrow_{stc} (\overline{N}^t, \Lambda - \{t\})$, where secure contraction of $t \in \Lambda$ is applied.

(2) $(N, \Lambda) \rightarrow_{rpd} (N', \Lambda)$ if N' is obtained from N by deleting a redundant place.

(3) $\rightarrow_{red} = \rightarrow_{stc} \cup \rightarrow_{rpd}$ □

Proposition 23. *Applying $\rightarrow_{red}$ preserves the marked graph properties (Definition 6) as well as liveness and boundedness.*

Proof. For boundedness refer to [VW02]. Deleting a redundant place does not change the firing sequences of the net and therefore liveness is preserved. Since the other places are not affected, the marked graph properties remain valid.

Let $p' = (p_1, p_2)$ be a place resulting from a secure transition contraction. Since p_1 has exactly one place in its preset, so has p', and analogously for the postset. Since the contraction of a transition t shortens each cycle c containing t but leaves $M_N(c)$ unchanged, the cycles of $\overline{N}^t$ still contain at least one token each, and thus $\overline{N}^t$ is live. □

Furthermore, $\rightarrow_{red}$ is a terminating reduction, as noted in [VW02] for general Petri nets: only finite nets are considered, $\rightarrow_{stc}$ reduces the number of transitions, this stays the same under $\rightarrow_{rpd}$, and $\rightarrow_{rpd}$ reduces the number of places.

Each normal form of $(N, \Lambda) \in MGR$ is a possible result of the decomposition algorithm; thus, by Lemma 21, it suffices to show that $\rightarrow_{rpd}$ is locally confluent in order to prove decomposition to be determinate, because in this case every element of MGR has a unique normal form.

To show the local confluence of $\rightarrow_{red}$, we need to show the local confluence for every of the three combinations of $\rightarrow_{stc}$ and $\rightarrow_{rpd}$ as shown in Figure 5.

$$\begin{tikzcd}
(N,\Lambda) \arrow[r, "stc"] \arrow[d, "stc"'] & (N_1,\Lambda_1) \arrow[d, "red"', "*"] \\
(N_2,\Lambda_2) \arrow[r, "red", "*"'] & (N',\Lambda')
\end{tikzcd}
\qquad
\begin{tikzcd}
(N,\Lambda) \arrow[r, "rpd"] \arrow[d, "stc"'] & (N_1,\Lambda) \arrow[d, "red"', "*"] \\
(N_2,\Lambda_2) \arrow[r, "red", "*"'] & (N',\Lambda')
\end{tikzcd}
\qquad
\begin{tikzcd}
(N,\Lambda) \arrow[r, "rpd"] \arrow[d, "rpd"'] & (N_1,\Lambda) \arrow[d, "red"', "*"] \\
(N_2,\Lambda) \arrow[r, "red", "*"'] & (N',\Lambda')
\end{tikzcd}$$

Fig. 5. The three possibilities for the local confluence of $\rightarrow_{red}$. The left and the upper application of a reduction rule is specified, the existence of an appropriate (N', Λ') has to be shown for each case.

Local Confluence of $\rightarrow_{stc}$

We will show now the local confluence for secure transition contractions in marked graphs. Before that, a result for arbitrary Petri nets similar to local confluence is given, namely Theorem 26, which is something like a weak diamond property.

Definition 24. Let N be an STG and N' an STG obtained from N by contracting arbitrary transitions. Each $p' \in P'$ is a structured tuple with components from $P \cup \{\star\}$. $\mathfrak{M}_N^{N'}(p')$ is defined as the multi-set of those places $p \in P$ occurring in p'. □

Lemma 25. *Let N be a Petri net, N' be obtained from N by two transition contractions and $p'_1, p'_2 \in P'$. From $\mathfrak{M}_N^{N'}(p'_1) = \mathfrak{M}_N^{N'}(p'_2)$ it follows that $p'_1 = p'_2$.*

Theorem 26. *Let N be a Petri net and $t_1, t_2 \in T$. If both $\overline{N}^{t_1,t_2}$ and $\overline{N}^{t_2,t_1}$ are defined then they are isomorphic.*

Proof. Let $N_1 = \overline{N}^{t_1,t_2}$ and $N_2 = \overline{N}^{t_2,t_1}$. Furthermore, $f \subseteq P_1 \times P_2 \cup T_1 \times T_2$ is defined by $f|_{T_1 \times T_2} = Id$ and $(p_1, p_2) \in f \Leftrightarrow \mathfrak{M}_N^{N_1}(p_1) = \mathfrak{M}_N^{N_2}(p_2)$. We will show that f is an isomorphism.

a) f is a function: Let $(p_1, p_2), (p_1, p'_2) \in f \Rightarrow \mathfrak{M}_N^{N_2}(p_2) = M_N^{N_2}(p'_2)$. Lemma 25 implies $p_2 = p'_2$.

b) f is injective: Let $f(p_1) = f(p'_1) \Rightarrow \mathfrak{M}_N^{N_1}(p_1) = \mathfrak{M}_N^{N_1}(p'_1)$. From Lemma 25 follows $p_1 = p'_1$.

c) f is surjective, since $p_2 \in P_2$ matches one from the cases in Table 3 and $\mathfrak{M}_N^{N_2}(p_2) = \mathfrak{M}_N^{N_1}(p_1)$ holds for all possible corresponding places p_1 in column 'reverse order' (where exactly one of them exists).

d) f preserves the structure, i.e. $W_1(p_1, t) = W_2(f(p_1), f(t))$, $W_1(t, p_1) = W_2(f(t), f(p_1))$ $\forall p_1 \in P_1, t \in T_1$. This follows rather obviously from the definition of transition contraction. Since the weight of an arc incident to a composite place is the sum of the related weights of the component places, we derive that $W_1(p_1, t_1) = \sum_{p \in \mathfrak{M}_N^{N_1}(p_1)} W(p, t_1) = \sum_{p \in \mathfrak{M}_N^{N_2}(f(p_1))} W(p, t_1) = W_2(f(p_1), f(t_1))$. Analogous for the second case. □

The proof for the following lemma uses Theorem 26; if this is not applicable, we show that – since $N \in MGR$ – in N_1 and N_2 loop-only places can be deleted such that the contraction of t_2 and t_1 resp. is applicable afterwards. After the contraction, extended duplicates can be deleted such that the results are isomorphic.

Lemma 27. *Let $(N, \Lambda) \in MGR$, $(N, \Lambda) \rightarrow_{stc} (N_1, \Lambda_1)$ and $(N, \Lambda) \rightarrow_{stc} (N_2, \Lambda_2)$. Then an $(N', \Lambda') \in MGR$ exists with $(N_1, \Lambda_1) \rightarrow^*_{red} (N', \Lambda')$ and $(N_2, \Lambda_2) \rightarrow^*_{red} (N', \Lambda')$.*

Local Confluence of $\rightarrow_{rpd}$

We will now proceed to the next part of the local confluence proof. Although the local confluence of redundant place deletion seems rather obvious, some effort is already needed to prove it at least for marked graphs.

Let p_1, p_2 be redundant places of $N \in MGR$ with $p_1 \neq p_2$. Due to Theorem 19 we can assume that p_1 and p_2 are shortcut places and the reference sets consist of the places of the corresponding paths.

We will distinguish three cases: 1) $p_1 \notin Q_2, p_2 \notin Q_1$, 2) $p_1 \notin Q_2, p_2 \in Q_1$ (w.l.o.g.) and 3) $p_1 \in Q_2, p_2 \in Q_1$.

The first case obviously fulfils the diamond property, since the deletion of one of the redundant places does neither affect the other one nor its reference set. Furthermore, it includes the case that one place, lets say p_1, is a loop-only place. Then $p_2 \notin Q_1 = \emptyset$ and $p_1 \notin Q_2$, because p_1 is only adjacent to one transition.

For the second case take a look at Figure 6. Since p_1 is not a loop-only place, p_2 lies on a Q_1-path $w_1 = {}^\bullet p_1 q_1^1 \dots q_1^m p_1{}^\bullet$. Since p_2 is not a loop-only place either, a Q_2-path $w_2 = {}^\bullet p_2 q_2^1 \dots q_2^n p_2{}^\bullet$ exists. This implies that there is a path w connecting ${}^\bullet p_1$ and $p_1{}^\bullet$ and using only places from $q_1^1 \dots q_1^m$ excluding p_2 and from $q_2^1 \dots q_2^n$. $M_N(p_1) \geq \sum_{i=1}^m M_N(q_1^i)$ and $M_N(p_2) \geq \sum_{i=1}^n M_N(q_2^i)$ (Definition 7(1)) directly imply that $M_N(p_1) \geq \sum_{i=1}^m M_N(q_1^i) - M_N(p_2) + \sum_{i=1}^n M_N(q_2^i)$; hence, w also shows that p_1 is redundant; the corresponding reference set does not contain p_2 and we are done by case (1).

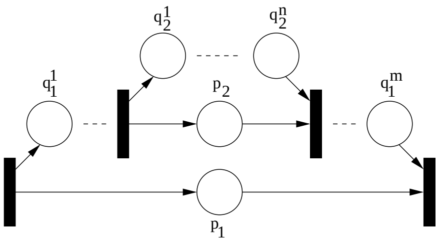


Fig. 6. Two redundant places p_1, p_2 with $p_1 \notin Q_2, p_2 \in Q_1$

The last case $p_1 \in Q_2, p_2 \in Q_1$ is impossible, because it implies

$$M_N(p_1) \geq \sum_{q \in Q_1 \setminus \{p_2\}} M_N(q) + M_N(p_2) \qquad M_N(p_2) \geq \sum_{q \in Q_2 \setminus \{p_1\}} M_N(q) + M_N(p_1)$$

From this we get immediately:

$$M_N(p_1) = M_N(p_2) \quad \text{and} \quad \sum_{q \in Q_1 \setminus \{p_2\}} M_N(q) = \sum_{q \in Q_2 \setminus \{p_1\}} M_N(q) = 0 \quad (*)$$

Since $p_1 \in Q_2$, there are Q_2-paths ${}^\bullet p_2 \dots {}^\bullet p_1$ and $p^\bullet{}_1 \dots p^\bullet{}_2$ not using p_1, and analogously there are Q_1-paths ${}^\bullet p_1 \dots {}^\bullet p_2$ and $p^\bullet{}_2 \dots p^\bullet{}_1$ not using p_2. Therefore, either a cycle c using only places from $(Q_1 \cup Q_2) \setminus \{p_1, p_2\}$ exists which contradicts N being live by Lemma 11, since $(*)$ implies $M_N(c) = 0$; or

$(Q_1 \cup Q_2) \setminus \{p_1, p_2\} = \emptyset$. In the latter case, p_1 and p_2 are extended duplicates of each other with the same initial marking; thus, removing either of them gives the same net up to isomorphism.

Altogether the following lemma holds.

Lemma 28. *Let $(N, \Lambda) \in MGR$, $(N, \Lambda) \rightarrow_{rpd} (N_1, \Lambda_1)$ and $(N, \Lambda) \rightarrow_{rpd} (N_2, \Lambda_2)$. Then an $(N', \Lambda') \in MGR$ exists with $(N_1, \Lambda_1) \rightarrow^{=}_{rpd} (N', \Lambda')$ and $(N_2, \Lambda_2) \rightarrow^{=}_{rpd} (N', \Lambda')$.*

Observe that two steps of $\rightarrow_{rpd}$ fulfil the diamond property or lead to isomorphic results; in particular we have not used $\rightarrow_{stc}$.

Local confluence of $\rightarrow_{stc}$ and $\rightarrow_{rpd}$

Lemma 29. *Let $(N, \Lambda) \in MGR$, $(N, \Lambda) \rightarrow_{rpd} (N_1, \Lambda_1)$ and $(N, \Lambda) \rightarrow_{stc} (N_2, \Lambda_2)$. Then an $(N', \Lambda') \in MGR$ exists with $(N_1, \Lambda_1) \rightarrow^{*}_{red} (N', \Lambda')$ and $(N_2, \Lambda_2) \rightarrow^{*}_{red} (N', \Lambda')$.*

Proof. Let p be the redundant place and t the transition to be contracted. In marked graphs p is either a loop-only place or a shortcut place.

In the first case t and p are not adjacent because the contraction of t is possible for (N, Λ), i.e. p forms a loop with another transition and the operations can be performed independently.

If p is not a loop-only place, there are the following possibilities: 1) t is neither adjacent to p nor part of the path making p redundant; then both operations are independent of each other again. 2) t is part of the path but not adjacent to p. The contraction of t shortens the path but does not interrupt it, and also the sum of the markings remains unchanged; therefore the two operations are independent. 3) t is adjacent to both the path and p – leading to two sub-cases, one of them shown in Figure 7(a). In the other one, analogously the path starts from t and $p \in t^\bullet$.

We will only consider the case depicted in (a), with the results of contraction and deletion shown in (b) and (c) resp. Each place (p_s, p_{xi}) in (b) is a shortcut place of $\{(p_1, *),\ \dots, (p_{n-1}, *),\ (p_n, p_{xi})\}$ because they give a path and the initially marking of this path as well as $M_N(p_s)$ are increased by the same value $M_N(xi)$. Therefore, these shortcut places can be deleted yielding a Petri net which also results from (c) when contracting t. □

Altogether, our results can be collected in the central theorem of this section.

Theorem 30. *The reduction rule $\rightarrow_{red}$ is confluent and terminating for marked graphs.*

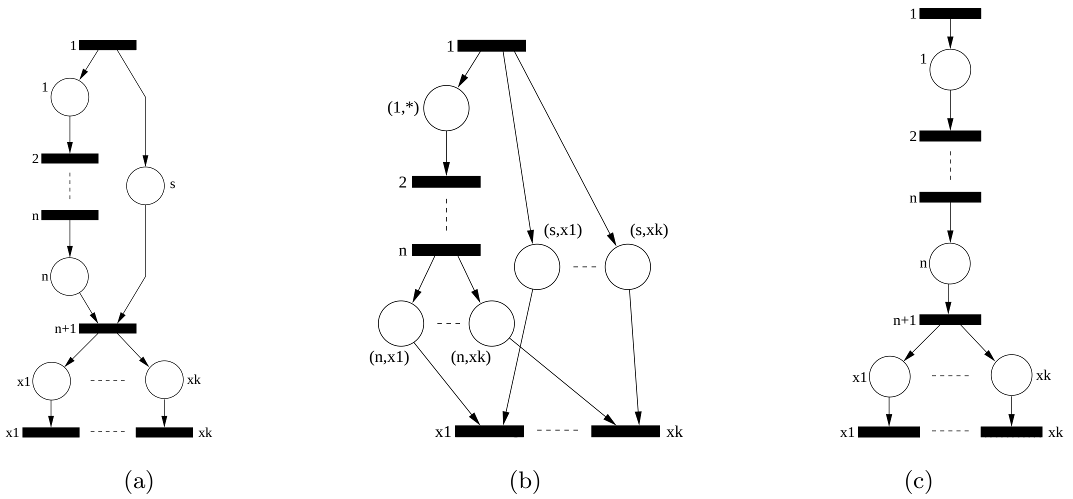


Fig. 7. Confluence of shortcut place deletion and transition contraction. (a) $p \equiv p_s$ is a shortcut place of $\{p_1, \ldots, p_n\}$ and $t \equiv t_{n+1}$ is the transition to be contracted. The net in (b) is obtained by contracting t_{n+1}, (c) by deleting p_s.

Corollary 31. *The decomposition algorithm of [VW02] is determinate for marked graphs.*

5 Conclusion

We have shown that the STG decomposition algorithm presented in [VW02] is determinate if applied to live and bounded marked graphs, a subclass of considerable interest in the area of circuit design. The proof of this result is based on several statements, and only one of them could be shown for general Petri nets. It would be clearly interesting to generalise at least some other of the partial results to other net classes. We currently look at nets where the marked-graph requirements are only violated 'in a few places'; such nets also turn up often in circuit design. A problematic point is that our proofs relied on the liveness characterization of marked graphs via the markings of cycles several times.

Related to the determinacy result, but also of independent interest is our conceptionally and algorithmically easy characterization of redundant places in live and bounded marked graphs, a rather old concept. Again, we would like to generalise this result; Until now, it is only clear that in S-Systems [DE95] — which coincide with finite automata — no place can be redundant if every place has at least one transition in its postset.

References

[Ber87] G. Berthelot. Transformations and decompositions of nets. In W. Brauer et al., editors, *Petri Nets: Central Models and Their Properties*, LNCS 254, 359–376. Springer, 1987.

[BN98] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, Cambridge, 1998.

[Chu87] T.-A. Chu. *Synthesis of Self-Timed VLSI Circuits from Graph-Theoretic Specifications.* PhD thesis, MIT, 1987.

[CKK$^+$97] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev. Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers. *IEICE Transactions on Information and Systems*, E80-D, No. 3:315–325, 1997. More information at http://www.lsi.upc.es/ jordic/petrify/petrify.html.

[DE95] J. Desel and J. Esparza. *Free Choice Petri Nets.* Cambridge University Press, Cambridge, 1995.

[STC98] M. Silva, E. Teruel, and J.M. Colom. Linear algebraic and linear programming techniques for the analysis of place/transition net systems. In *Lectures on Petri Nets I; Basic Models*, LNCS 1491, 309–373. Springer, 1998.

[VW02] W. Vogler and R. Wollowski. Decomposition in asynchronous circuit design. In J. Cortadella et al., editors, *Concurrency and Hardware Design*, Lect. Notes Comp. Sci. 2549, 152 – 190. Springer, 2002.

Appendix

A Proofs

A.1 Proof of Lemma 25

Proof. This proof works with the tables 1 and 2. In the first one, all possibilities for the structure of a place after two transition contractions are listed. In the latter one these 6 cases are instantiated resulting in 30 possible combinations of places from the original net.

In the following we will show the impossibility of different cases - esp. the more complicated ones. For most cases this becomes clear at first sight, e.g. if (p_1, p_1) is part of the place, which means that $p_1 \in {}^\bullet t$ and $p_1 \in t^\bullet$ for one of the contracted t. For such a configuration an extension of the definition of transition contraction seems possible. But for something like $((p_1, \star), (p_1, p_2))$ - p_1 is treated in two incompatible ways during the first contractions - a sensible extension is not in sight.

Hence, the cases 2, 4, 6-9, 11-14, 16-22, 26 and 29 instantly drop out. The remaining impossible cases 23-25, 27, 28 and 30 are considered in more detail.

Case 23 drops out, because p_1 is part of the preset of the first transition due the occurrence of (p_1, p_2) or otherwise p_1 has to be element of the postset, too, due to the occurrence of (p_2, p_1). Therefore p_1 forms a loop with the first contracted transition. With the same argumentation cases 24 and 28 are impossible.

Case 25 leads to a circle or an arc with weight 2, see figure 8. Case 27 is very similar to the previous one, only the pre - and postsets of t_1 are exchanged.

At last case 30 remains which is more complicated but nevertheless turns out to be impossible, see figure 9.

Obviously we can restrict our considerations to the cases in table 3, second column. We can distinguish three cases for $\mathfrak{M}_N^{N'}(p'_1)$.

(1) $\mathfrak{M}_N^{N'}(p'_1) = \{p_1\} = \mathfrak{M}_N^{N'}(p'_2)$. This is only possible if both p'_1 and p'_2 are in the form of case 1 which implies $p'_1 = p'_2$.

(2) $\mathfrak{M}_N^{N'}(p'_1) = \{p_1, p_2\} = \mathfrak{M}_N^{N'}(p'_2)$. This implies $p'_1 \in \{((p_1, p_2), \star), ((p_2, p_1), \star), ((p_1, \star), (p_2, \star)), ((p_2, \star), (p_1, \star))\}$. Each of these cases excludes the others e.g. if $p'_1 = ((p_1, p_2), \star)$ there is no place $p''_1 = ((p_2, p_1), \star)$, since the existence of p'_1 implies that p_1 is an element of the first contracted transition but the existence of p''_1 implies p_1 is an element of the postset. This can be true but then the contraction would not be possible. Therefore $\mathfrak{M}_N^{N'}(p'_1) = \mathfrak{M}_N^{N'}(p'_2)$ implies $p'_1 = p'_2$ for this case.

(3) $\mathfrak{M}_N^{N'}(p_1') = \{p_1, p_2, p_3\} = \mathfrak{M}_N^{N'}(p_2')$. Analogous to the second case we obtain twelve possibilities for p_1 which all exclude each other.

1	$((p_1, p_2), (p_3, \star))$	$((p_1, \star), (p_2, p_3))$
2	$((p_1, p_3), (p_2, \star))$	$((p_1, \star), (p_3, p_2))$
3	$((p_2, p_1), (p_3, \star))$	$((p_2, \star), (p_1, p_3))$
4	$((p_2, p_3), (p_1, \star))$	$((p_2, \star), (p_3, p_1))$
5	$((p_3, p_1), (p_2, \star))$	$((p_3, \star), (p_1, p_2))$
6	$((p_3, p_2), (p_1, \star))$	$((p_3, \star), (p_2, p_1))$

To see this, it is not necessary to consider all 66 cases. It suffices to show that both cases in the first line exclude all other places since a suitable renaming of the places from N results in them. $((p_1, p_2), (p_3, \star))$ is in conflict with all places not containing $(p_3, \star)$ as a sub-place. Since $(p_3, \star)$ implies that p_3 is not adjacent to the first contracted transition the occurrence of e.g. (p_1, p_3) implies the opposite. From the remaining three cases (ll. 3,5,6) we can exclude the ones containing (p_2, p_1) as a component since p_1 would be a loop place. The last place (l. 5) cannot exist since (p_1, p_2) would be a loop place. Analogous for $((p_1, \star), (p_2, p_3))$. □

Group	Structure
1	$((p, \star), \star)$
2	$((p, p), \star)$
3	$((p, \star), (p, \star))$
4	$((p, \star), (p, p))$
5	$((p, p), (p, \star))$
6	$((p, p), (p, p))$

Table 1. Structures of possible places. This table is obtained from all syntactically possible places by omitting cases which contains a leading $\star$, e.g. $(\star, (p, \star))$. Here a p is only a placeholder; in table 2 all possible allocations are considered.

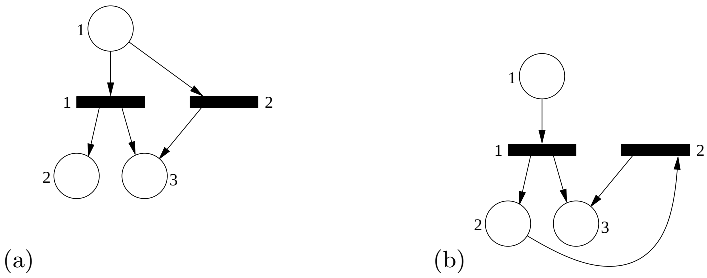
A.2 Proof of Lemma 27

Proof. If both $\overline{N}^{t_1,t_2}$ and $\overline{N}^{t_2,t_1}$ are defined, Theorem 26 implies that the results are isomorphic. In this case even the diamond property is fulfilled.

Therefore assume that w.l.o.g. $\overline{N}^{t_1,t_2}$ is not defined. Since $N_1 = \overline{N}^{t_1}$ is defined by hypothesis, the contraction of t_2 is not possible in N_1, although it is possible in N. Since N_1 is a marked graph — in particular no arc weight becomes greater than 1 —, the contraction of t_1 in N must have generated a loop place adjacent to t_2, because t_1 and t_2 form a cycle with two places in N.

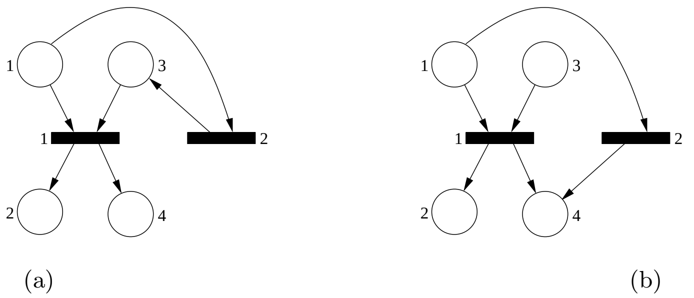
No.	Group	# Places	Example	Possible	If not, why?
1	1	1	$((p_1, \star), \star)$	●	
2	2	1	$((p_1, p_1), \star)$	-	loop
3	2	2	$((p_1, p_2), \star)$	●	
4	3	1	$((p_1, \star), (p_1, \star))$	-	loop
5	3	2	$((p_1, \star), (p_2, \star))$	●	
6	4	1	$((p_1, \star), (p_1, p_1))$	-	` definition
7	4	2	$((p_1, \star), (p_1, p_2))$	-	` definition
8	4	2	$((p_1, \star), (p_2, p_1))$	-	` definition
9	4	2	$((p_2, \star), (p_1, p_1))$	-	loop
10	4	3	$((p_1, \star), (p_2, p_3))$	●	
11	5	1	$((p_1, p_1), (p_1, \star))$	-	` definition
12	5	2	$((p_1, p_1), (p_2, \star))$	-	loop
13	5	2	$((p_1, p_2), (p_1, \star))$	-	` definition
14	5	2	$((p_2, p_1), (p_1, \star))$	-	` definition
15	5	3	$((p_1, p_2), (p_3, \star))$	●	
16	6	1	$((p_1, p_1), (p_1, p_1))$	-	loop
17	6	2	$((p_1, p_1), (p_1, p_2))$	-	loop
18	6	2	$((p_1, p_1), (p_2, p_1))$	-	loop
19	6	2	$((p_1, p_2), (p_1, p_1))$	-	loop
20	6	2	$((p_2, p_1), (p_1, p_1))$	-	loop
21	6	2	$((p_1, p_1), (p_2, p_2))$	-	loop
22	6	2	$((p_1, p_2), (p_1, p_2))$	-	loop
23	6	2	$((p_2, p_1), (p_1, p_2))$	-	loop
24	6	3	$((p_1, p_2), (p_3, p_1))$	-	loop
25	6	3	$((p_1, p_2), (p_1, p_3))$	-	loop or weight 2
26	6	3	$((p_1, p_1), (p_2, p_3))$	-	loop
27	6	3	$((p_2, p_1), (p_3, p_1))$	-	weight 2
28	6	3	$((p_2, p_1), (p_1, p_3))$	-	weight 2
29	6	3	$((p_2, p_3), (p_1, p_1))$	-	loop
30	6	4	$((p_1, p_2), (p_3, p_4))$	-	loop

Table 2. All combinatory possible places (up to isomorphism) after two transition contractions. This table is obtained from table 1 by instantiating p. The places p_i are pairwise different. The places which have an '` definition'-entry are absolutely not possible, since a place is treated in two incompatible ways. For the other impossible cases (with a 'circle' or 'weight 2'-entry) an extension of the definition is supposable which covers such cases.



(a) (b)

Fig. 8. Case 25 - $p' = ((p_1, p_2), (p_1, p_3))$. p_1 has to be an element of the preset of the first contracted transition (t_1), p_2 and p_3 have to be elements of the postset. To obtain p', p_1 has to be element of ${}^{\bullet}t_2$ and p_2 or p_3 have to be element of $t_2{}^{\bullet}$ (a) leading to a circle when contracting in reverse order. Alternatively, p_2 can be element of ${}^{\bullet}t_2$ and p_3 element of $t_2{}^{\bullet}$ (b) leading to an arc with weight 2 when contracting in reverse order.



(a) (b)

Fig. 9. Case 30 - $((p_1, p_2), (p_3, p_4))$. p_1 and p_3 have to be in the preset of the first transition to be contracted (t_1), p_2 and p_4 in the postset. For the connection to t_2 there are several possibilities; all of them satisfy that p_1 or p_2 (or both) are in the preset and p_3 or p_4 (or both) are in the postset, which leads to 9 sub-cases. Exemplary two of them are considered. (a) leads to an arc with weight 2 when t_2 is contracted first and (b) leads to a circle. The other cases are similar to these ones or contain them.

Since N is a live marked graph, this cycle contains at least one token making the loop place redundant.

This situation is schematically shown in Figure 10(a): each place represents a set of places connected to t_1 and t_2 in the same way, e.g. places of type 1 are in the preset of t_1 and not adjacent to t_2. Figure 10(b) and (c) depict the results of contracting t_1 and t_2 resp. in the same way, e.g. places of type $(2, 4)$ are pairs (p, p') with p of type 2 and p' of type 4.

Places of type $(2, 5)$ and $(5, 2)$ are loop-only places, which can be removed as noted above; afterwards, the other transition contraction becomes possible. These contractions give places of types $((1, 4), *)$, $((1, 5), (3, \star))$, $((1, 5), (2, 4))$, $((6, *), (2, 4))$, $((6, *), (3, *))$ in the first case and $((1, *), (4, *))$, $((1, *), (5, 3))$, $((6, 2), (5, 3))$, $((6, 2), (4, *))$, $((6, 3), *)$ in the second. We will argue that the resulting nets are isomorphic after removal of some redundant places.

As noted in the proof of Theorem 26, the connections of these places to the remaining transitions are determined by their at most four components, and analogously for the initial marking. In particular, places of type $((1, 5), (2, 4))$

No.	Given Order	Reverse Order
1	$((p_1,\star),\star)$	$((p_1,\star),\star)$
2	$((p_1,p_2),\star)$	$((p_1,\star),(p_2,\star))$
3	$((p_1,\star),(p_2,\star))$	$((p_1,p_2),\star)$
4	$((p_1,\star),(p_2,p_3))$	$((p_1,p_2),(p_3,\star))$ / $((p_2,\star),(p_1,p_3))$
5	$((p_1,p_2),(p_3,\star))$	$((p_1,\star),(p_2,p_3))$ / $((p_1,p_3),(p_2,\star))$

Table 3. Possible places after two transition contractions. These are the cases from table 2 which turned out to be possible for Definition 10. In the column 'reverse order' the places resulting from contracting the transitions in reverse order are written.

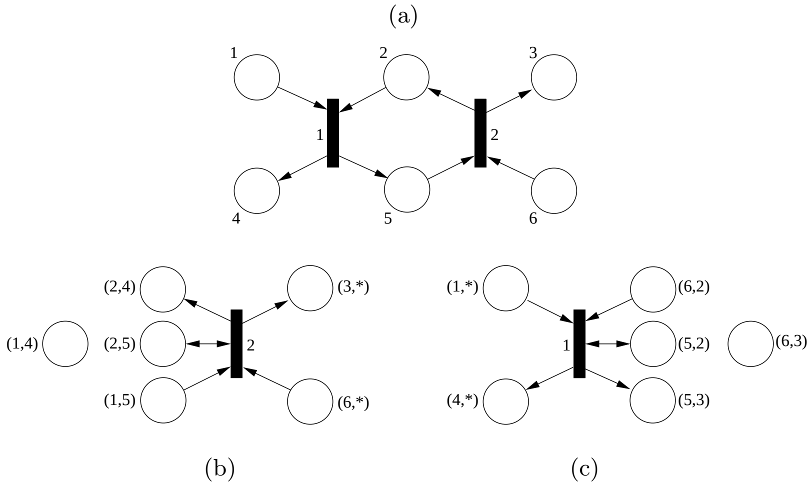


Fig. 10. (a) Scheme of a net fragment where contraction generates a loop (b) After t_1-contraction (c) After t_2-contraction.

are connected in the same way as places of type $((1,4),*)$ in the first case – since t_1 and t_2 are not present anymore – and they carry even more tokens, since at least one of a type-2 and a type-5 place is marked in N. Therefore, places of type $((1,5),(2,4))$ are extended duplicates, and so are places of type $(6,2),(5,3))$; we remove them in the two nets.

For the other types, we find a matching between $((1,4),*)$ and $((1,*),(4,*))$, $((1,5),(3,*))$ and $((1,*),(5,3))$ etc., which matches each place of type $((1,4),*)$ to the place of type $((1,*),(4,*))$ with the same component-places etc. By the above, this gives an isomorphism between the remaining nets when the above extended duplicates are removed. □