Determinate STG Decomposition of Marked Graphs[⋆]

Mark Schäfer[1], Walter Vogler[1], and Petr Jančar[2,⋆⋆]

[1] Institut für Informatik, Universität Augsburg
`{mark.schaefer, walter.vogler}@informatik.uni-augsburg.de`
[2] Centre for Applied Cybernetics, Technical University of Ostrava
`petr.jancar@vsb.cz`

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 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 characterise redundant places in these marked graphs as shortcut places; this easy-to-apply graph-theoretic characterisation 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. Signals are subdivided into input signals, which are produced by the environment, and output signals, which the circuit should produce as specified by the STG. 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

⋆ This work was partially supported by the DFG-project 'STG-Dekomposition' Vo615/7-1.

⋆⋆ This author is supported by the Czech Ministry of Education, Grant No. 1M6840770004.

STGs in [Chu87], the improved decomposition algorithm given in [VW02] works under – comparatively moderate – restrictions on the labelling only.

Roughly, the decomposition algorithm works as follows; see [VW02] for details. 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 is 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. To construct a 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, resulting in the final component.

The aim is to find components with small reachability graphs. In principle, this requires to consider all possible sequences of contractions and deletions; but if the algorithm is determinate, i.e. nondeterminism does not influences the result, it is sufficient to consider only one sequence, which greatly increases efficiency. Our main contribution is a determinacy result for a subclass of STGs, where a part of the result applies to STGs in general.

In general, one might find during the processing of a component 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 a general determinacy-result, we will not consider backtracking in this paper; we will mostly concentrate on the subclass of live marked graphs, for which backtracking is never needed as already noted in [VW02–p. 178].

Although marked graphs are a rather restricted subclass of Petri nets, our results for this subclass are non-trivial. Marked graphs are definitely of practical importance for asynchronous circuit and particularly prominent in benchmark examples studied in the respective 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.

We will show that for live marked graphs the algorithm is determinate, i.e. it produces a unique component (up to isomorphism). Part of this result applies to general Petri nets, for which we show that secure transition contractions satisfy a weak diamond property. We give an easy-to-apply graph-theoretic characterisation of redundant places in marked graphs as so-called shortcut places; our result is a small generalisation of a result in [CCJS94] and our contribution is a much simpler proof. This result is an important ingredient to prove our main result.

The paper is organised 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, we characterise redundant places in marked graphs as shortcut places. The other contribution is proven in Section 4. We conclude with Section 5.

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; for a (sub)set Q of places we define $M(Q) = \sum_{p \in Q} M(p)$ (where the sum is zero if $Q = \emptyset$). A *node* is a place or a transition. $\square$

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. $\square$

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. We will regard isomorphic Petri nets as equal.

Definition 3. Let N be a Petri net. A *path* w is a sequence $x_0 x_1 \dots x_n$, $n \geq 0$ of different nodes such that $W(x_i, x_{i+1}) > 0$ $\forall i = 0, \dots, n-1$. A *cycle* c is a sequence $x_0 x_1 \dots x_n x_0$, $n \geq 1$ 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. By the marking of a path (cycle resp.) we mean the marking (i.e. the sum of the tokens) of the set of its places.

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 $v = t_0 t_1 \dots t_n$ 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 sequence* 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 transition t is called *live under a marking* M if for every $M' \in [M\rangle$ there exists an $M'' \in [M'\rangle$ with $M''[t\rangle$, t is *live* if it is live under M_N and N is live if every $t \in T$ is live. A transition t is *dead under a marking* M if there is no $M' \in [M\rangle$ with $M'[t\rangle$. □

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* 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. Let N be a Petri net and $p \in P$. Place p is called *implicit* if it can be removed from N without changing the set of firing sequences.

Place p is *(structurally) redundant* [Ber87] if there is a set of places Q – called *reference set* – with $p \notin Q$, a valuation $V : Q \cup \{p\} \to \mathbb{N}$ 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. $V(p)W(p,t) - \sum_{q \in Q} V(q)W(q,t) \leq d$

We call V *balanced* if, for all transitions $t \in T$, $V(p)(W(t,p) - W(p,t)) - \sum_{q \in Q} V(q)(W(t,q) - W(q,t)) = 0$. □

When constructing a component in our STG decomposition, we would like to remove implicit places. Implicitness is hard to decide, and therefore we actually consider only structural redundancy, since checking this does not require to generate the reachability graph.

Remark: It is well-known that the reachability problem (RP) for Petri nets is EXPSPACE-hard. This even holds for SPZ-RP (Single-Place-Zero RP) where we ask if a given place p can be emptied; we can also assume arc-weights to be 1. Given an instance of SPZ-RP, we can add a fresh t and the arcs (p, t), (t, p), and observe that p is implicit iff no marking with zero tokens in p is reachable. This shows EXPSPACE-hardness of the implicitness problem. On the other hand, redundancy can be solved by linear programming.

The proof that a redundant place p is indeed implicit argues that initially the valuated token number of p is at least d greater than the valuated token sum on Q by the first item, and that this difference can only get greater when firing transitions by the second item; 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.

Since deletion of p preserves the firing sequences it also preserves liveness. In general implicitness does not imply redundancy, but we will show that these notions coincide for live marked graphs.[1]

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 containing at least one place but not p and satisfying $p \notin w$ and $M_N(p) \geq M_N(w \cap P)$. □

Definition 9. Let N be a Petri net and $t \in T$. If t is not incident to an arc with weight greater than 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:

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

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 $\overline{N}^{t_1,t_2}$.

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]. In this paper, arbitrary contractions in general Petri nets are considered in Theorem 20; otherwise, we consider marked graphs where all contractions are secure.

[1] [CCJS94] shows that the second redundancy item characterizes that p is *structurally implicit*, i.e. each marking of the other places can be extended to p such that p is implicit.

3 Redundant Places in Marked Graphs

This section deals with redundant and implicit places in live marked graphs. The main result will be that redundant and implicit places coincide in live marked graphs and furthermore they are either loop-only places or shortcut places.

We start with two propositions about redundant places in general.

Proposition 10.

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 (in)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. $\square$

The first part of the following proposition was used in an alternative proof of Theorem 13, and we think that it is of independent interest. The second part will be applied below.

Proposition 11. *1. Let p be a redundant place of a live Petri net N with at least one home marking. Then V is balanced.*
2. If, in an arbitrary net N, p is redundant under a marking $M \in [M_N\rangle$ with a balanced valuation, it is also redundant under M_N with the same valuation. In particular, if p is a shortcut place under M, it is also one under M_N.

Proof. 1) 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:

$$V(p)M_H(p) - \sum_{q\in Q} V(q)M_H(q)$$

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

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:

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

This implies directly that V is balanced.

2) Items 2 and 3 of Definition 7 do not depend on the marking and item 1 follows directly from the valuation being balanced. If p is a shortcut place then the respective path induces a balanced valuation V (as observed in the proof of 10) and, since item 1 can be transferred from M to M_N, the marking of this path is at most the marking of p also under M_N. □

Before we prove the main theorem of this section, we note an easy lemma about liveness in marked graphs.

Lemma 12. *Let c be a cycle of a marked graph N. For every reachable marking M, $M(c) = M_N(c)$. If N is live, c is initially marked.*

Proof. Let $M[t\rangle M'$. We show that $M(c) = M'(c)$. For $t \in c$ this is trivially true, since all edge weights are 1 in marked graphs. Otherwise, t is not adjacent to any place of c, since N is a marked graph.

The second statement now follows easily; if c is not marked under M_N it is not marked under any reachable marking and therefore no transition of c can ever fire, a contradiction. □

Actually, marked graphs are live if and only if every cycle is initially marked, see e.g. [DE95]. This is a deeper result, which we do not need here. In fact, our proof of the next theorem has the advantage that it does not require profound knowledge about marked graphs, and we only proved the above lemma to demonstrate that our proof of Theorem 13 is indeed elementary.

Theorem 13. *Let N be a live marked graph and $p \in P$. The following properties are equivalent:*

1. *p is a redundant place*
2. *p is an implicit place*
3. *p is a loop-only place or a shortcut place*

Proof. "1$\rightarrow$2" even holds for arbitrary Petri nets – as we observed already –, and "3$\rightarrow$1" follows from Proposition 10.

"2$\rightarrow$3": Let p be an implicit place but not a loop-only one. We define $\{t_i\} = {}^\bullet p$ and $p^\bullet = \{t_o\}$, obviously $t_i \neq t_o$, otherwise $M_N(p) = 0$, which contradicts with liveness. Let N' be the net obtained from N by deleting t_i and all incident arcs. Observe that p is also implicit in N', since the set of firing sequence of N' coincides with the set of those firing sequences of N which do not contain t_i.

In N', starting from the initial marking we fire transitions until a maximal set D of transitions is dead.[2] From this marking fire every transition not in D at least once; we denote the marking reached by M. Observe that $(*)$ M can be reached in N by the same firing sequence.

Since t_o can fire at most $M_N(p)$ times in N', we must have $t_o \in D$. Furthermore, there exists a $p_1 \in {}^\bullet t_o$, $p_1 \neq p$ with $M(p_1) = 0$. If not, p would be the only place in ${}^\bullet t_o$ preventing the firing of t_o, hence would not be implicit in N'.

This implies ${}^\bullet p_1 \in D$; otherwise p_1 would have been marked when every transition not in D fired once. Now there is an unmarked place p_2 in ${}^\bullet({}^\bullet p_1)$ and so on. This leads either to a cycle not containinig any tokens, which is by $(*)$ a contradiction to N being live (cf. Lemma 12); or ends up in a place p' with an empty preset in N', hence $p' \in t_i{}^\bullet$ and so we have constructed an unmarked path from t_i to t_o not containing p. Therefore p is a shortcut place under M in N, cf. $(*)$, and we are done by Proposition 11.2. $\square$

Remark: Javier Esparza pointed out to us that a weaker version of this theorem could be proved as follows. 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 [Bes87]; 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.

A result very close to Theorem 13 can be found in [CCJS94]. The difference is that strong connectedness is assumed there – an assumption that we do

[2] D does not neccessarily contain all transitions, since we do not assume boundedness or connectedness.

not need. Furthermore, the proof in [CCJS94] makes heavy use of deep results about marked graphs, while our direct proof only needs elementary knowledge. [CCJS94] also considers some form of decomposition of marked graphs; we will discuss the relationship to our approach at the end of the next section.

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 marked graphs as already noted in [CCJS94]: 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.

Actually, in [CCJS94] the addition of implicit places is considered; for deciding whether a given place is redundant we note the following improvement.

Dijkstra's algorithm determines all distances from ${}^\bullet p$ in increasing order; hence, the algorithm can already be finished with a negative answer, if all transitions with a distance of no more than $M_N(p)$ have been found and if $p^\bullet$ 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 14. 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 $a_1 \rightarrow^* a', a_2 \rightarrow^* a'$ for some a'
   (c) is *locally confluent* if $a \rightarrow a_1, a \rightarrow a_2$ implies $a_1 \rightarrow^* a', a_2 \rightarrow^* a'$ for some 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 15.

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 marked graphs, although the reduction rules below are actually defined for general nets; Theorem 20 gives a result for general Petri nets.

Definition 16. Let $MGR := \{(N, \Lambda) | N$ is a live 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. $\rightarrow_{red} = \rightarrow_{stc} \cup \rightarrow_{rpd}$ □

These reductions are well-defined according to the following proposition.

Proposition 17. *Applying $\rightarrow_{red}$ preserves the marked graph properties (Definition 6) as well as liveness.*

Proof. 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 transition 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 15, it suffices to show that $\rightarrow_{red}$ is locally confluent in order to prove decomposition to be determinate, because in this case every element of MGR has a unique normal form; recall that we regard isomorphic nets as equal.

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 $\rightarrow_{rpd}$.

Local Confluence of $\rightarrow_{stc}$

We will show now the local confluence for secure transition contractions in live marked graphs. Before that, a result for arbitrary transition contractions in arbitrary Petri nets similar to local confluence is given, namely Theorem 20, which is something like a weak diamond property.

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

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	$((p, p), (p, p))$

Definition 18. Let N be a Petri net and N' a Petri net obtained from N by arbitrary transition contractions. 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'. $\square$

As an example: Let N be a Petri net with $P = \{p_1, p_2, \ldots, p_n\}$, then $\mathfrak{M}_N^{N'}(((((p_1, \star), (p_2, \star)), \star), (((p_1, \star), (p_3, p_4)), \star))) = \{2 \cdot p_1, p_2, p_3, p_4\}$.

Lemma 19. *Let N be a Petri net, $N' = \overline{N}^{t_1,t_2}$ and $p'_1, p'_2 \in P'$. If $\overline{N}^{t_2,t_1}$ is defined as well, $\mathfrak{M}_N^{N'}(p'_1) = \mathfrak{M}_N^{N'}(p'_2)$ implies $p'_1 = p'_2$.*

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 combinations of places from the original net.

As indicated in Table 2 many of the combinations are actually not posssible for simple reasons. For example, if (p_1, p_1) is part of the place then $p_1 \in {}^\bullet t_1$ and $p_1 \in t_1{}^\bullet$, a contradiction since a contraction of a transition with a loop is not defined. As another example, case 23 drops out, because p_1 belongs to the preset of t_1 due the occurrence of (p_1, p_2), and on the other hand p_1 is element of its postset, 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.

The remaining impossible cases 25, 27, and 30 are considered in more detail.

Case 25 leads either to a loop after contracting t_1 or to an arc with weight 2 after contracting t_2, see Figure 1. 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 2.

In summary, it sufficies to consider the cases 1, 3, 5, 10 and 15 (also shown in Table 3, middle column, the last column is used later). We 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 $p'_1 = p'_2$.

Table 2. All combinatory possible places (up to isomorphism) after contraction of t_1 and then t_2. This table is obtained from Table 1 by instantiating p. The places p_i are pairwise different. The places which have an 'type error'-entry are not possible, since a place is treated as being and at the same time as not being adjacent to a contracted transition; 'initial loop' means that there is a loop at one of the transitions initially. Rows with a leading ▲ are considered in greater detail in the text

No.	Group	# Places	Example	Possible	If not, why?
▲ 1	1	1	$((p_1,\star),\star)$	●	
2	2	1	$((p_1,p_1),\star)$	-	initial loop $p_1 - t_1$
▲ 3	2	2	$((p_1,p_2),\star)$	●	
4	3	1	$((p_1,\star),(p_1,\star))$	-	initial loop $p_1 - t_2$
▲ 5	3	2	$((p_1,\star),(p_2,\star))$	●	
6	4	1	$((p_1,\star),(p_1,p_1))$	-	type error
7	4	2	$((p_1,\star),(p_1,p_2))$	-	type error
8	4	2	$((p_1,\star),(p_2,p_1))$	-	type error
9	4	2	$((p_2,\star),(p_1,p_1))$	-	initial loop $p_1 - t_1$
▲ 10	4	3	$((p_1,\star),(p_2,p_3))$	●	
11	5	1	$((p_1,p_1),(p_1,\star))$	-	type error
12	5	2	$((p_1,p_1),(p_2,\star))$	-	initial loop $p_1 - t_1$
13	5	2	$((p_1,p_2),(p_1,\star))$	-	type error
14	5	2	$((p_2,p_1),(p_1,\star))$	-	type error
▲ 15	5	3	$((p_1,p_2),(p_3,\star))$	●	
16	6	1	$((p_1,p_1),(p_1,p_1))$	-	initial loop $p_1 - t_1$
17	6	2	$((p_1,p_1),(p_1,p_2))$	-	initial loop $p_1 - t_1$
18	6	2	$((p_1,p_1),(p_2,p_1))$	-	initial loop $p_1 - t_1$
19	6	2	$((p_1,p_2),(p_1,p_1))$	-	initial loop $p_1 - t_1$
20	6	2	$((p_2,p_1),(p_1,p_1))$	-	initial loop $p_1 - t_1$
21	6	2	$((p_1,p_1),(p_2,p_2))$	-	initial loop $p_1 - t_1$ and $p_2 - t_1$
22	6	2	$((p_1,p_2),(p_1,p_2))$	-	loop after contracting t_1
23	6	2	$((p_2,p_1),(p_1,p_2))$	-	initial loop $p_1 - t_1$
24	6	3	$((p_1,p_2),(p_3,p_1))$	-	initial loop $p_1 - t_1$
▲ 25	6	3	$((p_1,p_2),(p_1,p_3))$	-	loop after contracting t_1 or weight 2 after contracting t_2 first
26	6	3	$((p_1,p_1),(p_2,p_3))$	-	initial loop $p_1 - t_1$
▲ 27	6	3	$((p_2,p_1),(p_3,p_1))$	-	loop after contracting t_1 or weight 2 after contracting t_2 first
28	6	3	$((p_2,p_1),(p_1,p_3))$	-	initial loop $p_1 - t_1$
29	6	3	$((p_2,p_3),(p_1,p_1))$	-	initial loop $p_1 - t_1$
▲ 30	6	4	$((p_1,p_2),(p_3,p_4))$	-	loop or weight 2 after contracting t_2 first

2. $\mathfrak{M}_N^{N'}(p'_1) = \{p_1,p_2\} = \mathfrak{M}_N^{N'}(p'_2)$. Hence, $p'_1, p'_2 \in \{((p_1,p_2),\star),\ ((p_2,p_1),\star),\ ((p_1,\star),\ (p_2,\star)), ((p_2,\star),(p_1,\star))\}$. If a fixed p'_1 from this set occurs in the net $\overline{N}^{t_1,t_2}$ it is not possible that a different element from this set occurs, too;

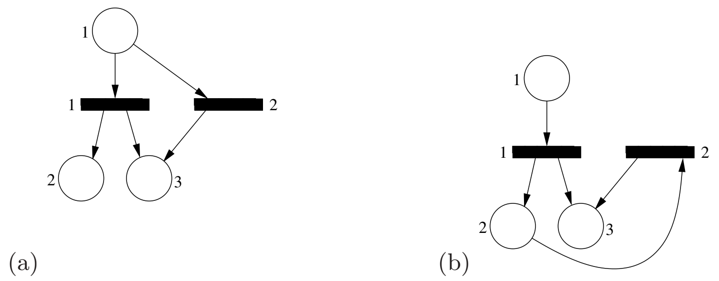

Fig. 1. Case 25 - $p' = ((p_1, p_2), (p_1, p_3))$. p_1 has to be an element of ${}^\bullet t_1$ p_2 and p_2, p_3 have to be elements of $t_1{}^\bullet$. Then there are 4 cases: 1) $p_1 \in {}^\bullet t_2, p_3 \in t_2{}^\bullet$: loop after contracting t_1, see (a) 2) $p_1 \in {}^\bullet t_2, p_1 \in t_2{}^\bullet$: initial loop $p_1 - t_1$ 3) $p_2 \in {}^\bullet t_2, p_1 \in t_2{}^\bullet$: loop after contracting t_1 4) $p_2 \in {}^\bullet t_2, p_3 \in t_2{}^\bullet$: weight 2 after contracting t_2, see (b)

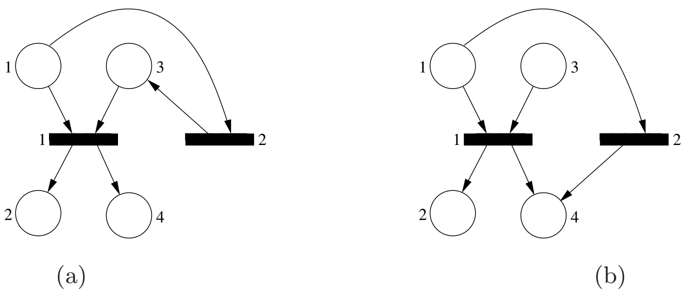

Fig. 2. Case 30 - $((p_1, p_2), (p_3, p_4))$. p_1 and p_3 have to be in the preset of the first transition to be contracted (t_1), p_2 and p_4 in the postset. For the connection to t_2 there are several possibilities; all of them satisfy that p_1 or p_2 (or both) are in the preset and p_3 or p_4 (or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

for example: 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 ${}^\bullet t_1$ but the existence of p_1'' implies p_1 is an element of $t_1{}^\bullet$; a contradiction, since the contraction was possible. With similar argumentations one can exclude the other combinations. Hence, $\mathfrak{M}_N^{N'}(p_1') = \mathfrak{M}_N^{N'}(p_2')$ 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 possible structures for p_1', p_2' resp. which all exclude each other as places of P', see the following table.

1	$((p_1, p_2), (p_3, \star))$	7	$((p_1, \star), (p_2, p_3))$
2	$((p_1, p_3), (p_2, \star))$	8	$((p_1, \star), (p_3, p_2))$
3	$((p_2, p_1), (p_3, \star))$	9	$((p_2, \star), (p_1, p_3))$
4	$((p_2, p_3), (p_1, \star))$	10	$((p_2, \star), (p_3, p_1))$
5	$((p_3, p_1), (p_2, \star))$	11	$((p_3, \star), (p_1, p_2))$
6	$((p_3, p_2), (p_1, \star))$	12	$((p_3, \star), (p_2, p_1))$

Without loss of generality, assume $p'_1 = ((p_1, p_2), (p_3, \star))$ (case 1) or $p'_1 = ((p_3, \star), (p_1, p_2))$ (case 11). $(p_3, \star)$ implies that p_3 is not adjacent to t_1, and therefore the existence of such a place excludes the existence of places 2,4-10. The remaining cases 3 and 12 can be excluded, since (p_2, p_1) implies $p_1 \in t_1{}^\bullet$ whereas p'_1 implies $p_1 \in {}^\bullet t_1$; in this case p_1 would be a loop place which is a contradiction. Case 1 cannot coexist with case 11, since the latter implies $(p_1, p_2) \in t_2{}^\bullet$ whereas the former case implies $(p_1, p_2) \in {}^\bullet t_2$ after contracting t_1, also a contradiction. □

Table 3. Possible places after two transition contractions. In the middle column one can find the places from Table 2 which turned out to be possible according to Definition 9. In each case there exists a place in $\overline{N}^{t_2,t_1}$ which uses the same places from N as the one in the middle column. This place is shown in the last column; for line 4 and 5 there are two possibilities, but only one of them exists

No.	$\overline{N}^{t_1,t_2}$	$\overline{N}^{t_2,t_1}$
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	$((p_1, p_2), (p_3, \star))$	$((p_1, \star), (p_2, p_3))$ / $((p_1, p_3), (p_2, \star))$

Theorem 20. *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 (even if the contractions are not secure).*

Proof. For this proof Table 3 is used again; the last column shows the place of $N_2 = \overline{N}^{t_2,t_1}$, which uses the same places from N as the place from $N_1 = \overline{N}^{t_1,t_2}$ in the middle column. If there are two possibilities, only one of them exists. For lines 1-3, it is quite clear that these places exist in N_2, for line 4 see Figure 4: since the place $((p_1, \star), (p_2, p_3))$ exists in $\overline{N}^{t_1,t_2}$, N must contain the net fragment (a); observe that exactly one of the dotted arcs exists but not both (in this case contracting t_1 would generate an arc with weight 2). Depending on which arc exist in $\overline{N}^{t_2,t_1}$, exactly one of the places in the last column exists. Line 5 is analogous.

We define a relation $f \subseteq P_1 \times P_2 \cup T_1 \times T_2$ 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 partial function: $(p'_1, p'_2), (p'_1, p''_2) \in f \Rightarrow \mathfrak{M}_N^{N_2}(p'_2) = \mathfrak{M}_N^{N_2}(p''_2)$. Lemma 19 implies $p'_2 = p''_2$.

b) f is total (surjective): After two contractions each place $p'_1 \in P_1$ has a structure shown in Table 3, middle column, and $\mathfrak{M}_N^{N_1}(p'_1) = \mathfrak{M}_N^{N_2}(p'_2)$ holds for the corresponding place p'_2 in the last column. Analogous for surjective.

c) f is injective: $f(p'_1) = f(p''_1) \Rightarrow \mathfrak{M}_N^{N_1}(p'_1) = \mathfrak{M}_N^{N_1}(p''_1)$. From Lemma 19 follows $p'_1 = p''_1$.

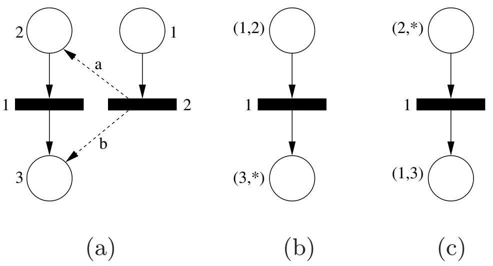

Fig. 3. For line 4 from Table 3. Since the place $((p_1,\star),(p_2,p_3))$ exists in $\overline{N}^{t_1,t_2}$, N must contain the net fragment (a); observe that exactly one of the dotted arcs exists but not both (in this case contracting t_1 would generate an arc with weight 2). If arc a, b resp. exists, contracting t_2 first results in (b), (c) respectively; the next contraction results in $((p_1,p_2),(p_3,\star))$, $((p_2,\star),(p_1,p_3))$ resp. as it is written in the last column

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 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 $W_1(p'_1,t_1) = \sum_{p\in\mathfrak{M}_N^{N_1}(p'_1)} W(p,t_1) = \sum_{p\in\mathfrak{M}_N^{N_2}(f(p'_1))} W(p,t_1) = W_2(f(p'_1),f(t_1))$. Observe that for every place p'_1 of N_1 shown in Table 3, $\mathfrak{M}_N^{N_1}(p'_1)$ is a *set*. Analogous for the second case. □

The proof for the following lemma uses Theorem 20; 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 21. *For $(N,\Lambda) \in MGR$, let $(N,\Lambda) \rightarrow_{stc} (N_1,\Lambda_1)$ and $(N,\Lambda) \rightarrow_{stc} (N_2,\Lambda_2)$. Then, there exists $(N',\Lambda') \in MGR$ with $(N_1,\Lambda_1) \rightarrow^*_{red} (N',\Lambda')$ and $(N_2,\Lambda_2) \rightarrow^*_{red} (N',\Lambda')$.*

Proof. Let the contractions concern transition t_1 and t_2. If both $\overline{N}^{t_1,t_2}$ and $\overline{N}^{t_2,t_1}$ are defined, Theorem 20 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. 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 4(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 4(b) and (c) depict the results of

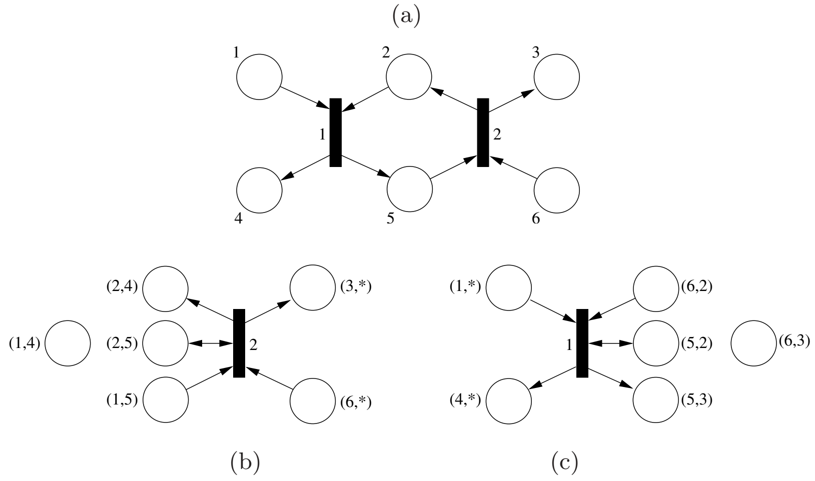

Fig. 4. (a) Scheme of a net fragment where contraction generates a loop (b) After t_1-contraction (c) After t_2-contraction

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 20, 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 $((1,5),(2,4))$ 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. $\square$

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 might seem rather obvious, in fact some effort is already needed to prove it at least for marked graphs.

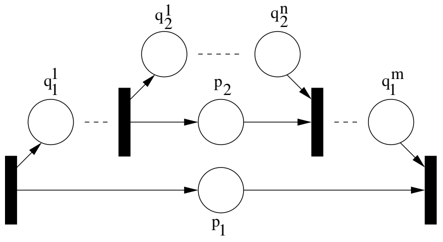

Fig. 5. Two redundant places p_1, p_2 with $p_1 \notin Q_2, p_2 \in Q_1$

Let p_1, p_2 be redundant places of $N \in MGR$ with $p_1 \neq p_2$. If one of them, 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. This 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.

Due to Theorem 13 we can now 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 is treated as above. For the second case take a look at Figure 5. 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).

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 $p_1{}^\bullet \dots p_2{}^\bullet$ not using p_1, and analogously there are Q_1-paths ${}^\bullet p_1 \dots {}^\bullet p_2$ and $p_2{}^\bullet \dots p_1{}^\bullet$ 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 12, 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 22. *Let* $(N, \Lambda) \rightarrow_{rpd} (N_1, \Lambda_1)$ *and* $(N, \Lambda) \rightarrow_{rpd} (N_2, \Lambda_2)$ *for some* $(N, \Lambda) \in MGR$*. 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 23. *Let* $(N, \Lambda) \rightarrow_{rpd} (N_1, \Lambda_1)$ *and* $(N, \Lambda) \rightarrow_{stc} (N_2, \Lambda_2)$ *for some* $(N, \Lambda) \in MGR$*. Then, there exists an* $(N', \Lambda') \in MGR$ *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 live 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 a shortcut 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; hence, the two operations are independent. 3) t is adjacent to the path *and* p – leading to two sub-cases, one of them shown in Figure 6(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

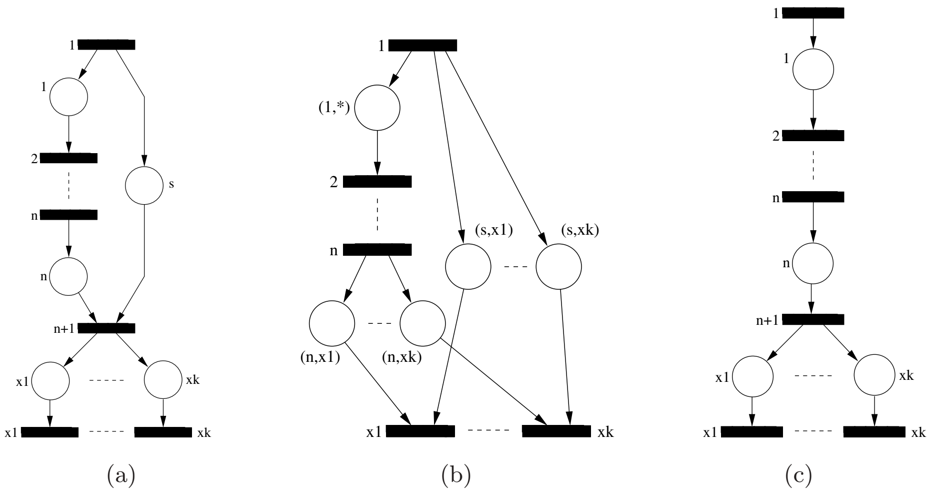

Fig. 6. Confluence of shortcut place deletion and transition contraction. (a) $p \equiv p_s$ is a shortcut place of $\{p_1, \ldots, p_n\}$ and $t \equiv t_{n+1}$ is the transition to be contracted. The net in (b) is obtained by contracting t_{n+1}, (c) by deleting p_s

place of $\{(p_1, *), \ldots, (p_{n-1}, *), (p_n, p_{xi})\}$ 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 24. *The reduction rule $\rightarrow_{red}$ is confluent and terminating for live marked graphs.*

Corollary 25. *The STG-decomposition algorithm of [VW02] is determinate for live marked graphs.*

In [CCJS94] a decomposition of strongly connected live marked graphs into two components is considered. In this approach the nets are unlabelled, while our STG decomposition is directed by the labelling with signal transitions; therefore the decomposition of [CCJS94] is not applicable in our setting.

What is interesting is that in the decomposition of [CCJS94] a whole subnet is removed and this could be used in our setting to remove several internal transitions together. A result of [CCJS94] implies that this removal preserves the language, but this does not immediately imply that subnet removal can be used to determine correct STG decompositions in the sense of [VW02]. In fact, the correctness criterion of [VW02] is of bisimulation type, but does not imply language equivalence. Furthermore, redundant place deletion and secure transition contractions always lead to a correct decomposition while subnet removal presupposes liveness and strong connectedness. Liveness is a precondition for determinacy of STG decomposition but not for its correctness.

Nevertheless subnet removal might be closely related to redundant place deletion and secure transition contractions. If one could show some sort of coincidence this might lead to an alternative proof of our determinacy result. Such a result would not imply that subnet removal is more efficient; the latter involves solving an all-pairs shortest paths problem, which takes time of $O(n^3)$ where n is the number of removed internal transitions *plus* the number of "neighbouring" non-internal transitions.

5 Conclusion

We have shown that the STG decomposition algorithm presented in [VW02] is determinate if applied to live 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 some other 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 rely several times on the liveness characterisation of marked graphs via the markings of cycles.

Related to the determinacy result, but also of independent interest, is the conceptionally and algorithmically easy characterisation of redundant places in live marked graphs, for which we provided a new easy proof. Again, we would like to generalise this result; it is 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, and we currently consider a generalisation to free-choice nets, which is not obvious at all.

Acknowledgement: We thank the anonymous referees for their comments which helped to improve the paper.

References

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

[Bes87] E. Best. Structure theory of Petri nets: The free choice hiatus. In W. Brauer et al., editors, *Petri Nets: Central Models and Their Properties*, Lect. Notes Comp. Sci. 254, 168–205. Springer, 1987.

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

[CCJS94] J. Campos, J. M. Colom, H. Jungnitz, and M. Silva. Approximate throughput computation of stochastic marked graphs. In *IEEE Transactions on Software Engineering 20*, pages 526–535, 1994.

[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 Trans. Information and Systems*, E80-D, 3:315–325, 1997.

[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.