Doctopic: Logic, semantics and theory of programming ARTICLE IN PRESS TCS:11342

Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Sufficient conditions for the marked graph realisability of labelled transition systems

Eike Best [1], Thomas Hujsa [1,*], Harro Wimmel [1]

*Department of Computing Science, Carl von Ossietzky Universität Oldenburg, D-26111 Oldenburg, Germany*

ARTICLE INFO

*Article history:*
Received 15 April 2017
Received in revised form 8 September 2017
Accepted 2 October 2017
Available online xxxx

*Keywords:*
Synthesis
Labelled transition system
Petri net
Realisability
Marked graph

ABSTRACT

This paper describes two results within the context of Petri net synthesis from labelled transition systems. We consider a set of structural properties of transition systems, and we show that, given such properties, it is possible to re-engineer a Petri net realisation into one which lies inside the set of marked graphs, a well-understood and useful class of Petri nets.

The first result originates from Petri net based workflow specifications, where it is desirable that $k$ customers can share a system without mutual interferences. In a Petri net representation of a workflow, the presence of $k$ customers can be modelled by an initial $k$-marking, in which the number of tokens on each place is a multiple of $k$. For any initial $k$-marking with $k \geq 2$, we show that other desirable assumptions such as reversibility and persistence suffice to guarantee marked graph realisability. For the case that $k = 1$, we show that the existence of certain cycles, along with other properties such as reversibility and persistence, again suffices to guarantee marked graph realisability.



## 1. Introduction

In order to be useful, a system is normally required to be well-behaved. For example, in a business workflow [1], a customer's activity should not impede other customers' concurrent (or future) activities [2,3]. Similarly, in a security operating system [4], it should be possible for several users to share a system without being aware of each other. Often, such systems are also required to be reversible, meaning that their initial states always remain reachable. If several such well-behavedness properties are postulated simultaneously, it may happen that they entail strong consequences. The present paper studies two such implications in the context of systems modelled by persistent Petri nets [5–7]. Persistence disallows true conflicts and is sometimes, but not always, required of workflow models [3].

In a Petri net representation of a workflow, the presence of $k$ customers can be modelled by initial markings in which the number of tokens on each place is a multiple of $k$. Such markings are called $k$-markings and are written as $k \cdot M_0$. For instance, Fig. 1 depicts a Petri net $\Sigma_1$ with an initial 4-marking. Intuitively, this might model four individual customers who are using a workflow simultaneously.

* Corresponding author.
*E-mail addresses:* eike.best@informatik.uni-oldenburg.de (E. Best), hujsa.thomas@gmail.com (T. Hujsa), harro.wimmel@informatik.uni-oldenburg.de (H. Wimmel).

[1] Funding: This work was supported by DFG (German Research Foundation) through grant Be 1267/16-1 ASYST (Algorithms for Synthesis and Pre-Synthesis Based on Petri Net Structure Theory).

https://doi.org/10.1016/j.tcs.2017.10.006
0304-3975/

Please cite this article in press as: E. Best et al., Sufficient conditions for the marked graph realisability of labelled transition systems, Theoret. Comput. Sci. (2017), https://doi.org/10.1016/j.tcs.2017.10.006

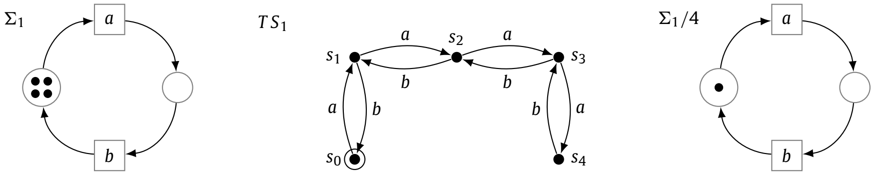


**Fig. 1.** A 4-marked Petri net $\Sigma_1$ (left-hand side) and its reachability graph (i.e., state space), represented by a labelled transition system (middle) with initial state $s_0$ (encircled). The system $\Sigma_1/4$ (defined structurally as $\Sigma_1$, but with a quarter of the initial marking) is shown on the right-hand side.

Such a system should be separable, that is, it should behave in the same way as if its initial state is divided by $k$ and the resulting system is executed $k$ times concurrently. For instance, in Fig. 1, $\Sigma_1$ has the same state space as four disjoint parallel instances of $\Sigma_1/4$. It has been proved in [8] that plain, bounded, reversible, and persistent Petri nets[2] are already guaranteed to be separable. It is also known [3,9] that marked graphs are separable. In the present paper, we shall augment this work by the following two results:

- Let a system be described by a persistent Petri net which is plain, bounded, reversible, and has an initial $k$-marking with $k \geq 2$. Then there exists a marked graph Petri net [10] with an isomorphic state space.
- Let a system be described by a persistent Petri net which is plain, safe,[3] reversible, and has, in its reachability graph, a cycle containing each transition once. Then there exists a marked graph Petri net with an isomorphic state space.

These results enrich the domain of Petri net synthesis from a labelled transition system, the latter being given as a specification to be implemented by a Petri net. Indeed, one can first check the existence of a Petri net realisation and build one when possible, and then exploit these new conditions to determine the existence of a marked graph solution satisfying the same specification. Moreover, both methods are constructive, the $k$-marked case (first item above) needing a prior result, and the safe case (second point above) being described in the sequel, so that they provide algorithms that re-engineer a Petri net solution into a marked graph satisfying the same specification.

The main part of the paper is organised as follows. Section 2 presents the technical background (labelled transition systems and Petri nets). In Section 3, we introduce some key behavioural notions necessary to understand the rest of the paper. Section 4 contains the proof of our first main theorem, and in Section 5, we proceed to proving the second main result, as sketched above. Section 6 concludes and presents some ideas for further research.

## 2. Transition systems and Petri nets

### 2.1. Labelled transitions systems

A labelled transition system (lts, for short) with initial state is a tuple $TS = (S, T, \to, s_0)$ with nodes (states) $S$, edge labels $T$, edges $\to \subseteq (S \times T \times S)$, and an initial state $s_0 \in S$. It is called finite when $S$ and $T$ (and hence also $\to$) are finite sets. A label $t$ is enabled at $s \in S$, written as $s[t\rangle$, if $\exists s' \in S\colon (s,t,s') \in \to$. We also write $s[t\rangle s'$ if $(s,t,s') \in \to$. A walk of length $\ell \in \mathbb{N}$ is a sequence

$$\eta \;=\; r_0[t_1\rangle r_1 \dots r_{\ell-1}[t_\ell\rangle r_\ell$$

where $r_0, \dots, r_\ell \in S$, and for $1 \leq j \leq \ell$, $r_{j-1}[t_j\rangle r_j$. The walk $\eta$ is elementary if it does not contain the same node twice, except perhaps $r_0 = r_\ell$, in which case the walk forms a cycle. We write $r_0[\sigma\rangle r_\ell$ (or $r_0 \xrightarrow{\sigma} r_\ell$), where $\sigma = t_1 \dots t_\ell \in T^*$, and say that $\sigma$ is enabled (or firable, or feasible) at $r_0$, and that $r_\ell$ is reachable from $r_0$ by $\sigma$ (or by $\eta$, in order to emphasise the intermediate states). The set of states reachable from $r_0$ is denoted by $[r_0\rangle$.

A function $\Phi$ is called a $T$-vector if $\Phi\colon T \to \mathbb{N}$, and a binary $T$-vector if $\Phi\colon T \to \{0,1\}$. The support of a $T$-vector $\Phi$ is $supp(\Phi) = \{t \in T \mid \Phi(t) > 0\}$. We denote by $\mathbf{1}^{|T|}$ (or by $\mathbf{1}$ when no confusion is possible) the binary T-vector whose support is $T$. Two $T$-vectors $\Phi_1, \Phi_2$ are label-disjoint if $\forall t \in T\colon \Phi_1(t) = 0 \vee \Phi_2(t) = 0$. For a sequence $\sigma \in T^*$, the Parikh vector $\Psi(\sigma)$ of $\sigma$ is a $T$-vector defined by $\Psi(\sigma)(t) =$ the number of occurrences of $t$ in $\sigma$.

An lts $TS = (S, T, \to, s_0)$ is called totally reachable if $[s_0\rangle = S$ (i.e., every state is reachable from $s_0$); (forward) deterministic if for any states $s, s', s'' \in [s_0\rangle$ and label $t \in T$, $(s[t\rangle s' \wedge s[t\rangle s'') \Rightarrow s' = s''$ (i.e., the state reached from $s$ after firing $t$ is unique); backward deterministic if for any states $s, s', s'' \in [s_0\rangle$ and label $t \in T$, $(s'[t\rangle s \wedge s''[t\rangle s) \Rightarrow s' = s''$; live if $\forall t \in T$ $\forall s \in [s_0\rangle$ $\exists s' \in [s\rangle\colon s'[t\rangle$ (i.e., transitions remain eventually firable); reversible if $\forall s \in [s_0\rangle\colon s_0 \in [s\rangle$ (i.e., $s_0$ always remains reachable); (forward) persistent if for all reachable states $s$, $s'$, $s''$, and labels $t, t'$, if $s[t\rangle s'$ and $s[t'\rangle s''$ with $t \neq t'$, there is

[2] Plainness means that there are no arc weights $> 1$ in the Petri net. Boundedness means that the state space is finite. Reversibility means that the initial state can be reached from every reachable state.

[3] Safeness means that the places of the net are binary: they may contain either 0 or 1 token.

Please cite this article in press as: E. Best et al., Sufficient conditions for the marked graph realisability of labelled transition systems, Theoret. Comput. Sci. (2017), https://doi.org/10.1016/j.tcs.2017.10.006