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

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Applying regions

Jetty Kleijn [a], Maciej Koutny [b,*], Marta Pietkiewicz-Koutny [b], Grzegorz Rozenberg [a]

[a] LIACS, Leiden University, P.O. Box 9512, NL-2300 RA Leiden, The Netherlands
[b] School of Computing Science, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom

ARTICLE INFO

Article history:
Received 17 October 2015
Accepted 28 January 2016
Available online xxxx

Keywords:
Concurrency
Theory of regions
Transition system
Synthesis problem
Petri net
Step semantics
A/sync connection
Whole-place operations net

ABSTRACT

In this paper we present a brief overview of a representative fragment of the theory of regions. Regions are a powerful tool for the synthesis of concurrent systems from a behavioural specification. To demonstrate the robustness of region based synthesis we survey some of the existing results for extensions of place/transition nets. We relate in particular to the general approach founded on τ-nets and τ-regions. A new extension of region theory to the case of Petri nets with whole-place operations is presented.

1. Introduction

Synthesising systems from behavioural specifications is a powerful way of constructing implementations which are correct-by-design and thus requiring no costly validation efforts. In this paper, we focus on the problem of synthesising a Petri net N from a specification provided by a step transition system $\mathcal{T}$. The latter specifies the desired state space of the net N i.e., the concurrent reachability graph[1] of N should be isomorphic to $\mathcal{T}$. The approach we follow is based on the notion of a *region* of a transition system and we will show the robustness of the concept.

Regions were introduced in the seminal paper [18] for the class of Elementary Net systems (EN-systems) with sequential execution semantics, arguably the simplest Petri net model. The aim was to characterise the sequential transition systems (essentially, finite state automata) that are isomorphic to the reachability graph of an EN-system. Regions of a transition system $\mathcal{T}$ were defined as subsets R of its nodes enjoying the so-called *crossing property*. This property simply states that, for every label t, the arcs $q \xrightarrow{t} r$ of $\mathcal{T}$ either all enter R (i.e., $q \notin R$ and $r \in R$), or all leave R (i.e., $q \in R$ and $r \notin R$), or none of them crosses the boundary of R (i.e., $q \in R$ iff $r \in R$). A region can thus be viewed as an abstract representation of a possible place of the net N to be synthesised. Based on this idea, $\mathcal{T}$ is realisable by an EN-system if and only if there are enough regions of $\mathcal{T}$ to satisfy two fundamental regional axioms. The first axiom (*separation*) requires that for every pair of distinct states of $\mathcal{T}$, there is a *witness* region R such that one state belongs to R and the other does not. The second axiom (*forward closure*) requires that if t does not label any transition outgoing from a state q, there must be a witness region R such that t

* Corresponding author.
E-mail address: maciej.koutny@ncl.ac.uk (M. Koutny).

[1] Synthesis from a sequential reachability graph is a special case of the concurrent case.

http://dx.doi.org/10.1016/j.tcs.2016.01.040
0304-3975/

leaves R while q does not belong to R. It was shown that if both regional axioms hold for $\mathcal{T}$, an EN-system realising $\mathcal{T}$ can be constructed by taking the witness regions and turning them into places [16] with appropriate connections to transitions derived from the crossing relationships.

Over the years, this original idea has been developed further and was extended in several different directions, including other Petri net classes (e.g., PT-nets [28], pure and bounded PT-nets [6], Flip-flop nets [32], nets with inhibitor arcs [8, 31], and nets with localities [26]); synthesis modules of tools (e.g., Petrify [12], ProM [34], VipTool [5], Genet [9], and Rbminer [33]); application areas (e.g., asynchronous VLSI circuits [12,9,33] and workflows [34]); other semantical execution models (e.g., step sequences [21,31], (local) maximal concurrency [26], and firing policies [15,25]); and specification formalisms other than transition systems (e.g., languages [13] and scenarios [5]).

In this paper we present to the TCS community a brief overview of a representative fragment of the theory of regions. In the concluding section we present a number of challenging problems in the area. More details concerning the importance and long term impact of the region concept can be found in [4] and the recent monograph [3] that provides a comprehensive overview of region-based net synthesis.

Intuitively, a region captures a single net place through essential behavioural characteristics as encoded in a transition system, including its marking information and connectivity with all the transitions. One of the key advances in the design of region based solutions for a variety of synthesis problems has been the development of a general approach [4] for dealing with region based synthesis of nets. It is founded on so-called τ-nets and corresponding τ-regions. The parameter τ is a convenient way of capturing the marking information and different connections between places and transitions of varying classes of Petri nets, removing the need to re-state and re-prove the main results every time a new kind of transitions or arcs is introduced. This approach can be applied once a class of Petri nets has been shown to be a class of τ-nets, i.e., to correspond to a class of τ-nets for some suitable τ. It should be kept in mind however, that although the theory provides necessary and sufficient conditions for the existence of a τ-net whose reachability graph is isomorphic to a given transition system, it does not provide ready answers for decidability and algorithmic concerns.

In this paper we demonstrate the robustness of regions as already known for several net classes. In addition, we introduce an extension of the original concept by defining a new type of regions for nets with whole-place operations (i.e., with arc weights defined in relation to all places and depending on the current marking). The nets are derived from *transfer/reset* nets [17] and *affine* nets [19], and executed under the step semantics rather than sequentially. This is yet another confirmation of the relevance and flexibility of the notion of region for the derivation of correct concurrent systems.

2. Nets with step sequence semantics

We start by presenting some basic notions concerning Petri nets; in particular, a general notion of nets defined over a transition system that captures relationships between places and transitions.

Throughout the paper, $\mathbb{Z}$ and $\mathbb{N}$ denote respectively the sets of all integers and non-negative integers. The absolute value of an integer n is denoted by $abs(n)$, e.g., $abs(2) = abs(-2) = 2$. The minimum of two integers, k and n, is denoted by $min\{k, n\}$.

2.1. Abelian monoids and multisets

An *abelian monoid* is a set $\mathbb{S}$ with a commutative and associative binary operation $+$, and a neutral element $\mathbf{0}$. The result of composing n copies of $s \in \mathbb{S}$ is denoted by $n \cdot s$, and so $\mathbf{0} = 0 \cdot s$. Two examples of abelian monoids are: (i) $\mathbb{S}_{PT} = \mathbb{N} \times \mathbb{N}$ with the pointwise arithmetic addition operation and $\mathbf{0} = (0, 0)$; and (ii) the free abelian monoid $\langle T \rangle$ generated by a set T. $\mathbb{S}_{PT}$ will represent arcs between places and transitions in PT-nets, whereas $\langle T \rangle$ will represent *steps* of nets with transition set T.

The free abelian monoid $\langle T \rangle$ can be seen as the set of all the multisets over T, e.g., $aab = aba = baa = \{a, a, b\}$. We use $\alpha, \beta, \gamma, \ldots$ to range over the elements of $\langle T \rangle$. For $t \in T$ and $\alpha \in \langle T \rangle$, $\alpha(t)$ denotes the multiplicity of t in α, and so $\alpha = \sum_{t \in T} \alpha(t) \cdot t$. Then $t \in \alpha$ whenever $\alpha(t) > 0$, and $\alpha \leq \beta$ whenever $\alpha(t) \leq \beta(t)$ for all $t \in T$. The size of α is $|\alpha| = \sum_{t \in T} \alpha(t)$.

2.2. Transition systems

A *(deterministic) transition system* $\langle Q, \mathbb{S}, \delta \rangle$ over an abelian monoid $\mathbb{S}$ consists of a set of *states* Q and a partial *transition function*[2] $\delta : Q \times \mathbb{S} \to Q$ such that $\delta(q, \mathbf{0}) = q$ for all $q \in Q$. An *initialised* transition system $\langle Q, \mathbb{S}, \delta, q_0 \rangle$ is a transition system with an *initial* state $q_0 \in Q$ such that each state $q \in Q$ is *reachable*, i.e., there are $s_1, \ldots, s_n$ and $q_1, \ldots, q_n = q$ ($n \geq 0$) with $\delta(q_{i-1}, s_i) = q_i$, for $1 \leq i \leq n$. For every state q of a transition system *TS*, we denote by $enb_{TS}(q)$ the set of all s which are *enabled* at q, i.e., $\delta(q, s)$ is defined. *TS* is *finite* if it has finitely many states, and the set of enabled elements of $\mathbb{S}$ at any of its states is finite. In the diagrams, an initial state is represented by a small square and all the remaining nodes by circles. The trivial $\mathbf{0}$-labelled transitions are omitted.

[2] Transition functions are not related to Petri net transitions.

Please cite this article in press as: J. Kleijn et al., Applying regions, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.01.040