COMPUTING SCIENCE

Regions of Petri Nets with a/sync Connections

Jetty Kleijn, Maciej Koutny and Marta Pietkiewicz-Koutny

TECHNICAL REPORT SERIES

No. CS-TR-1297 November 2011

TECHNICAL REPORT SERIES

No. CS-TR-1297 November, 2011

Regions of Petri Nets with a/sync Connections

J. Kleijn, M. Koutny and M. Pietkiewicz-Koutny

Abstract

Automated synthesis from behavioural specifications, is an attractive way of constructing correct concurrent systems.

In this paper, we investigate the synthesis of Petri nets which use special connections between transitions and places. Along these a/sync connections tokens can be transferred instantaneously between two transitions executed in a single step. We show that for Place/Transition nets with a/sync connections the synthesis problem can be treated within the general approach based on regions of step transition systems. Moreover, we demonstrate that the problem is decidable for finite transition systems, and identify a subclass of nets for which a polynomial decision procedure and construction algorithm exists.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

KLEIJN, J., KOUTNY, M., PIETKIEWICZ-KOUTNY, M.

Regions of Petri Nets with a/sync Connections
[By] J. Kleijn, M. Koutny, M. Pietkiewicz-Koutny
Newcastle upon Tyne: Newcastle University: Computing Science, 2011.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1297)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1297

Abstract

Automated synthesis from behavioural specifications, is an attractive way of constructing correct concurrent systems.

In this paper, we investigate the synthesis of Petri nets which use special connections between transitions and places. Along these a/sync connections tokens can be transferred instantaneously between two transitions executed in a single step. We show that for Place/Transition nets with a/sync connections the synthesis problem can be treated within the general approach based on regions of step transition systems. Moreover, we demonstrate that the problem is decidable for finite transition systems, and identify a subclass of nets for which a polynomial decision procedure and construction algorithm exists.

About the authors

Jetty Kleijn is a visiting fellow within the School of Computing Science, Newcastle University.

Maciej Koutny obtained his MSc (1982) and PhD (1984) from the Warsaw University of Technology. In 1985 he joined the then Computing Laboratory of the University of Newcastle upon Tyne to work as a Research Associate. In 1986 he became a Lecturer in Computing Science at Newcastle, and in 1994 was promoted to an established Readership at Newcastle. In 2000 he became a Professor of Computing Science.

Marta Pietkiewicz-Koutny received her M.Sc. in Applied Mathematics from the Warsaw University of Technology in 1982. In 1984 she joined the Department of Operational Research, Institute of Econometrics in the Warsaw University of Economics where she worked as a junior lecturer until 1986. In 1987 she joined the Computing Laboratory of the University of Newcastle upon Tyne first as a research associate and then as a demonstrator (1988-1997). In the period 1997-2000 she was a Ph.D. student at the Department of Computing Science of the University of Newcastle upon Tyne, and in December 2000 she was awarded her Ph.D. degree. In the period 2000-2003 she was a researcher on the EU-funded DSoS (Dependable Systems of Systems) project, and in 2003 she was appointed a lecturer in the School of Computing Science. Dr. Pietkiewicz-Koutny's research interests concentrate on modelling and validation of concurrent systems. The main topic of her research is the synthesis of Petri nets from transition systems.

Suggested keywords

ONCURRENCY
TRANSITION SYSTEM
PETRI NET
THEORY OF REGIONS
SYNTHESIS PROBLEM
PLACE TRANSITION NET
ELEMENTARY NET SYSTEM
STEP SEQUENCE SEMANTICS
SYNCHRONOUS AND ASYNCHRONOUS COMMUNICATION
A/SYNC CONNECTION

Regions of Petri Nets with a/sync Connections

Jetty Kleijn[1], Maciej Koutny[2], and Marta Pietkiewicz-Koutny[2]

[1] LIACS, Leiden University
P.O.Box 9512, NL-2300 RA Leiden, The Netherlands
kleijn@liacs.nl

[2] School of Computing Science, Newcastle University
Newcastle upon Tyne, NE1 7RU, United Kingdom
{maciej.koutny,marta.koutny}@ncl.ac.uk

Abstract. Automated synthesis from behavioural specifications, is an attractive way of constructing correct concurrent systems. In this paper, we investigate the synthesis of Petri nets which use special connections between transitions and places. Along these a/sync connections tokens can be transferred instantaneously between two transitions executed in a single step. We show that for Place/Transition nets with a/sync connections the synthesis problem can be treated within the general approach based on regions of step transition systems. Moreover, we demonstrate that the problem is decidable for finite transition systems, and identify a subclass of nets for which a polynomial decision procedure and construction algorithm exists.
Keywords: concurrency, transition system, Petri net, theory of regions, synthesis problem, place transition net, elementary net system, step sequence semantics, synchronous and asynchronous communication, a/sync connection

1 Introduction

In standard Petri net classes, including Place/Transition-nets (PT-nets) and Elementary Net Systems (EN-systems), a fundamental assumption concerning the production and consumption of tokens (resources) is that in a single computational step a token can be produced or consumed, but not both. (Note that when a place belongs to a self-loop, as a result of the transition execution, an available token is consumed and a new token is produced in that place.) This assumption appears to be justified in case only one transition can be executed at a time (as a step), but is less compelling if we consider computational steps in which several transitions can be executed simultaneously. The step semantics of Petri nets is defined through multisets of transitions that may be executed simultaneously when initially enough resources are available for all executions of transitions in that step. Hence, while it is easy to express asynchronous communication between transitions (by messages left by one transition in the form of a token in a place to be picked up later by another transition), there is no structural way to express that tokens may be directly picked up in the same step in a

form of synchronous communication. This was recognised also in e.g., [5], where zero-safe nets are introduced in which — to force synchronisation — sequences of transitions are collapsed into so-called transactions. The causality observed in communication in the structured occurrence net model [20] was our motivation in [17, 18] for allowing tokens to be simultaneously produced and consumed. Essentially, it was proposed there to extend the standard PT-nets with special places (called a/sync places in [18]) that can be used for the instantaneous (or synchronous) transfer of tokens from an input transition to an output transition. These places moreover allow asynchronous communication, because tokens that are not consumed instantaneously remain available as ordinary tokens. In this paper we take this idea further, by introducing PT-nets with a/sync connections (or PTASC-nets). Rather than having special a/sync places, there can be a/sync arcs between places and transitions and between transitions and places. An input a/sync arc from a transition to a place and an output a/sync arc from that place to a second transition can effect a synchronous transfer of tokens between these transitions when executed in a single computational step.

In this paper, we investigate the problem of synthesising a PTASC-net N from a behavioural specification given in the form of a step transition system TS which specifies the desired state space of the net N i.e., the reachability graph of N should be isomorphic to TS. (In fact, the construction of N is a by-product of solving the corresponding decision problem.) Automated synthesis from behavioural specifications is an attractive and powerful way of constructing correct concurrent systems. Our solution of the synthesis problem considered here will be based on the notion of a region of a transition system. Intuitively, a region captures a single net place through essential behavioural characteristics as encoded in TS, including its marking information and connectivity with all the transitions. Regions were introduced by Andrzej Ehrenfeucht and Grzegorz Rozenberg in the seminal paper [15] for the class of EN-systems with sequential execution semantics. Over the following two decades, the original idea has been developed and extended in several different directions, including: other Petri net classes (e.g., PT-nets [13, 21], Flip-flop nets [24], nets with inhibitor arcs [6, 23], and nets with localities [19]); synthesis tools (e.g., Petrify [9], ProM [26], VipTool [4], Genet [7], and Rbminer [25]); application areas (e.g., asynchronous VLSI circuits [9, 7, 25] and workflows [26]); other semantical execution models (e.g., step sequences [16, 23], (local) maximal concurrency [19], and firing policies [12]); and specification formalisms other than transition systems (e.g., languages [10] and scenarios [4]). More details concerning the importance and long term impact of the region concept as introduced by Andrzej Ehrenfeucht and Grzegorz Rozenberg can be found in the monograph article [3], and the proceedings of the recently held workshop Applications of Region Theory [14].

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 [3] for dealing with region based synthesis. 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

several 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 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 feasibility of the synthesis problem, it does not provide ready answers for decidability and algorithmic concerns).

In this paper, we take advantage of this general region theory when investigating the synthesis problem for PTASC-nets. First we show that they are indeed a class of τ-nets. Then we demonstrate that for PTASC-nets the synthesis problem from finite transition systems is decidable, and we identify a subclass for which a polynomial decision procedure and construction algorithm exists. We regard this as another confirmation of the robustness of the notion of region and its importance for the derivation of correct concurrent systems.

2 Preliminaries

We denote by $\mathbb{Z}$ and $\mathbb{Q}$ the set of all integer and rational numbers, respectively; moreover, $\mathbb{Z}^+ = \{n \in \mathbb{Z} \mid n \geq 0\}$, $\mathbb{Z}^- = \{n \in \mathbb{Z} \mid n \leq 0\}$, $\mathbb{Q}^+ = \{x \in \mathbb{Q} \mid x \geq 0\}$ and $\mathbb{Q}^- = \{x \in \mathbb{Q} \mid x \leq 0\}$. 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 $min\{k, n\}$.

A multiset over a finite set X is a function $U : X \to \mathbb{N} = \mathbb{Z}^+$, and $\mathcal{M}(X)$ is the set of all multisets over X. Sets may be treated as multisets and multisets may be represented by listing their elements with repetitions, e.g., $U = \{y, y, z\}$ is a multiset such that $U(y) = 2$, $U(z) = 1$, and $U(x) = 0$ otherwise. The cardinality of a multiset U is defined as $|U| = \sum_{x \in X} U(x)$.

A (labelled) transition system is triple $TS = (Q, A, \Delta)$ where Q is a set of states, A is a set of labels, and $\Delta : Q \times A \to Q$ is a partial function. In diagrams, states are represented as graph nodes, and the function Δ by arcs annotated with labels. For every state q, the set $enbld_{TS}(q) = \{a \mid \Delta(q, a) \text{ is defined}\}$ consists of all elements from A that are enabled at q.

An initialised step transition system is a tuple $TS = (Q, A, \Delta, q_0)$ such that (Q, A, Δ) is a transition system, $q_0 \in Q$ is the initial state, and $A = \mathcal{M}(T)$ for a finite set T. We assume that each $t \in T$ occurs in the label of at least one arc in the graph of TS, and that each state $q \in Q$ is reachable from q_0, i.e., in the graph of TS there is a directed path from q_0 to q.

3 Place/Transition nets with a/sync connections

A PT-*net with a/sync connections* (or PTASC-net) is a tuple:

$$PTASC = (P, T, W, AS, M_0) \ ,$$

where: P and T are finite disjoint sets of *places* and *transitions*, respectively; $W : (T \times P) \cup (P \times T) \to \mathbb{Z}^+$ is the *(standard) arc weight* function; $AS : T \times P \to \mathbb{Z}$

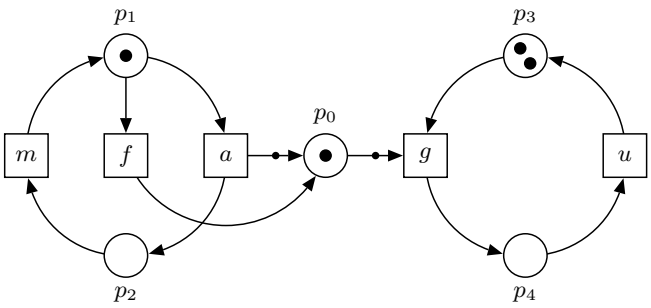


Fig. 1. A PTASC-net modelling a one-producer/two-consumers system with two a/sync connections.

is the *a/sync connection* function; and M_0 is a multiset over P called the *initial* marking (in general, any multiset of places is a *marking*). We refer to the places and transitions as *nodes* (of *PTASC*). We also assume that, for every transition t, there is at least one place p such that $W(p,t) > 0$.

In diagrams, as usual, places are represented by circles; transitions by rectangles; the arc weight function by directed arcs with the weight n annotated if $n \geq 2$ while arcs with weight 0 are omitted; and a marking by *tokens* (small black dots) drawn inside places. To represent an a/sync connection $AS(t,p) = n$ with $n > 0$ (or $n < 0$) we use a directed arc from t to p (resp. from p to t) with a dot in the middle; if $abs(n) > 1$ the arc is annotated by its weight $abs(n)$. Figure 1 depicts a PTASC-net modelling a producer (left cycle), an unbounded buffer (p_0) in the middle, and two consumers (right cycle). The producer can execute one of three transitions: m (making item(s)), a (adding a new item to the buffer), and f (failure in which a new item is added to the buffer, but the producer terminates). Each of the two consumers represented by the two tokens in place p_3 can cyclically execute: g (getting an item), and u (using the item). The feature distinguishing this net from a standard PT-net are two a/sync connections, from a to p_0, and from p_0 to g with $AS(a,p_0) = 1$ and $AS(g,p_0) = -1$. This means that tokens produced along the a/sync a to p_0 connection can be in the same step (i.e., instantaneously) consumed along the a/sync p_0 to g connection. Intuitively, the corresponding items are not stored in the buffer but rather handed over directly to the consumer (if it is ready to get items). In any other context, these two a/sync connections lose their special status and behave as if they were standard ones.

A multiset of transitions (a *step*) U is *enabled* at a marking M if, for every place p, $M(p) - \sum_{t \in T} U(t) \cdot W(p,t) + min\{0, \sum_{t \in T} U(t) \cdot AS(t,p)\} \geq 0$, i.e., if p contains enough tokens for all standard connections from p to transition occurrences in U and, in addition, also enough tokens to compensate for all a/sync connections from p to transition occurrences in U that will not be supplied synchronously to p by a/sync connections to p. The idea is that transitions that have p as an a/sync input place will have to consume tokens deposited in p

earlier on, if not all 'immediate' tokens can be delivered in the current step. In this situation the a/sync connection may (partially) lose its special meaning and be treated as a standard connection. On the other hand, a possible surplus of tokens deposited in p by the transitions connected to it by a/sync arcs will be kept for later use.

A step enabled at M can be *executed* leading to the new marking M' given, for every place p, by $M'(p) = M(p) + \sum_{t \in T} U(t) \cdot (W(t,p) - W(p,t) + AS(t,p))$. In Figure 1, for example, the step $\{a, g, g\}$ is enabled at the initial marking, since the second g can consume instantaneously the token produced by a. The step $\{f, g, g\}$ however cannot be executed at the initial marking, as the token produced along the connection from f to p_0 may only be consumed 'asynchronously', i.e., in some future step. The execution of an enabled step follows the usual rules of token manipulation and yields, for $\{a, g, g\}$ from the initial marking, the new marking $\{p_2, p_4, p_4\}$.

Clearly, a PT-net $PT = (P, T, W, M_0)$ can be treated simply as a PTASC-net $PTASC = (P, T, W, AS, M_0)$ with $AS(T \times P) = \{0\}$. Furthermore, every EN-system can be seen as a PTASC-net $PTASC = (P, T, W, AS, M_0)$ such that $AS(T \times P) = \{0\}$, $W((T \times P) \cup (P \times T)) \subseteq \{0, 1\}$, $W(p,t) = 1$ always implies $W(t,p) = 0$, all markings and steps are sets, and a step U is enabled at a marking M if, for every place p, $M(p) \geq \sum_{t \in U} W(p,t)$ and $M(p) + \sum_{t \in U} W(t,p) \leq 1$.

4 τ-nets: a framework for defining Petri net classes

Surprisingly many Petri net classes can be defined as instances of τ-nets [3]. In this general set-up, nets are defined in terms of connections between places and transitions. The ways in which these connections influence markings of places are captured by special transition systems, called *net-types*. Moreover, the effect of executing a step U on a place p is calculated using a *connection monoid* that determines the composite connection between p and U. Then, each concrete net-type τ together with the corresponding connection monoid define a class of nets with step sequence semantics, the τ-nets.

A *connection monoid* is a set $\mathbb{S}$ of *connections* with a commutative and associative binary composition operation $\oplus$, and a neutral element (identity) $\mathbf{0}$. For each $s \in \mathbb{S}$ we let $\bigoplus^0 s = \mathbf{0}$, and $\bigoplus^{n+1} s = (\bigoplus^n s) \bigoplus s$, for all $n \in \mathbb{N}$. The same symbol $\mathbb{S}$ will be used for a connection monoid and for its underlying set of connections. Then, a *net-type over* $\mathbb{S}$ is a transition system $\tau = (Q, \mathbb{S}, \Delta)$ where $\Delta : Q \times \mathbb{S} \to Q$ is a partial function such that $\Delta(q, \mathbf{0}) = q$, for all $q \in Q$.

Given a net-type $\tau = (Q, \mathbb{S}, \Delta)$, a *$\tau$-net* is a tuple $N = (P, T, F, M_0)$, where P and T are, respectively, disjoint sets of places and transitions (T is assumed to be finite), $F : (P \times T) \to \mathbb{S}$ is a *connection mapping*, and M_0 is the *initial marking* of N (in general, a marking of a τ-net is a mapping from the places of the net to the states of τ). For a place p of N and a step U of transitions of N, we define the composite connection between U and p by $F(p, U) = \bigoplus_{t \in T} (\bigoplus^{U(t)} (F(p,t)))$. Then U is *enabled* at a marking M if $F(p, U) \in enbld_\tau(M(p))$, for every place $p \in P$. The *execution* of U produces

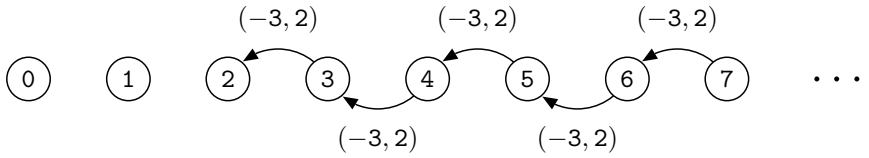
the marking M' such that $M'(p) = \Delta(M(p), F(p,U))$, for every place $p \in P$. The *concurrent reachability graph* $CRG(N)$ of N is a transition system formed by executing inductively from M_0 all possible enabled steps of N. The nodes of $CRG(N)$ are the *reachable* markings of N.

To see how a PT-net $PT = (P, T, W, M_0)$ can be regarded as a τ-net, we consider the connection monoid $\mathbb{S}_{PT} = (\mathbb{Z}^- \times \mathbb{Z}^+, \oplus, \mathbf{0})$ with $\mathbf{0} = (0,0)$ and $\oplus$ being point-wise arithmetic addition. Using this monoid, the connections between places and multisets of transitions in PT-nets can be expressed through the net-type $\tau_{PT} = (\mathbb{Z}^+, \mathbb{S}_{PT}, \Delta_{PT})$ over $\mathbb{S}_{PT}$, where:

$$\Delta_{PT} = \{(\ell, (m,n)) \mapsto \ell + m + n \mid \ell + m \geq 0\} .$$

Intuitively, this states that a place p containing ℓ tokens enables steps which take no more than ℓ tokens, and that the resulting number of tokens in p is $\ell + m + n$ where $abs(m)$ and n are the numbers of tokens taken and produced, respectively, by all occurrences of transitions in that step together. Then, to encode PT as a τ_{PT}, all we need to do is define $F(p,t) = (-W(p,t), W(t,p))$.

The following is a fragment of τ_{PT} with arcs labelled by the connection $(-3, 2)$.



This particular connection describes the situation that there is a combination of transitions (a step) and arcs pointing from a place to these transitions with accumulated weight 3, and arcs from these transitions to that place with 2 as their combined weight. For example, we can have a step of three transitions: two connected by self-loops and one removing a single token from the place. We then have an arc, e.g., from 5 to 4 because $5 - 3 = 2 \geq 0$ and $5 - 3 + 2 = 4$.

To see how an EN-system (P, T, W, M_0) can be regarded as a τ-net, we first observe that, in EN-systems, there are three basic connections between places and transitions: an arc from a transition to a place denoted here as `out`; an arc from a place to a transition denoted as `in`; moreover $\top$, the identity element of the connection monoid, indicates that a place and a transition are disconnected (independent). In addition, there is the composite and technically useful 'blocking' connection $\bot$ (used to indicate that a step is not enabled). Figure 2 defines the connection monoid $\mathbb{S}_{EN} = (\{\top, \mathtt{out}, \mathtt{in}, \bot\}, \oplus_{EN}, \top)$ together with the net-type τ_{EN}. Note that $\mathtt{out} \oplus_{EN} \mathtt{out} = \mathtt{out} \oplus_{EN} \mathtt{in} = \mathtt{in} \oplus_{EN} \mathtt{out} = \mathtt{in} \oplus_{EN} \mathtt{in} = \bot$ which essentially means that the neighbourhoods of transitions forming an enabled step must be disjoint. Then, to encode the EN-system as a τ_{EN}, all we need to do is define: $F(p,t) = \mathtt{out}$ if $W(p,t) = 0$ and $W(t,p) = 1$, $F(p,t) = \mathtt{in}$ if $W(p,t) = 1$ and $W(t,p) = 0$, and $F(p,t) = \top$ otherwise.

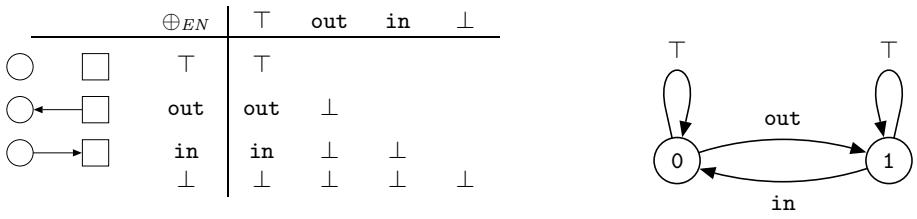


Fig. 2. The table of $\oplus_{EN}$ and net-type τ_{EN}.

5 The τ-net synthesis problem and τ-regions

Let $\mathbb{S}$ be a connection monoid and let $\tau = (Q, \mathbb{S}, \Delta)$ be a net-type over $\mathbb{S}$. The generic synthesis problem for τ-nets can be formulated as follows.

SYNTHESIS PROBLEM
Let $TS = (\widehat{Q}, \mathcal{M}(T), \delta, q_0)$ be an initialised step transition system. Provide necessary and sufficient conditions for TS to be *realised* by some τ-net N (i.e., $TS \cong CRG(N)$ where $\cong$ is transition system isomorphism preserving the initial states and arc labels).

To solve the SYNTHESIS PROBLEM, the notion of τ-region can be employed. A *τ-region* of TS is a pair of mappings $(\sigma : \widehat{Q} \to Q \ , \ \eta : T \to \mathbb{S})$ such that:

$$\eta(U) \in \mathit{enbld}_\tau(\sigma(q)) \ \text{ and } \ \Delta(\sigma(q), \eta(U)) = \sigma(\delta(q, U)) \ ,$$

for all $q \in \widehat{Q}$ and $U \in \mathit{enbld}_{TS}(q)$, where $\eta(U) = \bigoplus_{t \in T}(\bigoplus^{U(t)} \ \eta(t))$.

Remark 1. The original concept of region, introduced for EN-systems [15, 22], can be defined as a set R of states of TS which has a consistent 'crossing' relationship with every transition t, i.e., all arcs labelled by t either leave R, or enter R, or do not cross the boundary of R. Such an R is easily seen as a τ-region by assuming: $\sigma(q) = \mathtt{1}$ if $q \in R$; $\sigma(q) = \mathtt{0}$ if $q \notin R$; $\eta(t) = \mathtt{out}$ if t always enters R; $\eta(t) = \mathtt{in}$ if t always leaves R; and $\eta(t) = \top$ otherwise.

The idea of τ-regions of TS is that they would correspond to net places (of the net to be synthesised) including information about markings (σ) and connectivities with all the transitions (η). Note that if TS is the concurrent reachability graph $CRG(N)$ of a τ-net N and p is a place of N, then p defines a τ-region (σ_p, η_p) of $CRG(N)$ as follows: $\sigma_p(q) = M(p)$, for every $q \in \widehat{Q}$ (where M is the marking corresponding to node q), and $\eta_p(t) = F(p, t)$, for every $t \in T$.

We can also state which steps are 'enabled' at the nodes of TS from the 'point of view' of the τ-regions of TS. For every state $q \in \widehat{Q}$, we denote by $\mathit{enbld}_{TS,\tau}(q)$ the set of all *region enabled* steps U satisfying $\eta(U) \in \mathit{enbld}_\tau(\sigma(q))$, for all τ-regions (σ, η) of TS. Intuitively, we treat here each τ-region as if it was a genuine place of a hypothetical τ-net with TS as its concurrent reachability graph. While the inclusion $\mathit{enbld}_{TS}(q) \subseteq \mathit{enbld}_{TS,\tau}(q)$ follows immediately from the definition

of a τ-region, the inverse inclusion is only satisfied for those transition systems TS that can be realised by τ-nets. In such a case, we are not only able to trace enabled steps of TS in the net-type τ, but as well to guarantee that for the steps not enabled at a state of TS, there will be τ-regions disallowing them.

The above observations are reflected in the following fundamental general synthesis result. In a nutshell, TS can be realised by a τ-net iff for every pair of distinct states there is always a place (defined by a τ-region of TS) distinguishing between them (state separation), and there are sufficiently many places/τ-regions to disallow steps which are not present in TS (forward closure).

Theorem 1 ([13]). *TS can be realised by a τ-net* iff *the following two regional axioms are satisfied:*

AXIOM I: STATE SEPARATION
For any pair of states $q \neq r$ of TS, there is a τ-region (σ, η) of TS such that $\sigma(q) \neq \sigma(r)$.
AXIOM II: FORWARD CLOSURE
For every state q of TS, $enbld_{TS}(q) = enbld_{TS,\tau}(q)$.

The above result provides necessary and sufficient conditions for realisability that hold for all transition systems TS. If we are interested in effective solutions to the SYNTHESIS PROBLEM, we first need to assume that TS is finite (in terms of nodes and of arcs). Then, an effective solution to the SYNTHESIS PROBLEM based on Theorem 1 is obtained if one can compute a finite set WR of τ-regions of TS *witnessing* the satisfaction of all instances of AXIOMS I and II [13]. As a *by-product* of checking the realisability of TS, a τ-net $N_{WR} = (P, T, F, M_0)$ satisfying $TS \cong CRG(N)$ can be then constructed by taking $P = WR$ and, for any τ-region (place) $p = (\sigma, \eta)$ in P and every $t \in T$, $F(p, t) = \eta(t)$ and $M_0(p) = \sigma(q_0)$ (recall that q_0 is the initial state of TS).

When it comes to effective solutions of the SYNTHESIS PROBLEM for τ-nets, the complexity of the resulting decision procedures and synthesis algorithms depend on the properties of specific net classes. For example, it has been shown in [24] that Flip-Flop nets can be synthesised in polynomial time. On the other hand, the problem for EN-systems is NP-complete [2]. For safe PT-nets (closely related to EN-systems) an efficient synthesis algorithm was developed and implemented in the PETRIFY tool [9]. Recently, a new algorithm was proposed for the synthesis of EN-systems [1].

6 PTASC-nets are τ-nets

In order to be able to apply Theorem 1 for the synthesis of PTASC-nets, we first need to show that they can be regarded as a class of τ-nets.

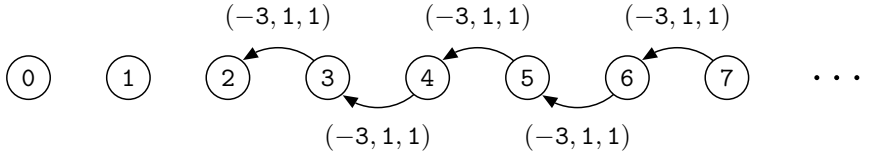
Let $PTASC = (P, T, W, AS, M_0)$ be a PTASC-net. Let $\mathbb{S}_{PTASC} = (\mathbb{Z}^- \times \mathbb{Z}^+ \times \mathbb{Z}, \oplus, \mathbf{0})$ be the connection monoid with $\mathbf{0} = (0, 0, 0)$ and point-wise arithmetic addition $\oplus$. Intuitively, this monoid is an extension of $\mathbb{S}_{PT}$ used in the case of PT-nets. The connections between places and transitions of PTASC-nets can be

expressed through the net-type $\tau_{PTASC} = (\mathbb{Z}^+, \mathbb{S}_{PTASC}, \Delta_{PTASC})$ over $\mathbb{S}_{PTASC}$, where:

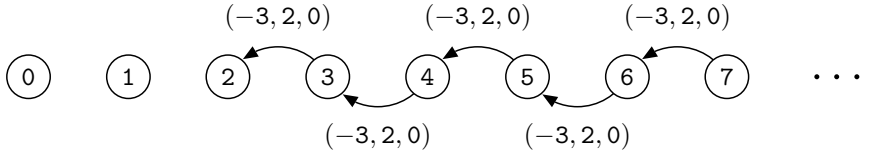
$$\Delta_{PTASC} = \{(\ell, (m, n, k)) \mapsto \ell + m + n + k \mid \ell + m + min\{0, k\} \geq 0\} .$$

This formalises the idea that a place containing ℓ tokens enables steps which require no more than $abs(m)$ standard tokens and, in addition, $abs(k)$ tokens moving along a/sync connections if $k < 0$. The resulting number of tokens in the place is $\ell + m + n + k$. Note that each $(m, n, 0)$ is a standard PT-net connection, and each $(0, 0, k)$ with $k \neq 0$ is a pure a/sync connection. To encode $PTASC$ as a τ_{PTASC}-net, we only have to define $F(p, t) = (-W(p, t), W(t, p), AS(t, p))$.

As an example, a fragment of the net-type τ_{PTASC} with the connection $(-3, 1, 1)$ looks as follows:

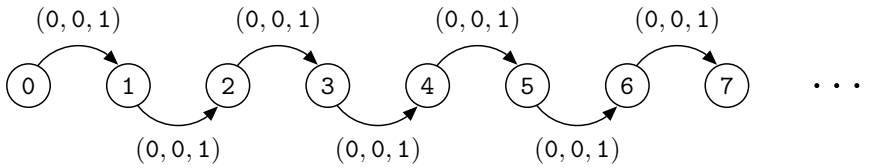


This fragment is isomorphic to the fragment of the net-type τ_{PTASC} with connection $(-3, 2, 0)$:



Indeed, if transitions were to be executed one at a time, the connections $(-3, 1, 1)$ and $(-3, 2, 0)$ would have exactly the same properties. However, this no longer holds in the case of steps. We have, for example, $(-3, 1, 1) \oplus (0, 0, -1) = (-3, 1, 0)$ and although $(-3, 2, 0) \oplus (0, 0, -1) = (-3, 2, -1)$ has the same net effect when executed (removing two tokens), the enabling is different. In the first case the place should contain at least 3 tokens and in the second case at least 4.

The connection between transition a and place p_0 in Figure 1 is captured by $(0, 0, 1)$ and the relevant fragment of the net-type τ_{PTASC} looks as follows:



7 Solving the synthesis problem for PTASC-nets

From the previous section, we know that the general approach to region based synthesis can be applied also in the case of PTASC-nets.

Let $TS = (\widehat{Q}, \mathcal{M}(T), \delta, q_0)$ be an initialised step transition system. To decide whether TS can be realised by a PTASC-net, we first determine its τ_{PTASC}-regions. Assume that $\widehat{Q} = \{q_0, \ldots, q_m\}$ and $T = \{t_1, \ldots, t_n\}$. The τ-regions of TS are pairs $(\sigma : \widehat{Q} \to \mathbb{Z}^+ \ , \ \eta : T \to \mathbb{S}_{PTASC})$. As a start, we restrict the problem to the case of a PTASC-net in which every place has either only standard connections $(y, z, 0)$ or only a/sync connections $(0, 0, w)$ with (multisets of) transitions. (This is the kind of nets we used in [17, 18].) Hence there are two types of places and so we can distinguish between two different kinds of regions: $\mathcal{R}_{PT}$, consisting of the regions of TS corresponding to standard places, and $\mathcal{R}_{AS}$, consisting of the regions of TS corresponding to a/sync places.

We use five tuples of variables: $\mathbf{x} = x_0 \ldots x_m$ and $\widehat{\mathbf{x}} = \widehat{x}_0 \ldots \widehat{x}_m$ over $\mathbb{Q}^+$; $\mathbf{y} = y_1 \ldots y_n$ over $\mathbb{Q}^-$; $\mathbf{z} = z_1 \ldots z_n$ over $\mathbb{Q}^+$; and $\mathbf{w} = w_1 \ldots w_n$ over $\mathbb{Q}$. Moreover, for a multiset α over T and a tuple $\mathbf{h} = h_1 \ldots h_n$ of n arithmetic expressions, we denote $\alpha \otimes \mathbf{h} = \alpha(t_1) \cdot h_1 + \cdots + \alpha(t_n) \cdot h_n$.

We then construct two homogeneous linear systems. The first one encodes the regions representing standard places of the PTASC-net being synthesised:

$$\mathcal{P}_{PT} \ : \ \begin{cases} x_i + \alpha \otimes \mathbf{y} \geq 0 \\ x_j = x_i + \alpha \otimes (\mathbf{y} \oplus \mathbf{z}) \end{cases} \qquad \text{for all } q_i \xrightarrow{\alpha} q_j \text{ in } TS$$

The second one does the same for the a/sync places:

$$\mathcal{P}_{AS} \ : \ \left\{ \widehat{x}_j = \widehat{x}_i + \alpha \otimes \mathbf{w} \right. \qquad \text{for all } q_i \xrightarrow{\alpha} q_j \text{ in } TS$$

Note that the enabling condition $\widehat{x}_i + \alpha \otimes \mathbf{w} \geq 0$ is implied as $\widehat{x}_j$ is a variable over $\mathbb{Q}^+$.

It now follows that the set $\mathcal{R}_{PT}$ is determined by the integer solutions $\mathbf{p} = \mathbf{xyz}$ of $\mathcal{P}_{PT}$ assuming that $\sigma(q_i) = x_i$ for $0 \leq i \leq m$, and $\eta(t_j) = (y_j, z_j, 0)$ for $1 \leq j \leq n$. Similarly, $\mathcal{R}_{AS}$ is determined by the integer solutions $\mathbf{v} = \widehat{\mathbf{x}}\mathbf{w}$ of $\mathcal{P}_{AS}$ assuming that $\sigma(q_i) = \widehat{x}_i$ for $0 \leq i \leq m$, and $\eta(t_j) = (0, 0, w_j)$ for $1 \leq j \leq n$.

Following [8] (see also [11]), one can find in time polynomial in the size of TS a finite set $\mathbf{p}^1, \ldots, \mathbf{p}^r$ of integer solutions from $\mathcal{R}_{PT}$ such that each integer solution $\mathbf{p}$ of $\mathcal{P}_{PT}$ can be expressed as a linear combination $\mathbf{p} = \sum_{l=1}^{r} a_l \cdot \mathbf{p}^l$ with non-negative rational coefficients a_l. And, similarly, one can find in time polynomial in the size of TS a finite set $\mathbf{v}^1, \ldots, \mathbf{v}^s$ of integer solutions from $\mathcal{R}_{AS}$ such that each integer solution $\mathbf{v}$ of $\mathcal{P}_{AS}$ can be expressed as a linear combination $\mathbf{v} = \sum_{l=1}^{s} b_l \cdot \mathbf{v}^l$ with non-negative rational coefficients b_l.

The $\mathbf{p}^l$'s and $\mathbf{v}^l$'s are fixed and (a representative selection of them) are turned into net places if the SYNTHESIS PROBLEM is feasible. More precisely, the initial marking of each $\mathbf{p}^l = \mathbf{x}^l\mathbf{y}^l\mathbf{z}^l$ is given by x_0^l and, for every t_i, we have $F(\mathbf{p}^l, t_i) = (y_i^l, z_i^l, 0)$; moreover, the initial marking of each $\mathbf{v}^l = \widehat{\mathbf{x}}^l\mathbf{w}^l$ is given by $\widehat{x}_0^l$ and, for every t_i, we have $F(\mathbf{v}^l, t_i) = (0, 0, w_i^l)$.

To check the feasibility of the instance of the SYNTHESIS PROBLEM we are considering, we need to verify that the state separation and forward closure properties hold. Checking state separation is carried out for each pair of distinct states, q_i and q_j, and amounts to deciding whether there exists an integer solution

$\mathbf{p}$ of $\mathcal{P}_{PT}$ with coefficients $a_1, \ldots, a_r$ such that $x_i \neq x_j$, or whether there exists an integer solution $\mathbf{v}$ of $\mathcal{P}_{AS}$ with coefficients $b_1, \ldots, b_s$ such that $\widehat{x}_i \neq \widehat{x}_j$. Since these are respectively equivalent to:

$$\sum_{l=1}^{r} a_l \cdot x_i^l \neq \sum_{l=1}^{r} a_l \cdot x_j^l \quad \text{and} \quad \sum_{l=1}^{s} b_l \cdot \widehat{x}_i^l \neq \sum_{l=1}^{s} b_l \cdot \widehat{x}_j^l \ ,$$

one simply checks whether there exists at least one $l \leq r$ such that $x_i^l \neq x_j^l$, or at least one $l \leq s$ such that $\widehat{x}_i^l \neq \widehat{x}_j^l$ (in such a case $\mathbf{p}^l$ or $\mathbf{v}^l$ becomes a witness).

Checking forward closure is carried out for each state q_i, and considers steps of $\mathcal{M}(T)$ that are not enabled at q_i in TS. One only needs to consider steps α with $|\alpha| \leq max$ where max is the maximum size of steps labelling arcs in TS since, as one can easily see,

$$\mathbf{p}_{max} = \underbrace{max \ \ldots \ max}_{m+1 \ times} \ \underbrace{-1 \ \ldots \ -1}_{n \ times} \ \underbrace{1 \ \ldots \ 1}_{n \ times}$$

is an integer solution of $\mathcal{P}_{PT}$, and any step of size greater than max will be disabled by $\mathbf{p}_{max}$. (Intuitively, $\mathbf{p}_{max}$ is a witness place with max tokens which is connected by a self-loop with each of the transitions in T.) Such a step is not region enabled at q_i *iff* for some integer solution $\mathbf{p}$ of $\mathcal{P}_{PT}$ with coefficients $a_1, \ldots, a_r$ we have that $x_i + \alpha \otimes \mathbf{y} < 0$, or for some integer solution $\mathbf{v}$ of $\mathcal{P}_{AS}$ with coefficients $b_1, \ldots, b_s$ we have that $\widehat{x}_i + \alpha \otimes \mathbf{w} < 0$. These are respectively equivalent to:

$$\sum_{l=1}^{r} a_l \cdot (x_i^l + \alpha \otimes \mathbf{y}^l) < 0 \quad \text{and} \quad \sum_{l=1}^{s} b_l \cdot (\widehat{x}_i^l + \alpha \otimes \mathbf{w}^l) < 0 \ ,$$

and so the step α is not region enabled at q_i *iff* there is $l \leq r$ such that $x_i^l + \alpha \otimes \mathbf{y}^l < 0$, or there is $l \leq s$ such that $\widehat{x}_i^l + \alpha \otimes \mathbf{w}^l < 0$ (in such a case $\mathbf{p}^l$ or $\mathbf{v}^l$ becomes a witness).

We can therefore conclude that in this case the SYNTHESIS PROBLEM is decidable and, if TS is realisable, a suitable PTASC-net can be constructed in polynomial time.

In the general case, when a place can have both standard and a/sync connections, we assume that $\widehat{Q}$, T, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and $\mathbf{w}$ are as before. Moreover, $\mathbf{p} = \mathbf{xyzw}$. Recall that in a PTASC-net, token transfer along a/sync connections may lead to either a surplus or a deficit of tokens. In τ_{PTASC}-nets this has been abstracted to the enabling condition $\ell + m + min\{0, k\} \geq 0$. Therefore we distinguish whether a step in TS takes a/sync tokens from the place ($k < 0$) which is being constructed (as a region), or adds ($k \geq 0$). The steps satisfying the former w.r.t. that place belong to a subset B of steps labelling arcs in TS, those satisfying the latter are outside B. For every subset B, we define a homogeneous linear

system:

$$\mathcal{P}_B \;:\; \begin{cases} x_j = x_i + \alpha \otimes (\mathbf{y} \oplus \mathbf{z} \oplus \mathbf{w}) & \text{for all } q_i \xrightarrow{\alpha} q_j \text{ in } TS \\ x_i + \alpha \otimes (\mathbf{y} \oplus \mathbf{w}) \geq 0 & \text{for all } q_i \xrightarrow{\alpha} q_j \text{ with } \alpha \in B \\ x_i + \alpha \otimes \mathbf{y} \geq 0 & \text{for all } q_i \xrightarrow{\alpha} q_j \text{ with } \alpha \notin B \\ \alpha \otimes \mathbf{w} < 0 & \text{for all } \alpha \in B \\ \alpha \otimes \mathbf{w} \geq 0 & \text{for all } \alpha \notin B \end{cases}$$

The τ_{PTASC}-regions (σ, η) of TS are determined by the integer solutions $\mathbf{p}$ of all systems $\mathcal{P}_B$, assuming that $\sigma(q_i) = x_i$ for $0 \leq i \leq m$, and $\eta(t_j) = (y_j, z_j, w_j)$, for $1 \leq j \leq n$.

Again, following [8], for each B one can find in time polynomial in the size of TS a finite set $\mathbf{p}_B^1, \ldots, \mathbf{p}_B^{r_B}$ of integer solutions of $\mathcal{P}_B$ such that each integer solution $\mathbf{p}$ of $\mathcal{P}_B$ can be expressed as a linear combination $\mathbf{p} = \sum_{l=1}^{r_B} a_l \cdot \mathbf{p}_B^l$ with non-negative rational coefficients a_l. The $\mathbf{p}_B^l$'s are fixed and turned into net places if the SYNTHESIS PROBLEM is feasible. More precisely, the initial marking of each $\mathbf{p}_B^l = \mathbf{xyzw}$ is given by x_0 and, for every t_i, we have $F(\mathbf{p}_B^l, t_i) = (y_i, z_i, w_i)$.

Checking state separation is carried out for each pair of distinct states, q_i and q_j, and amounts to deciding whether there exists a B and an integer solution $\mathbf{p} = \sum_{l=1}^{r_B} a_l \cdot \mathbf{p}_B^l$ of $\mathcal{P}_B$ such that $x_i \neq x_j$. Since the latter is equivalent to:

$$\sum_{l=1}^{r_B} a_l \cdot x_{Bi}^l \neq \sum_{l=1}^{r_B} a_l \cdot x_{Bj}^l \;,$$

one simply checks whether there exists at least one l such that $x_{Bi}^l \neq x_{Bj}^l$. Hence, to check state separation for q_i and q_j, one needs to see whether there is B and $l \leq r_B$ satisfying $x_{Bi}^l \neq x_{Bj}^l$.

When checking forward closure, similarly as in the previous case we only need to consider steps α with $|\alpha| \leq max$, where max is the maximum size of steps occurring as arc labels in TS. Given a state q_i and a step α which is not enabled at q_i in TS, α is not region enabled at q_i *iff* for some B and some integer solution $\mathbf{p} = \mathbf{xyzw} = \sum_{l=1}^{r_B} a_l \cdot \mathbf{p}_B^l$ of $\mathcal{P}_B$, we have that $x_i + \alpha \otimes \mathbf{y} + min\{0, \alpha \otimes \mathbf{w}\} < 0$. We then consider two possibilities (Case 2 is only attempted if Case 1 was unsuccessful), and we proceed by considering in turn all sets B, stopping if a witness of forward closure is found for α:

Case 1: $\alpha \otimes \mathbf{w} < 0$ and $x_i + \alpha \otimes (\mathbf{y} \oplus \mathbf{w}) < 0$. This leads to a system of two equations with variables $a_1, \ldots, a_{r_B}$ of the form:

$$\begin{cases} \sum_{l=1}^{r_B} (\alpha \otimes \mathbf{w}_B^l) \cdot a_l < 0 \\ \sum_{l=1}^{r_B} (x_{Bi}^l + \alpha \otimes (\mathbf{y}_B^l \oplus \mathbf{w}_B^l)) \cdot a_l < 0 \end{cases}$$

The feasibility of this system can be checked following [8], and if an integer solution exists, we add the corresponding witness place to the net being constructed.

Case 2: $\alpha \otimes \mathbf{w} \geq 0$ and $x_i + \alpha \otimes \mathbf{y} < 0$. We then proceed similarly as in Case 1, considering a linear system of the form:

$$\begin{cases} \sum_{l=1}^{r_B} (\alpha \otimes \mathbf{w}_B^l) \cdot a_l \geq 0 \\ \sum_{l=1}^{r_B} (x_{Bi}^l + \alpha \otimes \mathbf{y}_B^l) \cdot a_l < 0 \end{cases}$$

Thus the SYNTHESIS PROBLEM in the general case is decidable though the solution presented is exponential, as we need to consider exponentially many sets B.

8 Conclusions

The seminal work by Andrzej Ehrenfeucht and Grzegorz Rozenberg about labelled partial 2-structures [15] introduced the notion of a region. This concept proved to be very useful for relating two different representations of concurrent systems: one given by a Petri net (representing explicitly local information), and the other given in the form of a transition system (capturing global behavioural information). In this paper, we take advantage of a generalised version of this notion, i.e., τ-region, to address the synthesis problem for PT-nets with a/sync connections. Solving this problem presented a new challenge as the interpretation of a/sync arcs can vary depending on the current marking. If, for a given place, the a/sync arcs are not 'balanced', then the 'unbalanced' arcs are interpreted as standard PT-net arcs. As a consequence, there may be exponentially many cases to consider when building a linear system used to find τ-regions while constructing PTASC-nets. However, we also identified an important class of PTASC-nets for which the problem is polynomial. In future work, we plan to investigate ways in which the number of cases considered for the general PTASC-net synthesis can be significantly reduced.

References

1. Badouel, E., Private communication (2011)
2. Badouel, E., Bernardinello, L., Darondeau, P.: The Synthesis Problem for Elementary Net Systems is NP-complete. Theoretical Computer Science **186** (1997) 107–134
3. Badouel, E., Darondeau, P.: Theory of Regions. In: Reisig, W., Rozenberg, G. (eds.): Lectures on Petri Nets I: Basic Models, Advances in Petri Nets. Lecture Notes in Computer Science **1491**, Springer (1998) 529–586
4. Bergenthum, R., Desel, J., Lorenz, R., Mauser, S.: Synthesis of Petri Nets from Scenarios with VipTool. Lecture Notes in Computer Science **5062**, Springer (2008) 388–398
5. Bruni, R., Montanari, U.: Zero-Safe Nets: Comparing the Collective and Individual Token Approaches. Information and Computation **156** (2000) 46–89
6. Busi, N., Pinna, G.M.: Synthesis of Nets with Inhibitor Arcs. Lecture Notes in Computer Science **1243**, Springer (1997) 151–165

7. Carmona, J., Cortadella, J., Kishinevsky, M.: Genet: A Tool for the Synthesis and Mining of Petri Nets. Proc. of ACSD'09, IEEE Computer Society (2009) 181–185
8. Chernikova, N.: Algorithm for Finding a General Formula for the Non-negative Solutions of a System of Linear Inequalities. USSR Computational Mathematics and Mathematical Physics **5** (1965) 228–233
9. Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., Yakovlev, A.: Logic Synthesis of Asynchronous Controllers and Interfaces. Springer (2002)
10. Darondeau, P.: Deriving Unbounded Petri Nets from Formal Languages. Lecture Notes in Computer Science **1466**, Springer (1998) 533–548
11. Darondeau, P.: On the Petri Net Realization of Context-free Graphs. Theoretical Computer Science **258** (2001) 573–598
12. Darondeau, P., Koutny, M., Pietkiewicz-Koutny, M., Yakovlev, A.: Synthesis of Nets with Step Firing Policies. LNCS 5062 (2008) 112–131
13. Desel, J., Reisig, W.: The Synthesis Problem of Petri Nets. Acta Informatica 33 (1996) 297–315
14. Desel, J., Yakovlev, A. (eds.): Applications of Region Theory 2011. Proc. of the Workshop Applications of Region Theory 2011 (ART-2011), CEUR-WS **725** (2011) `http://ceur-ws.org/Vol-725/`
15. Ehrenfeucht, A., Rozenberg, G.: Partial 2-structures; Part I: Basic Notions and the Representation Problem, and Part II: State Spaces of Concurrent Systems. Acta Informatica **27** (1990) 315–368
16. Hoogers, P.W., Kleijn, H.C.M., Thiagarajan, P.S.: A Trace Semantics for Petri Nets. Information and Computation **117** (1995) 98–114
17. Kleijn, J., Koutny, M.: Causality in Structured Occurrence Nets. Lecture Notes in Computer Science **6875**, Springer (2011) 283–297
18. Kleijn, J., Koutny, M.: Localities in Systems with a/sync Communication. Submitted (2011)
19. Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Petri Nets with Localities. Sci. Ann. Comp. Sci. **19** (2009) 1–23
20. Koutny, M., Randell, B.: Structured Occurrence Nets: A Formalism for Aiding System Failure Prevention and Analysis Techniques. Fundamenta Informaticae **97** (2009) 41–91
21. Mukund, M.: Petri Nets and Step Transition Systems. International Journal of Foundations of Computer Science **3** (1992) 443–478
22. Nielsen, M., Rozenberg, G., Thiagarajan, P.S.: Elementary Transition Systems. Theoretical Computer Science **96** (1992) 3–33
23. Pietkiewicz-Koutny, M.: Synthesising Elementary Net Systems with Inhibitor Arcs from Step Transition Systems. Fundamenta Informaticae **50** (2002) 175–203
24. Schmitt, V.: Flip-Flop Nets. Lecture Notes in Computer Science **1046**, Springer (1996) 517–528
25. Solé, M., Carmona, J.: Rbminer: A Tool for Discovering Petri Nets from Transition Systems. Lecture Notes in Computer Science **6252**, Springer (2010) 396–402
26. Verbeek, H.M.W., et. al: ProM 6: The Process Mining Toolkit. BPM 2010 Demo (2010)