The Petri net synthesis problem for automatic graphs

Eric Badouel, Philippe Darondeau

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The Petri net synthesis problem for automatic graphs

Eric Badouel — Philippe Darondeau

N° 4661

Décembre 2002

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--4661--FR+ENG

The Petri net synthesis problem for automatic graphs

Eric Badouel* , Philippe Darondeau †

Thème 1 — Réseaux et systèmes
Projet S4

Rapport de recherche n° 4661 — Décembre 2002 — 23 pages

Abstract: Automatic graphs are possibly infinite graphs with a rational presentation by finite automata; they include grids as a particular case, and Sénizergues has shown in a seminal work that they are an extension of deterministic pushdown graphs. Here, we propose a procedure that decides from the automatic presentation of a graph whether it may be realized isomorphically by the reachable state graph of some finite Petri net, with transitions in bijection with labels of edges. As automatic graphs cover more concurrent systems than pushdown graphs, this procedure improves over, and indeed subsumes, the net synthesis procedure studied earlier for pushdown graphs.

Key-words: automatic graphs, Petri nets, regions, synthesis

* ENSP, BP 8390, Yaoundé, Cameroun <ebadouel@polytech.unimet.cm>
† IRISA, campus de Beaulieu, F35042 Rennes Cedex <darondeau@irisa.fr>

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex (France)
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Le problème de synthèse des réseaux de Petri pour les graphes automatiques

Résumé : Les graphes automatiques sont des graphes finis ou infinis donnés par une présentation rationnelle, faite d'automates finis à une ou deux bandes; ces graphes contiennent en particulier les grilles, et Sénizergues a montré dans un travail fondateur qu'ils forment une extension propre des graphes de transitions d'automates déterministes à pile. Nous proposons ici une procédure qui décide, à partir de la présentation automatique d'un graphe fini ou infini, s'il peut être réalisé à isomorphisme près par le graphe des marquages accessibles d'un réseau de Petri fini, avec des transitions en bijection avec les étiquettes des arcs du graphe. Comme les graphes automatiques couvrent plus de systèmes concurrents que ne le font les graphes de transitions des automates à pile, cette procédure améliore, et de fait recouvre, la procédure proposée antérieurement pour la synthèse de réseaux de Petri à partir des graphes de transitions d'automates déterministes à pile.

Mots-clés : graphes automatiques, réseaux de Petri, régions, synthèse

1 Introduction

The problem of net synthesis consists in deciding whether there exists some Petri net satisfying given specifications. In the event of a positive decision, such a Petri net should be constructed; otherwise feedback should hopefully be provided giving indications why the given specifications cannot be realized by a Petri net. In front of this general scheme, the algorithmic solutions provided so far in the literature apply to much restricted classes of specifications. In a first stage, the synthesis problem has been dealt with assuming that specifications are finite labelled graphs given by explicit sets of states and labelled transitions. Labelled graphs are said to be realized by finite Petri nets, labelled injectively on transitions, if they are isomorphic to their reachable marking graphs. Such specifications have a unique model and they cannot be stated without a complete knowledge of the states of this model, which is a strong requirement seldom met in practical systems design. Moreover, only finite transition systems are considered whereas the marking graphs of finite Petri nets may well be infinite. These limitations have been lifted partially in subsequent studies. On the one hand, a decision has been brought to the problem of synthesizing nets from infinite graphs given by graph grammars, that is to say context-free graphs (or pushdown graphs) [8]. This is far from sufficient, since e.g. the complete grid on the positive quarter of the plane is a typical example of a non context-free graph with regular structure that may be realized by a finite Petri net. One might reasonably discard this example by imposing bounded asynchronism on Petri nets, but even under that proviso the context-free graphs are far from exhausting the class of marking graphs of Petri nets. On the other hand, a decision has been brought to the problem of synthesizing nets from a disjoint pair of regular languages, one expressing required behaviours and the other expressing forbidden behaviours without making any reference to states [7]. A Petri net realizes such specifications if its language contains all required behaviours and no forbidden behaviour. This also is far from being sufficient, since most state-based properties of the system under design cannot be formulated as language inclusions.

In this paper, we improve over previous work by lifting further the two restrictions mentionned above. The main result is an effective solution to the net synthesis problem for automatic graphs. Automatic graphs are generally infinite graphs whose set of states and labelled transition relations are represented by finite automata and transducers (or two-tape automata). The class of automatic graphs contains strictly the class of context-free graphs, hence the new synthesis algorithm for automatic graphs improves over the former algorithm. Moreover, a straigtforward adaptation of the new algorithm allows to extend net synthesis from automatic graphs to *automatic specifications*. Automatic specifications resemble much automatic graphs, but they need not define a unique labelled graph. In fact, automatic specifications may have an arbitrary number of models ranging from zero to infinity. As a result, one may now search automatically for a Petri net with marking graph in an infinite family of labelled graphs, but there is a price to pay since automatic specifications are admittedly difficult to write. We leave open for further study the question of producing automatic specifications by translation from more convenient specification languages.

The remaining sections of the paper are organized as follows. Sections 2 and 3 bring the necessary background of automatic structures and Petri net synthesis. Sections 4 and 5 propose a solution to the Petri net synthesis problem for automatic graphs. Section 6 and 7 show a full example and a counter-example. Section 8 extends the results from automatic graphs to automatic specifications. The paper is followed by a short appendix on semi-linear subsets of $\mathbb{N}^n$ and $\mathbb{Z}^n$.

2 The basics of automatic graphs

Graphs with an *automatic structure*, also called *automatic graphs*, were evolved by Sénizergues as a smooth extension of Cayley graphs of automatic groups. In order to reach an efficient decision of classical problems on groups, automatic structures of Cayley graphs were defined as follows by Epstein *et al.* [10]: a regular language on the set of group generators supplies the representatives for vertices (i.e. group elements) through the evaluation of words in the group; pairs of equivalent representatives are accepted by a finite automaton with two tapes (called the *equalizer*); and the right action of each generator is defined similarly by a finite automaton with two tapes (called a *multiplier*). For precisions, the reader is referred to [10].

Sénizergues's definition of automatic graphs [23] retains from this framework as much as possible to accomodate also non homogeneous graphs such as the deterministic pushdown graphs. Setting focus on reachable graphs with one root, and with a deterministic labelling of edges, he essentially adapts the above definition by substituting sequences of labels on rooted paths in the graph for sequences of group generators. The equalizer and the multipliers (one for each symbol in the alphabet of labels) are defined as above by finite two-tape automata, however with some significant differences. In [10], two alternative types of two-tape automata were considered, namely the *synchronous* automata that read their tapes in alternation, and the *asynchronous* automata. Synchronous rational languages form a boolean algebra, entailing decidability of the first order theory of synchronous automatic groups, but asynchronous regular languages do not form a boolean algebra, although they are *closed under complementation*. It is worth noting that *boundedly asynchronous* automata, where one tape cannot be read ahead from the other by more than fixed k letters, are actually sufficient for supplying asynchronous automatic groups with automatic structures [10]. The two-tape automata used in [23] to define automatic structures of graphs are basically of the bounded asynchronous type, but the boundedness constraint is lifted after the first tape has been exhausted in the specific case of the multiplier automata. In spite of the boundedness constraints, the first order theory of automatic graphs is not decidable [20].

In this work, we adopt Sénizergues's view of automatic graphs but we discard all boundedness constraints from his definitions. Before stating the simplified definitions, let us briefly comment on the relationship between automatic graphs and the wider class of *automatic structures* introduced in [17] and further studied in [6]. Automatic structures are relational structures over an underlying set, whose elements are represented as words of a regular language. Equality and other relations are decided by finite multi-tape automata of the

synchronous type, and so the first order theories of automatic structures are decidable. In particular, labelled graphs may be represented with automatic structures in which the alphabet used to represent vertices as words differs from the alphabet of labels. The same remark applies to the asynchronous automatic structures that underlie the *rational graphs* studied in [18]. We do not know whether the Petri net realization problem can be decided for these larger classes of automatic graphs: our present technique of net synthesis relies heavily on the assumption that vertices are represented by labels of incoming paths. Definitions now.

Definition 2.1 (excerpt from Epstein) *An* asynchronous deterministic two-tape automaton*, or simply* asynchronous automaton*, over A is a partial deterministic automaton over $A \cup \{\$\}$, in which the set of states is partitioned into five subsets, denoted by S_L, $S_L^\$$, S_R, $S_R^\$$ and $S^\$$. The only state in $S^\$$ is the unique accept state, called $s_\$$; no arrows originate from it. An arrow labelled by an element of A and having its source in $S_L \cup S_R$ has its target in $S_L \cup S_R$; if its source is in $S_L^\$$ or $S_R^\$$, its target is in the same set. A \$ arrow with source in S_L has target in $S_R^\$$; one with source in S_R has target in $S_L^\$$; and one with source in $S_L^\$ \cup S_R^\$$ has target $s_\$$. The start state is in $S_L \cup S_R$. We say that an asynchronous automaton* accepts *a pair of strings $(w_L, w_R) \in A^* \times A^*$ if there is a shuffle w of $(w_L\$, w_R\$)$ which is accepted by the automaton.*

Lemma 2.2 (Epstein et al.) *Complementation in $A^* \times A^*$ preserves the class of languages accepted by asynchronous automata.*

In the sequel, $L(M)$ denotes the rational relation formed of all pairs of words $(w_L, w_R) \in A^* \times A^*$ accepted by asynchronous automaton M. For convenience, in the illustrative parts, the generators (a, ε) and (ε, a) of $A^* \times A^*$ will be written respectively as a and $\overline{a}$. Thus, for instance, $\overline{a}b\overline{a}$ means (b, aa). Throughout the paper, $G = (V, E, v_0, \lambda)$ will denote a graph in which all vertices $v \in V$ may be reached from the root $v_0 \in V$ by sequences of directed edges in $E \subseteq V \times V$, with a surjective edge labelling map $\lambda : E \to A$, such that distinct edges with common source are mapped to distinct labels in $A = \{a_1, \ldots, a_n\}$. We let $v \cdot a = v'$ if $\lambda(v, v') = a$ and be undefined otherwise; by extension, we let $v \cdot aw = v \cdot a \cdot w$ for $w \in A^*$ different from ε (the empty word).

Definition 2.3 (adapted from Sénizergues) *G is an* automatic graph *if it may be given a* rational structure *$\mathcal{S} =< W, w_0, Eq, M_1, \ldots, M_n >$ as follows:*
i) *W is a finite (one-tape) automaton over A, accepting a regular language $L(W)$ such that, for all $w \in L(W)$, $v_0 \cdot w$ is defined, and for every vertex $v \in V$, $v = v_0 \cdot w$ for some $w \in L(W)$,*
ii) *$w_0 \in L(W)$ and $v_0 = v_0 \cdot w_0$,*
iii) *Eq is an asynchronous automaton over A, such that*
$L(Eq) = \{(w, w') \in L(W) \times L(W) \mid v_0 \cdot w = v_0 \cdot w'\}$,
iv) *for each $i \in [1, n]$, M_i is an asynchronous automaton over A, such that $L(M_i) = \{(w, w') \in L(W) \times L(W) \mid v_0 \cdot wa_i = v_0 \cdot w'\}$.*

One might assume w.l.o.g. that $L(W)$ is closed under prefix. We shall not use this assumption. Deterministic pushdown graphs are a particular case of automatic graphs with bounded tree-width. A typical example of an automatic graph with unbounded tree-width is the complete grid on the positive quarter of the plane (see Fig. 1). The grid, which is

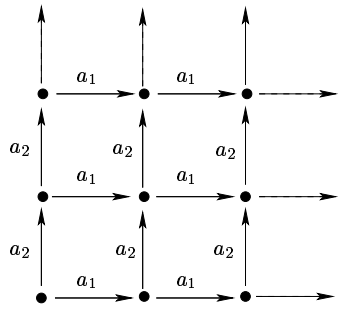

Figure 1: The grid

obviously isomorphic to the reachable marking graph of a one-safe Petri net, may be given a rational structure $\mathcal{S} =< W, w_0, Eq, M_1, M_2 >$ as follows:
1. $L(W) = a_1^* a_2^*$ and $w_0 = \varepsilon$,
2. $L(Eq) = \{(w, w) \mid w \in L(W)\}$,
3. $L(M_1) = (a_1\overline{a_1})^*\,\overline{a_1}\,(a_2\overline{a_2})^*$,
4. $L(M_2) = (a_1\overline{a_1})^*\,\overline{a_2}\,(a_2\overline{a_2})^*$.

3 The principles of Petri net synthesis

In [19] and [9], labelled graphs isomorphic to reachable state graphs of P/T-nets, with as many transitions as labels, were shown to satisfy two characteristic axioms, reminiscent of the regional axioms for Elementary Transition Systems [11] [12]. Based on these axioms, procedures deciding upon the P/T-net definability of finite graphs and of context-free graphs were proposed, respectively, in [1] [2] and in [8]. The goal of this paper is to extend the decision to automatic graphs. We recall in this section the basic principles of P/T-net synthesis, which the development presented in this paper rests upon. We refer the reader to [3] for a more substantial presentation of net synthesis, for various classes of nets, taken as actual parameters of the net synthesis functor. Definitions follow.

Definition 3.1 (P/T-nets) *A P/T-net is a triple $N = (P, T, F)$ where P and T are finite disjoint sets of* places *and* transitions*, respectively, and F is a function, $F : (P \times T) \cup (T \times P) \to \mathbb{N}$. A* marking *of N is a map $m : P \to \mathbb{N}$. The* state graph *of N is a labelled multi-graph, with markings as vertices, where there is an edge labelled with transition t from m to m' (in notation: $m[t\rangle m'$) if and only if, for every place $p \in P$, $m(p) \geq F(p,t)$ and $m'(p) = m(p) - F(p,t) + F(t,p)$. The* reachable state graph *of an* initialized *P/T-net*

$\mathcal{N} = (P, T, F, m_0)$ *with* initial marking m_0 *is the restriction of its state graph reachable from* m_0.

Definition 3.2 (Implicit places of a graph) *Given a graph* $G = (V, E, v_0, \lambda)$*, labelled in* A*, an* implicit place *of* G *is a* P/T*-net* $p = (\{p\}, A, F, m_0)$ *with one place* p*, compatible with* G *in the sense that there exists a map* $\sigma : V \to \mathbb{N}$*, sending vertices to place values, such that* $\sigma(v_0) = m_0(p)$ *and every edge* (v, v') *projects through this map to a firing step* $m[\lambda(v, v')\rangle m'$ *of the net, with markings* m *and* m' *respectively given by* $m(p) = \sigma(v)$ *and* $m'(p) = \sigma(v')$*. Map* σ*, which is unique, is called the* companion map *of place* p.

Definition 3.3 (Separation and inhibition) *Let* $p = (\{p\}, A, F, m_0)$ *be an implicit place of* G*. Place* p *is said to* separate *vertices* v *and* v' *if the companion map* $\sigma : V \to \mathbb{N}$ *satisfies* $\sigma(v) \neq \sigma(v')$*. Place* p *is said to* inhibit *an action with label* $a \in A$ *at vertex* v *if the companion map satisfies* $\sigma(v) < F(p, a)$.

Proposition 3.4 *Graph* G *is isomorphic to the reachable state graph of a* P/T*-net if and only if it has a finite subset of implicit places* P *such that all distinct vertices are separated by places in* P*, and for every vertex* v*, every action* a *labelling no edge with origin* v *is inhibited at this vertex by some place in* P*. When both conditions are satisfied, graph* G *is isomorphic to the reachable state graph of the* synthesized net $\mathcal{N} = (P, A, F, M_0)$ *obtained by gluing together on transitions in* A *all implicit places in* P*, with pointwise definition of the flow relation* F *and the initial marking* M_0.

4 The implicit places of an automatic graph

Let $G = (V, E, v_0, \lambda)$ be an automatic graph over $A = \{a_1, \ldots, a_n\}$, and let $\mathcal{S} =< W, w_0, Eq, M_1, \ldots, M_n >$ be a rational structure for G. Each implicit place $p = (\{p\}, A, F, m_0)$ of graph G may be identified with the $(2n+1)$-vector

$$\vec{p} =< m_0(p), F(p, a_1), \ldots, F(p, a_n), F(a_1, p), \ldots, F(a_n, p) > \quad (1)$$

Under this representation, the implicit places of G are the integral points of a polyhedral cone whose generators may be computed from $\mathcal{S}$. The purpose of the section is to show an effective computation of this cone.

By definition of implicit places, vector $\vec{x} =< x_o, x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n} >$ defines an implicit place of G if and only if all entries x_i are non-negative integers and there exists a map $\sigma : V \to \mathbb{N}$ such that $\sigma(v_0) = x_0$ and for every edge (v, v') of G, if $\lambda(v, v') = a_i$ then $\sigma(v) \geq x_i$ and $\sigma(v') = \sigma(v) - x_i + x_{(n+i)}$. Using the rational structure $\mathcal{S}$ of G, the latter condition may be reformulated as follows: for all $i \in [1, n]$ and for every pair of words $(w_L, w_R) \in L(M_i)$,

$$\sigma(v_0 \cdot w_L) \geq x_i \ \wedge \ \sigma(v_0 \cdot w_R) = \sigma(v_0 \cdot w_L) - x_i + x_{(n+i)}$$

At this stage, one can take advantage of a crucial fact: for any non-empty prefix ua_i of a word $w \in L(W)$, both $v_0 \cdot u$ and $v_0 \cdot ua_i$ are defined, and $(v_0 \cdot u, v_0 \cdot ua_i)$ is an edge of

G, labelled with a_i. Thus, if one chooses in $L(W)$ two representatives w' and w'' for the respective vertices $v_0 \cdot u$ and $v_0 \cdot ua_i$, it comes:

$$\sigma(v_0 \cdot ua_i) = \sigma(v_0 \cdot w'') = \sigma(v_0 \cdot w') - x_i + x_{(n+i)} = \sigma(v_0 \cdot u) - x_i + x_{(n+i)}$$

It follows by an induction on prefixes u of words $w \in L(W)$ that, if $|w|_i$ denotes the occurrence count of letter a_i in w, then

$$\sigma(v_0 \cdot w) = x_0 + \sum_{i=1}^{n} |w|_i \times (x_{(n+i)} - x_i) \tag{2}$$

Using this equation, a possibly infinite set of linear equality or inequality contraints on vector $\vec{x}$ may be derived from the rational structure of G. The constraints are the following, where j, (w', w'') and (w_L, w_R) range respectively over $[1, n]$, $L(M_j)$ and $L(Eq)$.

$$\sum_{i=1}^{n} |w_0|_i \times (x_{(n+i)} - x_i) = 0 \tag{3}$$

$$\sum_{i=1}^{n} |w_L|_i \times (x_{(n+i)} - x_i) = \sum_{i=1}^{n} |w_R|_i \times (x_{(n+i)} - x_i) \tag{4}$$

$$\sum_{i=1}^{n} |w'|_i \times (x_{(n+i)} - x_i) \geq x_j - x_0 \tag{5}$$

$$\sum_{i=1}^{n} |w''|_i \times (x_{(n+i)} - x_i) = \sum_{i=1}^{n} |w'|_i \times (x_{(n+i)} - x_i) - x_j + x_{(n+j)} \tag{6}$$

It is readily observed that any vector $\vec{x}$ satisfying all constraints defines an implicit place of G, with companion map given by equation 2. The problem now is to reduce the infinite set of constraints to an equivalent but *finite* set of constraints. Before we give the reduction, let us recall definitions and facts.

Definition 4.1 (Commutative image) *The* commutative image *of a word $w \in A^*$ is the n-vector $\psi(w)$ whose respective entries $\psi(w)[i]$ count for each $i \in [1, n]$ the occurences of letter a_i in w.*

Definition 4.2 (Rational subset) *Let $\mathcal{M} = (\mathcal{M}, \cdot, \vec{1})$ a monoid. A subset $\mathcal{R}$ of $\mathcal{M}$ is* rational *if there exists a finite automaton R over $\mathcal{M}$, i.e. a finite graph $(V, E, v_0, V_F, \lambda)$ with a distinguished subset V_F of accepting vertices and with edges labelled in $\mathcal{M}$, such that $L(R) = \mathcal{R}$ when concatenation is interpreted as product in $\mathcal{M}$ and ε is interpreted as the neutral element $\vec{1}$.*

Definition 4.3 (Semi-linear subset) *Let $\mathcal{M} = (\mathcal{M}, \cdot, \vec{1})$ a monoid. A subset of $\mathcal{M}$ is* linear *if it may be expressed as $m \cdot \mathcal{F}^*$ where $m \in \mathcal{M}$, $\mathcal{F}$ is a finite subset of $\mathcal{M}$, and $\mathcal{F}^*$ is the least subset $\mathcal{S}$ of $\mathcal{M}$ such that $\vec{1} \in \mathcal{S}$ and $\mathcal{S} \cdot \mathcal{F} \subseteq \mathcal{S}$. A finite union of linear subsets of $\mathcal{M}$ is called a* semi-linear *subset of $\mathcal{M}$.*

Proposition 4.4 (Eilenberg, Schützenberger) *Rational subsets and semi-linear subsets coincide in any commutative monoid.*

The correspondence stated in the above proposition, established in [13], is moreover effective. Coming back to our infinite set of constraints, it should be clear that the following are rational subsets of $\mathbb{Z}^n$ (the monoid with coordinate-wise addition of vectors as product operation and the null vector as neutral element):

$$\Delta(Eq) = \{\psi(w_R) - \psi(w_L) \mid (w_L, w_R) \in L(Eq)\}$$
$$\Delta(M_j) = \{\psi(w'') - \psi(w') \mid (w', w'') \in L(M_j)\}$$
$$\Sigma_1(M_j) = \{\psi(w') \mid \ (\exists w'')\ (w', w'') \in L(M_j)\}$$

For instance, the asynchronous automaton Eq (see Def. 2.1) may be transformed into an automaton over $\mathbb{Z}^n$, accepting $\Delta(Eq)$, as follows: for each arrow with label a_i, change the label to $\psi(a_i)$ if the arrow has its source in $S_R \cup S_R^\$$, or to $-\psi(a_i)$ if it has its source in $S_L \cup S_L^\$$; change label \$ to $\vec{0}$ wherever it occurs. Thus, by proposition 4.4, the set $\Delta(Eq)$ is a finite union of linear subsets:

$$\Delta(Eq) = \cup_{k=1}^{K}\ \vec{y_k} + (Z_k)^*$$

where $\vec{y_k} \in \mathbb{Z}^n$, Z_k is a finite subset of vectors in $\mathbb{Z}^n$, and $(Z_k)^*$ is the set of linear combinations of vectors in Z_k with non-negative integral coefficients. Therefore, the infinite set of constraints imposed by equation 4 on $\vec{x} \in \mathbb{N}^{2n+1}$ is equivalent to the finite set of constraints:

$$\sum_{i=1}^{n} \vec{y_k}[i] \times (x_{(n+i)} - x_i) \quad = \quad 0 \tag{7}$$
$$\sum_{i=1}^{n} \vec{z_k}[i] \times (x_{(n+i)} - x_i) \quad = \quad 0 \tag{8}$$

where k ranges over $[1, K]$ and $\vec{z_k}$ ranges over (Z_k). A similar reduction applies to constraints imposed by equation 6. We are left with reducing, for each fixed $j \in [1, n]$, the infinite set of inequality constraints imposed by equation 5.

Observe that $\Sigma_1(M_j)$ is a rational subset of $\mathbb{N}^n$ (when M_j is transformed to an automaton accepting this set, labels a_i are changed either to $\psi(a_i)$ or to $\vec{0}$, depending on whether the source of the arrow is in $S_L \cup S_L^\$$ or in $S_R \cup S_R^\$$). Thus, by proposition 4.4, $\Sigma_1(M_j)$ is a finite union of linear subsets:

$$\Sigma_1(M_j) = \cup_{l=1}^{L}\ \vec{b_l} + (C_l)^*$$

where $\vec{b_l} \in \mathbb{N}^n$, C_l is a finite subset of vectors in $\mathbb{N}^n$, and $(C_l)^*$ is the set of all sums of vectors taken from C_l. Therefore, the infinite set of linear constraints imposed by equation

5 on $\vec{x} \in \mathbb{N}^{2n+1}$ for the considered j is equivalent to the finite set of constraints:

$$\sum_{i=1}^{n} \vec{b_l}[i] \times (x_{(n+i)} - x_i) \quad \geq \quad x_j - x_0 \tag{9}$$

$$\sum_{i=1}^{n} \vec{c_l}[i] \times (x_{(n+i)} - x_i) \quad \geq \quad 0 \tag{10}$$

where l ranges over $[1, L]$ and $\vec{c_l}$ ranges over (C_l). Altogether, equation 3 and the reduced sets of constraints that arise from 4, 5, and 6 form a finite system $\mathcal{P}$ of linear homogeneous equations and inequalities on the x_i's, with integral coefficients. The implicit places of G are the solutions of $\mathcal{P}$ in $\mathbb{N}^{2n+1}$. Therefore, the extremal rays of the rational cone defined by $\mathcal{P}$ and inequalities $x_i \geq 0$ (see [22]) determine a *finite* set $P = \{\vec{p_1}, \ldots, \vec{p_m}\}$ of implicit places of G, generating all other places of G as linear combinations

$$\vec{p} = \sum_{j=1}^{m} q_j \times \vec{p_j} \tag{11}$$

with non-negative rational coefficients q_j. Chernikova's algorithm (see [22]) may be used to compute the set P, that we assume fixed from now on.

On account of equation 2, the companion map σ of a non-negative linear combination of places $\vec{p_j}$ is a non-negative linear combination of the respective companion maps σ_j. Therefore, vertices v and v' are separated by an implicit place of G if and only if they are separated by an implicit place in P. In view of relation 11, it follows similarly that action a is inhibited at vertex v by an implicit place of G if and only if a is inhibited at v by an implicit place in P. Therefore, graph G may be realized by some P/T-net if and only if it is realized by the P/T-net defined in the obvious way from set P. The goal of the next section is to make this decision effective.

5 The decision of the synthesis problem

Taking advantage of the rational structure $\mathcal{S} =< W, w_0, Eq, M_1, \ldots, M_n >$ of graph $G = (V, E, v_0, \lambda)$, we propose in this section procedures able to decide:
i) whether any two distinct vertices v, v' in the infinite set V are separated by some place $p \in P$, and
ii) whether, for every vertex v, each action $a \in A$ labelling no edge with source v is inhibited at v by some place $p \in P$.
We indicate also the way to remove redundant places, if any exist, from the finite set $P = \{\vec{p_1}, \ldots, \vec{p_m}\}$ computed in section 4.

5.1 Separating vertices

Recall from the definition of automatic graphs that Eq is an asynchronous automaton over A, such that $L(Eq) = \{(w, w') \in L(W) \times L(W) \mid v_0 \cdot w = v_0 \cdot w'\}$. We construct first another

asynchronous automaton Neq, such that $L(Neq) = \{(w, w') \in L(W) \times L(W) \mid v_0 \cdot w \neq v_0 \cdot w'\}$. In view of lemma 2.2, and seeing that W is a finite automaton, this can be done by synchronizing a two-tape automaton, accepting the complement of $L(Eq)$, separately on each tape with two independent copies of W.

Consider distinct vertices v and v'. Let w and w' be their respective representatives in $L(W)$, thus $(w, w') \in L(Neq)$. Let $p_j \in P$ be an implicit place of graph G. From relation 1 and equation 2, p_j separates v and v' if and only if

$$\sum_{i=1}^{n} (|w'|_i - |w|_i) \times (F(a_i, p_j) - F(p_j, a_i)) \neq 0$$

Therefore, if we define for all $j \in [1, m]$

$$Y_j = \{\vec{y} \in \mathbb{Z}^n \mid \sum_{i=1}^{n} \vec{y}[i] \times (F(a_i, p_j) - F(p_j, a_i)) \neq 0\}$$

then all pairs of distinct vertices are separated by places in P if and only if $\Delta(Neq) \subseteq \cup_{j=1}^{m} Y_j$. At this stage, let us recall two useful facts following from classical results: *i)* the solutions in $\mathbb{Z}^n$ of a linear equation with coefficients in $\mathbb{Z}$ form an effective semi-linear subset of $\mathbb{Z}^n$, *ii)* the semi-linear subsets of $\mathbb{Z}^n$ form an effective boolean algebra (see the appendix). From *(i)*, each set Y_j is the complement of a semi-linear subset of $\mathbb{Z}^n$. As $\Delta(Neq)$ is a semi-linear subset of $\mathbb{Z}^n$, the decision follows by *(ii)*.

5.2 Inhibiting actions at vertices

Recall from the definition of automatic graphs that for each $k \in [1, n]$, M_k is an asynchronous automaton over $A = \{a_1, \ldots, a_n\}$, such that $L(M_k) = \{(w, w') \in L(W) \times L(W) \mid v_0 \cdot wa_k = v_0 \cdot w'\}$. Fix a_k in A. The set of vertices which are sources of edges labelled with a_k is therefore

$$V_k = \{v_0 \cdot w \mid (\exists w')\ (w, w') \in L(M_k)\}$$

We must construct a regular set of representatives of the complementary set $V \setminus V_k$. This is pretty easy as $L(M_k)$ and $L(Eq)$ are rational relations: the left projection of $L(M_k)$ is a regular language, whose image under equalization is regular (see e.g. [5]). A regular language R_k containing all and only representatives of $V \setminus V_k$ may be obtained by computing the complement of this image in $L(W)$.

Consider now a fixed vertex $v \in V \setminus V_k$. Let $w \in R_k$ be some representative of v, and let $p_j \in P$ be an implicit place of graph G. From relation 1 and equation 2, place p_j inhibits action a_k at vertex v if and only if

$$m_0(p_j) + \sum_{i=1}^{n} (|w|_i) \times (F(a_i, p_j) - F(p_j, a_i)) < F(p_j, a_k)$$

As R_k is a regular language, its commutative image $\psi(R_k) = \{\psi(w) \mid w \in R_k\}$ is an effective semi-linear subset of $\mathbb{N}^n$. Likewise, if we define, for all $j \in [1, m]$

$$B_j^k = \{\vec{b} \in \mathbb{N}^n \mid \sum_{i=1}^{n} \vec{b}[i] \times (F(a_i, p_j) - F(p_j, a_i)) < F(p_j, a_k) - m_0(p_j)$$

it may be observed that the considered sets B_j^k are effective semi-linear subsets of $\mathbb{N}^n$ (see the appendix). Now, it should be clear that action a_k is indeed inhibited by implicit places of graph G at all vertices $v \in V \setminus V_k$ if and only if $\psi(R_k) \subseteq \cup_{j=1}^{m} B_j^k$. Since the semi-linear subsets of $\mathbb{N}^n$ form an effective boolean algebra, the decision follows.

5.3 Removing redundancy from synthesized nets

Adding up proposition 3.4, the effective construction of set P given in section 4, and the decision procedures proposed above, we have established the following.

Theorem 5.1 *One can decide whether an automatic graph G is isomorphic to the reachable state graph of some P/T-net and in this case, one can construct a P/T-net realizing G.*

Notice that some automatic graphs G may be realized by P/T-nets with sets of places strictly smaller than the set P computed in section 4. They may even be realized by subnets of the P/T-net synthesized from the considered set of places P. In order to remove redundant places from the synthesized net, it suffices obviously to search for minimal subsets $\{p_1, \ldots, p_l\}$ of P satisfying $\Delta(Neq) \subseteq \cup_{j=1}^{l} Y_j$ and $\psi(R_k) \subseteq \cup_{j=1}^{l} B_j^k$ for all $k \in [1, n]$. Since P is finite and all inclusions can be decided, this procedure is effective.

6 A detailed example

The infinite graph G depicted in Fig.2 is an automatic graph since it may be given the rational structure $\mathcal{S} =< W, w_0, Eq, M_a, M_b, M_c >$ which is elaborated hereafter.

1. $L(W) = (ab)^* a^* + (ab)^* b + (ab)^* bb$ and $w_0 = \varepsilon$, thus W is the deterministic automaton shown in Fig. 3.

2. $L(Eq) = \{(w, w) \mid w \in L(W)\}$, i.e. the diagonal of $L(W)$, hence the equalizer Eq may be derived from W by splitting transitions as shown in Fig. 4.

3. $L(M_a) = \{(w, w') \in L(W) \times L(W) \mid v_0 \cdot wa = v_0 \cdot w'\}$, therefore either $w = (ab)^n a^m$ and $w' = (ab)^n a^{m+1} = wa$, or $w = (ab)^n b$ and $w' = (ab)^{n+1}$, or $w = (ab)^n bb$ and $w' = (ab)^{n+1} b$. The multiplier M_a is shown in Fig. 5.

4. $L(M_b) = \{(w, w') \in L(W) \times L(W) \mid v_0 \cdot wb = v_0 \cdot w'\}$, hence either $w = (ab)^n a^m$ with $m \geq 1$ and $w' = (ab)^{n+1} a^{m-1}$, or $w = (ab)^n b^k$ with $k = 0, 1$ and $w' = (ab)^n b^{k+1} = wb$. The multiplier M_b is depicted in Fig. 6.

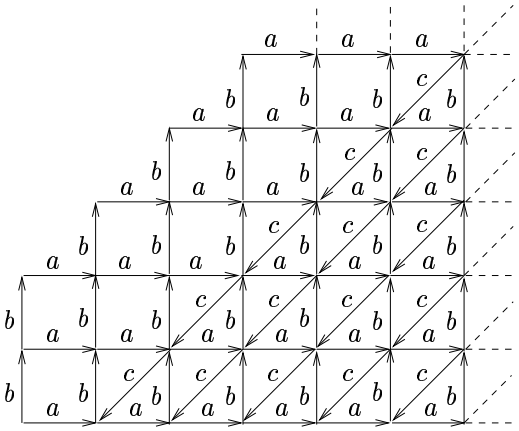

Figure 2: Automatic graph G

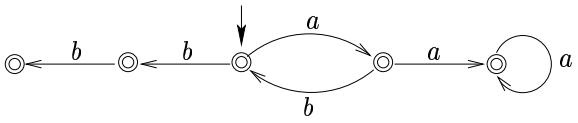

Figure 3: Automaton W

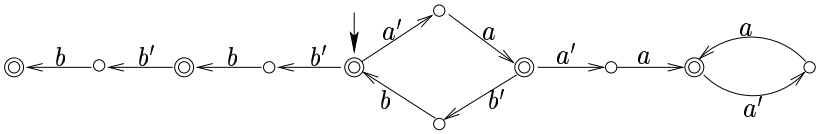

Figure 4: Equalizer Eq

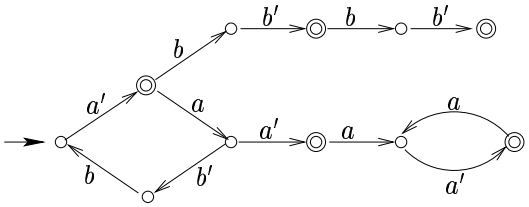

Figure 5: Multiplier M_a

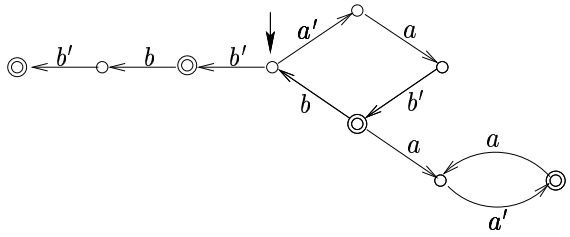

Figure 6: Multiplier M_b

5. $L(M_c) = \{(w, w') \in L(W) \times L(W) \mid v_0 \cdot wc = v_0 \cdot w'\}$, hence necessarily $w = (ab)^n a^m$ with $n, m \geq 1$ and $w' = (ab)^{n-1} a^m$. The multiplier M_c is shown in Fig. 7.

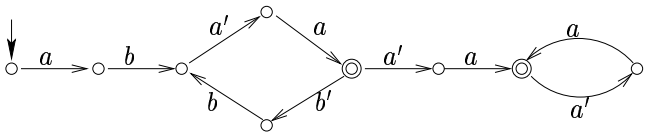

Figure 7: Multiplier M_c

Let us now compute the implicit places of graph G from the rational structure $\mathcal{S}$, following the instructions given in section 4. In order to alleviate the notation, let implicit places p be identified with the corresponding vectors $\vec{p} = \langle m_0, {}^\bullet a, {}^\bullet b, {}^\bullet c, a^\bullet, b^\bullet, c^\bullet \rangle$ such that $m_0 = m_0(p)$, ${}^\bullet x = F(p, x)$ and $x^\bullet = F(x, p)$. For convenience, let us also identify the vectors in $\mathbb{Z}^{\{a,b,c\}}$ with vectors in $\mathbb{Z}^3$ (with entries given in alphabetic order). As $w_0 = \varepsilon$, equation 3 is trivial, and similarly, all instances of equation 4 are trivial, seeing that $L(Eq)$ is the diagonal of $L(W)$ (therefore, $\Delta(Eq) = \{\psi(w') - \psi(w) \mid (w, w') \in L(Eq)\} = \{\langle 0, 0, 0 \rangle\}$). Regarding now equation 6, one obtains $\Delta(M_a) = \{\langle 1, 0, 0 \rangle\}$, $\Delta(M_b) = \{\langle 0, 1, 0 \rangle\}$, and $\Delta(M_c) = \{\langle -1, -1, 0 \rangle\}$, hence all instances of this equation are trivial for $j = 1$ or $j = 2$, and any instance of this equation for $j = 3$ yields the constraint

$$a^\bullet - {}^\bullet a + b^\bullet - {}^\bullet b + c^\bullet - {}^\bullet c = 0$$

It remains to deal with equation 5 and to produce thereof a finite set of instances of the inequality constraints 9 and 10.

$$\Sigma_1(M_a) = \langle 0, 0, 0 \rangle \{\langle 1, 0, 0 \rangle, \langle 1, 1, 0 \rangle\}^* \cup \langle 0, 1, 0 \rangle \{\langle 1, 1, 0 \rangle\}^* \cup \langle 0, 2, 0 \rangle \{\langle 1, 1, 0 \rangle\}^*$$

produces the constraints

$$\begin{array}{l} 0 \geq {}^\bullet a - m_0 \\ b^\bullet - {}^\bullet b \geq {}^\bullet a - m_0 \\ 2\,(b^\bullet - {}^\bullet b) \geq {}^\bullet a - m_0 \\ a^\bullet - {}^\bullet a \geq 0 \\ a^\bullet - {}^\bullet a + b^\bullet - {}^\bullet b \geq 0 \end{array}$$

$\Sigma_1\,(M_b) = \langle 0,0,0\rangle\,\{\langle 1,0,0\rangle, \langle 1,1,0\rangle\}^* \cup \langle 0,1,0\rangle\,\{\langle 1,1,0\rangle\}^*$ adds two constraints

$$\begin{array}{l} 0 \geq {}^\bullet b - m_0 \\ b^\bullet - {}^\bullet b \geq {}^\bullet b - m_0 \end{array}$$

$\Sigma_1\,(M_c) = \langle 2,1,0\rangle\,\{\langle 1,0,0\rangle, \langle 1,1,0\rangle\}^*$ brings one final constraint

$$2\,(a^\bullet - {}^\bullet a) + b^\bullet - {}^\bullet b \geq {}^\bullet c - m_0$$

Chernikova's algorithm, run with these linear homogeneous constraints at input, produces at output the extremal rays of the polyhedral cone defined by the constraints. There are 15 extremal rays, listed below as formal sums which may be read as implicit places of G, e.g. $ray_1 = 2 + a^\bullet +^\bullet b + 3.{}^\bullet c + 3.c^\bullet$ means the place $\vec{p_1} = \langle 2,0,1,3,1,0,3\rangle$. The list is as follows:

$$\begin{array}{lcl} ray_1 & = & 2 + a^\bullet +^\bullet b + 3\,{}^\bullet c + 3\,c^\bullet \\ ray_2 & = & 1 +^\bullet a + a^\bullet +^\bullet c + c^\bullet \\ ray_3 & = & 1 +^\bullet b + b^\bullet +^\bullet c + c^\bullet \\ ray_4 & = & 1 +^\bullet a + a^\bullet +^\bullet b + b^\bullet +^\bullet c + c^\bullet \\ ray_5 & = & a^\bullet +^\bullet c \\ ray_6 & = & 2 + a^\bullet + 2\,{}^\bullet b + b^\bullet + 3\,{}^\bullet c + 3\,c^\bullet \\ ray_7 & = & 1 +^\bullet c + c^\bullet \\ ray_8 & = & a^\bullet + 2\,{}^\bullet c + c^\bullet \\ ray_9 & = & b^\bullet +^\bullet c \\ ray_{10} & = & 2 + a^\bullet +^\bullet b \\ ray_{11} & = & 1 +^\bullet a + a^\bullet \\ ray_{12} & = & 1 +^\bullet a + a^\bullet +^\bullet b + b^\bullet \\ ray_{13} & = & 2 + a^\bullet + 2\,{}^\bullet b + b^\bullet \\ ray_{14} & = & 1 +^\bullet b + b^\bullet \\ ray_{15} & = & 1 \end{array}$$

We are ready to decide whether graph G is the reachable marking graph of some Petri net, following the instructions given in section 5. We show below that the two implicit places p_1 and p_9 defined by the respective rays ray_1 and ray_9 suffice to separate all states of G and to inhibit actions a, b, and c wherever necessary.

Separation of vertices.

For $j = 1$ or 9, the set variable $Y_j \subseteq \mathbb{Z}^3$ defined in section 5.1 evaluates to the respective subsets $Y_1 = \{\langle y_a, y_b, y_c\rangle \mid y_a \neq y_b\}$ and $Y_9 = \{\langle y_a, y_b, y_c\rangle \mid y_b \neq y_c\}$. Now $\Delta(Neq) = \mathbb{Z} \times \mathbb{Z} \times \{0\} \setminus \{\langle 0,0,0\rangle\}$, as $L(W) \subseteq \{a,b\}^*$ and because two words in $L(W)$ lead to the same vertex in G if and only if their commutative image is the same. We show that $\Delta(Neq) \subset Y_1 \cup Y_9$. To this effect, consider any vector $\vec{y} = \langle y_a, y_b, y_c\rangle$ in $\Delta(Neq)$, hence $y_c = 0$. If $y_b \neq y_c$ then $\vec{y} \in Y_9$, else $y_b = y_c = 0$, and necessarily $y_a \neq 0$ since otherwise $\vec{y} = \langle 0,0,0\rangle \notin \Delta(Neq)$. Thus finally $y_a \neq y_b$, showing that $\vec{y} \in Y_1$.

Inhibition of actions at vertices.

Regular languages $R_a, R_b, R_c \subseteq L(W)$ containing representatives of all vertices where the respective actions a, b, or c should be inhibited are the following: $R_a = \emptyset$, $R_b = (ab)^* bb$, $R_c = a^* + (ab)^*(\varepsilon + b + bb)$. The commutative images of these languages are respectively: $\psi(R_a) = \emptyset$, $\psi(R_b) = \{\langle n, n+2, 0\rangle \mid n \in \mathbb{N}\}$, $\psi(R_c) = \{\langle n, 0, 0\rangle \mid n \in \mathbb{N}\} \cup \{\langle n, n+p, 0\rangle \mid n \in \mathbb{N},\ p \in \{0,1,2\}\}$. In order to show that places p_1 and p_9 defined from ray_1 and ray_9 inhibit actions b, and c wherever necessary, it suffices to check that $\psi(R_b) \subseteq B_1^b$ and $\psi(R_c) \subseteq B_1^c \cup B_9^c$, where the sets $B_j^k \subseteq \mathbb{N}^3$ defined in section 5.2 are the following($k = 2$ for b, or $k = 3$ for c): $B_1^b = \{\langle x,y,z\rangle \mid x - y < -1\}$, $B_1^c = \{\langle x,y,z\rangle \mid x - y < 1\}$, $B_9^c = \{\langle x,y,z\rangle \mid y - z < 1\}$. Now $\psi(R_b) \subseteq B_1^b$ is obvious, and $\psi(R_c) \subseteq B_1^c \cup B_9^c$ since $\{\langle n,0,0\rangle \mid n \in \mathbb{N}\} \subseteq B_9^c$ and $\{\langle n, n+p, 0\rangle \mid n \in \mathbb{N},\ p \in \{0,1,2\}\} \subseteq B_1^c$.

Thus graph G must be isomophic to the reachable marking graph of the Petri net shown in Fig.8, with two places p_1 and p_9 as defined by rays ray_1 and ray_9.

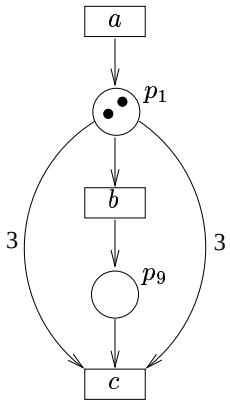

Figure 8: The synthetised Petri net

7 A counter-example

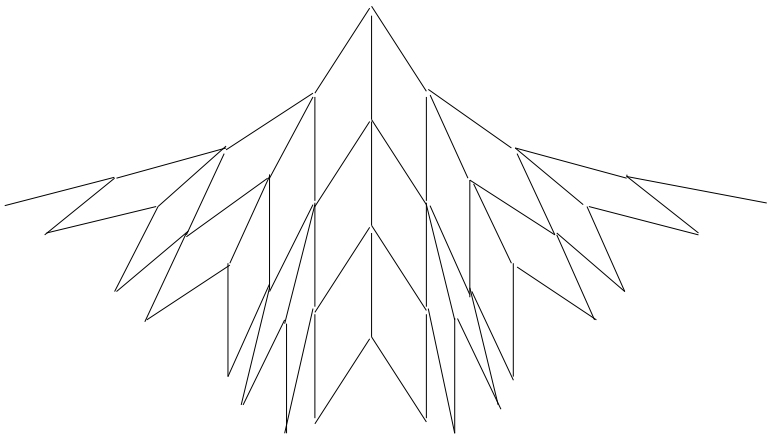

Figure 9: A graph which is not a Petri net graph

Consider the infinite graph depicted in Fig. 9, with edges going downwards. We claim that edges may be labelled in a concurrent alphabet (Σ, I) such that the labelled graph $G = (V, E, v_0, \lambda)$ represents a recognizable language of Mazurkiewicz traces $\mathcal{L}$, meaning that vertices in V are mapped bijectively to traces in $\mathcal{L}$, v_0 is mapped to the empty trace and an edge $(v, v') \in E$ labelled with $\sigma \in \Sigma$ is mapped to a pair of traces $([w], [w\sigma]) \in \mathcal{L} \times \mathcal{L}$. The claim follows from facts established in [4] and in [16] –namely, the graph given in Fig. 9 is isomorphic to the ordered set of traces of a finite and stable trace automaton, and any finite and stable trace automaton is effectively covered by a finite asynchronous transition system that generates the same coherent dI-domain.

We claim that any graph $G = (V, E, v_0, \lambda)$ induced as above from a recognizable trace language $\mathcal{L}$ (closed under prefix) is an automatic graph. Actually, the Foata normal forms of the traces in $\mathcal{L}$ are accepted by a finite automaton W over Σ, and for each $\sigma \in \Sigma$, the pairs of Foata normal forms of contiguous traces $[w], [w\sigma] \in \mathcal{L}$ are accepted by an asynchronous (in fact synchronous) automaton.

Now consider any labelling of the graph in Fig. 9, such that the labelled graph $G = (V, E, v_0, \lambda)$ may be seen as a recognizable trace language. The grid (Fig. 1) cannot be embedded in G. Nevertheless, G is not a pushdown graph, for cuts at constant depth yield infinitely many types of (infinite) connected subgraphs. Observe that the number of vertices at depth n in G grows faster than 2^n. Based on this observation, we claim that G cannot be isomorphic to the reachable state graph of any Petri net. The same claim applies to all graphs in which the number of vertices at distance n from the root grows exponentially with

n, and in particular to the complete binary tree (emblematic of pushdown graphs).

In order to establish this claim, we observe an immediate consequence of the fundamental equation of Petri net theory: given a Petri net with m transitions, the number of different markings that may be reached after firing n transitions from its initial marking is bounded by the number K_n^m of non-negative integral m-vectors with sum of entries equal to n. Clearly $K_n^1 = 1$, $K_1^m = m$, and for all $n, m \geq 2$, $K_n^m = K_n^{m-1} + K_{n-1}^m$. An easy induction shows that $K_n^m = C_{n+m-1}^n$. Thus, for fixed m, the number of markings that may be reached by firing n transitions has the polynomial order $\mathcal{O}(n^{m-1})$.

8 Automatic Specifications

We show in this final section that the effective solution to the net synthesis problem proposed in this paper can easily be extended to deal, instead of automatic *graphs*, with *families of graphs* given by automatic specifications.

Definition 8.1 *An* automatic specification *over alphabet* $A = \{a_1, \ldots, a_n\}$ *is a* rational structure $\mathcal{S} = < W, Eq, Dis, M_1, \ldots, M_n, I_1 \ldots, I_n >$ *as follows:*
i) *W is a finite (one-tape) automaton over A, accepting a non-empty prefix-closed (regular) language $L(W)$,*
ii) *Eq is an asynchronous automaton over A, such that $L(Eq) \subseteq L(W) \times L(W)$ and $L(Eq)$ is an equivalence relation on $L(W)$,*
iii) *Dis is an asynchronous automaton over A, such that $L(Dis) \subseteq L(W) \times L(W)$ and $L(Dis)$ does not intersect $L(Eq)$,*
iv) *for each $i \in [1, n]$, M_i is an asynchronous automaton over A, such that $(w, wa_i) \in L(M_i)$ whenever $wa_i \in L(W)$,*
v) *for each $i \in [1, n]$, I_i is a finite (one-tape) automaton over A, such that $w \notin L(I_i)$ whenever $wa_i \in L(W)$.*

Definition 8.2 *A graph $G = (V, E, v_0, \lambda)$ is a* model *of the specification $\mathcal{S}$ if the following conditions are satisfied for all $i \in [1, n]$:*

1. $(\forall w \in L(W))\,(\exists v \in V)\,(v = v_0 \cdot w)$ and $(\forall v \in V)\,(\exists w \in L(W))\,(v = v_0 \cdot w)$
2. $(w, w') \in L(Eq) \implies v_0 \cdot w = v_0 \cdot w'$
3. $(w, w') \in L(Dis) \implies v_0 \cdot w \neq v_0 \cdot w'$
4. $(w, w') \in L(M_i) \implies v_0 \cdot wa_i = v_0 \cdot w'$
5. $w \in L(I_i) \implies v_0 \cdot wa_i$ *is undefined.*

The net synthesis problem for automatic specifications is to decide whether some model of the specifications is the reachable marking graph of a Petri net, and to construct such a Petri net when it exists. The decision procedure presented in sections 4 and 5 can solve this

problem with minor modifications. A finite set P of implicit places common to *all* models of $\mathcal{S}$ is computed as previously in section 4 from the semi-linear sets $\Delta(Eq)$, $\Delta(M_i)$, and $\Sigma_1(M_i)$, whose definition does not change. In view of the positive conditions in Def. 8.2 (conditions 1,2 and 4), *any* implicit place of *any* model of $\mathcal{S}$ must be a non-negative linear combination of places in P. Therefore, some model of $\mathcal{S}$ is the reachable marking graph of a Petri net *if and only if* the reachable marking graph of the Petri net synthesized from set of places P is a model of $\mathcal{S}$. In view of the negative conditions in Def. 8.2 (conditions 3 and 5), deciding on this amounts to apply the two procedures from section 5 to *smaller* sets of instances $L(Dis) \subseteq L(Neq)$ and $L(I_i) \subseteq L(W) \setminus \pi_1(M_i)$. In case when $L(Dis) = (L(W) \times L(W)) \setminus L(Eq)$ and $L(I_i) = L(W) \setminus \pi_1(M_i)$ for all $i \in [1, n]$, the automatic specification $\mathcal{S}$ has *one and only one* model, and the net synthesis algorithm for automatic specifications coincides in this case with the net synthesis algorithm for automatic graphs.

Automatic specifications may sometimes be *inconsistent*, i.e. they may have *no* model at all. Rational presentations of families of graphs differ in that from rational presentations of graphs. Two conditions against inconsistency have been inserted in Def. 8.1 (namely $L(Dis) \cap L(Eq) = \emptyset$ and $L(I_i) \cap \pi_1(L(M_i)) = \emptyset$). These conditions are necessary but not sufficient to ensure consistency. It is not difficult to exhibit extra conditions that would reject all and only inconsistent specifications, but this has little interest since no effective procedure could decide on such conditions. Notice moreover that the effective procedure we have proposed for net synthesis does always terminate; just, it ends with no solution when it is applied to specifications that do not have any net model and in particular to inconsistent specifications.

We conclude the paper by elaborating on the following remark: the net synthesis procedure for automatic specifications depends alone on the commutative images of the sets $L(Eq)$, $L(Dis)$, $L(M_i)$, and $L(I_i)$. As we are interested exclusively in graphs likely to be marking graphs of Petri nets labelled injectively on transitions, the following condition may moreover be imposed on graphs.

Definition 8.3 *A graph $G = (V, E, v_0, \lambda)$ labelled in alphabet A satisfies the* Parikh condition *if, for all vertices v and v', and for every element w of $F(A)$ (the free group generated by A):*

$$[\, v \overset{w}{\rightsquigarrow} v' \ \wedge \psi(w) = 0 \,] \ \Rightarrow \ v = v'$$

where $\psi : F(A) \to \mathbb{Z}^A$ is the (unique) morphism of groups such that $\psi(a) = a$ for every $a \in A$, and $v \overset{w}{\rightsquigarrow} v'$ means that there exists a generalized path from v to v' in G, i.e. a sequence $e_1^{\sim} \ldots e_m^{\sim}$ of edges e_i or reversed edges e_i^{-1}, such that $\lambda(e_1^{\sim}) \ldots \lambda(e_m^{\sim}) = w$ in $F(A)$, letting $\lambda(e_i^{-1}) = -\lambda(e_i)$.

Restricting our attention to graphs that satisfy the Parikh condition, we can allow ourselves to abuse the notation by writing $v' = v \cdot \psi(w)$ in place of $v \overset{w}{\rightsquigarrow} v'$. In doing so, no ambiguity is introduced since the derived graph labelled in $\mathbb{Z}^A$ is deterministic. These remarks incite us to propose a variant definition for automatic specifications.

Definition 8.4 *An* automatic specification *over* $A = \{a_1, \ldots, a_n\}$ *is a structure* $\mathcal{S} = < W, Eq, Dis, M_1, \ldots, M_n, I_1 \ldots, I_n >$ *with components as follows:*
i) W *is a non-empty downwards closed semi-linear subset of* $\mathbb{N}^n$,
ii) Eq *is a semi-linear subset of* $\mathbb{Z}^n$,
iii) Dis *is a semi-linear subset of* $\mathbb{Z}^n$, *disjoint from* Eq,
iv) *for each* $i \in [1,n]$, M_i *is a semi-linear subset of* $W \times W$, *such that* $(w, w + a_i) \in M_i$ *whenever* $w \in W$ *and* $w + a_i \in W$,
v) *for each* $i \in [1,n]$, I_i *is a semi-linear subset of* W, *disjoint from* $\pi_1(M_i)$.

Definition 8.5 *A graph* $G = (V, E, v_0, \lambda)$ *labelled in* A *is a model of* $\mathcal{S}$ *if it satisfies the Parikh condition and the following are satisfied for all* $i \in [1,n]$*:*

1. $(\forall w \in W)\,(\exists v \in V)\,(v = v_0 \cdot w)$ and $(\forall v \in V)\,(\exists w \in W)\,(v = v_0 \cdot w)$
2. $w \in Eq \implies v_0 \cdot w = v_0$
3. $w \in Dis \implies v_0 \cdot w \neq v_0$
4. $(w, w') \in M_i \implies v_0 \cdot (w + a_i) = v_0 \cdot w'$
5. $w \in I_i \implies v_0 \cdot (w + a_i)$ *is undefined.*

Note that in conditions (2) and (3) the root v_0 could be replaced by any other vertex $v \in V$. Actually, if we let $H_1(G) = \{w \in \mathbb{Z}^A \mid v_0 \cdot w = v_0\}$, i.e. the homology group of the graph G, the conditions (2) and (3) read respectively as

$$Eq \subset H_1(G) \quad \text{and} \quad H_1(G) \cap Dis = \emptyset$$

Similarly, the conditions (4) and (5) read respectively as

$$\{w + a_i - w' \mid (w, w') \in M_i\} \subseteq H_1(G) \qquad \{w + a_i - w' \mid w \in I_i \ \wedge \ w' \in W\} \cap H_1(G) = \emptyset$$

Therefore, if we introduce now the two semi-linear sets

$$Equal = Eq \cup \{w + a_i - w' \mid (w, w') \in M_i\}$$
$$Distinct = Dis \cup \{w + a_i - w' \mid w \in I_i \ \wedge \ w' \in W\}$$

then the specification $\mathcal{S}$ is consistent if and only if $Equal \cap Distinct = \emptyset$, and this *can effectively be decided*. The models of $\mathcal{S}$, that must (by Def. 8.5) be graphs satisfying the Parikh condition, are in a bijective correspondence with the subgroups $H \subset \mathbb{Z}^A$ such that $Equal \subset H$ and $H \cap Distinct = \emptyset$. The correspondence sends each subgroup H to the full restriction of the Cayley graph $Cay(\mathbb{Z}^A/Equal, A)$ on the set of classes $\{Equal + w \mid w \in W\}$. Of course there is nothing new with this reformulation: the marking graph of a Petri net is a model of an automatic specification in the sense given by Def. 8.1 and Def. 8.2 if and only if it is a model, in the sense of Def. 8.5, of the automatic specification derived from the latter by taking the commutative images of all its components so as to conform to Def. 8.4. The net synthesis algorithm is not affected anyway.

References

[1] Badouel, E., Bernardinello, L., Darondeau, Ph.: Polynomial Algorithms for the Synthesis of Bounded Nets. Proc. CAAP, LNCS **915** (1995) 647-679.

[2] Badouel, E., Caillaud, B., Darondeau, Ph.: Distributing finite automata through Petri net synthesis. Formal Aspects of Computing **13** (2002) 447–470.

[3] Badouel, E., Darondeau, Ph.: Theory of regions. Lectures on Petri Nets I: Basic Models, LNCS **1491** (1998) 529-586.

[4] Badouel, E., Darondeau, Ph., Raoult, J.C.: Context-Free Event Domains are Recognizable. Information and Computation **149** (1999) 134-172.

[5] Berstel, J.: Transductions and Context-Free Languages. Teubner Verlag (1979).

[6] Blumensath, A., Grädel, E.: Automatic Structures. Proc. LICS, IEEE (2000) 51-62.

[7] Darondeau, Ph.: Deriving Petri Nets from Formal Languages. Proc. CONCUR, LNCS **1466** (1998) 533-548.

[8] Darondeau, Ph.: On the Petri net realization of context-free graphs. Theoretical Computer Science **258** (2001) 573-598.

[9] Droste, M., Shortt, R.M.: Petri Nets and Automata with Concurrency Relations - an Adjunction. In "Semantics of Programming Languages and Model Theory", M. Droste and Y. Gurevich eds (1993) 69-87.

[10] Epstein, D., Cannon, J., Holt, D., Levy, S., Paterson, M., Thurston, W.: Word Processing in Groups. Jones and Bartlett Publishers (1992).

[11] Ehrenfeucht, A., Rozenberg, G.: Partial (Set) 2-Structures; *Part I:* Basic Notions and the Representation Problem. Acta Informatica **27** (1990) 315-342.

[12] Ehrenfeucht, A., Rozenberg, G.: Partial (Set) 2-Structures; *Part II:* State Spaces of Concurrent Systems. Acta Informatica **27** (1990) 343-368.

[13] Eilenberg, S., Schützenberger, M.P.: Rational Sets in Commutative Monoids. Journal of Algebra **13** (1969) 173-191.

[14] Ginsburg, S., and Spanier, E.H.: Bounded Algol-like Languages. Trans. Amer. Math. Soc. (1964) 333-368.

[15] Ginsburg, S., and Spanier, E.H.: Semigroups, Presburger formulas, and Languages. Pacific Journal of Mathematics **16** (1966) 285-296.

[16] Husson, J.F., and Morin, R.: On Recognizable Stable Trace Languages. Proc. ETAPS, Springer-Verlag LNCS **1784** (2000) 177-191.

[17] Khoussainov, B., Nerode, A.: Automatic Presentations of Structures. Proc. LCC, Springer-Verlag LNCS **960** (1994) 367-392.

[18] Morvan, Ch.: Les graphes rationnels. PhD thesis, University of Rennes I (2002).

[19] Mukund, M.: Petri Nets and Step Transition Systems. International Journal of Foundations of Computer Science, **3**, 4 (1992) 443-478.

[20] Pélecq, L.: Isomorphismes et automorphismes des graphes context-free, équationnels et automatiques. PhD thesis, University of Bordeaux I (1997).

[21] Reutenauer, Ch.: Aspects mathématiques des réseaux de Petri. Masson (1979). English translation by I. Craig: The Mathematics of Petri Nets. Prentice-Hall (1990).

[22] Schrijver, A.: Theory of Linear and Integer Programming. John Wiley (1986).

[23] Sénizergues, G.: Definability in weak monadic second order logic of some infinite graphs. Dagstuhl seminar on Automata theory: Infinite computations, Wadern **28** (1992) 16.

9 Appendix

Ginsburg and Spanier proved in [14] that semi-linear subsets of $\mathbb{N}^n$ form an effective boolean algebra: intersections and complements of semi-linear subsets are semi-linear subsets, and they may be computed. The same authors gave in [15] an effective correspondence between semi-linear subsets and Presburger subsets, i.e. subsets of $\mathbb{N}^n$ definable in Presburger's arithmetic. The algebraic results established in [14] were extended soon after by Eilenberg and Schützenberger, who proved that semi-linear subsets form a boolean algebra in any finitely generated commutative monoid [13]. This covers the case of $\mathbb{Z}^n$, but the constructions given in [13] were not immediately effective.

An effective construction of the intersection of semi-linear subsets of $\mathbb{Z}^n$ is given in [21], relying on a lemma from [14] strengthened in [13]. The import of the lemma is as follows: given a homomorphism of monoids $\phi : \mathbb{N}^n \to \mathbb{Z}^m$ and a vector $\vec{z} \in \mathbb{Z}^m$, the inverse image $\phi^{-1}(\vec{z})$ of $\vec{z}$ under ϕ is an effective semi-linear subset of $\mathbb{N}^n$. From this lemma follows an important corollary: the set of solutions in $\mathbb{N}^n$ of a system of linear inequalities with coefficients in $\mathbb{Z}$ is semi-linear and it may be computed [21]. As semi-linear sets are closed under intersection, this may be shown by a simple observation: whenever, for some vector of natural numbers $\vec{k} =< k_1, \ldots, k_p >$, a linear form $\vec{y} \cdot \vec{k} = y_1 k_1 + \ldots + y_p k_p$ takes a positive value l greater than all integers y_i, vector $\vec{k}$ is the sum of two smaller vectors $\vec{k'}$ and $\vec{k''}$ such that $\vec{y} \cdot \vec{k'} = l' \geq 0$, $\vec{y} \cdot \vec{k''} = l'' \geq 0$, and $l = l' + l''$. As a direct consequence, solutions in $\mathbb{Z}^n$ of a system of linear inequalities with integral coefficients form an effective semi-linear subset of $\mathbb{Z}^n$: solutions with a fixed sign for each coordinate form indeed a semi-linear set.

A similar decomposition may be used to show that complementation is an effective operation

on semi-linear subsets of $\mathbb{Z}^n$. For each sign vector $\vec{\dagger} \in \{-1,+1\}^n$, for any subset Y of $\mathbb{Z}^n$, and for any vector $\vec{y}$ in $\mathbb{Z}^n$, define

$$Y \oslash \vec{\dagger} = \{\vec{y} \in Y \mid (\forall i)\ \vec{y}[i]\,\vec{\dagger}[i] \geq 0\} \quad (12)$$

$$\vec{y} \otimes \vec{\dagger} = < \vec{y}[1]\,\vec{\dagger}[1], \ldots, \vec{y}[n]\,\vec{\dagger}[n] > \quad (13)$$

$$Y \otimes \vec{\dagger} = \{\vec{y} \otimes \vec{\dagger} \mid \vec{y} \in Y\} \quad (14)$$

Thus, Y is a finite union of subsets $Y \oslash \vec{\dagger} = ((Y \oslash \vec{\dagger}) \otimes \vec{\dagger}) \otimes \vec{\dagger}$. By de Morgan's law, the complement of Y in $\mathbb{Z}^n$ is the intersection of sets $(\mathbb{N}^n \setminus ((Y \oslash \vec{\dagger}) \otimes \vec{\dagger})) \otimes \vec{\dagger}$. If Y is a semi-linear subset of $\mathbb{Z}^n$, each set $(Y \oslash \vec{\dagger}) \otimes \vec{\dagger}$ is a semi-linear subset of $\mathbb{N}^n$, hence its complement in $\mathbb{N}^n$ is effective. It follows therefrom that $\mathbb{Z}^n \setminus Y$ is an effective semi-linear subset of $\mathbb{Z}^n$ *q.e.d.*

Contents

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399