Acta Informatica 33, 297-315 (1996)

The synthesis problem of Petri nets

Jörg Desel, Wolfgang Reisig

Institut für Informatik der Humboldt-Universität zu Berlin, Unter den Linden 6, D-10099 Berlin, Germany

Received October 5, 1992 / April 11, 1995

Abstract. The synthesis problem of concurrent systems is the problem of synthesizing a concurrent system model from sequential observations. The paper studies the synthesis problem for elementary Petri nets and transition systems. A characterization of the class of transition systems which correspond to elementary Petri nets is proven. It is shown how to generate *all* elementary Petri nets corresponding to a given transition system. If there is any such elementary Petri net, it is proven that there always exists a small one which has only polynomially many elements in the size of the transition system.

Introduction

The synthesis problem of concurrent system models is the problem of synthesizing a concurrent system model from sequential observations. This problem has been attacked for various different formalisms, including parallel programs [Lengauer, Hehner 82], COSY-expressions [Janicki 85] and Petri Nets [Krieg 77], [Ehrenfeucht, Rozenberg 90], [Nielsen, Rozenberg, Thiagarajan 92], [Mukund 92], [Bernadinello 93].

The sequential observations of a concurrent system are usually given as a transition system, i.e. a directed, arc-labelled graph $\mathcal{S}$ with a distinguished "initial" node. Nodes and arcs of $\mathcal{S}$ denote global system states and transitions, respectively. The initial node represents the initial state. Any path of $\mathcal{S}$ starting with the initial node represents a sequential observation of a system run.

In the setting of this paper, the synthesis problem for a transition system $\mathcal{S}$ is the problem of constructing an elementary Petri net such that the state graph of this net is isomorphic to $\mathcal{S}$. Each such net is called a solution of the synthesis problem. For this setting, a basic solution of the synthesis problem has been suggested by [Ehrenfeucht, Rozenberg 90], employing the theory of partial 2-structures. Necessary and sufficient conditions are given to decide whether to a given transition system there exists an elementary Petri net which solves the synthesis problem. For each accepted transition system, a corresponding Petri net is constructed.

Based on this work, six conditions are given in [Nielsen, Rozenberg, Thiagarajan 92] which characterize the transition systems with solvable synthesis problem. This paper employs a categorical framework and also establishes relations between behaviour preserving mappings of elementary transition systems and, respectively, elementary Petri nets.

Instead of proving the existence of a solution before constructing it, we suggest to construct a canonical candidate Petri net Σ and then to check whether Σ in fact is a solution. If not, we show that there exists no solution at all. The existence checks of [Ehrenfeucht, Rozenberg 90] and [Nielsen, Rozenberg, Thiagarajan 92] are hence replaced by a simple isomorphism check between transition systems.

The elementary Petri net constructed in [Ehrenfeucht, Rozenberg 90] and [Nielsen, Rozenberg, Thiagarajan 92] is a canonical solution, satisfying a maximality property w.r.t. its conditions. However, this solution is in general exponential in the size of the given transition system. We show how to construct *all* solutions and in particular small ones, which are polynomial in the size of the transition system.

Sections 1 to 3 provide the (few) standard notations on graphs and elementary net systems, to be employed in the sequel. We adopt the core idea of "regions" from [Ehrenfeucht, Rozenberg 90] in Sect. 4. Section 5 gives the basic solution to the synthesis problem; we provide an algorithm to decide if there exists a solution. This result could have been proven using the theory developed in [Nielsen, Rozenberg, Thiagarajan 92]. Instead, we present an independent direct proof. We consider *all* solutions of the synthesis problem in Sect. 6. As one central result of this section, we show that each elementary Petri net which solves the synthesis problem for a given graph $\mathcal{G}$ is isomorphic to a Petri net constructed from an "admissible" set of regions of $\mathcal{G}$. Moreover, admissibility of sets of regions is characterized directly, employing the structure of the graph $\mathcal{G}$. Based on this results, Sect. 7 provides rules for identifying "redundant" regions. Small solutions of the synthesis problem can then be obtained by constructing small admissible sets without redundant regions.

1 Directed, arc-labelled, initialized graphs

Sequential system observation, consisting in global state occurrences and state transitions is conveniently described by help of *transition systems*. A transition system can formally be represented as a graph. Its nodes and its directed arcs represent states and state transitions, respectively. As different state transitions may be caused by equal events, different arcs may be labelled by equal symbols. The initial state is identified by a distinguished "initial" node.

1.1 Definition. *Let K and L be finite disjoint sets, let $G \subseteq K \times L \times K$, and let $k_0 \in K$. Then $\mathcal{G} = (K, L, G, k_0)$ is a* **finite, directed, arc-labelled, initialized graph.**
K and L are the sets of **nodes** *and* **labels** *of $\mathcal{G}$, respectively. Each $(h, l, k) \in G$ is an* **arc** *of $\mathcal{G}$, labelled by l and leading from h to k. The node k_0 is the* **initial node** *of $\mathcal{G}$.*

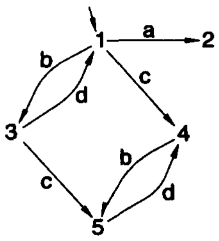
It is often assumed that every node of an initialized graph can be reached from the initial node:

1.2 Definition. *Let $\mathcal{G} = (K, L, G, k_0)$ be a finite, directed, arc-labelled, initialized graph. A node k is* **reachable** *iff $k = k_0$ or there exists a sequence of arcs $(k_0, l_1, k_1), (k_1, l_2, k_2), \ldots, (k_{n-1}, l_n, k_n)$ such that $k = k_n$.*

1.3 Convention. *In this paper, the term "graph" always denotes a finite, directed, arc-labelled, initialized graph with every node being reachable.*

We employ the usual graphical conventions for graphs, indicating the initial node by an extra arc without source and label.

1.4 Example.



This graph has nodes 1, . . . , 5 and labels a, . . . , d. Its initial node is 1.

Isomorphism of graphs is defined as follows:

1.5 Definition. *Let $\mathcal{G} = (K, L, G, k_0)$ and $\mathcal{G}' = (K', L', G', k_0')$ be graphs.*
A bijection $f: K \to K'$ is an **isomorphism** *from $\mathcal{G}$ to $\mathcal{G}'$ (written $f: \mathcal{G} \to \mathcal{G}'$) iff*

$f(k_0) = k_0'$ *and*
$(h, l, k) \in G$ *iff* $(f(h), l, f(k)) \in G'$.

We call two graphs $\mathcal{G}$ and $\mathcal{G}'$ isomorphic *(written $\mathcal{G} \simeq \mathcal{G}'$) iff there exists an isomorphism $f: \mathcal{G} \to \mathcal{G}'$.*

Note that we do not employ a bijection from L to L' but rather demand that the set of labels of L appearing at arcs of $\mathcal{G}$ coincides with the set of labels of L' appearing at arcs of $\mathcal{G}'$.

An immediate consequence of Definition 1.6 is that $\simeq$ is an equivalence relation on graphs.

2 Elementary net systems

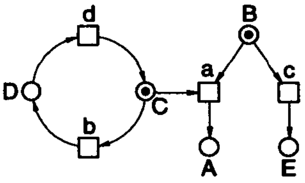
We recall the basic definition of the fundamental class of Petri nets, usually called *elementary net systems* ([Thiagarajan 87]). We only consider elementary net systems with finitely many conditions and events.

2.1 Definition. *Let B and E be disjoint finite sets, let $F \subseteq (B \times E) \cup (E \times B)$, and let $c_0 \subseteq B$. Then $\Sigma = (B, E, F, c_0)$ is called* **elementary net system** *(en-system, for short). The elements of B and E are called* **conditions** *and* **events**, *respectively. F is the flow relation and c_0 is the initial state.*

2.2 Notation. *Let $\Sigma = (B, E, F, c_0)$ be an en-system. For $x \in B \cup E$, we denote the pre-set $\{y \mid (y, x) \in F\}$ of x by ${}^\bullet x$ and the post-set $\{y \mid (x, y) \in F\}$ of x by $x^\bullet$.*

We employ the usual graphical representation for en-systems: Circles denote conditions and squares denote events. The flow relation is represented by arrows. The elements of the initial state are distinguished by a *token* (a dot) in each corresponding circle.

2.3 Example.



This en-system has five conditions A, . . . , E, and four events a, . . . , d. Its initial state is {B, C}. $^\bullet a = \{B, C\}$ is the pre-set of a and $C^\bullet = \{a, b\}$ is the post-set of C.

Every state c of an en-system is determined by a subset of its conditions. A condition is said to hold at the state c if and only if it is an element of c. An event is enabled at c if all its pre-conditions (i.e. conditions in its pre-set) hold at c and moreover none of its post-conditions holds at c. The occurrence of an enabled event removes its pre-conditions from c and adds the post-conditions to c. Formally, the dynamics of an en-system is described by its *state transitions*:

2.4 Definition. *Let $\Sigma = (B, E, F, c_0)$ be an en-system, let $c, d \subseteq B$, and let $e \in E$.*

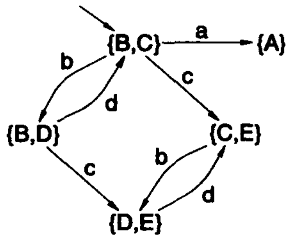
i. *c **enables** e iff $^\bullet e \subseteq c$ and $e^\bullet \cap c = \emptyset$.*
ii. *(c, e, d) is a **state transition** (written $c \xrightarrow{e} d$) iff c enables e and $d = (c \setminus {}^\bullet e) \cup e^\bullet$.*
iii. *A set $c \subseteq B$ is a **reachable state** iff either $c = c_0$ or there exists a sequence of state transitions $c_0 \xrightarrow{e_1} c_1 \xrightarrow{e_2} \cdots \xrightarrow{e_r} c_r$ such that $c_r = c$.*
iv. *A state transition $c \xrightarrow{e} d$ is **reachable** iff c is a reachable state.*

2.5 Example. The initial state {B, C} in 2.3 enables the events a, b and c. The corresponding state transitions are: $\{B, C\} \xrightarrow{a} \{A\}$, $\{B, C\} \xrightarrow{b} \{B, D\}$ and $\{B, C\} \xrightarrow{c} \{C, E\}$. {D, E} is another reachable state and $\{D, E\} \xrightarrow{d} \{C, E\}$ is another reachable state transition.

Starting with the initial state c_0, sequences $c_0 \xrightarrow{e_1} c_1 \xrightarrow{e_2} c_2 \xrightarrow{e_3} \cdots$ of state transitions form the sequential behaviour of an en-system Σ. Since we only consider en-systems with finite sets of conditions, the set of states and in particular the set of reachable states of an en-system is also finite. Hence the sequential behaviour can finitely be represented as the paths in a graph, the *state graph* of Σ.

2.6 Definition. *Let $\Sigma = (B, E, F, c_0)$ be an en-system, let C be the set of reachable states, and let G be the set of reachable state transitions of Σ. Then the graph $sg(\Sigma) = (C, E, G, c_0)$ is the **state graph of** Σ.*

2.7 Example.



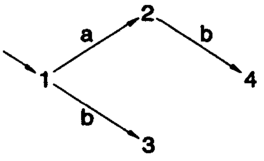
The state graph of the en-system given in 2.3

An *abstract state graph* is a "potential" state graph, where the identity of the involved states is unknown:

2.8 Definition. *A graph $\mathcal{G}$ is an* **abstract state graph** *iff there exists an en-system Σ such that $\mathcal{G} \simeq sg(\Sigma)$.*

2.9 Examples.

i. The state graph of the en-system in 2.3, shown in 2.7, is isomorphic to the graph in 1.4. Hence 1.4 shows an abstract state graph.
ii. There exists no en-system with a state graph isomorphic to (this will later be



proven). Hence this graph is not an abstract state graph.

The central issue of this paper can now be stated as follows:

The synthesis problem:

Is a given graph $\mathcal{G}$ an abstract state graph?
If yes, construct some ("small") en-system Σ satisfying $\mathcal{G} \simeq sg(\Sigma)$.

3 Isomorphism and reduction of en-systems

Let Σ and Σ' be en-systems which coincide up to the names of conditions. Then the behaviours of Σ and Σ' are closely related; their state graphs are isomorphic. If Σ is a solution to the synthesis problem for a given graph $\mathcal{G}$, then so is Σ'.

3.1 Definition. *Let $\Sigma = (B, E, F, c_0)$ and $\Sigma' = (B', E, F', c'_0)$ be en-systems with identical sets of events. A bijection $f: B \rightarrow B'$ is an* **isomorphism from Σ to Σ'** *iff for every condition $b \in B$ and every event $e \in E$ holds:*

$(b, e) \in F$ iff $(f(b), e) \in F'$,
$(e, b) \in F$ iff $(e, f(b)) \in F'$ and
$b \in c_0$ iff $f(b) \in c'_0$.

Σ and Σ' are **isomorphic** *iff there exists an isomorphism from Σ to Σ'.*

Usually, two en-systems are considered isomorphic if the sets of conditions and the sets of events are related by suitable bijections. The definition used here is the restricted case that the bijection between events is the identity mapping.

3.2 Proposition. *If Σ and Σ' are isomorphic en-systems then $sg(\Sigma) \simeq sg(\Sigma')$.*

Proof. Let $\Sigma = (B, E, F, c_0)$ and $\Sigma' = (B', E, F', c'_0)$ and let f be an isomorphism from Σ to Σ'. We extend f to sets of conditions c in the canonical way, i.e., for $c \subseteq B$, $f(c)$ denotes the set $\{f(b) \in B' \mid b \in c\}$.

(i) $f(c_0) = c'_0$, by Definition 3.1.
(ii) Let e be an event of Σ. Then, by Definition 3.1, a condition b is in the pre-set of e in Σ if and only if $f(b)$ is in the pre-set of e in Σ'. The same holds for the respective post-sets.
(iii) $c \xrightarrow{e} d$ is a state transition of Σ iff $f(c) \xrightarrow{e} f(d)$ is a state transition of Σ', by the occurrence rule and (ii).
(iv) A set c of conditions of Σ is a reachable state of Σ iff $f(c)$ is reachable in Σ'. This follows inductively from (i) and (iii).

This way f induces a bijection between the set of reachable states of Σ and the set of reachable states of Σ'. It remains to be shown that this bijection is an isomorphism from $sg(\Sigma)$ to $sg(\Sigma')$:

$f(c_0) = c'_0$ by (i). $c \xrightarrow{e} d$ is a reachable state transition of Σ iff $f(c) \xrightarrow{e} f(d)$ is reachable in Σ', by (iii) and (iv). □

Events not occurring in any state transition do not affect the behaviour of an en-system. In an en-system without "useless events", two conditions which agree on all reachable states have identical pre-sets, post-sets and initial tokens. Hence they can be interpreted as identical conditions. Since we will be interested in small en-systems, we define *reduced* en-systems without such conditions.

3.3 Definition. *An en-system is called* **reduced** *iff*

every event occurs in a reachable state transition and
for each two distinct conditions b and b' there exists a reachable state c satisfying either $b \in c$ and $b' \notin c$, or $b' \in c$ and $b \notin c$.

All en-systems used in examples of this paper are reduced.

3.4 Proposition. *For every en-system Σ there exists a reduced en-system Σ' such that $sg(\Sigma) \simeq sg(\Sigma')$.*

Proof. An event of Σ which does not occur in any reachable state transition does not contribute to the state graph of Σ. Hence assume w.l.o.g. that Σ has no such events.

Let b, b' be distinct conditions of Σ which agree on all reachable states. Let $e \in {}^\bullet b$. By assumption, there exists a reachable state transition $c \xrightarrow{e} d$. Then $b \notin c$

and $b \in d$. Since b and b' agree on all reachable states, $b' \notin c$, $b' \in d$ and therefore $e \in {}^\bullet b'$. Hence ${}^\bullet b \subseteq {}^\bullet b'$. By symmetry, ${}^\bullet b' \subseteq {}^\bullet b$ and so ${}^\bullet b = {}^\bullet b'$. It is shown similarly that $b^\bullet = b'^\bullet$.

Let $\overline{\Sigma}$ be the en-system obtained from Σ by removing b', its adjacent arcs, and possibly its initial token. Since b and b' have identical pre- and post-sets and agree on all reachable states, an event e is enabled by a reachable state c of Σ iff it is enabled by $c \setminus \{b'\}$ in $\overline{\Sigma}$. Moreover, e leads to a state c' in Σ iff e leads to $c \setminus \{b'\}$ in $\overline{\Sigma}$. Hence there is an isomorphism $f: sg(\Sigma) \rightarrow sg(\overline{\Sigma})$ defined by $f(c) = c \setminus \{b'\}$ for every reachable state c of Σ.

As Σ is finite, exhaustive repetition of this transformation finally yields a reduced en-system Σ' satisfying $sg(\Sigma) \simeq sg(\Sigma')$. □

4 Regions

The notion of "regions", introduced in [Ehrenfeucht, Rozenberg 90], is one of the essential concepts of this paper. Roughly speaking, a region R of a graph is a set of nodes corresponding to an existing or a potential condition b of an associated en-system: R is the set of the reachable states containing b. More concretely, every condition b of an en-system Σ can be assigned the set R_b of nodes of $sg(\Sigma)$ defined by $c \in R_b$ if and only if $b \in c$. As an event e of Σ is related to b in exactly one of four manners (1. $b \in {}^\bullet e$ and $b \notin e^\bullet$, 2. $b \notin {}^\bullet e$ and $b \in e^\bullet$, 3. $b \notin {}^\bullet e$ and $b \notin e^\bullet$ or 4. $b \in {}^\bullet e$ and $b \in e^\bullet$), the set R_b relates to e by one of the following alternatives: In case 1, all source nodes and no target node of e-labelled arcs belong to R_b. In case 2, no source node, but all target nodes of e-labelled arcs belong to R_b. In case 3, each e-labelled arc has either both its source and target node in R_b, or none of them. Finally, in the 4th case, e belongs to no state transition of Σ since, in every state of Σ, b is either an unmarked pre-condition or a marked post-condition of e. Hence, in this case there exists no e-labelled arc in $\mathcal{G}$.

Formulated in terms of state graphs, each set R_b of nodes which is induced by a condition b meets the above properties. Those properties can be formulated not only for state graphs but for arbitrary graphs. So we define:

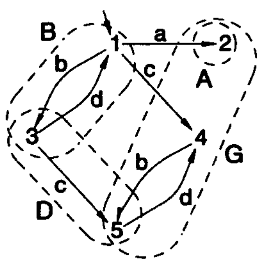
4.1 Definition. *Let $\mathcal{G}$ be a graph. A set R of nodes of $\mathcal{G}$ is a* **region** *of $\mathcal{G}$ iff for equally labelled arcs (h, l, k) and (h', l, k') holds:*

if $h \in R$ and $k \notin R$ then $h' \in R$ and $k' \notin R$, and
if $h \notin R$ and $k \in R$ then $h' \notin R$ and $k' \in R$.

$\emptyset$ and K are called **trivial regions** *of $\mathcal{G}$. $reg(\mathcal{G})$ denotes the set of all regions of $\mathcal{G}$.*

4.2 Examples.

i. The graph in 1.4 has 10 nontrivial regions:
A={2}, B={1,3}, C={1,4}, D={3,5}, E={4,5}, F={1,3,4,5}, G={2,4,5}, H={2,3,5}, I={1,2,4} and J={1,2,3}.
Some of them are outlined in the following figure:



ii. The graph in 2.9.ii has 4 nontrivial regions:
A={1,2}, B={1,3}, C={2,4} and D={3,4}.

The essential of a region is the uniform treatment of equally labelled arcs: Given a region R and an arc label l, either all l-labelled arcs "enter" R or all l-labelled arcs "leave" R, or l does not touch R. This motivates the following definition.

4.3 Definition. *Let l be a label of a graph $\mathcal{G}$. The* **pre-set** *${}^{\circ}l$ and the* **post-set** *l° are the sets of regions of $\mathcal{G}$ satisfying*

$$R \in {}^{\circ}l \text{ iff for all arcs } (h,l,k) \text{ of } \mathcal{G}, h \in R \text{ and } k \notin R, \text{ and}$$
$$R \in l^{\circ} \text{ iff for all arcs } (h,l,k) \text{ of } \mathcal{G}, h \notin R \text{ and } k \in R.$$

4.4 Example. For the graph in 1.4 and the notions of 4.2.i we have ${}^{\circ}\mathrm{c} = \{\mathrm{B}, \mathrm{J}\}$ and $\mathrm{c}^{\circ} = \{\mathrm{E}, \mathrm{G}\}$.

The following propositions state that the conditions of an en-system and the regions of its state graph are tightly related.

4.5 Proposition. *Let Σ be an en-system and let b be a condition of Σ. Let R_b be the set of reachable states c of Σ satisfying $b \in c$. Then R_b is a region of $sg(\Sigma)$.*

Proof. Let (c, e, d) and (c', e, d') be arcs of $sg(\Sigma)$.
Assume $c \in R_b$ and $d \notin R_b$. Then $c \xrightarrow{e} d$ and $c' \xrightarrow{e} d'$ are reachable state transitions of Σ and $b \in c$, $b \notin d$. Therefore $b \in {}^{\bullet}e$ and $b \notin e^{\bullet}$. Hence $b \in c'$, because c' enables e. By the definition of state transition, $b \notin d'$. So $c' \in R_b$ and $d' \notin R_b$.
It is shown similarly that $c \notin R_b$ and $d \in R_b$ implies $c' \notin R_b$ and $d' \in R_b$. □

Conversely, a region does not necessarily correspond to an existing condition but to a "potential" condition. Adding this condition to the en-system preserves behaviour, i.e., the augmented en-system has an isomorphic state graph.

4.6 Definition. *Let $\Sigma = (B, E, F, c_0)$ be an en-system and let R be a region of $sg(\Sigma)$. Let b_R be a new condition, $b_R \notin B \cup E$. Then the en-system $\Sigma^{+R} = (B', E', F', c_0')$ is defined by*

$$B' = B \cup \{b_R\},$$
$$E' = E,$$
$$F' = F \cup \{(b_R, e) \mid R \in {}^{\circ}e\} \cup \{(e, b_R) \mid R \in e^{\circ}\},$$
$$c_0' = c_0 \cup \{b_R\} \text{ if } c_0 \in R \text{ and } c_0' = c_0 \text{ if } c_0 \notin R.$$

4.7 Proposition. *Let Σ be an en-system and let R be a region of $sg(\Sigma)$. Then $sg(\Sigma) \simeq sg(\Sigma^{+R})$.*

Proof. Let $\Sigma = (B, E, F, c_0)$ and let $\Sigma^{+R} = (B \cup \{b_R\}, E, F', c'_0)$. Let C be the set of reachable states of Σ and let C' be the set of reachable states of Σ^{+R}. Define, for each $c \in C$, $f(c) = c \cup \{b_R\}$ if $c \in R$ and $f(c) = c$ if $c \notin R$. We show that f is an isomorphism from $sg(\Sigma)$ to $sg(\Sigma^{+R})$.

(i) $f(c_0) = c'_0$ holds by the definition of f and c'_0.
(ii) $c \in C$ enables an event e in Σ iff $f(c)$ enables e in Σ^{+R}.
Assume c enables e and let $c \xrightarrow{e} d$. If, in Σ^{+R}, $b_R \in {}^\bullet e$, then $R \in {}^\circ e$ and hence $c \in R$ and $b_R \in f(c)$. If, in Σ^{+R}, $b_R \in e^\bullet$, then $R \in e^\circ$ and hence $c \notin R$ and $b_R \notin f(c)$. Therefore $f(c)$ enables e in Σ^{+R}. If $f(c)$ enables e in Σ^{+R}, then c enables e in Σ, by definition of f and since ${}^\bullet e$ in Σ is a subset of ${}^\bullet e$ in Σ^{+R} and $e^\bullet$ in Σ is a subset of $e^\bullet$ in Σ^{+R}.
(iii) Let $c \in C$.
$c \xrightarrow{e} d$ is a state transition of Σ iff $f(c) \xrightarrow{e} f(d)$ is a state transition of Σ^{+R}.
Assume c enables e in Σ and – which is by (ii) equivalent – $f(c)$ enables e in Σ'.
Let $c \xrightarrow{e} d$ and $f(c) \xrightarrow{e} d'$ be the respective state transitions. For proving $d' = f(d)$ it suffices to show that $b_R \in d'$ iff $b_R \in f(d)$:

$$\begin{aligned} b_R \in d' \quad & \text{iff} \quad \text{either } b_R \in f(c) \text{ and } b_R \notin {}^\bullet e \text{ or } b_R \in e^\bullet \\ & \text{iff} \quad \text{either } c \in R \text{ and } R \notin {}^\circ e \text{ or } R \in e^\circ \\ & \text{iff} \quad d \in R \\ & \text{iff} \quad b_R \in f(d). \end{aligned}$$

(iv) f is a bijection from C to C', by induction using (i) and (iii).
(v) $c \xrightarrow{e} d$ is a reachable state transition of Σ iff $f(c) \xrightarrow{e} f(d)$ is a reachable state transition of Σ^{+R}, by (iii) and (iv).

By definition, the sets of nodes of $sg(\Sigma)$ and $sg(\Sigma^{+R})$ are C and C', and the sets of arcs are the reachable state transitions of Σ and Σ^{+R}, respectively. By (iv), (i) and (v), f is an isomorphism from $sg(\Sigma)$ to $sg(\Sigma^{+R})$. □

5 The basic solution of the synthesis problem

In this section, we develop a procedure to decide whether or not a given graph $\mathcal{G}$ is an abstract state graph. In the positive case, the procedure provides an en-system with a state graph isomorphic to $\mathcal{G}$.

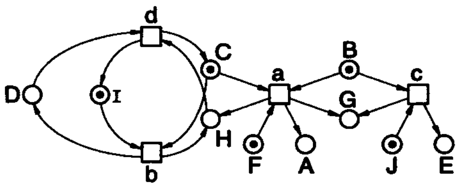
Since every condition corresponds to a region and every region generates a potential condition we can construct an en-system from a graph, using only generated conditions.

5.1 Definition. *Let $\mathcal{G} = (K, L, G, k_0)$ be a graph and let m be a set of regions of $\mathcal{G}$. Then the* **m-generated en-system** *is $sy(\mathcal{G}, m) = (m, L, F, c_0)$ where for each $R \in m$ and each $l \in L$:*

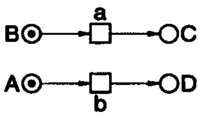
$(R, l) \in F$ iff $R \in {}^\circ l$,
$(l, R) \in F$ iff $R \in l^\circ$ and
$R \in c_0$ iff $k_0 \in R$.

5.2 Examples.

i. The graph in 1.4 with the regions A, …, J of 4.2.i generates the en-system



ii. The graph in 2.9.ii with the regions A, …, D of 4.2.ii, generates the en-system



In the rest of this section, we consider en-systems generated by *all* regions of a graph.

The basic solution to the synthesis problem can now be formulated as follows: Given an arbitrary graph $\mathcal{G}$, construct the en-system generated by the regions of $\mathcal{G}$. If the state graph of this en-system is isomorphic to $\mathcal{G}$, then this en-system is a basic solution to the synthesis problem and we are finished. Otherwise, there exists no en-system which corresponds to $\mathcal{G}$ and so $\mathcal{G}$ is no abstract state graph, as is shown in the following theorem.

5.3 Theorem. *A graph $\mathcal{G}$ is an abstract state graph iff $\mathcal{G} \simeq sg(sy(\mathcal{G}, reg(\mathcal{G})))$.*

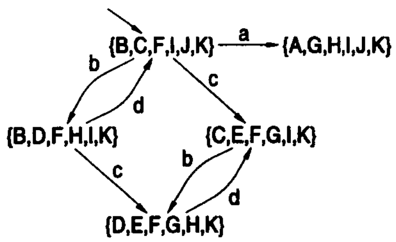
Proof. If $\mathcal{G} \simeq sg(sy(\mathcal{G}, reg(\mathcal{G})))$ then $\mathcal{G}$ is an abstract state graph by definition.

Let conversely $\mathcal{G}$ be an abstract state graph. By Proposition 3.4, $\mathcal{G} \simeq sg(\Sigma)$ for some reduced en-system Σ. Since Σ is reduced, there exists for each region R of $sg(\Sigma)$ at most one condition b which belongs exactly to the states of R. If there is a region R of $sg(\Sigma)$ with no corresponding condition then the addition of such a condition to Σ yields the system Σ^{+R}. By Proposition 4.7, Σ and Σ^{+R} have isomorphic state graphs. Let Σ' be an en-system obtained by exhaustive addition of conditions which have no corresponding regions. Then $sg(\Sigma) \simeq sg(\Sigma')$ and for every region of $sg(\Sigma)$ there is exactly one corresponding condition of Σ' and vice versa. Hence the en-systems Σ' and $sy(\mathcal{G}, reg(\mathcal{G}))$ are isomorphic. By Proposition 3.2, they have isomorphic state graphs. So we get

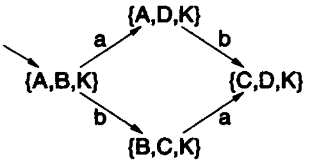
$$\mathcal{G} \simeq sg(\Sigma) \simeq sg(\Sigma') \simeq sg(sy(\mathcal{G}, reg(\mathcal{G}))). \qquad \square$$

5.4 Examples.

i. The graph $\mathcal{G}$ in 1.4 is an abstract state graph. The state graph of $sy(\mathcal{G}, reg(\mathcal{G}))$ is shown below (using the notions of 4.2.i and K = $\{1, 2, 3, 4, 5\}$). It is isomorphic to $\mathcal{G}$.



ii. The graph $\mathscr{G}$ in 2.9.ii is not an abstract state graph. The state graph of $sy(\mathscr{G}, reg(\mathscr{G}))$ is shown below (using the notions of 4.2.ii and K = $\{1,2,3,4\}$). It is not isomorphic to $\mathscr{G}$.



It is in general a nontrivial problem to decide if two graphs are isomorphic. However, for our procedure, to decide if a graph $\mathscr{G}$ is an abstract state graph we have to decide if $\mathscr{G} \simeq sg(sy(\mathscr{G}, reg(\mathscr{G})))$. This is easy since there exists at most one such isomorphism, as shown in the following proposition:

5.5 Proposition. *Let $\mathscr{G} = (K, L, G, k_0)$ be an abstract state graph. Then there is exactly one isomorphism f from $\mathscr{G}$ to $sg(sy(\mathscr{G}, reg(\mathscr{G})))$ which is defined by $f(k) = \{R \in reg(\mathscr{G}) \mid k \in R\}$.*

Proof. By Theorem 5.3, there exists an isomorphism $g: \mathscr{G} \to sg(sy(\mathscr{G}, reg(\mathscr{G})))$. We wish to prove $g = f$ where f is defined as above.

By definition of $sy(\mathscr{G}, reg(\mathscr{G}))$, its initial state is the set of regions of $\mathscr{G}$ that contain k_0. Hence, by the definition of f, $g(k_0) = f(k_0)$.

Next we show that, for every arc (h, l, k) of $\mathscr{G}$, $f(h) = g(h)$ implies $f(k) = g(k)$.

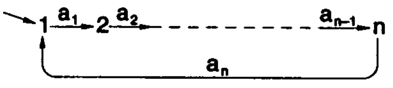
Let (h, l, k) be an arc of $\mathscr{G}$. Then $(f(h), l, f(k))$ and $(g(h), l, g(k))$ are arcs of $sg(sy(\mathscr{G}, reg(\mathscr{G})))$, because f and g are isomorphisms. Therefore $f(h) \xrightarrow{l} f(k)$ and $g(h) \xrightarrow{l} g(k)$ are state transitions of $sy(\mathscr{G}, reg(\mathscr{G}))$. Now assume $f(h) = g(h)$. By the occurrence rule, the resulting state of a state transition is uniquely determined by the current state and the occuring event. Hence $f(k) = g(k)$.

The result follows inductively because every node of $\mathscr{G}$ is reachable. □

6 The general solution of the synthesis problem

Theorem 5.3 provides a decision procedure as well as (together with Definition 5.1) a constructive solution of the synthesis problem. This way the distinguished en-system $sy(\mathscr{G}, reg(\mathscr{G}))$ is assigned to each abstract state graph $\mathscr{G}$. This en-system is reduced in the sense of Definition 3.3. However, the number of its conditions can grow exponentially with the number of nodes of $\mathscr{G}$. Consider the following example:

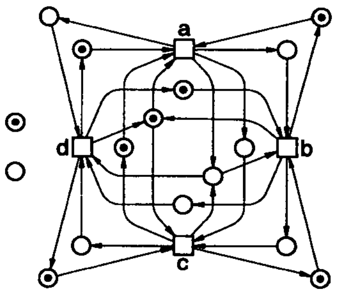
6.1 Example.



The graph $\mathscr{G}_n$ for $n \geq 2$

In 6.1, no label occurs more than once. Therefore, by the definition of regions, every subset of nodes is a region of the graph. Hence the number of regions grows exponentially with the number of nodes. The same holds for the number of conditions of the system generated by all regions:

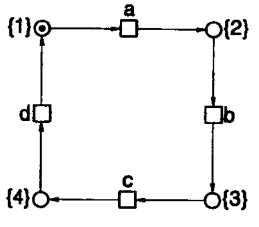
6.2 Example.

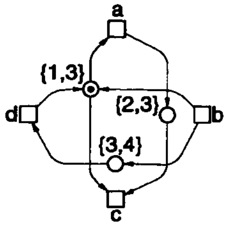


$sy(\mathscr{G}_4, reg(\mathscr{G}_4))$ has $2^4 = 16$ conditions (instead of a_1, a_2, a_3, a_4 we use the labels a, b, c, d).

The aim of this section is to show that, for every abstract state graph $\mathscr{G}$ with n nodes and m labels, there exists an en-system Σ satisfying $sg(\Sigma) \simeq \mathscr{G}$ with at most $n \cdot (n+m)$ conditions. Since every condition of Σ corresponds to a region of its state graph, and consequently to a region of $\mathscr{G}$, an en-system Σ can be generated by a subset of regions of $\mathscr{G}$. In order to achieve the polynomial growth we characterize sets of regions which suffice to construct suitable en-systems. Example 6.3 shows such en-systems for the graph $\mathscr{G}_4$ of 6.1.

6.3 Example.





The left en-system is generated from $\mathcal{G}_4$ by the set of regions $\{\{1\},\{2\},\{3\},\{4\}\}$. The right en-system is generated from $\mathcal{G}_4$ by the set of regions $\{\{1,3\},\{2,3\},\{3,4\}\}$.

Somewhat surprising, both en-systems exhibit the same behaviour; their state graphs are isomorphic to $\mathcal{G}_4$. This fact is obvious for the en-system on the left hand side. For the en-system on the right hand side we recall that an event can only occur when the current state contains none of the conditions in its post-set; so b is not enabled at the initial state (this situation is sometimes called "contact").

This en-system proves that a set m of three (of the sixteen) regions suffices to construct an en-system $sy(\mathcal{G}_4, m)$ satisfying $sg(sy(\mathcal{G}_4, m)) \simeq sg(sy(\mathcal{G}_4, reg(\mathcal{G})))$ (where in this case, by Theorem 5.3, $sg(sy(\mathcal{G}_4, reg(\mathcal{G}))) \simeq \mathcal{G}$ since $\mathcal{G}$ is an abstract state graph). Of course, not every set of regions has this property. So, we distinguish:

6.4 Definition. *Let $\mathcal{G}$ be a graph and let $m \subseteq reg(\mathcal{G})$. m is called* **admissible** *iff $sy(\mathcal{G}, reg(\mathcal{G}))$ and $sy(\mathcal{G}, m)$ have isomorphic state graphs.*

6.5 Example. Let $\mathcal{G}_4$ be as in 6.1. The sets $m_1 = \{\{1\},\{2\},\{3\},\{4\}\}$ and $m_2 = \{\{1,3\},\{2,3\},\{3,4\}\}$ are admissible. No proper subsets of these sets are admissible.

The set of conditions of an en-system corresponds to an admissible set of regions of its state graph. Conversely, for each reduced en-system, the conditions are generated by an admissible set of regions:

6.6 Theorem. *Let $\mathcal{G}$ be an abstract state graph. A reduced en-system Σ satisfies $\mathcal{G} \simeq sg(\Sigma)$ iff there exists an admissible set $m \subseteq reg(\mathcal{G})$ such that Σ and $sy(\mathcal{G}, m)$ are isomorphic en-systems.*

Proof. ($\Rightarrow$): By Proposition 4.5, every condition of Σ corresponds to a region of $sg(\Sigma)$ and, hence, to a region of $\mathcal{G}$. Let m be the set of region corresponding to conditions of Σ. Then, for every $R \in m$, there exists *at least* one condition b such that R corresponds to b. There exists *at most* one such condition b because Σ is reduced. Again since Σ is reduced, every event of Σ occurs in some reachable state transition. Therefore, and by construction of $sy(\mathcal{G}, m)$, ${}^\bullet b$ in Σ equals ${}^\bullet R$ in $sy(\mathcal{G}, m)$ and the same holds for the respective post-sets. b belongs to the initial state of Σ iff the initial state is in the corresponding region R_b iff R_b belongs to the initial state of $sy(\mathcal{G}, m)$. Hence Σ and $sy(\mathcal{G}, m)$ are isomorphic.

m is admissible because $sg(\Sigma) \simeq sg(sy(\mathcal{G}, reg(\mathcal{G})))$ by Theorem 5.3 and $sg(\Sigma) \simeq sg(sy(\mathcal{G}, m))$.

($\Leftarrow$) follows from Definition 6.4, Proposition 3.2 and Theorem 5.3. □

This result gives a complete picture of all solutions of the synthesis problem. It implies that each reduced en-system is entirely characterized (up to isomorphism) by an admissible set of regions of an abstract state graph.

Now let us return to our original task: How can we find "small" solutions of the synthesis problem? Since we construct en-systems from admissible sets of regions, we study necessary conditions for admissibility.

Let $\mathcal{G}$ be an abstract state graph with an admissible set of regions m. Then m satisfies the following properties:

Assume h and k are distinct nodes of $\mathcal{G}$. The corresponding states of the en-system generated by m are also distinct, because m is admissible. So there exists a place of this en-system which is contained in exactly one of the states. By definition of the generated en-system, this place is generated by a region which separates h and k, i.e. contains exactly one of these nodes.

Consider a node k and a label l. Then k corresponds to a reachable state of the generated en-system and l is a transition of that en-system. Either l is enabled at that state – then there is an l-labelled arc of $\mathcal{G}$ with source k – or l is not enabled – then the state contains not all pre-conditions or it contains some post-conditions of l. In other words, if there is no l-labelled arc with source k, then m contains a region of the pre-set of l which does not contain k or a region of the post-set of l which does contain k.

The following theorem shows that these conditions are not only necessary but also sufficient for admissibility.

6.7 Theorem. *Let $\mathcal{G}$ be an abstract state graph. A set $m \subseteq reg(\mathcal{G})$ is admissible iff for every two different nodes h and k of $\mathcal{G}$, there exists a region $R \in m$ satisfying*

(a) *$h \in R$ and $k \notin R$ or*
(b) *$h \notin R$ and $k \in R$,*

and, for every node k of $\mathcal{G}$ and every label l of $\mathcal{G}$, at least one of the following conditions is true:

(c) *there exists an arc (k, l, k') of $\mathcal{G}$ for some node k',*
(d) *there exists a region $R \in m$ satisfying $R \in {}^\circ l$ and $k \notin R$,*
(e) *there exists a region $R \in m$ satisfying $R \in l^\circ$ and $k \in R$.*

Proof. Define $\Sigma = sy(\mathcal{G}, reg(\mathcal{G}))$ and $\Sigma' = sy(\mathcal{G}, m)$. Since $\mathcal{G}$ is an abstract state graph, and by Theorem 5.3, there is an isomorphism $f: \mathcal{G} \rightarrow sg(\Sigma)$.

(i) The isomorphism f maps each node k of $\mathcal{G}$ to the set of regions of $\mathcal{G}$ containing k, by Proposition 5.5.
(ii) Let c_0 be the initial state of Σ. Then $c_0 \cap m$ is the initial state of Σ', by construction of Σ and Σ'.
(iii) Let $c \xrightarrow{e} d$ be a state transition of Σ. We show that $(c \cap m) \xrightarrow{e} (d \cap m)$ is a state transition of Σ'. .
The pre-set of e in Σ' equals the intersection of m and the pre-set of e in Σ. The same holds respectively for the post-set of e. Hence, $c \cap m$ enables e in Σ' and the occurrence of e yields $d \cap m$.
(iv) Let $c \xrightarrow{e} d$ be a reachable state transition of Σ. From (ii) and (iii) inductively follows that $(c \cap m) \xrightarrow{e} (d \cap m)$ is a reachable state transition of Σ'.

($\Rightarrow$): Since m is admissible, there is an isomorphism $g: sg(\Sigma) \rightarrow sg(\Sigma')$. By (ii) and (iv), each node c of $sg(\Sigma)$ is mapped to $c \cap m$.

Let h and k be two different nodes of $\mathcal{G}$. Since f is bijective, $f(h) \neq f(k)$. Since g is bijective, $g(f(h)) \neq g(f(k))$ and, hence, $(f(h) \cap m) \neq (f(k) \cap m)$. Therefore, by the definition of f, at least one region of m contains h but not k or k but not h. So either (a) or (b) holds.

Let k be a node and let l be a label of $\mathcal{G}$ such that (d) and (e) do not hold for k and l. Then $g(f(k))$ is a reachable state of Σ', l is an event of Σ', every condition in ${}^\bullet l$ belongs to $g(f(k))$ (since (d) does not hold) and no condition in $l^\bullet$ belongs to $g(f(k))$ (since (e) does not hold). Hence, $g(f(k))$ enables l in Σ'. $\mathcal{G} \simeq sg(\Sigma')$, so (c) does hold.

($\Leftarrow$): We prove that $g: sg(\Sigma) \rightarrow sg(\Sigma')$, defined by $g(c) = c \cap m$ for every reachable state c of Σ, is an isomorphism.

By (ii) and (iv), g is surjective. g is injective because, by (a) or (b), for each two different nodes $f(h)$ and $f(k)$ of $sg(\Sigma)$, there is a region of m which is contained in exactly one of these nodes and, hence, $g(f(h)) \neq g(f(k))$.

By (ii), g maps the initial node of $sg(\Sigma)$ to the initial node of $sg(\Sigma')$.

Let (c, e, d) be an arc of $sg(\Sigma)$. By (iv), $(g(c), e, g(d))$ is an arc of $sg(\Sigma')$.

Let conversely $(g(c), e, g(d))$ be an arc of $sg(\Sigma')$. Let $c = f(k)$. Neither (d) nor (e) holds for k and e since $g(c)$ enables e in Σ'. Hence (c) holds for k and e. So there exists an e-labelled arc leaving k in $\mathscr{G}$ and, by $\mathscr{G} \simeq sg(\Sigma)$, there exists an arc (c, e, d') in $sg(\Sigma)$. By (iv), there exists an arc $(g(c), e, g(d'))$ in $sg(\Sigma')$. So $g(c) \xrightarrow{e} g(d)$ and $g(c) \xrightarrow{e} g(d')$ are state transitions of Σ'. Since the resulting state of a state transition is uniquely determined by the current state and the occurring event we obtain $g(d) = g(d')$. Since g is bijective, $d = d'$. Hence (c, e, d) is an arc of $sg(\Sigma)$. □

We will use the characterization provided by this theorem in the following section. An immediate consequence is the following corollary:

6.8 Corollary. *For every abstract state graph $\mathscr{G} = (K, L, G, k_0)$, there exists an en-system Σ satisfying $\mathscr{G} \simeq sg(\Sigma)$ with at most*

$$|K| \cdot (|K| - 1) + |K| \cdot |L| - |G|$$

conditions. An upper bound of this value is $|K| \cdot (|K| + |L|)$.

7 Construction of solutions of the synthesis problem

Given an abstract state graph $\mathscr{G}$, we obtained with Theorem 5.3 a method to construct a (reduced) en-system which solves the synthesis problem. However, in general this system has unnecessarily (in general exponentially) many conditions. Exploiting Theorem 6.7, small solutions with a polynomial number of conditions can be constructed as follows:

7.1 Construction. Let $\mathscr{G} = (K, L, G, k_0)$ be an abstract state graph.
Step 1: For each node $k \in K$ and each label $l \in L$ such that there is no arc $(k, l, k') \in G$, choose a region R of $\mathscr{G}$ satisfying $R \in {}^\circ l$ and $k \in R$ (or take $K \setminus R$, then $R \in l^\circ$ and $k \notin R$).
Step 2: For each two different nodes $h, k \in K$ choose a region R which contains exactly one of these two nodes.
By Theorem 4.7, we find such regions and the set of all chosen regions is admissible.

Consider a sequence of all pairs (k, l) of nodes and labels of an abstract state graph. Starting with the empty set, a set of regions according to Step 1 of Construction 7.1 is obtained by successively adding regions to the current set, if necessary for the current pair (k, l). However, this method does not necessarily yield *minimal* admissible sets of regions. Consider the following example:

7.2 Example. Let the graph $\mathscr{G}_4$ be defined as in 6.1. Then Step 1 of Construction 7.1 may yield an admissible set as follows

list of pairs		current set of regions
$(1, a_1)$	there is an arc $(1, a_1, 2)$	$\emptyset$
$(1, a_2)$	add $\{2\}$	$\{\{2\}\}$
$(1, a_3)$	add $\{3\}$	$\{\{2\}, \{3\}\}$
$(1, a_4)$	add $\{4\}$	$\{\{2\}, \{3\}, \{4\}\}$
$(2, a_1)$	use $\{2\}$	$\{\{2\}, \{3\}, \{4\}\}$
$(2, a_2)$	there is an arc $(2, a_2, 3)$	$\{\{2\}, \{3\}, \{4\}\}$
$(2, a_3)$	use $\{3\}$	$\{\{2\}, \{3\}, \{4\}\}$
$(2, a_4)$	use $\{4\}$	$\{\{2\}, \{3\}, \{4\}\}$
$(3, a_1)$	add $\{2, 3\}$	$\{\{2\}, \{3\}, \{4\}, \{2, 3\}\}$
$(3, a_2)$	use $\{2\}$ or $\{3\}$	$\{\{2\}, \{3\}, \{4\}, \{2, 3\}\}$
$(3, a_3)$	there is an arc $(3, a_3, 4)$	$\{\{2\}, \{3\}, \{4\}, \{2, 3\}\}$
$(3, a_4)$	use $\{4\}$	$\{\{2\}, \{3\}, \{4\}, \{2, 3\}\}$
$(4, a_1)$	add $\{1\}$	$\{\{1\}, \{2\}, \{3\}, \{4\}, \{2, 3\}\}$
$(4, a_2)$	use $\{2\}$	$\{\{1\}, \{2\}, \{3\}, \{4\}, \{2, 3\}\}$
$(4, a_3)$	use $\{3\}$ or $\{4\}$	$\{\{1\}, \{2\}, \{3\}, \{4\}, \{2, 3\}\}$
$(4, a_4)$	there is an arc $(4, a_4, 1)$	$\{\{1\}, \{2\}, \{3\}, \{4\}, \{2, 3\}\}$

The set $\{\{1\}, \{2\}, \{3\}, \{4\}, \{2, 3\}\}$ is not a minimal admissible set because its proper subset $\{\{1\}, \{2\}, \{3\}, \{4\}\}$ is admissible as well.

In Example 7.2, the region $\{2, 3\}$ is *redundant*, i.e. its removal preserves admissibility. The obtained set of regions $\{\{1\}, \{2\}, \{3\}, \{4\}\}$ is a minimal admissible set, i.e. none of its regions is redundant. It corresponds to a net with a more intuitive choice of places. Minimal admissible sets are not necessarily those with minimal cardinality. For the graph $\mathcal{G}_4$ in 6.1, Example 6.3 shows an admissible set of three regions.

Redundant regions can be detected by help of Theorem 6.7. Sometimes it is easier to identify redundant regions using simple sufficient conditions for redundancy. Such conditions will be considered in the rest of this section.

7.3 Definition. *Let $\mathcal{G}$ be an abstract state graph and let m be an admissible set of regions. $R \in m$ is* **redundant** (w.r.t. *m) iff $m \setminus \{R\}$ is admissible.*

A region R is redundant w.r.t. a set m if the systems $sy(\mathcal{G}, m)$ and $sy(\mathcal{G}, m \setminus \{R\})$ have the same behaviour, i.e. they have isomorphic state graphs.

It is easily seen that trivial regions are always redundant. In terms of associated reduced en-systems, the empty region corresponds to an isolated condition which never holds. Likewise, the region consisting of all nodes corresponds to the condition which always holds.

7.4 Proposition. *Let $\mathcal{G} = (K, L, G, k_0)$ be an abstract state graph and let m be an admissible set of regions. The trivial regions K and $\emptyset$ are redundant w.r.t. m.*

Proof. We apply Theorem 6.7.

Neither (a) nor (b) holds for K and $\emptyset$.

Since, for $l \in L$, $K \notin {}^\circ l$, $\emptyset \notin {}^\circ l$, $K \notin l^\circ$ and $\emptyset \notin l^\circ$, neither (d) nor (e) holds for K and $\emptyset$. □

A first nontrivial criterion for redundancy of regions is the presence of *complements*: a region $R \in m$ is redundant if the set of all nodes not in R (which clearly is a region as well) is in m. In terms of reduced en-systems, two conditions b_0 and b_1 satisfying

${}^\bullet b_0 = b_1^\bullet$, $b_0^\bullet = {}^\bullet b_1$ and b_0 is in the initial marking iff b_1 is not in the initial marking, are complements. One of them may be skipped without affecting the behaviour.

7.5 Proposition. *Let $\mathcal{S} = (K, L, G, k_0)$ be an abstract state graph and let m be an admissible set of regions. If m contains a region R and its complement $\overline{R} = K \setminus R$, then $\overline{R}$ is redundant (and, by $\overline{\overline{R}} = R$, R is also redundant).*

Proof. We apply Theorem 6.7.

If (a) holds for a pair of nodes h and k and the region $\overline{R}$, then (b) holds for h and k and the region R and vice versa.

If (d) holds for a node k, a label l and $\overline{R}$ then (e) holds for k, l and R and vive versa. □

The next rules concern regions which are intersections of other regions. Note that this condition is in general not sufficient for redundancy. But, if moreover the complement of a region R (which does not need to be in m) is the intersection of regions of m as well, then R is redundant.

7.6 Proposition. *Let $\mathcal{S} = (K, L, G, k_0)$ be an abstract state graph and let m be an admissible set of regions. If m contains regions R, R_1, R_2, R_3, R_4 such that $R = R_1 \cap R_2$ and $\overline{R} = R_3 \cap R_4$ then R is redundant.*

Proof. We apply Theorem 6.7.

If (a) holds for a pair of nodes h and k and the region R then h is in R_1 and in R_2 and k is either not in R_1 or not in R_2. Hence (a) holds either for R_1 or for R_2. (b) is preserved by a similar argument.

Assume that (d) holds for a node k, a label l and R. Then $k \in \overline{R}$. Hence $k \in R_3$ and $k \in R_4$. Since $R \in {}^\circ l$, $\overline{R} \in l^\circ$. Hence either $R_3 \in l^\circ$ or $R_4 \in l^\circ$. So (e) holds either for k, l and R_3 or for k, l and R_4.

Assume that (e) holds for a node k, a label l and R. Then $k \in R_1$ and $k \in R_2$. Since $R \in l^\circ$, either $R_1 \in l^\circ$ or $R_2 \in l^\circ$. So (e) holds either for k, l and R_1 or for k, l and R_2. □

7.7 Corollary. *Let $\mathcal{S} = (K, L, G, k_0)$ be an abstract state graph and let m be an admissible set of regions. If m contains regions R, R_1, R_2, R_3, R_4 such that $R = R_1 \cup R_2$ and $\overline{R} = R_3 \cup R_4$ then R is redundant.*

Proof. Define $m' = m \cup \{\overline{R}, \overline{R_1}, \overline{R_2}, \overline{R_3}, \overline{R_4}\}$. Using Proposition 7.5, R is redundant w.r.t. m'. By Proposition 7.6 and since $\overline{R} = \overline{R_1} \cap \overline{R_2}$ and $R = \overline{R_3} \cap \overline{R_4}$, $\overline{R}$ is redundant w.r.t. $m' \setminus \{R\}$. Applying Proposition 7.5 again, $m \setminus \{R\}$ is admissible. □

7.8 Example. The set $\{2,3\}$ in the admissible set considered in 7.2 is redundant: $\{2,3\} = \{2\} \cup \{3\}$ and $\overline{\{2,3\}} = \{1,4\} = \{1\} \cup \{4\}$.

Finally, we introduce a sufficient condition for redundancy which does not only depend on the regions but also takes the arcs of $\mathcal{S}$ into account.

7.9 Proposition. *Let $\mathcal{S} = (K, L, G, k_0)$ be an abstract state graph and let m be an admissible set of regions. If m contains regions R, R_1, R_2 such that $R = R_1 \cap R_2$ and every arc $(k, l, k') \in R \times L \times \overline{R}$ satisfies $k' \notin R_1$ and $k' \notin R_2$, then R is redundant.*

Proof. We apply Theorem 6.7.

(a) and (b) is preserved as in Proposition 7.6.

Assume that (d) holds for a node k, a label l and R. Then $R_1 \in {}^{\circ}l$ and $R_2 \in {}^{\circ}l$ by assumption. Since $k \notin R_1$ or $k \notin R_2$, (d) holds either for k, l and R_1 or for k, l and R_2.

(e) is preserved as in Proposition 7.6. □

7.10 Example. Consider the graph $\mathscr{G}_4$ defined in 6.1. The set $\{\{3\},\{1,3\},\{2,3\},\{3,4\}\}$ is admissible. $\{3\}$ is redundant since $\{3\} = \{1,3\} \cap \{2,3\}$, the only arc leaving $\{3\}$ is $(3, a_3, 4)$, and $4 \notin \{1,3\}$ as well as $4 \notin \{2,3\}$.

Conclusion

We have given a direct proof that a graph is an abstract state graph if and only if it is isomorphic to the state graph of its associated en-system. This en-system is constructed using the concept of regions of graphs, introduced in [Ehrenfeucht, Rozenberg 90]. In that paper, abstract state graphs (there called "abstract sequential case graphs") are characterized by a number of requirements, including (in our terminology) isomorphy of $\mathscr{G}$ and $sg(sy(\mathscr{G}, reg(\mathscr{G})))$. Theorem 5.3 shows that this condition is sufficient.

[Ehrenfeucht, Rozenberg 90] and [Nielsen, Rozenberg, Thiagarajan 92] restrict their studies to en-systems where different events with equal pre- and post-sets are excluded. We have shown that this restriction is not necessary. Even side-conditions, modelled by a different interpretation of self-loops, can be tackled, demanding however a more involved formalism.

For simplicity we only considered finite en-systems. This restriction is not necessary; [Ehrenfeucht, Rozenberg 90] and [Nielsen, Rozenberg, Thiagarajan 92] study also infinite en-systems. In a quite early contribution, [Krieg 77] has shown that, for the case of place/transition systems (a generalisation of elementary net systems), it is decidable whether a given graph is an abstract state graph. [Mukund 92] solves the synthesis problem for the case of finite place/transition systems, by construction of one solution. In a recent contribution, [Bernadinello 93] also strives for small solutions of the synthesis problem for en-systems. In this paper, the structure of regions is investigated; only "small regions" are used to generate an en-system. Thus, instead of conditions, the number of tokens in reachable states is minimized.

Further considerations concern contact-free, small solutions. A *contact situation* is a reachable state which contains all conditions in the pre-set of an event e but does not enable e (since it also contains post-conditions). Taking again Example 6.3, the left system is contact-free whereas the right system is not (${}^{\bullet}b = \emptyset$, hence the initial state includes ${}^{\bullet}b$ but does not enable b). Contact-free systems are in general much more intuitive than systems exhibiting a contact. The left system of 6.3 can be seen as a "canonical" solution of the synthesis problem whereas the right system has less conditions. The admissible sets yielding contact-free solutions can be characterized by skipping line (e) in Theorem 4.7. An en-system can be considered "optimal" if it is contact-free and has no redundant conditions, i.e. each condition is either crucial for contact-freeness or for the behaviour of the system. The left system of 6.3 is optimal. However, not every abstract state graph has corresponding optimal en-systems. We leave it an open problem if the existence of optimal solutions can be characterized in terms of state graphs. It is also unknown if there always exists at most one optimal solution (up to isomorphism), which then could be considered canonical.

References

[Bernadinello 93] L. Bernadinello: "Synthesis of Net Systems", *Proc. Application and Theory of Petri Nets, Lecture Notes in Computer Science 691*, (Springer-Verlag, 1993) 89–105

[Ehrenfeucht, Rozenberg 90] A. Ehrenfeucht, G. Rozenberg: "Partial 2-structures; Part II, State Space of Concurrent Systems", Acta Inf. **27** (1990) 348-368

[Janicki 85] R. Janicki: "Transforming Sequential Systems into Concurrent Systems", Theoret. Comp. Sci. **36** (1985) 27–58

[Krieg 77] B. Krieg: "Petrinetze und Zustandsgraphen", IFI-Bericht B-29/77, Institut für Informatik, Universität Hamburg, Germany, 1977

[Lengauer, Hehner 82] C. Lengauer, E.C.R. Hehner: "A Methodology for Programming with Concurrency: An Informal Presentation", Sci. of Comp. Progr. **2** (1982) 1–18

[Mukund 92] M. Mukund: "Petri Nets and Step Transition Systems", Int. Journal of Found. of Comp. Sci. **3** No. 4 (1982) 443–478

[Nielsen, Rozenberg, Thiagarajan 92] M. Nielsen, G. Rozenberg, P.S. Thiagarajan: "Elementary Transition Systems", Theo. Comp. Sci. **96**, 1 (1992) 3–33

[Thiagarajan 87] P.S. Thiagarajan: "Elementary Net Systems", *Advances in Petri Nets 1986 Part I, Lecture Notes in Computer Science 254*, (Springer-Verlag, 1987) 26–59