STOC(Milwaukee 1981),238-246.

An Algorithm for the General Petri Net Reachability Problem

Ernst W. Mayr *
Visiting Scientist
Laboratory for Computer Science
Massachusetts Institute of Technology
545 Technology Square, Cambridge, Mass. 02139

Abstract. An algorithm is presented for the general Petri net reachability problem based on a generalization of the basic reachability construction which is symmetric with respect to the initial and final marking. Sets of transition sequences described by finite automata are used for approximations to firing sequences, and the approximation error is assessed by uniformly constructable Presburger expressions. The approximation algorithm is iterated until a sufficient criterion for reachability can be given, notwithstanding the remaining uncertainty.

Key words and phrases: Petri net, vector addition system, reachability problem, decidability

1. Introduction

Petri nets are a mathematical model for the representation and analysis of parallel processes which is due to C.A. Petri [33] and, later on, was widely generalized and investigated [4,13,16,17,21]. Especially in [13], the connections between different variants of the basic model—which can also be formulated in a purely algebraic way as vector addition system—were investigated. In [28,32] summaries can be found showing the relation of Petri nets to numerous other models of parallel computation.

Problems analyzed in modelling parallel systems by Petri nets usually are dealing with dynamic aspects of the control structure as Petri nets tend to heavily abstract from the individuality of data objects. Such problems are (partial or total) deadlock freeness, or liveness of the system. While these properties, in a sense, state absence of problems in all states reachable in the system, other questions were concerned with the problems whether some arbitrary state is reachable from some fixed initial state, or whether there is some reasonable effective description of the set of all reachable states. The former, the so-called general reachability problem, soon proved to be of basic importance to many others. In [12], the recursive equivalence of the reachability and the liveness problem were shown, i.e. an algorithm for one of the problems automatically solves the other. In addition, a number of other problems in the representation of parallel and concurrent systems, but also in language generating systems, in algebra and in number theory could be shown to be effectively reducible to or equivalent with the reachability problem. On the other hand, the proof of the undecidability of the inclusion problem for Petri net reachability sets [2]—extended in [14] to the equality problem—implies that there cannot be any reasonable closed effective representation for reachability sets in general. For restricted classes of Petri nets, however, such effective representations have been given [1,9, 26,29].

For the reachability problem, more or less restricted subclasses of Petri nets have been investigated (a summary of results can be found in [19]), or sometimes quite heuristic methods have been proposed to simplify given Petri nets in order to obtain sufficient conditions [3,15,22,24]. In [36], the decidability of the reachability problem for Petri nets with at most three places was shown, and in [18], this result was extended to up to five places. Whereas these methods fail for the general case as they rely on semilinear reachability sets—and there are Petri nets with six places and non-semilinear reachability sets—the general reachability problem was claimed to be decidable in [35], no correct proof was given, however.

The undecidability of the reachability set inclusion problem suggested a treatment of the reachability problem symmetric

*This work was supported in part by the Deutsche Forschungsgemeinschaft, Grant No. 13 Ma 870/1-1, and by The National Science Foundation, Grant No. MCS 8010707

wrt. the initial and final state, as presented in the sequel. After giving the notation and basic concepts, a generalization of the fundamental reachability tree construction [20] is introduced which uses finite automata to restrict the set of transistion sequences possible in the original construction, and which can be symmetrized in the above respect. In the iterative algorithm, starting with the trivial automaton accepting all transition sequences, each automaton is a refinement of its predecessor, and is (also) used to carry over from one step of the iteration to the next the information obtained so far. By means of regular sets of transition sequences, possible firing sequences from the initial to the final state are thus being approximated. The approximation error is determined by uniformly constructable Presburger expressions, and the approximation algorithm is iterated until, in spite of the remaining uncertainty, a sufficient criterion for reachability can be given.

The algorithm is formulated as a nondeterministic one. In a deterministic implementation, every nondeterministic choice is replaced by branching to all possibilities the number of which always is finite.

Finally, some decision problems reducible to or equivalent with the reachability problem, and some open research problems are listed.

2. Notation and basic concepts

A *Petri net* P is a triple (S, T, K) with

i) $S = \{s_1, \ldots, s_v\}$ a finite set of *places*;

ii) $T = \{t_1, \ldots, t_w\}$ a finite set of *transitions*, disjoint from S;

iii) $K : S \times T \cup T \times S \to \mathbf{N}$ a mapping indicating the *multiplicity* of *directed edges* between places and transitions,

where $\mathbf{N} = \{0, 1, \ldots\}$ denotes the set of nonnegative integers.

Further, $\overline{\mathbf{N}} =_{\text{def}} \mathbf{N} \cup \{\omega\}$ stands for $\mathbf{N}$ augmented by the "infinite" number ω with $\pm n + \omega = \omega \pm n = \omega$ and $n < \omega$ for all $n \in \mathbf{N}$, $\mathbf{Z}$ for the set of integers, and I_n for the (index) set $\{1, 2, \ldots, n\}, n \in \mathbf{N}$.

A *marking* (*pseudomarking*) of P is a mapping $m : S \to \mathbf{N}$ ($\overline{m} : S \to \overline{\mathbf{N}}$, resp.) which usually gets represented as a vector $m \in \mathbf{N}^v$ ($\overline{m} \in \overline{\mathbf{N}}^v$, resp.) such that its value at place $s_i \in S$ can be written interchangeably as $m(s_i)$ ($\overline{m}(s_i)$, resp.) or m_i ($\overline{m}_i$, resp.), $i \in I_v$.

The following relations are defined on pseudomarkings:

i) $\overline{m} \leq \overline{m}' \Leftrightarrow_{\text{def}} (\forall i \in I_v)[\overline{m}_i \leq \overline{m}'_i]$;

ii) $\overline{m} < \overline{m}' \Leftrightarrow_{\text{def}} \overline{m} \leq \overline{m}' \wedge \overline{m} \neq \overline{m}'$;

iii) $\overline{m} \subset \overline{m}' \Leftrightarrow_{\text{def}} (\forall i \in I_v)[\overline{m}'_i < \omega \Rightarrow \overline{m}_i = \overline{m}'_i]$;

iv) $\overline{m} \simeq \overline{m}' \Leftrightarrow_{\text{def}} (\forall i \in I_v)[\overline{m}_i < \omega \wedge \overline{m}'_i < \omega \Rightarrow \overline{m}_i = \overline{m}'_i]$

(note: ad iii) $\overline{m} \subset \overline{m}'$ means that $\overline{m}$ can be obtained by replacing some ω-components of $\overline{m}'$ by finite values;
ad iv) $\overline{m} \simeq \overline{m}'$ means that there is some $\overline{m}''$ s.t. $\overline{m}'' \subset \overline{m} \wedge \overline{m}'' \subset \overline{m}'$).

Further, let $H \in \mathbf{N}$, $m, \overline{m}' \in \overline{\mathbf{N}}^v$. Then

$$F(\overline{m}, H) =_{\text{def}} \{m \in \mathbf{N}^v; (\forall i \in I_v)[m_i = \overline{m}_i \vee (\overline{m}_i = \omega \wedge m_i \geq H)]\}$$

(i.e. $F(\overline{m}, H)$ is the set of all finite vectors obtained by replacing ω-components of $\overline{m}$ by finite values $\geq H$).

The *marking difference* $\delta t_i \in \mathbf{Z}^v$ effected by $t_i \in T$ is given by

$$(\delta t_i)_j = K(t_i, s_j) - K(s_j, t_i), \text{ for } i \in I_w, j \in I_v.$$

$t \in T$ is *firable* at pseudomarking $\overline{m}$ (written $f(t, \overline{m})$) iff $\overline{m} \geq \delta t^- =_{\text{def}} (K(s_1, t), \ldots, K(s_v, t))$. If t is firable at $\overline{m}$, the firing of t takes $\overline{m}$ to $\overline{m} + \delta t$: $\overline{m} \xrightarrow{t} \overline{m} + \delta t$.

For sequences $\tau = t^1 \ldots t^r \in T^*$, $\delta\tau$, $f(\tau, \overline{m})$, $\delta\tau^-$, and $\overline{m} \xrightarrow{\tau} \overline{m}'$ are defined inductively:

i) $\delta\tau =_{\text{def}} \sum_{i=1}^{r} \delta t^i$;

ii) $f(\tau, \overline{m}) =_{\text{def}} r = 0 \vee (f(t^1, \overline{m}) \wedge f(t^2 \ldots t^r, \overline{m} + \delta t^1))$;

iii) $\delta\tau^- =_{\text{def}} \max\{\delta(t^1)^-, \delta(t^2 \ldots t^r)^- - \delta t^1, 0\}$ (max componentwise);

iv) $\overline{m} \xrightarrow{\tau} \overline{m}' =_{\text{def}} f(\tau, \overline{m}) \wedge \overline{m}' = \overline{m} + \delta\tau$.

Note that $\delta\tau^-$ is the minimal marking at which τ is firable.

The *reachability set* $R(P, \overline{m})$ of the pseudomarked Petri net $(P, \overline{m})$ is

$$R(P, \overline{m}) =_{\text{def}} \{\overline{m}'; \overline{m} \xrightarrow{*} \overline{m}'\} = \{\overline{m}'; (\exists \tau \in T^*)[\overline{m} \xrightarrow{\tau} \overline{m}']\}.$$

Let $\Phi : T^* \to \mathbf{N}^w$ denote the Parikh mapping, i.e. $(\Phi(\tau))_i$ indicates, for each $i \in I_w$ and $\tau \in T^*$, the number of occurrences of t_i in τ, and let $V \in \mathbf{Z}^{v,w}$ be the integer matrix whose i-th column is given by δt_i. Note as an immediate consequence that $\delta\tau = V\Phi(\tau)$ for all $\tau \in T^*$.

A *linear set* $L \subseteq \mathbf{N}^w$ is a set of the form

$$L = \{b + \sum_{i=1}^{r} n_i p^i; (n_1, \ldots, n_r) \in \mathbf{N}^r\}$$

for some $r \in \mathbf{N}$, $b, p^1, \ldots, p^r \in \mathbf{N}^w$. b is called the *base* of L.

A *semilinear set* is a finite union of linear sets.

Semilinear sets are exactly those sets definable by expressions in Presburger Arithmetic, i.e. the first order theory of the nonnegative integers with addition [8,34]. Semilinear sets are, therefore, closed under Boolean operations, and there are effective procedures to construct semilinear representations of the sets defined by Presburger expressions and to decide Presburger formulae [8,30].

Let G be a digraph, and α a path in G. Then G_{rev} denotes the *reversed graph* of G obtained by reversing the orientation of all edges in G, α_{rev} denotes the *reversed path* in G_{rev} corresponding to α. $a(\alpha)$ is the initial and $d(\alpha)$ the final node of α. If, in particular, G is a Petri net $P = (S,T,K)$, then we refer to the transition in P_{rev} corresponding to t in P as t_{rev}, we have $\delta t_{rev} = -\delta t$, and set, for $\tau = t^1 \ldots t^r \in T^*$, $\tau_{rev} =_{\mathrm{def}} t^r_{rev} \ldots t^1_{rev}$ (this is a slight misuse of notation as actually t as a node does not get changed when reversing P to P_{rev}).

We now introduce the concept of a *regularly controlled reachability graph* which is a generalization of the basic reachability construction introduced in [20] and which uses finite automata to restrict the set of paths possible in the original construction. Let A be a finite digraph with edge labels in T which we consider the state transition diagram of a (nondeterministic) finite automaton, with some node z_0 designated as initial state. In the final algorithm, A will initially be the trivial automaton consisting of one state only and accepting T^*. Consecutive A's will be obtained by refining their predecessor in such a way that possible firing sequences from m to m' don't get thrown away.

The *regularly controlled reachability graph* $RCRG(P, \overline{m}, A)$ for some Petri net P with initial pseudomarking $\overline{m} \in \bar{\mathbb{N}}^n$ is a digraph with two node labels, $\overline{m}(k) \in \bar{\mathbb{N}}^n$ and $z(k)$ in the node set (= state set) of A, for every node k, and with edge label $t(e) \in T$ for every edge e. It is constructed as follows by

Algorithm 1:

start with the "root" r of $RCRG(P, \overline{m}, A)$, set $\overline{m}(r) := \overline{m}$; $z(r) := z_0$, and declare r as "unfinished".
repeat as long as there are unfinished nodes
  for all unfinished nodes k in the graph constructed so far **do**
    for each $t \in T$ s.t. $f(t, \overline{m}(k))$ and each state z in A reachable from $z(k)$ by an edge labelled t **do**
      add a new edge e to $RCRG(P, \overline{m}, A)$ with $t(e) := t$;
      let k' denote the endpoint $d(e)$ of e;
      set $z(k') := z$; $\overline{m}(k') := \overline{m}(k) + \delta t$
    od;
    declare k finished
  od;
  define an *augmenting path* in $RCRG(P, \overline{m}, A)$ to be a path α s.t.
    α is a subsegment of a simple path from r not containing r
    co the latter condition is for convenience only, it assures that $\overline{m}(r) = \overline{m}$ throughout the algorithm **oc**;
    all $\overline{m}(k)$ for nodes k on α have the same set of ω-coordinates,
    $a(\alpha)$ is not on an ω^+-*cycle* (see below)
    co this condition merely serves to avoid that the ω^+-cycle is destroyed later on by the creation of an ω^+-node **oc**;
    $\overline{m}(a(\alpha)) < \overline{m}(d(\alpha))$, and
    $z(a(\alpha)) = z(d(\alpha))$;
  set $AP :=$ the set of augmenting paths in $RCRG(P, \overline{m}, A)$;
  for all $\alpha \in AP$ **do**
    for all $i \in I_v$ s.t. $(\overline{m}(a(\alpha)))_i < (\overline{m}(d(\alpha)))_i$ **do**
      replace $(\overline{m}(a(\alpha)))_i$ in $RCRG(P, \overline{m}, A)$ by ω
    od;
    delete in $RCRG(P, \overline{m}, A)$ all edges leaving $a(\alpha)$
  od;
  delete all nodes and edges no longer reachable from r;
  for all $\alpha \in AP$ s.t. $a(\alpha)$ is still in $RCRG(P, \overline{m}, A)$ **do**
    let $t^1, \ldots, t^l$ be the edge labelling sequence of α;
    append to $a(\alpha)$ a cycle $k^0 = a(\alpha), k^1, \ldots, k^l = k^0$ with the same edge labelling sequence $t^1 \ldots t^l$ and the same node labelling sequence $z(k^0), \ldots, z(k^l)$ as α, and
    $\overline{m}(k^j) := \overline{m}(a(\alpha)) + \delta(t^1 \ldots t^j)$ for all $j \in I_l$
    co this cycle is called an ω^+-*cycle*, and k^0 an ω^+-*node* **oc**;
    declare $k^0 = a(\alpha)$ unfinished
  od;
  identify all nodes k which have the same node markings $\overline{m}(k)$ and $z(k)$;
  declare all newly generated nodes not identified with nodes already finished as unfinished
end repeat
end Algorithm 1

The proof for the termination of Algorithm 1 is along the same lines as in [20] taking into account that there are only finitely many states in A; it won't be given here.

Note the following properties of $R := RCRG(P, \overline{m}, A)$:

P1: If k, k' are nodes in R and k' is reachable from k then

$$W(k) =_{\mathrm{def}} \{i \in I_{v_i} ; (\overline{m}(k))_i = \omega\} \subseteq W(k').$$

P2: If k is an ω^+-*node* and k has some immediate predecessor k' s.t. $W(k') \subsetneq W(k)$ then there is an effectively obtainable ω^+-*cycle* α from k to k (which is not necessarily simple because of identification of nodes with the same marking) whose edge labelling sequence $\tau(\alpha)$ satisfies

$$(\delta\tau(\alpha))_i > 0 \text{ for all } i \in W(k) - W(k'), \text{ and}$$
$$f(\tau(\alpha), \overline{m}(k') + \delta t(e)),$$

where edge e goes from k' to k.

P3: For any path in A starting from z_0 with edge labelling sequence τ s.t. $f(\tau, \overline{m})$ there is a path α in R starting from r with $\tau(\alpha) = \tau$
(i.e. R contains all firing sequences not forbidden by A).
$d(\alpha)$ satisfies $\overline{m} + \delta\tau \sqsubset \overline{m}(d(\alpha))$.

P4: For any node k in R and any $H \in \mathbb{N}$, there is a path α from r to k s.t.

$$f(\tau(\alpha), \overline{m}) \text{ and } \overline{m} + \delta\tau(\alpha) \in F(\overline{m}(k), H)$$

(This characterizes, in a certain respect, the meaning of ω-components; also cf. [13,14]).

P5: Let A' be the transition diagram given by R without the node markings, with r as initial state. Then R equals $RCRG(P, \overline{m}, A')$ modulo the node marking z.
A' is a restriction or refinement of A, i.e. there is a homomorphism from A' into A (given by the node marking z of R).

P6: Let k be some node in R, $\overline{m}' \sqsupseteq \overline{m}(k)$, A'' the transition diagram given by all paths from r to k, and k the initial state in A''_{rev}. Then $RCRG(P_{rev}, \overline{m}', A''_{rev})$ — stripped of the node markings—equals A''_{rev}.

P1–P6 can be proved by induction on the steps of Algorithm 1, respectively the length of τ. The proof is omitted here.

Let $\overline{m}, \overline{m}' \in \overline{\mathbb{N}}^v$, and $B := RCRG(P, \overline{m}, A)$ for some A. A subgraph B' of B is called *admissible* wrt. $\overline{m}'$ iff it is obtained by starting with some simple path α from the root r of B to some node k with $\overline{m}(k) \simeq \overline{m}'$, and adding to it all nodes on cycles through nodes on α. Nodes r and k are called the *end nodes* of B'. If B' (resp., B'_{rev}) is used as the transition diagram of a finite automaton it is stripped of the node markings, and r (resp., k) is defined its initial state. In the sequel, any admissible B' will wlg. be assumed *separated*, i.e. every simple cycle in B' has at most one node in common with the simple path α chosen in the beginning. If B' does not yet have this form it can be transformed to it by attaching to each node k on α a copy of the strongly connected component (SCC) of k in B' s.t. it has only k in common with α. This transformation does not alter the set of edge labelling sequences of paths between the two end nodes.

3. The construction of two-sided RCRG's

As $\overline{m}' \in R(P, \overline{m})$ iff $\overline{m} \in R(P_{rev}, \overline{m}')$ the construction of Algorithm 1 is generalized to the concept of a two-sided RCRG. Let $\overline{m}, \overline{m}' \in \overline{\mathbb{N}}^v$, A some finite automaton, and $B := RCRG(P, \overline{m}, A)$.

Algorithm 2:

a) Nondeterministically select some subgraph B' of B, B' admissible wrt. $\overline{m}'$, with end nodes r and k. Set $\overline{m}' := \min\{\overline{m}', \overline{m}(k)\}$ (min componentwise).
b) Set $B'' := RCRG(P_{rev}, \overline{m}', B'_{rev})$;
c) Nondeterministically select some subgraph B''' of B'', B''' admissible wrt. $\overline{m}$, with end nodes r' and k', s.t. $z(r') = k, z(k') = r$.
d) If (as transitions diagrams) $B'''_{rev} \neq B'$ set $\overline{m} := \min\{\overline{m}, \overline{m}(k')\}$ and $B := RCRG(P, \overline{m}, B'''_{rev})$ and go to a), otherwise stop.

To show the termination of Algorithm 2, first note that if k is an ω^+-node in any B'' or B constructed in step b) or d), resp., of the algorithm introducing the i-th coordinate as new ω-coordinate for some $i \in I_v$, then $i \in W(z(k))$ because the ω^+-cycle α in k corresponds to some cycle through $z(k)$, and as $(\delta\tau(\alpha))_i > 0$.

Let B' with end nodes r and k and B''' with end nodes r' and k' be the admissible RCRG's determined in the first run of Algorithm 2's loop. Call some coordinate $i \in I_v$ *essential* iff

$$(\overline{m}(r))_i < \omega \wedge (\overline{m}(k))_i = \omega \text{ or}$$
$$(\overline{m}(r'))_i < \omega \wedge (\overline{m}(k'))_i = \omega.$$

If $i \in I_v$ is a non-essential coordinate it can be discarded from the construction after the first run of the loop because if both $\overline{m}_i$ and $\overline{m}'_i$ equal ω then all $(\overline{m}(k''))_i$ of all nodes k'' of all RCRG's constructed in Algorithm 2 equal ω, and if, say, $(\overline{m}(k'))_i < \omega$, then $(\overline{m}(k''))_i < \omega$ for all nodes k'' in all RCRG's constructed by the algorithm henceforth, and $(\overline{m}(k''))_i = (\overline{m}(z(k'')))_i$, and can, thus, be determined inductively from B'''. The proof of termination is now by induction on the number E of essential coordinates.
Assume that the loop in the algorithm is traversed at least twice, and let now B^{-1}, B^1, and B^3 denote the admissible RCRG's determined in step c) of the first and steps a) resp. c) of the second run of the loop, in this order.
If $E = 0$ the algorithm obviously stops after the second execution of the loop.
If $E = 1$ assume that coordinate i is essential, and also assume wlg. that B^1 contains some ω^+-node k'' (for coordinate i). Then, because of the above remark, $(\overline{m}(z(k'')))_i = \omega$ in B^{-1}, and the ω^+-node in B^{-1} is before or at $z(k'')$ in B^{-1} (if there is none, obviously $B^3 = B^1_{rev}$ because of property P6). But then the subgraph of B^1 reachable from k'' equals the subgraph of B^{-1}_{rev} reachable from $z(k'')$ because of P6, and, for the same reason, $B^3 = B^1_{rev}$, i.e. Algorithm 2 stops.
For general E, let B^{-1}, B^1, and B^3 be as above, let $W'(k)$ be the set of essential ω-coordinates of $\overline{m}(k)$, and let k'' be the last ω^+-node in B^1.
(i) If $W'(k'') = W'(z(k''))$, Algorithm 2 stops after the second run through the loop because of the same reasoning as above.
(ii) If $W'(k'') \subsetneq W'(z(k''))$, the number of essential coordinates decreases as k'' is the last ω^+-node in B^1.
(iii) If $W'(z(k'')) \subsetneq W'(k'')$, let k''' be the last node on the simple path between the two end nodes in B^1 (with edge e

leaving k''' on this path) such that $z(k''')$ is the last ω^+-node in B^{-1}. This implies that $W'(k''') \subseteq W'(z(k'''))$ as any ω-coordinate $(\overline{m}(k'''))_i$, $i \in W'(k''') - W'(z(k'''))$, would not be essential.
Now, if

$$W'(k''') = W'(z(k'''))$$

we reason as above.
If

$$W'(k''') \subsetneq W'(z(k'''))$$

there is an essential coordinate i_r s.t. $(\overline{m}(z(k)))_{i_r}$ is finite for all k in B^1 reachable from $d(e)$. because $z(k''')$ is ω^+-node in B^{-1}. and there is an essential coordinate i_l s.t. $(\overline{m}(k))_{i_l}$ is finite for all nodes k in B^1 from which k''' is reachable. Now assume that the algorithm does not terminate. Then there would be an infinite chain of B' 's generated in subsequent executions of step a), and in each such B'^j exactly one edge e^j would correspond to e (as e connects two SCC's in B^1). Because $\overline{\mathbb{N}}^v$ under $\subset$ is well-founded, each of the sequences $(\overline{m}(z(a(e^j))))_{j \in \mathbb{N}}$, and $(\overline{m}(d(e^j)))_{j \in \mathbb{N}}$ would have to become stationary. But this would mean that Algorithm 2 doesn't stop for some problem given by $\overline{m}$, some $\overline{m}(z(a(e^j)))$, and the subgraph of B'^j from which $a(e^j)$ is reachable, or given by $\overline{m}(d(e^j))$, $\overline{m}'$, and the subgraph of B'^j reachable from $d(e^j)$. However, both of these problems have less than E essential coordinates.
(iv) If $W'(z(k'')) \not\subseteq W'(k'')$ and $W'(k'') \not\subseteq W'(z(k''))$, a construction analogous to the second part of (iii) works, this time using the edge e on the simple path between the end nodes of B^1 with $d(e) = k''$.
This then concludes the proof for the termination of Algorithm 2.

Let B' be the graph determined in the last execution of step a) in Algorithm 2. As $B' = RCRG(P,\overline{m},B') = (RCRG(P_{rev},\overline{m}',B'_{rev}))_{rev}$ there are two possibilities to attach a pseudomarking $\overline{m}(e)$ to every edge e in B' :

$$\overline{m}(e) =_{\text{def}} \overline{m}(a(e)) - (\delta t(e))^-$$

taken from $RCRG(P,\overline{m},B')$, or

$$\overline{m}(e_{rev}) =_{\text{def}} \overline{m}(a(e_{rev})) - (\delta t(e_{rev}))^-$$

taken from $RCRG(P_{rev},\overline{m}',B'_{rev})$
(note that $\delta t(e_{rev}) = -\delta t(e)$).

Let the edge marking $m2(e)$ be the componentwise minimum of $\overline{m}(e)$ and $\overline{m}(e_{rev})$, for all edges E in B' (note that $\overline{m}(e) \simeq \overline{m}(e_{rev})$).

A graph B' output in this way by Algorithm 2 together with edge marking $m2$ is called a *two-sided regularly controlled reachability graph* $2SRCRG(P,\overline{m},\overline{m}',A)$. We note the following basic property (corresponding to P3 for RCRG's)

P7: Let $m, m' \in \mathbb{N}^v$. $\overline{m}, \overline{m}' \in \overline{\mathbb{N}}^v$, $m \subset \overline{m}, m' \subset \overline{m}'$. For any path in A starting from the initial state z_0 with edge labelling sequence τ s.t. $m \xrightarrow{\tau} m'$ there is a $2SRCRG(P,\overline{m},\overline{m}',A)$ B' with end nodes r and k s.t. there is a path α in B' from r to k with $\tau(\alpha) = \tau$. For every initial segment $\alpha'e$ of α ending with edge e we have

$$m + \delta\tau(\alpha') - (\delta t(e))^- \subset m2(e)$$

(P7 essentially means that every firing sequence accepted by A is also an edge labelling sequence in some 2SRCRG constructed by Algorithm 2, and gives a compatibility property for the node markings $\overline{m}$).

A proof for P7 can be obtained by an iterated and alternating application of P3 to P and P_{rev}.

4. Formally admissible paths

Let B be some $2SRCRG(P,m,m',A)$ with $m, m' \in \mathbb{N}^v$, end nodes r and k, α the simple path from r to k, $e^1, \ldots, e^p$ the sequence of edges on α, and $r = k^0 = a(e^1), k^1 = d(e^1), \ldots, k^p = d(e^p) = k$ the sequence of nodes on α. Let β be a path from r to k, $i \in I_p$. Then

a) β_i denotes the (uniquely determined) initial segment of β ending with e^i.
b) $\delta^-\tau(\beta_i) =_{\text{def}} \delta\tau(\beta_i) - (\delta t(e) + (\delta t(e))^-)$;
c) β is *formally admissible* iff
$$m + \delta\tau(\beta) = m', \text{ and}$$
$$(\forall i \in I_p)[m + \delta^-\tau(\beta_i) \geq 0]$$
(i.e. the total effect of $\tau(\beta)$ is $m' - m$, as desired, and all initial segments β' of β ending with edges on the simple path from r to k satisfy the nonnegativity condition $m + \delta\tau(\beta') \geq 0$, while for other initial segments β'' , $m + \delta\tau(\beta'')$ may still contain negative coordinates);
$$FAP(B) =_{\text{def}} \{\beta; \beta \text{ formally admissible path in } B \text{ from } r \text{ to } k\}.$$
d) Let β, β', β'' be paths in B, $n \in \mathbb{N}$.
$\beta \supset n\beta' + \beta'' \Leftrightarrow_{\text{def}} \beta$ contains each edge of B at least as often as do β'' and n copies of β' together.
e) Let $\beta, \beta' \in FAP(B)$. Define $\beta' \geq \beta$ iff
$$\beta' \supset \beta \wedge (\forall i \in I_p)[\delta^-\tau(\beta'_i) \geq \delta^-\tau(\beta_i)].$$

Let now q be the number of edges in B, and Φ' the Parikh mapping from the set of paths in B to $\mathbb{N}^q$. Then, by Parikh's Lemma [31] and the closure properties of semilinear sets [8],

$$PP_B := \Phi'(FAP(B))$$

is an effectively obtainable semilinear set, and so is every *component* L of PP_B (i.e. subset of PP_B of the form $L = \{\Phi'(\beta) \in PP_B; \beta \geq \alpha\}$, where $\Phi'(\alpha)$ is some base of PP_B). PP_B is the union of finitely many components. Note that if L is a component of PP_B, and $\beta, \beta', \beta'' \in FAP(B)$, $\beta' \geq \beta$, $\beta'' \geq \beta$, and $\Phi'(\beta), \Phi'(\beta'), \Phi'(\beta'') \in L$, then

$$\Phi'(\beta) + (\Phi'(\beta') - \Phi'(\beta)) + (\Phi'(\beta'') - \Phi'(\beta)) \in L.$$

Let $i \in I_v$ be an ω-component of some $m2(e^j)$, for some $j \in I_p$. This ω-component is called *formally justified* (wrt. L) if the i-th coordinate of the constructable semilinear set

$$\{m + \delta^- \tau(\beta_i); \beta \in FAP(B) \wedge \Phi'(\beta) \in L\}$$

is unbounded, otherwise the (because of the above property of L unique) finite value of the i-th coordinate of all its elements is called its *finite replacement.*

It is possible that b_i is constant for all $b \in L$ and some $i \in I_q$ corresponding to some edge *within* an SCC of B. In this case, every admissible path β with $\Phi'(\beta) \in L$ has to contain this edge exactly b_i times. *Modifying* B wrt. L means modifying B as a finite automaton by standard techniques in such a way that it still allows all paths in $\Phi'^{-1}(L)$ but forces all paths (from one end node to the other) to contain every such edge the appropriate number of times. This can be achieved by coding finite counters into B and then selecting, nondeterministically, an admissible subgraph which w.l.g. is assumed separated.

Finally, if e and e' are two edges on the simple path in B between the two end nodes, let $C_B(e, e')$ denote the subgraph of B consisting of all paths from $d(e)$ to $a(e')$, together with e and e'.

The following algorithm eliminates formally unjustified ω-components in 2SRCRG's by segmenting these at those edges on the simple path between the end nodes containing formally unjustified ω-components, applying Algorithm 2 to each of the segments, and then recombining the segments. For the last step, we have to make sure that the $m2$-markings of the edges being combined fit together. This may require iterating Algorithm 2 for a segment some bounded number of times.
Let P and $m, m' \in \mathbb{N}^v$ be given.

Algorithm 3:

a) Construct $B := 2SRCRG(P, m, m', E)$, where E accepts T^*.
Remove cycles through either one of the end nodes of B.
Let SE be the set of edges on the simple path between the end nodes in B.

b) Determine PP_B and select nondeterministically some component L of PP_B. If all ω-components of all edge markings $m2(e)$ of all edges on the simple path between the two end nodes in B are formally justified wrt. L, and if B needn't be modified wrt. L then go to i).

c) Modify B wrt. L, obtaining B'. Let $(e^j; j = 1, \ldots, p)$ be the ordered sequence of edges on the simple path between the two end nodes in B which are also in SE, and let e'^j be the (unique) edge in B' corresponding to e^j, $j \in I_p$. Let, further, $m2'(e^j)$ be $m2(e^j)$ with formally unjustified ω-components finitely replaced, for $j \in I_p$.
Set $j := 1$, and for $l \in I_{p-1}$, $ma^l := m2'(e^l)$, and $md^l := m2'(e^{l+1})$.

d) Construct $B^j := 2SRCRG(P, ma^j + \delta^- t(d^j),$
$$md^j + \delta t(e^{j+1}) + \delta^- t(e^{j+1}), C_{B'}(e'^j, e'^{j+1}));$$
let ea^j be the initial edge in B^j (corresponding to e'^j), ed^j the final edge (corresponding to e'^{j+1}), and set $ma^j := m2(ea^j)$; $md^j := m2(ed^j)$.

e) If $j > 1$ and $ma^j < md^{j-1}$ then set $j := j - 1$ and go to d).

f) If $j < p - 1$, set $ma^{j+1} := md^j; j := j + 1$, and go to d).

g) Construct B'' : combine the segments B^j by identifying ed^j with ea^{j+1}, for $j \in I_{p-1}$.

h) Add, for all $j \in I_{p-1}$, to SE all edges on the simple path between the two end nodes in B^j if the maximal number of ω-coordinates of markings in B^j has decreased wrt. $C_B(e^j, e^{j+1})$ or, if not, if the number of edges in some SCC with markings with a maximal number of ω-coordinates has decreased.
Set $B := B''$ and go to b).

i) Reattach the cycles removed in step a), calling the resulting graph B again.

end Algorithm 3

It can easily be seen that the loops formed in steps e) and f), resp., of the algorithm terminate. To show termination of the loop formed in step h) it can be proven that in one pass through that loop a segment B^j is not changed at all and all ω-components within it are formally justified if the number of ω-components in the $m2$-markings of the two end nodes, the maximal number of ω-coordinates of $m2$-markings in B^j, and the number of edges in some SCC C_{max} with $m2$-markings with a maximal number of ω-coordinates all are preserved. The reason for this is that in this case C_{max} is a copy of the corresponding SCC in B and, therefore, contains a cycle with the same edge labelling sequence as some appropriate cycle (i.e. increasing new ω-components, if there are any) containing every edge of, say, the first SCC C' in B^j. Thus, there is a (not connected) cycle which covers C' as well as C_{max} and whose Parikh image is contained in L as L is a component of PP_B. Induction on the number of SCC's in B^j concludes this argument the details of which are left to the reader. Let the four numbers above be, in order, nl_j, nr_j, nm_j, and ne_j. Therefore, attaching to each segment B^j the quadruple

(nm_j, ne_j, nl_j, nr_j), each of these quadruples in one run through the loop either remains the same and the corresponding segment is not changed, or is replaced by a finite multiset of quadruples which are all strictly smaller in the lexicographic ordering than the one being replaced. As this ordering is well-founded the algorithm must terminate [6].

A graph B produced by Algorithm 3 is called *reduced regularly controlled reachability graph* $RR(P, m, m')$.

Theorem 1:
Let $m, m' \in \mathbb{N}^v$.
If $m \xrightarrow{T} m'$ then there is some $RR(P, m, m')$ B s.t.
$\tau = \tau(\beta)$ for some $\beta \in FAP(B)$ with $\Phi'(\beta) \in L$
(L as determined in the last execution of step b) of Algorithm 3).

Proof:
Apply P7 to all segments constructed in Algorithm 3. ∎

The converse of Theorem 1 also holds, i.e. $FAP(B)$ contains some β s.t. $\tau(\beta)$ is a firing sequence from m to m' The proof of this is the subject of the next section.

5. A sufficient condition for reachability

Let B be some $RR(P, m, m')$ for $m, m' \in \mathbb{N}^v, e^1, \ldots, e^p$ the sequence of edges on the simple path between the two end nodes, and L the component of PP_B determined in the last execution of step b) in Algorithm 3. For $b \in L$ and $j \in I_p$, set $m(b, j) =_{\text{def}} m + \delta\tau(\beta_j)$ where $\Phi'(\beta) = b$ (note that $\delta\tau(\beta)$ only depends on $\Phi'(\beta)$).
Let $b, b + b^* \in L$ s.t.

i) $\Phi'(e) \leq b^*$ for every edge e on a cycle in B;
ii) $b^n =_{\text{def}} b + nb^* \in L$ for all $n \in \mathbb{N}$;
iii) $(m(b^1, j))_i > (m(b, j))_i$, for all $j \in I_p$, and all ω-coordinates i of $m2(e^j)$.

Let, finally, for $j \in I_p \cup \{0\}$, β^j and α^j be paths from $d(e^j)$ to $d(e^j)$ (where $d(e^0) =_{\text{def}} a(e^1)$) s.t.

$$\sum_{j=0}^{p} \Phi'(\beta^j) = b - \sum_{j=1}^{p} \Phi'(e^j) \quad \text{and} \quad \sum_{j=0}^{p} \Phi'(\alpha^j) = b^*$$

($b, b^*, \beta^j, \alpha^j$ all exist, e.g. b^* because of the properties of L determined by Algorithm 3 and the closure property of L noted in the last section).

Lemma:
Let $j \in I_{p-1}$.
Then $(\exists n \in \mathbb{N}, \forall n' \geq n)[m(b^{n'}, j) \xrightarrow{*} m(b^{n'}, j+1)]$.

Proof:
Set, for $j \in I_p$, $W_j =_{\text{def}}$ the set of ω-coordinates of $m2(e^j)$, and $\overline{W}_j =_{\text{def}}$ the maximal set of ω-coordinates of $m2(e)$ for edges e incident on $d(e^j)$.
For $\tau \in T^*$ and $n \in \mathbb{N} - \{0\}$, let τ^n denote the n-fold iteration of τ : $\tau^1 = \tau, \tau^{n+1} = \tau\tau^n$.

Case 1: $W_j = \overline{W}_j = W_{j+1}$

Choose n big enough s.t. $m(b^n, j) \geq \delta(\tau(\alpha^j)\tau(\beta^j))^-$.
Then, because of iii) above,

$$f((\tau(\alpha^j))^{n'}\tau(\beta^j), m(b^{n'}, j))$$

for all $n' \geq n$.

Case 2: $W_j \subsetneq \overline{W}_j = W_{j+1}$

Let γ be the ω^+-path in $d(e^j)$, and γ' some path from $d(e^j)$ to $d(e^j)$ s.t. $\Phi'(\gamma) + \Phi'(\gamma') = n''\Phi'(\alpha^j)$ for some $n'' \in \mathbb{N}$ (such a γ' exists!).
Choose $\bar{n}$ big enough s.t.

$$(m(b^{\bar{n}}, j) + \bar{n}\delta\tau(\gamma))_i \geq (\delta(\tau(\alpha^j)\tau(\beta^j))^-)_i$$

for all $i \in \overline{W}_j - W_j$, and $n > n''\bar{n}$ big enough s.t.

$$m(b^n, j) \geq \delta((\tau(\gamma))^{\bar{n}}\tau(\alpha^j)(\tau(\gamma'))^{\bar{n}}\tau(\beta^j))^-.$$

Then, because of iii) above,

$$f((\tau(\gamma))^{\bar{n}}(\tau(\alpha^j))^{n'-n''\bar{n}}(\tau(\gamma'))^{\bar{n}}\tau(\beta^j), m(b^{n'}, j))$$

for all $n' \geq n$.

The remaining two cases,

$$W_j \subsetneq \overline{W}_j \supsetneq W_{j+1} \text{ and } W_j = \overline{W}_j \supsetneq W_{j+1},$$

are dealt with similarly. In the first case, appropriate "short" pieces are extracted at both ends. ∎

Theorem 2:
Let $m, m' \in \mathbb{N}^v$. If there is some $RR(P, m, m')$ then $m \xrightarrow{*} m'$.

Proof:
Choose n big enough for all $j \in I_{p-1}$ and apply the above Lemma. ∎

Corollary:
The General Reachability Problem for Petri nets is decidable.

6. Conclusion

There follows a list of some problems which have been shown to be effectively reducible to ($\leq$) or equivalent with ($\equiv$) the general reachability problem for Petri nets (also see [27]):

a) ($\equiv$) the liveness problem for Petri nets [14];
b) ($\equiv$) the zero marking reachability problem [14];
c) ($\equiv$) the persistence problem for one transition [14];
d) ($\leq$) the containment and equivalence problem for sets of firing sequences [14];
e) ($\equiv$) the word problem for commutative Semi-Thue-Systems [21];
f) ($\equiv$) the word problem for $PBLIND$ [10];
g) ($\equiv$) the emptiness problem for the intersection of two Szilard languages [5];
h) ($\equiv$) the recursiveness of subsets of finitely presented commutative semigroups compatible with relations with finitely many minimal elements [11].

As shown in [23], the reachability problem for Petri nets is exp-space hard. In [25], Petri nets with finite, non-primitive recursive reachability sets are exhibited implying that Algorithm 3 is non-primitive recursive, too, in this case, as the reachability graph construction enumerates the whole reachability set. However, as shown in [27], necessary conditions derived from the equation $m' - m = V\Phi(\tau)$ can be built into the Petri net, and it is not clear how this modification affects the complexity of the algorithm. Also, it is still open whether Dickson's Lemma [7] which is used implicitly several times and which implies that every infinite sequence in $\mathbf{N}^n$ has an infinite nondecreasing subsequence, can be replaced by a different argument providing effectively given upper bounds.
Other open problems concerning the reachability sets of Petri nets are, e.g.:

a) Is there a live marking in $R(P, m)$?
b) Is there a marking m s.t. $R(P, m)$ is live?
c) Is $R(P, m)$ semilinear?
d) Is there a "small" bound $S_P(m, m')$ for $m' \in R(P, m)$ s.t. m' is reachable via intermediate markings which are all bounded by $S_P(m, m')$?

It is hoped that the techniques shown in this paper can also be applied to some other word problems (e.g., in monotonous systems characterized by the property that transitions possible in some state are also possible in all "bigger" states) where no a priori upper bounds on the length of shortest derivations are known so far.

Acknowledgement: I would like to thank R. Kannan and J.C. Lagarias for pointing out some errors in an earlier version and making invaluable suggestions to improve the presentation.

7. References

1. Araki, T., Kasami, T.: Decidable Problems on the Strong Connectivity of Petri Net Reachability Sets. Theoret. Comp. Sci. 4, 1 (Febr. 1977), p.99–119
2. Baker, H.G.: Rabin's Proof of the Undecidability of the Reachability Set Inclusion Problem of Vector Addition Systems. M.I.T., Project MAC, CSGM 79 (July 1973)
3. Berthelot, G. Roucairol, G.: Reduction of Petri Nets. In: Mazurkiewicz, A. (Ed.): Mathematical Foundations of Computer Science 1976, Gdansk. Berlin-Heidelberg-New York: Springer 1976, p.202–209
4. Commoner, F.: Deadlocks in Petri Nets. Applied Data Research, Inc. (Wakefield), CA-7206-3211 (1972)
5. Crespi-Reghizzi, S., Mandrioli, D.: Petri Nets and Szilard Languages. Information and Control 33, 2 (Febr. 1977), p.177–192
6. Dershowitz, N., Manna, Z.: Proving Termination with Multiset Orderings. Comm. ACM 22, 8 (August 1979), p.465–476
7. Dickson, L.E.: Finiteness of the Odd Perfect and Primitive Abundant Numbers with n Distinct Prime Factors. Amer. J. Math. 35 (1913), p.413–422
8. Ginsburg, S., Spanier, E.H.: Semigroups, Presburger Formulas, and Languages. Pacific J. of Math. 16 (1966), p.285–296
9. Grabowski,J.: The Decidability of Persistence for Vector Addition Systems. IPL 11, 1 (1980), p.20–23
10. Greibach, S.A.: Remarks on Blind and Partially Blind One-Way Multicounter Machines. Theoret. Comp. Sci. 7, 3 (Dec. 1978), p.311–324
11. Hack, M.: Petri Nets and Commutative Semigroups. M.I.T., Project MAC, CSGN 18 (July 1974)
12. Hack, M.: The Recursive Equivalence of the Liveness Problem and the Reachability Problem for Petri Nets and Vector Addition Systems. Proceedings of the 15th Ann. Symp. on SWAT, New Orleans, La. (Oct. 1974), p.156–164
13. Hack, M.: Decision Problems for Petri Nets and Vector Addition Systems. M.I.T., Project MAC, MAC-TM 59 (March 1975)
14. Hack, M.: Decidability Questions for Petri Nets. M.I.T., LCS, TR 161 (June 1976)
15. Herzog, O.: Liveness of Extended Control Structure Nets. University of Utah (Salt Lake City), Dept. of CS, TR CSUU-77-107 (August 1977)
16. Holt, A.W., et al.: Final Report of the Information Systems Theory Project. Griffiss Air Force Base, Rome Air Development Center (N.Y.), RADC-TR-68-305 (1968)
17. Holt, A.W., Commoner, F.: Events and Conditions. Record of the Project MAC Conference on Concurrent Systems and Parallel Computation. New York: ACM (1970), p.3–52
18. Hopcroft, P., Pansiot, J.J.: On the Reachability Problem for 5-Dimensional Vector Addition Systems. Theoret. Comp. Sci. 8 (1979), p.135–159
19. Jones, N.D., Landweber, L.H., Lien, Y.E.: Complexity of Some Problems in Petri Nets. Theoret. Comp. Sci. 4 (1977), p.277–299
20. Karp, R.M., Miller, R.E.: Parallel Program Schemata. JCSS 3, 2 (May 1969), p.147–195
21. Keller, R.M.: Vector Replacement Systems: A Formalism for Modelling Asynchronous Systems. Princeton University, CSL, TR 117 (Dec. 1972)
22. Lautenbach, K.: Exakte Bedingungen der Lebendigkeit für eine Klasse von Petri-Netzen. GMD Bonn (St. Augustin). Bericht Nr. 82 (1973)
23. Lipton, R.: The Reachability Problem is Exponential-Space-Hard. Yale University, Dept. of CS, Report No. 62 (Jan. 1976)
24. Mayr, E.W.: Einige Sätze über Umformungen und verklemmungsfreie Führbarkeit bei bewerteten Petrinetzen. Technical University of Munich, Institut für Informatik, Diplomarbeit (August 1975)
25. Mayr, E.W.: The Complexity of the Finite Containment Problem for Petri Nets. M.I.T., LCS, TR 181 (June 1977)
26. Mayr, E.W.: Persistence of Vector Replacement Systems is Decidable. M.I.T., LCS, TM 189 (Dec. 1980); also to appear in Acta Informatica
27. Mayr, E.W.: Ein Algorithmus für das allgemeine Erreichbarkeitsproblem bei Petrinetzen und damit zusammenhängende Probleme. Technical University of Munich, TUM-I8010 (Sept. 1980)
28. Miller, R.E.: A Comparison of Some Theoretical Models of Parallel Computation. IEEE Trans. C-22, 8 (1973), p.710–717
29. Müller, H.: Decidability of Reachability in Persistent Vector Replacement Systems. Proc. 9th Symp. on MFCS (1980). LNCS 88. Springer 1980, p.426–438
30. Oppen, D.C.: A $2^{2^{2^{pn}}}$ Upper Bound on the Complexity of Presburger Arithmetic. JCSS 16 (1978), p.323–332
31. Parikh, R.J.: On Context-free Languages. JACM 13 (1966), p.570–581
32. Peterson, J.L., Bredt, T.H.: A Comparison of Models of Parallel Computation. Proc. IFIP Congress 74. North Holland Publ. Comp. (1974), p.466–470
33. Petri, C.A.: Kommunikation mit Automaten. Institut für Instrumentelle Mathematik (Bonn), Schriften des IMM Nr. 2 (1962)
34. Presburger, M.: Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. Compte-Rendus du I. Congrès des Mathématiciens des pays Slavs, Warsaw (1930), p.92–101
35. Sacerdote, G., Tenney, R.L.: The Decidability of the Reachability Problem for Vector Addition Systems. Proc. of the 9th Ann. STOC (1977), p.61–76
36. van Leeuwen, J.: A Partial Solution to the Reachability Problem for Vector Addition Systems. Proceedings of the 6th Ann. STOC (1974), p.303–309