Acta Informatica 15, 309–318 (1981)



# Persistence of Vector Replacement Systems is Decidable

Ernst Mayr*

Institut für Informatik, Technische Universität München, Arcisstr. 21,
8000 München 2, Germany (Fed. Rep.)

**Summary.** In a persistent vector replacement system (VRS) or Petri net, an enabled transition can become disabled only by firing itself. Here, an algorithm is presented which allows to decide whether an arbitrary VRS is persistent or not, and if so, to construct a semilinear representation of the set of states reachable in the system.

## 1. Introduction

In this paper, an effective algorithm is presented which for any given arbitrary vector replacement system (VRS) [7] allows to decide whether it is persistent or not. This algorithm is self-contained, based on a recursive construction of semilinear representations for subsets of the counter set of the VRS. If the VRS is persistent, the whole counter set is obtained by the algorithm thus also giving a semilinear representation of the reachability set. For further motivation to study persistent systems the reader is referred to [6, 10, 12, 16]. In [5], it is proved that the decision problem for the persistence of one transition in a VRS or Petri net is recursively equivalent to the decidability of the reachability problem, but it is also conjectured that the persistence of a VRS or Petri net can be decided completely independent from the reachability problem. The algorithm presented in the sequel does not rely on an algorithm for the general reachability problem. Throughout the paper, VRS terminology is used; the transition to and from Petri nets is straightforward [4].

* *Author's present address:* Laboratory for Computer Science, M.I.T., Cambridge, MA 02139, USA

0001-5903/81/0015/0309/$02.00

## 2. Notation and Preliminaries

A *Vector Replacement System* (VRS) [7] is a pair $(T, m)$ where $T=\{t_1, \ldots, t_w\}$ is a finite set of transitions $t_i=(u_i, v_i)\in \mathbb{N}^v \times \mathbb{Z}^v$ [1] with $u_i+v_i \geqq 0$ and $m\in \bar{\mathbb{N}}^v$ is the *initial vector* ($v\in\mathbb{N}$). $t_i\in T$ is *applicable at* $m'\in\bar{\mathbb{N}}^v$ iff $u_i \leqq m'$ (written $a(t_i, m')$), application of $t_i$ at $m'$ takes $m'$ to $m'+v_i$ (written $m' \xrightarrow{t_i} m'+v_i$).

For $\tau=t_{i_1}\ldots t_{i_r}\in T^*$ we define inductively

i) $a(\tau, m') :\Leftrightarrow r=0 \vee a(t_{i_1}, m') \wedge a(t_{i_2}\ldots t_{i_r}, m'+v_{i_1})$;

ii) $\delta\tau := \sum_{j=1}^{r} v_{i_j}$;

iii) $m' \xrightarrow{\tau} m'' :\Leftrightarrow a(\tau, m') \wedge (m''=m'+\delta\tau)$.

The *reachability set* $R(T, m)$ of $(T, m)$ is $R(T, m) := \{m'; (\exists \tau\in T^*)[m \xrightarrow{\tau} m']\}$.

Let $\Phi$: $T^* \to \mathbb{N}^w$ denote the Parikh mapping. The *counter set* $C(T, m)$ of $(T, m)$ is given by $C(T, m) := \{\Phi(\tau); \tau\in T^* \wedge a(\tau, m)\}$.

Let $V\in\mathbb{Z}^{v, w}$ be the integer matrix whose $i$-th column is $v_i$, $i\in I_w$. Obviously, we have

i) $(\forall \tau\in T^*)[\delta\tau = V\Phi(\tau)]$;

ii) $R(T, m)=\{m+Vc; c\in C(T, m)\}$.

A *linear set* $L\subseteq\mathbb{N}^w$ is a set of the form $L=\left\{a+\sum_{i=1}^{r} n_i b_i; (n_1, \ldots, n_r)\in\mathbb{N}^r\right\}$ for some $r\in\mathbb{N}$, $a, b_1, \ldots, b_r\in\mathbb{N}^w$. 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 [15]. There is an effective procedure to construct semilinear representations of the sets defined by Presburger expressions [3, 13].

*Definition 1.* A vector $m'\in\bar{\mathbb{N}}^v$ is *persistent* iff

$$(\forall t_i, t_j\in T)[(i\neq j \wedge a(t_i, m') \wedge a(t_j, m')) \Rightarrow a(t_i t_j, m')].$$

$(T, m)$ is persistent iff all $m'\in R(T, m)$ are persistent.

It is known that for a persistent VRS $(T, m)$, $R(T, m)$ and $C(T, m)$ are effectively constructable semilinear sets [9, 11].

The following algorithm for the construction of the *reachability graph* $RG(T, m)$ works for arbitrary VRS's, it does not assume persistence. In this algorithm, which is a slight modification of one originally given in [6], a digraph with labelled nodes and edges is constructed: The label $t(e)$ of an edge $e$ is an element of $T$, and each node $k$ obtains a label $\bar{m}(k)\in\bar{\mathbb{N}}^v$. (In pictures, parallel edges are merged into one which receives the union of the edge labels.) From these labels, marks have to be distinguished which in the algorithm serve to decide which nodes still have to be dealt with.

---

[1] $\mathbb{N}$ denotes the set of nonnegative integers, $\mathbb{Z}$ the set of integers, and $\bar{\mathbb{N}} := \mathbb{N}\cup\{\omega\}$ the set $\mathbb{N}$ augmented by the "infinite" number $\omega$ with $\pm n+\omega=\omega\pm n=\omega$ and $n<\omega$ for all $n\in\mathbb{N}$. For $i\in\mathbb{N}$, $I_i$ stands for the set $\{1, \ldots, i\}$

## Algorithm 1

```
begin
   start with an unmarked node r (the "root") with m̄(r) := m;
   while there is an unmarked node do
      select nondeterministically an unmarked node k;
      mark k;
      for all j∈I_w with a(t_j, m̄(k)) do
         add to the graph constructed so far a new unmarked node k' and an
         edge e from k to k' with t(e) := t_j;
         for i := 1, ..., v do
            if there is a node k'' on a (not necessarily simple) path from r to k
               with m̄(k'') ≦ m̄(k) + v_j and (m̄(k''))_i < (m̄(k) + v_j)_i
            then
               (m̄(k'))_i := ω
            else
               (m̄(k'))_i := (m̄(k) + v_j)_i
            fi
         od;
         if there is a node k'' ≠ k' in the graph constructed so far with m̄(k'') = m̄(k')
        then
            identify k' with this k''
         fi
      od
   od
end Algorithm 1.
```

Note that new $\omega$-components are introduced independently for different coordinates as for each coordinate $i$ a different node $k''$ may be found.

The proof of the termination of Algorithm 1 is very similar to the one given in [6] and won't be presented here.

*Example.* For the (persistent) VRS $(T, m)$ with

$$T = \{(0^6, 10^5), (10^5, 012100 - 10^5), (0210^3, 0^4 10 - 0210^3),$$
$$(0^2 1^2 0^2, 0^5 1 - 0^2 1^2 0^2), (0^4 1^2, 10^5 - 0^4 1^2)\}$$

and $m = 10^5$ (which is short for $(1,0,0,0,0,0) \in \mathbb{N}^6$) some run of Algorithm 1 produces the graph $RG(T, m)$:

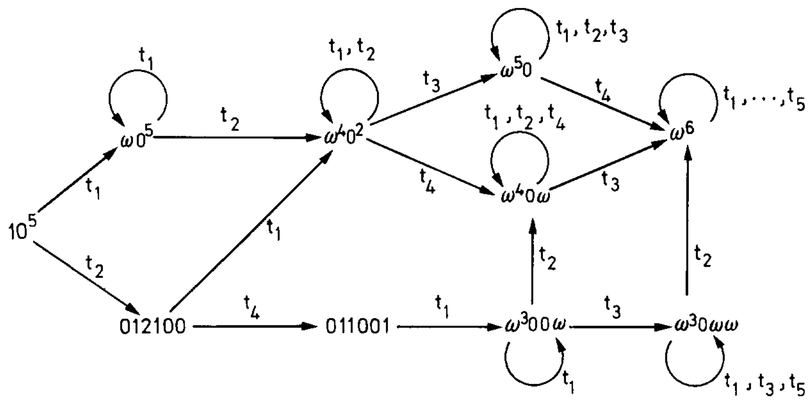


**Fig. 1**

**Lemma 1.** *Given $(T, m)$, the set $NPC(T, m) := \{c \in \mathbb{N}^w; m + Vc$ is $\geqq 0$ and not persistent$\}$ is an effectively constructable semilinear set.*

*Proof.* By Definition 1, $NPC(T,m)$ equals the set $\{c \in \mathbb{N}^w; (\exists t_i, t_j \in T)\ [i \neq j \wedge m + Vc \geqq u_i \wedge m + Vc \geqq u_j \wedge m + Vc + v_i \ngeqq u_j]\}$. Hence, $NPC(T,m)$ can be defined in Presburger Arithmetic, and a semilinear representation can effectively be found. q.e.d.

## 3. Some Properties of $RG(T, m)$

Let $k$ be a node in $RG(T, m)$ for some arbitrary VRS $(T, m)$.

**Lemma 2.** *The sets*

$$CT_k := \{\Phi(\tau); \tau \text{ is the edge-labelling sequence of a path in } RG(T, m) \text{ from } k \text{ to } k\}, \text{ and}$$
$$CT_k^+ := \{c \in CT_k;\ Vc \geqq 0\}$$

*are effectively constructable semilinear sets.*

*Proof.* Regarding the strongly connected component (SCC) of $k$ in $RG(T, m)$, stripped of the node marking $\bar{m}$, as the transition diagram of a finite automaton over $T$, the set $CT_k$ corresponds to the Parikh image of the regular language accepted by that finite automaton with $k$ as initial and single final state. Hence, by Parikh's Lemma [14], $CT_k$ is an effectively constructable semilinear set. As $CT_k^+ = \{c \in CT_k;\ Vc \geqq 0\}$, and as systems of linear inequalities are expressible in Presburger Arithmetic the claim of the lemma follows from the fact that semilinear sets are effectively closed under Boolean operations [3]. q.e.d.

*Definition 2.* For $\bar{m} \in \bar{\mathbb{N}}^v$ and $N \in \mathbb{N}$ set

$$F(\bar{m}, N) := \{m' \in \mathbb{N}^v; (\forall i \in I_v)\ [(\bar{m}_i = \omega \wedge m'_i \geqq N) \vee (\bar{m}_i = m'_i)]\}.$$

In the sequel, we shall make use of the following basic properties of $RG(T, m)$, proved in [4, 5]:

a) For any given $N \in \mathbb{N}$ and node $k$ in $RG(T, m)$, one can effectively find some $\tau \in T^*$ (and hence $\Phi(\tau) \in \mathbb{N}^w$) s.t. $a(\tau, m)$ and $m + \delta\tau \in F(\bar{m}(k), N)$.

b) For any given node $k$ in $RG(T, m)$, there is a path from the root $r$ to $k$ with edge-labelling sequence $\tau$ s.t. $a(\tau, m)$.

**Lemma 3.** *Let $(T, m)$ be a persistent VRS and $k$ a node in $RG(T, m)$. Let further $p$ be the projection of $\mathbb{N}^v \cup \mathbb{Z}^v$ on those coordinates where $\bar{m}(k)$ is not equal $\omega$. (Note that two transitions $p(t_i)$ and $p(t_j)$ with $i \neq j$ and $p(t_i) = p(t_j)$ are considered different.) Then $(p(T), p(\bar{m}(k)))$ is persistent.*

*Proof.* Assume that $(p(T), p(\bar{m}(k)))$ is not persistent. Then there are $\tau \in T^*$ and $t_i, t_j \in T$ with $i \neq j$ s.t. (with $m' := \bar{m}(k) + \delta\tau$)

$$a(p(t_i), p(m')) \wedge a(p(t_j), p(m')) \wedge \neg a(p(t_i)\, p(t_j), p(m')). \qquad (*)$$

Now, effectively find some $\tau \in T^*$ s.t. $a(\tau, m)$ and $m + \delta\tau \in F(\bar{m}(k), N)$. Choosing $N$ big enough one could obtain $m'' \in R(T, m) \cap F(\bar{m}(k), N)$ s.t. $a(\tau t_i, m'') \wedge a(\tau t_j, m'')$.

From (*) then follows $\neg a(\tau t_i t_j, m'')$, contradicting the persistence of $(T, m)$. q.e.d.

*Definition 3.* Let $(T, m)$ be an arbitrary VRS.

a) A transition $t \in T$ is *bounded* in $(T, m)$ iff $(\exists N \in \mathbb{N}, \nexists c \in C(T, m))\ [N\Phi(t) \leqq c]$.

b) A strongly connected component (SCC) CC in $RG(T, m)$ is called *distinguished* iff the labels of all edges leaving CC do not appear as labels of edges within any SCC.

It has been shown in [5] that it is decidable whether $t \in T$ is bounded. As a matter of fact, $t \in T$ is not bounded iff it is the label of some edge within some SCC of $RG(T, m)$. Let, in the sequel, $BT(T, m) \subseteq T$ denote the set of bounded transitions in $(T, m)$.

If $(T, m')$ is a persistent VRS, and if $\tau, \tau' \in T^*$ are sequences s.t. $a(\tau, m')$, $a(\tau', m')$, and $\delta\tau \geqq 0$ then $\Phi(\tau) \in C(T, m' + \delta\tau')$. This follows from the fact (proven in [8]) that $c, c' \in C(T, m')$ implies $\max\{c, c'\} - c \in C(T, m' + Vc)$ and the observation that $a(\tau', m' + \delta\tau)$ because of $\delta\tau \geqq 0$.

Now, let $(T, m)$ be a persistent VRS, $k$ a node in some distinguished SCC of $RG(T, m)$, $k' \neq k$ a node on a cycle through $k$, and $\tau$ the edge-labelling sequence of a cycle through $k'$ but not $k$. Note that the node markings $\bar{m}(k'')$ have the same set of $\omega$-coordinates for all nodes $k''$ in the SCC of $k$. If $p$ denotes the projection on those coordinates where $\bar{m}(k)$ is not equal to $\omega$, we have $a(p(\tau), p(\bar{m}(k')))$ and $p(\delta\tau) = 0$. It is also clear from the construction in Algorithm 1 that $p(\bar{m}(k)) \in R(p(T), p(\bar{m}(k')))$. Therefore, using Lemma 3 and the above observation, there must be a path starting from $k$ with some edge-labelling sequence $\tau'$ s.t. $\Phi(\tau') = \Phi(\tau)$. But then all transitions in $\tau'$ are unbounded, $p(\delta\tau') = 0$, and this path must end in $k$. From this observation, one easily obtains

**Lemma 4.** *If $(T, m)$ is a persistent VRS and $k$ a node in some distinguished SCC of $RG(T, m)$ then $CT_{k'}$ (resp., $CT_{k'}^+$) is linear and equal for all $k'$ in the SCC of $k$.*

*Proof.* From the above discussion, one deduces that the sets $CT_{k'}$ are linear and equal for all $k'$ in the SCC of $k$. But as $CT_{k'}^+ = \{c \in CT_{k'};\ Vc \geqq 0\}$, so are the $CT_{k'}^+$. q.e.d.

On the other hand, one may observe that if $(T, m)$ is an arbitrary VRS and CC some distinguished SCC in $RG(T, m)$ s.t. all $\bar{m}(k)$ for $k$ in CC are persistent, then $CT_k$ (resp. $CT_k^+$) is linear and equal for all $k$ in CC because $\{\bar{m}(k);\ k \text{ in CC}\} = R(T - BT(T, m), \bar{m}(k'))$ for any $k'$ in CC, and $(T - BT(T, m), \bar{m}(k'))$ still is persistent.

The following theorem states the basic properties of distinguished SCC's in the reachability graphs of persistent VRS's. Let, in the sequel, $w' := |BT(T, m)|$ denote the cardinality of $BT(T, m) \subseteq T$, and $p_{BT}$ the projection of $\mathbb{N}^w$ onto those coordinates which correspond to transitions in $BT$.

**Theorem 1.** *Let $(T, m)$ be a persistent VRS.*

a) *There is exactly one maximal SCC (i.e. no other SCC can be reached from it) in $RG(T, m)$.*

b) *For each $\bar{c} \in \{p_{BT}(c);\ c \in C(T, m)\}$, there is exactly one distinguished SCC $CC(\bar{c})$ in $RG(T, m)$ s.t. $(\forall \tau \in T^*)\ [(a(\tau, m) \wedge \tau$ determines the edge-labelling sequence of some path in $RG(T, m)$ from the root $r$ to some node in $CC(\bar{c})) \Rightarrow p_{BT}(\Phi(\tau)) = \bar{c}]$.*

*Proof.* a) Assume that there are two nodes $k$ and $k'$ in two different maximal SCC's of $RG(T,m)$. Then there are $\tau, \tau' \in T^*$ s.t. $a(\tau, m)$, $a(\tau', m)$, and $\tau$ (resp., $\tau'$) determine a path in $RG(T,m)$ from the root to $k$ (resp., $k'$). Because of the result in [8] and its consequence mentioned above, we may assume w.l.g. that $\Phi(\tau)=\Phi(\tau')$ and $\bar{m}(k)$ and $\bar{m}(k')$ have the same set of $\omega$-coordinates, and hence, that $k=k'$ contradicting the assumption.

b) First note that in $RG(T,m)$ an edge the label of which is a bounded transition always leads from one SCC to a different one.

Let $\bar{c} \in \{p_{BT}(c); c \in C(T,m)\} \subseteq \mathbb{N}^{w'}$. Obtain from $(T,m)$ a modified $(T',m')$ in the following way (where $BT=\{t_{i_1}, \ldots, t_{i_{w'}}\}$):

Add to all $(u_i, v_i) \in Tw'$ new coordinates the $j$-th of which is 1 for $u_{i_j}$, $-1$ for $v_{i_j}$, for $j \in I_{w'}$, and zero in all other cases, and $m'$ equals $(m, \bar{c}) \in \mathbb{N}^{w+w'}$. It is easy to see from the definition that $(T',m')$ still is persistent, and that $C(T',m') = \{c \in C(T,m); p_{BT}(c) \leqq \bar{c}\}$. Also, it follows from the remark made above that for all $\tau' \in T'^*$ with $a(\tau', m')$ $p_{BT}(\Phi(\tau'))=\bar{c}$ if $\tau'$ determines a path in $RG(T',m')$ from the root to some node in the maximal SCC.

If one now observes that - again because of the above remark - $RG(T',m')$ is isomorphic to a subgraph of (some, because Algorithm 1 is nondeterministic) $RG(T,m)$ (with the canonical mapping between the edge and node labels, resp.) then let the maximal SCC in $RG(T',m')$ correspond to $CC(\bar{c})$ in $RG(T,m)$, and b) follows from a). q.e.d.

*Definition 4.* Let $(T,m)$ be a VRS and $k$ a node in $RG(T,m)$ s.t. $CT_k^+$ is linear. Let, further, $\tau^j, j \in I_h$, be edge-labelling sequences of paths in $RG(T,m)$ from $k$ to $k$ s.t.

$$CT_k^+ = \left\{ \sum_{j=1}^{h} n_j \Phi(\tau^j); (n_1, \ldots, n_h) \in \mathbb{N}^h \right\}.$$

A *hurdle* for $k$ is then a number $H_k \in \mathbb{N}$ s.t.

$$m' \in F(\bar{m}(k), H_k) \Rightarrow (\forall j \in I_h) [a(\tau^j, m')].$$

Given $RG(T,m)$ for some $(T,m)$, an $H_k$ for some $k$ with linear $CT_k^+$ can effectively be determined.

Finally, we note a property of linear sets which is used by the algorithms discussed in the next section. Let, for sets $A, B \subseteq \mathbb{N}^v$, $A+B$ denote the set $\{a+b; a \in A, b \in B\}$. Then we have

**Lemma 5.** *Let $L \subseteq \mathbb{N}^v$ be linear, $0 \in L$, and $L'$ some subset of $L$. Then there is a finite $B \subseteq L'$ s.t. $L' \subseteq B+L$. If $L'$ is semilinear, and $L$, $L'$ are effectively given, such a $B$ can effectively be obtained.*

The proof is left to the reader.

## 4. A Decision Procedure for Persistence

Let $k$ be a node in the reachability graph $RG(T,m)$ of some VRS $(T,m)$, $m \in \mathbb{N}^v$, s.t. $CT_{k'}$ is equal and linear for all nodes $k'$ in the SCC of $k$. We are now going to describe a procedure *slset* which for SCC's as above constructs a semilinear representation of the set $\{c+\Phi(\tau); a(\tau, m+Vc)$ and $\tau$ is the edge-labelling sequence of a path in the SCC of $k\}$ where $c$ is some counter s.t. $m+Vc \in F(\bar{m}(k), H_k)$.

**procedure** slset ($k$: **node**; $SL$: **repr of semilinear set**; $c$: **counter**);
**begin**
  **co** it is required that $k$ is a node in the reachability graph $RG(T, m)$ of some VRS $(T, m)$, $m \in \mathbb{N}^v$, s.t. $CT_{k'}$ is equal and linear for all $k'$ in the SCC of $k$. $(T, m)$ etc. are global for slset **oc**;
  **var** $HK$: **integer**; $CTK$: **repr of semilinear set**;
  **co** $SL$, $CTK$ refer to representations of semilinear sets **oc**;
  **procedure** complete ($k$: **node**; $c$: **counter**);
  **begin**
    **var** $i$: **integer**; *bset*: **finite set of counter**; $L$:**repr of semilinear set**;
    **for** $i := 1, \ldots, w$ **do**
      **if** there is an edge labelled $t_i$ from $k$ to some $k'$ (possibly $k = k'$) in the SCC of $k$ s.t.
        $(\exists c' \in c + CTK)\,[a(t_i, m+Vc') \wedge (c' + \Phi(t_i) \notin SL)]$
        **co** this can be written as a Presburger expression **oc**
      **then**
        $L := \{c' \in c + CTK;\ a(t_i, m+Vc') \wedge (c' + \Phi(t_i) \notin SL)\}$;
        **co** the right hand side defines a semilinear set **oc**;
        $bset :=$ some finite subset of $L$ s.t. $L \subseteq bset + CTK$;
        **co** because of Lemma 5, this assignment is effective **oc**;
        **for** all $c'$ in *bset* **do**
          $SL := SL \cup (c' + \Phi(t_i) + CTK)$;
          complete $(k', c' + \Phi(t_i))$
        **od**
      **fi**
    **od**
  **end** complete;
  $CTK :=$ **co** a representation of **oc** $CT_k^+$;
  **co** note Lemma 2 and the remark after Lemma 4 **oc**;
  $HK :=$ some hurdle $H_k$ for $k$;
  $c :=$ some counter $\in C(T, m)$ s.t. $m+Vc \in F(\bar{m}(k), HK)$;
  $SL :=$ **co** a representation of **oc** $c + CT_k^+$;
  complete $(k, c)$
**end** *slset*;

**Theorem 2.** a) *The procedure* **slset** *terminates and yields a set* $SL \subseteq C(T, m)$.

b) *If* $(T, m)$ *is persistent then*

$SL = \{c + \Phi(\tau);\ \tau$ *is the edge-labelling sequence of a path in the SCC of $k$ in $RG(T, m)$ starting from $k$* s.t. $a(\tau, m+Vc)\}$,

*where $c$ is the counter determined in slset.*

*Proof.* a) From the selection of $HK$ and $c$, it is clear that $c + CT_k^+ \subseteq C(T, m)$. When some $c' + \Phi(t_i) + CTK$ is added to $SL$ one can assume by induction on the depth of recursion that $SL \subseteq c + C(T, m+Vc)$ before that step. Let $\tau' \in T^*$ be some sequence s.t. $a(\tau', m+Vc) \wedge \Phi(\tau') = c' - c$. Furthermore, $SL + CTK \subseteq SL$. But by the definition of $HK$ and $CTK$, there is, for any $c'' \in CTK$, a $\tau'' \in T^*$ s.t. $a(\tau'', m+Vc) \wedge \Phi(\tau'') = c'' \wedge \tau''$ determines a path from $k$ to $k$. Hence, $a(\tau', m+V(c+c'')) \wedge a(t_i, m+V(c'+c''))$ as $\delta\tau'' = Vc'' \geqq 0$. As $c''$ has been chosen arbitrary in $CTK$ this shows that $c' + \Phi(t_i) + CTK \subseteq c + C(T, m+Vc)$, and, by induction, that invariantly $SL \subseteq c + C(T, m+Vc) \subseteq C(T, m)$.

Now assume that *slset* does not terminate. Then, by König's Infinity Lemma, there must be an infinite chain of nested recursive calls of procedure *complete* and a subchain of this chain such that all calls in this subchain have the same first parameter. Let $(c^i)_{i \in \mathbb{N}}$ be the sequence of counters in the second parameter position of this subchain. Because each infinite sequence in $\mathbb{N}^v$ has a nondecreasing infinite subsequence (this is a corollary to what is sometimes referred to as Dickson's Lemma [2, Lemma A]) there is a subsequence $(\bar{c}^i)_{i\in\mathbb{N}}$ of $(c^i)_{i\in\mathbb{N}}$ s.t. $(m+V\bar{c}^i)_{i\in\mathbb{N}}$ is nondecreasing. As has been shown above, for each $\bar{c}^i$ there is a $\tau^i\in T^*$ s.t. $a(\tau^i, m+Vc)\wedge\Phi(\tau^i)=\bar{c}^i-c\wedge(\tau^i$ is the edge-labelling sequence of a path $\alpha^i$ in the SCC of $k$ in $RG(T,m)$ starting from $k$ and ending in some fixed node $k'$). The last observation follows from the choice of $(c^i)_{i\in\mathbb{N}}$. Considering the multiplicity with which the edges of $RG(T,m)$ appear in $\alpha^i$, $i\in\mathbb{N}$, and applying once more Dickson's Lemma, one obtains indices $j<j'$ s.t. $\alpha^{j'}$ contains each edge of $RG(T,m)$ at least as often as does $\alpha^j$. As $CT_{k'}$ is linear and equal for all $k'$ in the SCC of $k$, and by the definition of the $\tau^i$, $\Phi(\tau^{j'})-\Phi(\tau^j)\in CTK$. Thus, after $\bar{c}^j+CTK$ has been added to $SL$, $\bar{c}^{j'}\in SL$, contradicting the assumption that *slset* does not terminate.

b) From the first part of this proof we know that $SL\subseteq\{c+\Phi(\tau);\ \tau$ is the edge-labelling sequence of a path in the SCC of $k$ in $RG(T,m)$ starting from $k$ s.t. $a(\tau, m+Vc)\}$. The other direction can easily be seen by induction on the length of $\tau$. q.e.d.

The procedure *slset* will now be used in the following main algorithm to decide persistence of an arbitrary VRS.

## Algorithm 2

```
begin
  var GSL, NPC: repr of semilinear set;
  procedure test ((T, m): VRS);
  begin
    var i: integer; SL, SL': repr of semilinear set; k: node; c', cmax: counter;
    procedure slset (k: node; ...); ...;
    construct RG(T, m) using Algorithm 1;
    if RG(T, m) doesn't satisfy the necessary condition of Theorem 1a) or contains a non-
        persistent node marking thus violating Lemma 3
    then stop '(T, m) is not persistent' fi;
    set CC_i, i = 1, ..., h, the distinguished SCC's of RG(T, m);
    cmax := 0 co ∈ℕ^w oc;
    SL := ∅;
    for i := 1, ..., h do
        k := some node in CC_i;
        slset (k, SL', c');
        cmax := max (c', cmax) co max component-wise oc;
        SL := SL ∪ SL'
    od;
   if SL ∩ NPC ≠ ∅ then stop '(T, m) is not persistent' fi;
    GSL := GSL ∪ SL;
    co GSL globally collects all counters in C(T, m) found by the algorithm oc;
    for i := 1, ..., w do
       if t_i ∈ T − BT(T, m) then
          construct from (T,m) a new VRS (T^i, m^i) where T^i is obtained from T by adding a
          w+1st coordinate which is 1 for u_i, −1 for v_i, and zero in all other cases, and m^i
          equals (m, cmax_i) ∈ ℕ^(w+1);
          co this means that C(T^i, m^i) = {c ∈ C(T, m); c_i ≦ cmax_i} oc;
          test((T^i, m^i));
          co note that GSL is global in this recursion oc
       fi
    od
  end test;
```

$NPC := NPC(T, m)$ **co** note Lemma 1 **oc**;
$GSL := \emptyset$;
test $((T, m))$;
**if** $(\exists c \in GSL, \exists i \in I_w)\,[a(t_i, m + Vc) \wedge (c + \Phi(t_i) \notin GSL)]$
**then stop** '$(T, m)$ is not persistent'
**else stop** '$(T, m)$ is persistent'
**fi**
**end** Algorithm 2.

**Theorem 3.** *Algorithm* 2 *terminates for every VRS* $(T, m)$ *and determines whether* $(T, m)$ *is persistent or not.*

**Corollary 1.** *Persistence is decidable for arbitrary VRS's.*

**Corollary 2.** *If* $(T, m)$ *is persistent Algorithm* 2 *yields GSL* s.t. $GSL = C(T, m)$.

**Corollary 3.** *There is an effective construction of semilinear representations of the reachability set of persistent VRS's.*

*Proof.* Because of the reduction of the number of unbounded transitions in successive recursive calls of the procedure *test*, it is clear that Algorithm 2 terminates. If Algorithm 2 stops within *test* the answer given is correct because of Lemma 3. Otherwise, if the condition

$$(\exists c \in GSL, \exists i \in I_w)\,[a(t_i, m + Vc) \wedge (c + \Phi(t_i) \notin GSL)]$$

in the last *if*-statement of Algorithm 2 evaluates to *false*, *GSL* equals $C(T, m)$ because of Theorem 2a), and $(T, m)$ is persistent because, in fact, $GSL \cap NPC = \emptyset$ has been verified by the algorithm. Conversely, if $(T, m)$ is persistent then so are all $(T^i, m^i)$ generated in the recursive calls of *test* as addition of a coordinate just bounding the number of times how often some transition can be applied doesn't hurt persistence. Furthermore, if $c \in C(T, m)$, then it follows from Theorem 1b), Theorem 2b), and Lemma 3.1 in [9] that

$$((\forall i \in I_w)\,[t_i \in T - BT(T, m) \Rightarrow c_i \geq cmax_i]) \Rightarrow c \in GSL.$$

Hence, by induction on the number of unbounded transitions and by the construction of the $(T^i, m^i)$, $C(T, m) \subseteq GSL$, and because of Theorem 2a), $C(T, m) = GSL$. Therefore, if $(T, m)$ is persistent, the condition in the last *if*-statement of Algorithm 2 evaluates to *false*. This proves the theorem. The corollaries are immediate consequences. q.e.d.

## 5. Conclusion

The algorithm presented in this paper answers some of the questions asked in [9]. Thus, persistence of arbitrary VRS's is decidable (by an algorithm which does not make use of a solution to the general reachability problem), and there is an effective method to construct semilinear representations for the reachability set of persistent VRS's. As far as the author knows, *m*-reversible [1] and persistent VRS's are the only classes of VRS's for which an effective representation of infinite reachability sets has been given so far. Thus there remains a number of open problems in extending this result to other classes of VRS's of interest as well as in establishing and improving the complexity of the algorithm given here. Another open problem concerns the characterization of the class of VRS's that have semilinear reachability sets.

*Acknowledgement.* I should like to thank the referees for their helpful suggestions to improve the presentation of this paper.

## References

1. Araki, T., Kasami, T.: Decidable problems on the strong connectivity of Petri net reachability sets. Theor. Comput. Sci **4**, 99–119 (1977)
2. Dickson, L.E.: Finiteness of the odd perfect and primitive abundant numbers with $n$ distinct prime factors. Amer. J. Math. **35**, 413–422 (1913)
3. Ginsburg, S., Spanier, E.H.: Semigroups, Presburger formulas, and languages. Pacific J. Math. **16**, 285–296 (1966)
4. Hack, M.: Decision problems for Petri nets and vector addition systems. M.I.T., Project MAC, MAC-TM 59 (1975)
5. Hack, M.: Decidability questions for Petri nets. M.I.T., LCS, TR 161 (1976)
6. Karp, R.M., Miller, R.E.: Parallel program schemata. J. Comput. System Sci. **3**, 147–195 (1969)
7. Keller, R.M.: Vector replacement systems: A formalism for modelling asynchronous systems. Princeton University, CSL, TR 117 (1972)
8. Keller, R.M.: A fundamental theorem of asynchronous parallel computation. In: Parallel processing. (T.Y. Feng, ed.) Proceedings Sagamore Computer Conference. Lecture Notes in Computer Sciences, Vol. 24, pp. 102–112. Berlin Heidelberg New York: Springer, 1975
9. Landweber, L.H., Robertson, E.L.: Properties of conflict free and persistent Petri nets. J. Assoc. Comput. Mach. **25**, 352–364 (1978)
10. Lipton, R.J., Miller, R.E., Snyder, L.: Synchronization and computing capabilities of linear asynchronous structures. Proc. 16th Ann. Symp. on FOCS. IEEE Computer Society 1975, pp. 19–28
11. Müller, H.: Decidability of reachability in persistent vector replacement systems. In: Mathematical Foundations of Computer Science 1980. Proceedings of the 9th Symposium in Rydzyna. (P. Dembinski, ed.) Lecture Notes in Computer Sciences, Vol. 88, pp. 426–438. Berlin Heidelberg New York: Springer (1980)
12. Muller, D.E., Bantky, M.S.: A theory of asynchronous circuits. Proc. Int. Symp. on Theory of Switching. Cambridge, MA: Harvard Univ. Press, p. 204–243, 1959
13. Oppen, D.C.: A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger arithmetic. J. Comput. System Sci. **16**, 323–332 (1978)
14. Parikh, R.J.: On context-free languages. J. Assoc. Comput. Mach. **13**, 570–581 (1966)
15. 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, p. 92–101, 1930
16. Rosen, B.: Tree manipulating systems and Church-Rosser theorems. J. Assoc. Comput. Mach. **20**, 160–187 (1973)

Received November 11, 1979 / November 18, 1980

**Note Added in Proof**

While this paper was being prepared, another algorithm for deciding persistence based on the nonconstructive proof in [9] was obtained independently by J. Grabowski (Information Processing Lett. **11**, 20–23 (1980))