Theoretical Computer Science 2 (1976) 77–95.

THE EQUALITY PROBLEM FOR VECTOR ADDITION SYSTEMS IS UNDECIDABLE*

Michel HACK
Project MAC, Massachusetts Institute of Technology, Cambridge, Mass. 02139, USA

Communicated by A. Meyer
Received December 1974
Revised April 1975

Abstract. We demonstrate the usefulness of Petri nets for treating problems about vector addition systems by giving a simple exposition of Rabin's proof of the undecidability of the inclusion problem for vector addition system reachability sets, and then proceed to show that the inclusion problem can be reduced to the equality problem for reachability sets.

1. Introduction

Vector addition systems were introduced by Karp and Miller in 1966 to represent the control aspects of their parallel program schemata [8]. They proved the decidability of certain problems concerning vector addition systems, such as boundedness, and reported that Rabin had proved the undecidability of the equality problem. In fact, what Rabin had proved (in 1965) was that the inclusion problem was undecidable, by reducing the unsolvable problem of finding integer roots for exponential polynomial equations to it. The equality problem, which is of course reducible to the inclusion problem, remained unsolved. We will show that the converse reducibility also holds, thus finally settling the problem.

In 1972, by which time it was known that Hilbert's tenth problem was undecidable, Rabin presented a new, simplified proof of his result at a talk at Massachusetts Institute of Technology [14], an account of which can be found in [1]. (Rabin never published his proof.) This proof forms the basis for the Petri net version presented in this paper.

Petri nets were introduced as a model for concurrent systems by Petri [13] and Holt [6]; their interpretation was later generalized to permit unboundedness [7]. A further generalization, made in 1972 independently by several people, including the author [3, 9], yielded generalized Petri nets. We shall use generalized Petri nets only as a useful graphical representation of vector addition systems; our proofs can be carried out using only ordinary Petri nets.

Finally, let us mention that the recursiveness problem and the uniform recursive-

* This work was supported by the National Science Foundation under Research Grant DCR74–21822.

ness problem (reachability problem) for vector addition systems are still open; some preliminary results can be found in [3, 4] and [15].

2. Vector addition systems and Petri nets

2.1. *Definition of vector addition systems*

We give Karp and Miller's original [8] definition of a vector addition system (VAS). Let $\mathbf{Z}$ = the integers, $\mathbf{N}$ = the non-negative integers.

Definition. An r-dimensional *vector addition system* (VAS) is a pair $A = \langle q, W \rangle$ in which q is an r-dimensional vector of non-negative integers and W is a finite set of r-dimensional integer vectors, $q \in \mathbf{N}^r$, $W \subseteq \mathbf{Z}^r$.

The *reachability set* $R(A)$ is the set of all vectors of the form $q + w_1 + w_2 + \cdots + w_n$ such that $\forall i \leqslant n$, $w_i \in W$, and $q + \Sigma_{j=1}^{i} w_j \geqslant 0$. Geometrically, in r-coordinate space, $R(A)$ is the set of points reachable from q by successive translations from the set W *without ever leaving the first orthant.*

Karp and Miller gave no name to the various objects they introduced. For convenience we shall introduce a terminology which will be consistent with that used for Petri nets.

The elements of W are called *transitions*, the elements of $R(A)$ are called *markings*, where q is the *initial marking*. Henceforth we shall use t or t_i as a generic name for transitions, and $a, b, c, \cdots$ as names for particular transitions. Markings will be designated by $M, M', M_i, \cdots$; we reserve M_0 for the initial marking.

If M is a marking in $R(A)$ such that transition $t \in W$ can be added to it to produce a new reachable marking $M' = M + t \in \mathbf{N}^r$, we say that t is *enabled* at M. The passage from M to M' is called a *firing* of t, provided that $M \geqslant -t$ and $M' = M + t$.

This is because the original motivation for studying VASs was to model parallel, asynchronous systems, where a system state is described by a marking, and where a state transition (transition firing) is possible (enabled) iff certain conditions, such as non-empty buffers, are satisfied ($M \geqslant -t$).

In our proofs we shall only use VASs whose transitions have coordinates restricted to $\{-1, 0, +1\}$. Such a VAS is called a *restricted* VAS.

2.2. *A graphical representation of restricted vector addition systems*

In a restricted VAS, each transition firing changes certain coordinates of the marking by increasing or decreasing the coordinate by one. We can imagine a counter for each coordinate i, or better yet, a *place* p_i into which or from which we may deposit or remove a *token*. A marking can thus be visualized as a distribution of tokens over a set of places $\{p_1, \cdots, p_n\}$. We shall draw each place as a circle with a corresponding number of dots inside to represent the tokens at some marking (see Fig. 1). Now we can represent a transition t by a vertex, traditionally drawn as a bar, connected by arrows to those places whose token count will be affected by a

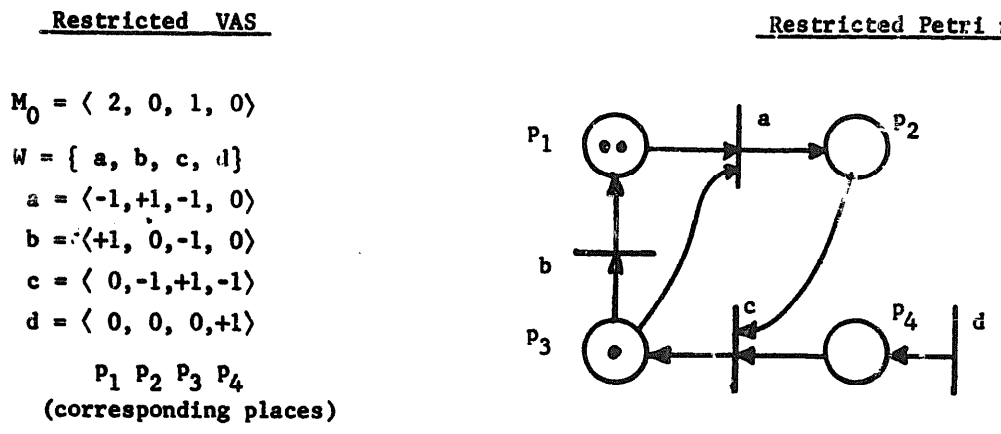


Fig. 1. An example.

firing of t: If the ith coordinate of t is -1, we direct an arrow from place p_i to t; if the ith coordinate of t is $+1$, we direct an arrow from t to p_i. A firing of t can now be visualized as the simultaneous transport of one token along each arrow attached to the transition. Note that the total number of tokens need not be conserved. A transition t is thus enabled iff each place which has an arrow pointing to t (*input place*) contains at least one token; the firing of t takes one token from each input place and adds one token to each *output place* of t.

We can thus represent an n-dimensional restricted VAS by a bipartite directed graph with an initial marking, which is a distribution of tokens over vertices of one type—the places $p_1 \cdots p_n$. The vertices of the other type are called transitions. Such a graph is called a *Petri net*. Fig. 1 shows a restricted VAS and the corresponding Petri net; Fig. 2 illustrates the firing of a transition by showing the result of firing transition a at the initial marking as in Fig. 1. The following definition of a Petri net is adapted from Hack [2], and agrees with the original definition in [7].

2.3. *Petri nets*

Definition. (a) A *Petri net* is a directed bipartite graph with an *initial marking*. The two distinguished types of vertices are called *places* $\{p_1, \cdots, p_n\}$ and *transitions*

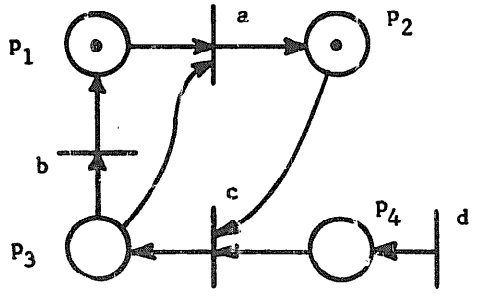


Fig. 2. Result of firing transition a.

$\{t_1, \cdots, t_l\}$. A *marking* M is a function which associates with each place p_i a non-negative number of *tokens* $M(p_i)$ in that place, and can be treated like a vector $M \in \mathbf{N}^n$ whose ith coordinate is $M(p_i)$.

(b) A *simulation* of a Petri net is a sequence of *firings* of transitions; only *enabled* transitions may fire at any time, and a transition is enabled iff each of its input places (those with arcs pointing to the transition) contains at least one token in the present marking. The firing of a transition *changes the marking* by removing one token from each of its input places and then adding one token to each of its *output places* (those places that are pointed to by arcs from the transition).

(c) A marking M' is said to be *reachable* from marking M iff there exists a firing sequence which transforms M into M'. The set of all markings reachable from the initial marking is the *reachability* set of the Petri net.

Actually, this definition permits two possibilities which do not arise when we represent a restricted VAS:

(i) there may be *self-loops*, i.e., pairs p, t such that p is both an input place and an output place of t;

(ii) there may be two transitions t_1 and t_2 which have exactly the same input and output places.

The first situation does not arise because in a VAS transition, a coordinate cannot be both positive and negative; the second does not arise because in a VAS a transition is defined by its input and output places: W is a set, where no element "occurs twice".

So let us call a *restricted Petri net* (RPN) a Petri net which has no self-loops and no duplicate transitions. Then it is easy to see that for every restricted Petri net with a given ordering of its places $p_1 \cdots p_n$ there is exactly one restricted VAS of which it is the graphical representation.

Fact.* The graphical representations of restricted vector addition systems are exactly all restricted Petri nets.

2.4. *Generalized Petri nets*

The notation of Petri nets can easily be extended to represent arbitrary vector addition systems, or a slight generalization thereof, namely Keller's vector replacement systems [9]. This is done by assigning a *size* to each arc in the Petri net, with the understanding that an arc of size $K \in \mathbf{N}$ transports K tokens per firing instead of one. These so-called generalized Petri nets have been found to be quite useful tools for investigating the properties of vector addition systems and related formalisms, mainly because their graphical form provides a better grasp for the intuition required to guide the researcher towards his goal [4]. For more information on the relationship between generalized Petri nets, vector addition systems, restricted Petri nets and related formalisms, see [3].

* This relationship is also discussed in [11].

3. The decision problems

We shall present a sequence of decision problems, each of which is recursively reducible to the next problem. The first problem in the sequence is Hilbert's tenth problem, which was shown to be undecidable by Matijas'evič [10] in 1970. This successively implies the undecidability of the problems to which it is reducible, and thus shows that the equality problem, to which the others are reducible, is undecidable.

3.1. *Hilbert's tenth problem* (H)

Hilbert's tenth problem [5] is the problem of deciding, given a polynomial $P(x_1, \cdots, x_r)$ with integer coefficients (diophantine polynomial), whether this polynomial has an integral root, that is, a solution to:

$$\langle \alpha_1, \cdots, \alpha_r \rangle \in \mathbf{Z}^r: \quad P(\alpha_1, \cdots, \alpha_r) = 0.$$

3.2. *The polynomial graph inclusion problem* (*PGIP*)

We define the graph $G(P)$ of a diophantine polynomial $P(x_1, \cdots, x_r)$ to be the set

$$G(P) = \{\langle x_1, \cdots, x_r, y \rangle \in \mathbf{N}^{r+1}: \ y \leqslant P(x_1, \cdots, x_r)\}.$$

The *polynomial graph inclusion problem* is the problem of deciding, given two polynomials $P(x_1, \cdots, x_r)$ and $Q(x_1, \cdots, x_r)$, whether $G(P) \subseteq G(Q)$.

We shall show that H is recursively reducible* to PGIP (Theorem 2).

3.3. *The VAS subspace inclusion problem* (SIP)

Instead of worrying about all coordinates in a VAS, we shall now restrict our attention to the subspace generated by some of the coordinates. To be specific, we shall study the *projections* of reachability sets of dimension n on the first r coordinates, $0 \leqslant r \leqslant n$. If $v \in \mathbf{N}^n$ is a vector, then its projection $P_r(v)$ is the vector $w \in \mathbf{N}^r$ whose coordinates are the first r coordinates of v, in the same order.

The *subspace inclusion problem* is the problem of deciding, given two VASs A and B of dimensions $\geqslant r$, whether the projection of one reachability set is included in the projection of the other, i.e., whether $P_r(R(A)) \subseteq P_r(R(B))$. In other words, is it true that for any $v \in R(A)$ there is a $w \in R(B)$ such that $P_r(v) = P_r(w)$?

We shall show that the PGIP is reducible to the SIP (Theorem 3).

3.4. *The VAS reachability set inclusion problem* (IP)

Given two VASs of the same dimension n, is the reachability set of one included in the reachability set of the other?

It is clear that the IP is an instance of the SIP; we shall show that the SIP is also reducible to, and thus equivalent to, the IP (Theorem 4).

* H and PGIP can, in fact, be shown to be recursively equivalent.

3.5. *The VAS reachability set equality problem* (EP)

Given two VASs of the same dimension n, are the two reachability sets equal?

Again, the EP is clearly reducible to the IP. We prove the converse, and thus establish the undecidability of the EP (Theorem 5).

3.6. *Other decision problems about VAS's*

It may be appropriate here to mention some related decision problems for VASs.

The *boundedness problem* is the problem of deciding whether the reachability set contains markings arbitrarily large in some given subset of the coordinates. It has been shown to be decidable by Karp and Miller [3, 8].

The *reachability problem* is the problem of deciding whether a given marking is in the reachability set or not. It is still open; partial results can be found in [3, 4, 9, 12, 15].

4. Computation by Petri nets

In order to relate Hilbert's tenth problem to Petri nets, we must show how Petri nets can compute polynomials, in some sense. Usually, an automaton used to compute a function is given its arguments in some form, and started in some "initial" state. If and when the automaton halts in some "final" state, we can recover the computed value, for example by reading the contents of a certain register. Such an automaton is usually thought to be deterministic, or at least functional in the sense that all halting computations produce the same result. But the non-determinism associated with the set of possible firing sequences in a Petri net is essential to the power of Petri nets. In fact, if we only consider nets whose firing sequences are monogenic ("deterministic" Petri net), then all the problems mentioned so far are decidable (the reachability sets will be ultimately periodic or finite).

So, in order to get any non-trivial functions, we have to modify our idea of a computation. Following Rabin, we shall say that a non-deterministic automaton *weakly computes* a function $f(x_1 \cdots x_r)$ iff the maximum output value over all computations starting with the argument $x_1, \cdots, x_r$ is $f(x_1, \cdots, x_r)$.

Now let us see how we can embed a weak computer in a Petri net. Some of the constraints we shall use are not essential, but they will simplify the exposition.

Definition. A *Petri net weak computer of r inputs* is a Petri net with at least $r+1$ places (r input places and one "*start*" place) and at least 2 transitions, called "*stop*" and "*count*", which satisfies the following conditions:

(1) The initial marking consists of x_i tokens in the ith input place, one or zero tokens in the "*start*" place, and zero tokens in all other places.

(2) If the "*start*" place contains zero tokens initially, no transition is firable.

(3) After the "*stop*" transition has fired, no transition is firable anymore.

(4) The "*count*" transition can fire *at most* $f(x_1, \cdots, x_r)$ times, where f is the function weakly computed by the Petri net.

(5) If the initial marking contains a token in the "*start*" place, then the "*stop*" transition can fire after any number of "*count*" firings up to $f(x_1, \cdots, x_r)$ times, including zero.

Some comments on the 5 points in the definition:

(1) This is the encoding of the argument for the weak computer.

(2) This permits the computer to be *switched on*, by putting a token into the "*start*" place. This token is consumed by the first transition firing which starts the computation.

(3) This is how the computer is switched off. Together with conditions (4) and (5), this assures that the computation can be stopped at any output between 0 and $f(x_1, \cdots, x_r)$.

(4) This means that, among all possible firing sequences from the initial marking with one token in the "*start*" place, at least one fires the "*count*" transition $f(x_1, \cdots, x_r)$ times, and *none* fires it more often. It does *not* mean that if we simply delay firing the "*stop*" transition, we will eventually fire the "*count*" transition that many times: we may get "*stuck*" before. Therefore, when explaining how such a computer works, we ought to provide a *strategy* for achieving the highest possible number of firings of "*count*".

Fig. 3 shows the general disposition of a Petri net weak computer. (The bottom portion of the box can be thought of as the "control" portion of the computer.) We shall now show how to construct such Petri net computers for addition and multiplication, and how to combine them to construct a weak computer for polynomials with non-negative coefficients.

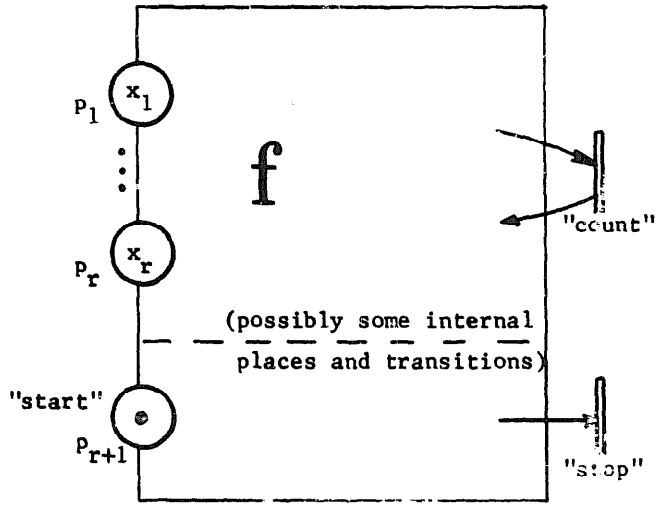


Fig. 3. A Petri net weak computer.

Lemma 1. *Addition of non-negative integers is weakly computable by Petri nets.*

Proof. The Petri net shown in Fig. 4 satisfies the conditions for a weak computer for addition.

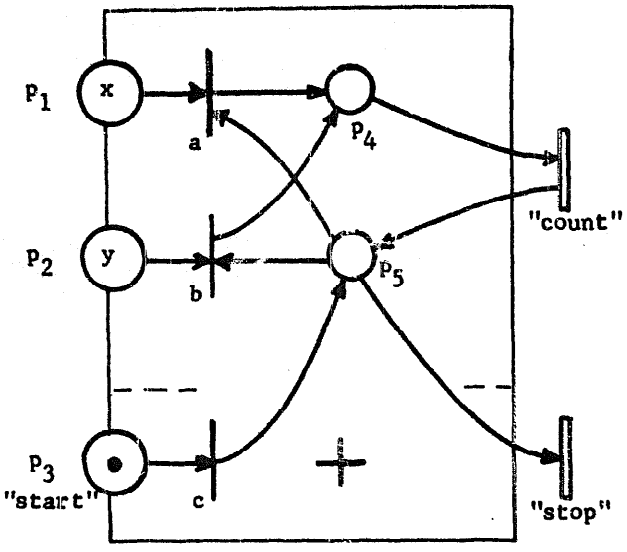


Fig. 4. A weak adder.

It is easy to verify points (1), (2), (3), and (5). To verify point (4), we note that the firing sequences are really a subset of the regular language described by $c \cdot (a \cdot count + b \cdot count)^*\ stop$, where the number of occurrences of a and b is limited by x and y, respectively. Thus, "*count*" will fire once for each token removed from one of the input places, up to $x + y$ times, and no more. □

Lemma 2. *Multiplication of non-negative integers is weakly computable by Petri nets.*

Proof. Consider the Petri net in Fig. 5. Again, it is easy to see that conditions (1), (2) and (3) are satisfied. Indeed, the four places labelled $p_4, \cdots, p_7$ can have only one token altogether, namely after s has fired (removing the token from "*start*") and before s' ("*stop*") has fired. Again, all "complete" firing sequences are selected from a regular set, described by

$$s(a(cc')^*a'(bb')^*)^*s'.$$

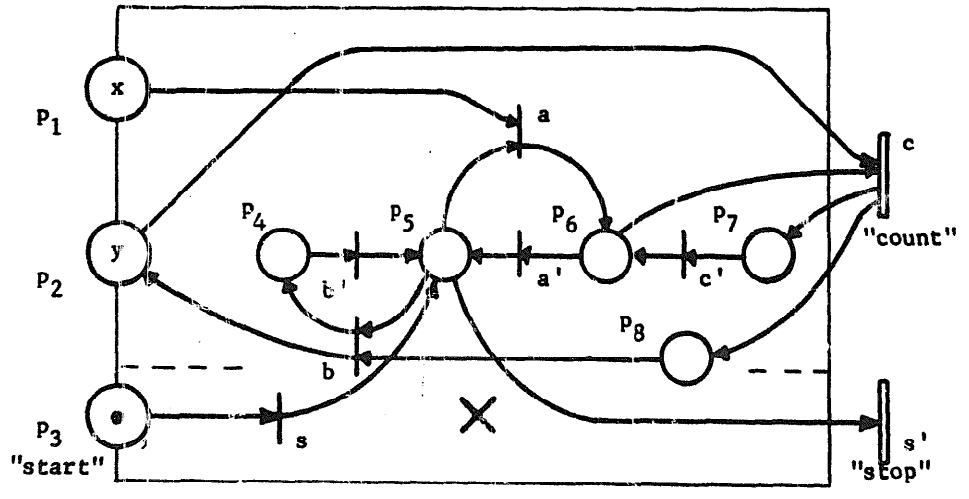


Fig. 5. A weak multiplier.

The restrictions imposed by the contents of the three "registers" p_1, p_2, p_8 (initialized to $x, y, 0$, respectively) are the following:

(i) the number of occurrences of a is bounded by x;

(ii) the number of occurrences of cc' between a firing of a and the following firing of a' is bounded by the contents of p_2 just after the firing of a;

(iii) the number of firings of bb' between a' and a is similarly bounded by the contents of p_8;

(iv) neither p_2 nor p_8 can ever hold more than y tokens; in fact, the y tokens originally in p_2 can only be shuttled back and forth between p_2 and p_8.

It follows that c ("*count*") can fire at most y times between firings of a, which itself can fire at most x times. "*Count*" can thus fire at most $x \cdot y$ times. A strategy to fire "*count*" the maximum number of times is the firing sequence

$$s(a(cc')^y a'(bb')^y)^x s'.$$

This clearly fires c ("*count*") $x \cdot y$ times, thus condition (4) is satisfied.

Finally, condition (5) is satisfied because after firing c the desired number of times, we can fire $c'a's'$ to stop. □

Now we shall see how to weakly compute an arbitrary polynomial of several variables with non-negative coefficients. We can write a "program" which uses only addition, multiplication, and substitution of the form $P(x_1, x_2, x_3)$, where $x_1 = x_2 = Q(y_1, y_2)$, for example. Here, P and Q are polynomials we have already constructed, and the result of the substitution is a new polynomial

$$R(y_1, y_2, x_3) = P(Q(y_1, y_2), Q(y_1, y_2), x_3).$$

Lemma 3. *If $P(x_1, x_2, x_3)$ and $Q(y_1, y_2)$ are polynomials with non-negative coefficients weakly computable by Petri nets, then the result of the substitution of $Q(y_1, y_2)$ for certain variables of $P(x_1, x_2, x_3)$ (say, x_1 and x_2) is weakly computable by a Petri net.*

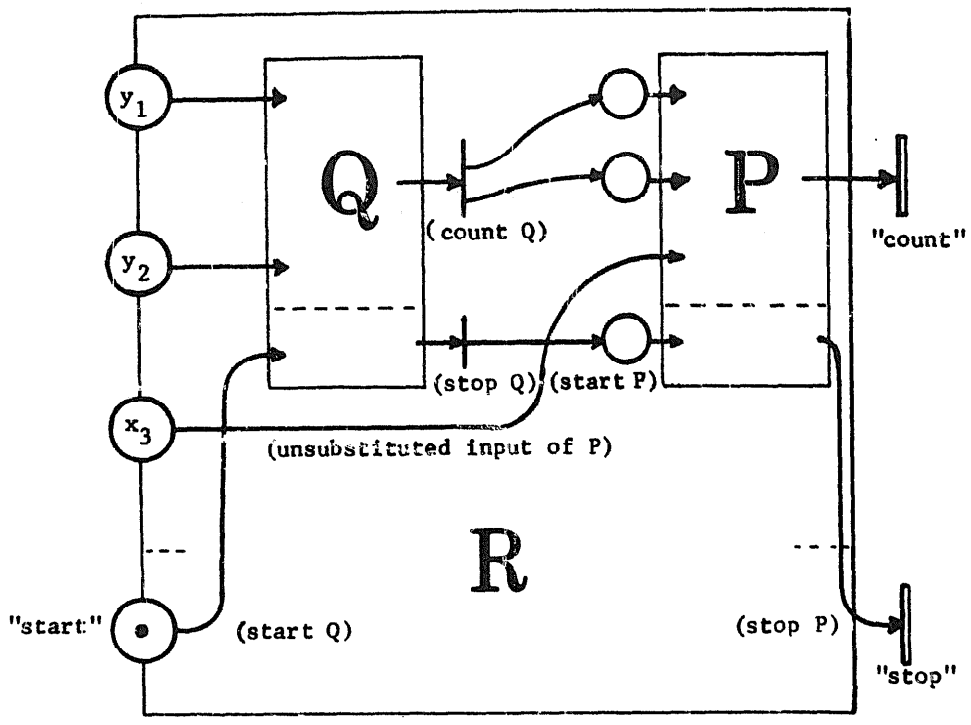
Proof. We construct a new Petri net out of the Petri nets for P and Q by connecting them as shown in Fig. 6. The input places of the Petri net for P that are used for substitution become internal "registers", filled by the firings of the "*count*" transition of the Petri net for Q; the "*stop*" transition of Q is connected to the "*start*" place of P which also becomes internal. The inputs to the new Petri net are the inputs to Q and the unsubstituted inputs of P; the output transitions "*count*" and "*stop*" are those of P.

We have to verify the 5 conditions of weak computability.

(1) The initial marking must leave all internal places blank, in particular the substitution inputs of P and "*start P*".

(2) If "*start*" (i.e. "*start Q*") has zero tokens, then "*stop Q*" can never fire (condition (2) for Q), hence "*start P*" never gets a token, and nothing in P can ever fire either.

(3) When "*stop*" (i.e. "*stop P*") has fired, no transition in P can fire anymore. But before "*stop P*" could have fired, there must have been a firing of "*stop Q*", hence no transition can fire in Q either.



$$R(y_1, y_2, x_3) = P(Q(y_1, y_2), Q(y_1, y_2), x_3)$$

Fig. 6. Composition of two polynomials.

(4) After "*stop Q*" has fired, the inputs to P are no greater than $Q(y_1, y_2)$ and x_3, respectively. Since we assumed that P is non-decreasing in all its variables (non-negative coefficients), the output of P (number of firings of "count") is no greater than $P(Q(y_1, y_2), Q(y_1, y_2), x_3)$. On the other hand, we may use the maximum strategy for Q before starting P, and then use the maximum strategy for P. This clearly produces the desired output.

(5) This condition is the one assumed for P. □

It should be easy to see how a polynomial with non-negative integer coefficients can actually be computed by iterating substitutions as described above. In particular, multiplication by integer coefficients can be done by repeated addition of monomials, since the substitution we described includes the possibility of producing several copies of intermediate results. (In the example, $Q(y_1, y_2)$ is copied twice, into x_1 and x_2.)

From this follows:

Theorem 1. *Polynomials of several variables with non-negative integer coefficients are weakly computable by Petri nets.*

Note. This does not nearly exhaust the computational power of Petri nets. Indeed, Rabin's first proof was based on exponential polynomials, and we have developed a method for weakly computing arbitrarily fast-growing primitive recursive functions by Petri nets.

5. The undecidability proofs

5.1. *The PGIP*

In the preceding section we have seen that Petri nets can weakly compute polynomials with non-negative integer coefficients. We therefore first present a variant of Hilbert's tenth problem.

Theorem 2. *Hilbert's tenth problem* (H) *is recursively reducible to the polynomial graph inclusion problem* (PGIP).

Proof. (a) We can restrict the arguments of the polynomials to the non-negative integers. Indeed, $P(x_1, \cdots, x_r) = 0$ has a solution in $\mathbf{Z}$ if and only if one of the 2^r polynomials obtained by replacing some variables by their negative has a solution in $\mathbf{N}$.

(b) Any root of $P(x_1, \cdots, x_r)$ is also a root of $P^2(x_1, \cdots, x_r)$, and vice versa. Hence we can restrict our attention to polynomials whose range is in $\mathbf{N}$.

(c) By separating the positive and the negative coefficients of a polynomial whose range is non-negative, we get two polynomials $Q_1(x_1, \cdots, x_r)$ and $Q_2(x_1, \cdots, x_r)$, each with non-negative integer coefficients, such that:

$$\forall x_1, \cdots, x_r \in \mathbf{N}: \quad Q_1(x_1, \cdots, x_r) \geqq Q_2(x_1, \cdots, x_r).$$

There exists an integral root to the original polynomial if and only if

$$\exists x_1, \cdots, x_r \in \mathbf{N}: \quad Q_1(x_1, \cdots, x_r) = Q_2(x_1, \cdots, x_r).$$

Now let us consider the following two polynomial graphs

$$G(Q_1) = \{\langle x_1, \cdots, x_r, y\rangle \in \mathbf{N}^{r+1}: \ y \leqq Q_1(x_1, \cdots, x_r)\},$$

$$G(Q_2 + 1) = \{\langle x_1, \cdots, x_r, y\rangle \in \mathbf{N}^{r+1}: \ y \leqq 1 + Q_2(x_1, \cdots, x_r)\}.$$

From this follows that

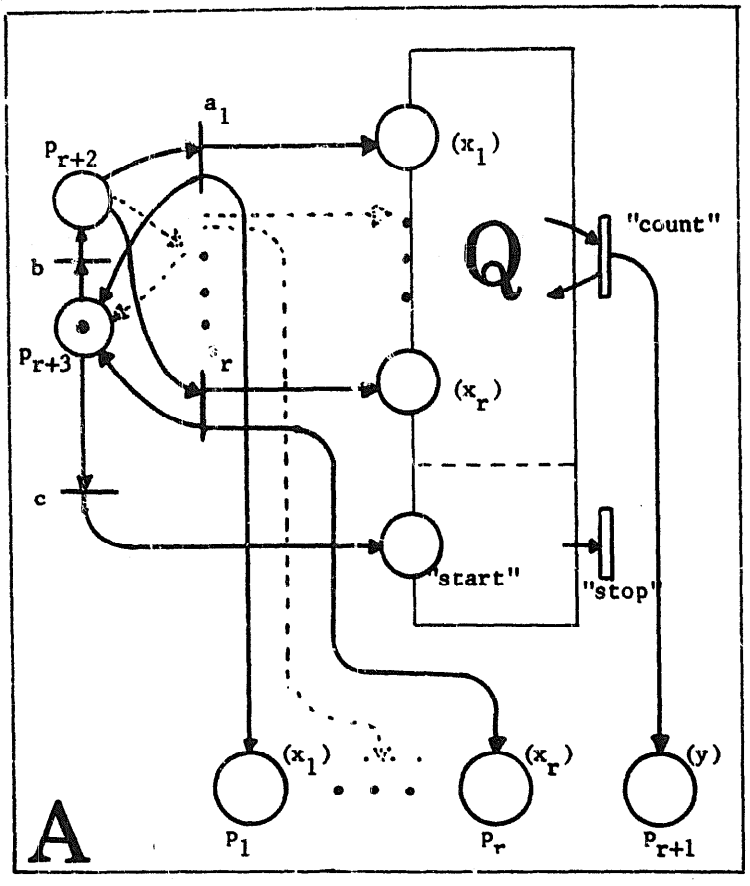
$$G(Q_2+1) \subseteq G(Q_1) \Leftrightarrow [\forall x_1, \cdots, x_r, y \in \mathbf{N}: \ (y \leqq Q_2(x_1, \cdots, x_r) + 1 \Rightarrow y \leqq Q_1(x_1, \cdots, x_r))]$$

$$\Leftrightarrow \neg\exists x_1, \cdots, x_r, y \in \mathbf{N}: \ Q_1(x_1, \cdots, x_r) < y \leqq 1 + Q_2(x_1, \cdots, x_r).$$

Combining this with the fact that Q_2 never exceeds Q_1, this implies:

$$G(Q_2+1) \subseteq G(Q_1) \Leftrightarrow \neg\exists x_1, \cdots, x_r, y \in \mathbf{N}: \ y = 1 + Q_1(x_1, \cdots, x_r) = 1 + Q_2(x_1, \cdots, x_r).$$

In other words, H is decided in the negative if and only if the corresponding PGIP is decided in the affirmative, thus proving the reducibility of H to the PGIP. □



$y \leq Q(x_1, \cdots, x_r)$ $\qquad$ $P_{r+1}(R(A)) = G(Q)$

Fig. 7. Simulation of a polynomial graph.

5.2. *The SIP*

Now we shall use the fact that Petri nets can weakly compute polynomials.

Lemma 4. *Given a polynomial $Q(x_1, \cdots, x_r)$ with non-negative integer coefficients, there exists a Petri net A such that the projection of its reachability set on the first $r+1$ coordinates is the graph of Q:*

$$P_{r+1}(R(A)) = G(Q).$$

Proof. Let B be a Petri net weak computer for polynomial Q, as described in the previous section. We construct from B a new Petri net A by adding $r+3$ new places and renumbering the places in B so that the new places are numbered $p_1 \cdots p_{r+3}$ as shown in Fig. 7. The initial marking consists of one token in p_{r+3} and zero tokens in all other places. The new transitions are $a_1 \cdots a_r$, one for each input to B, transition b, which connects p_{r+3} to p_{r+2}, and transition c, which connects p_{r+3} to the "*start*" place of B. Before c fires and switches the computer on, each a_i may fire an arbitrary number of times and thus generates an arbitrary argument $\langle x_1, \cdots, x_r \rangle$ for polynomial Q; this argument is also saved in $p_1 \cdots p_r$. After c has fired, B weakly computes $Q(x_1, \cdots, x_r)$, and at no time do we have more than $Q(x_1, \cdots, x_r)$

tokens in p_{r+1}; yet there exists a computation which puts that many tokens on p_{r+1}. The markings of these places, $p_1 \cdots p_{r+1}$, are thus precisely the points in the graph $G(Q)$. □

From this follows:

Theorem 3. *The PGIP is recursively reducible to the subspace inclusion problem (SIP)*

Proof. Given two polynomials with non-negative integer coefficients we construct two Petri nets whose projected reachability sets are the graphs of the respective polynomials as indicated by Lemma 4. Then a test for the SIP will also decide the corresponding PGIP. □

5.3. *The IP*

Now we shall show how we can modify Petri nets of n places to "forget" the marking in $n - r$ "uninteresting" places and thus reduce the SIP to a comparison of complete reachability sets, the IP.

Theorem 4. *The SIP is recursively reducible to the IP.*

Proof. Suppose we are given two Petri nets of n and m places, respectively, and we wish to test, for the two projections on the first r coordinates of the respective reachability sets, whether one is a subset of the other.

First, we note that we can always add $|n - m|$ places to the smaller net (without renumbering the original places) to get two nets with the same number of places, say n. If we do not connect these new places to any transitions, we will not change the reachability set as far as the old places are concerned, and thus the problem is reduced to the following: Given two Petri nets A and B of $n \geq r$ places each, is $P_r(R(A)) \subseteq P_r(R(B))$?

We will modify both nets by adding four places to each net, numbered $p_{n+1} \cdots p_{n+4}$, and we will modify the reachability sets in such a way as to make the inclusion depend only on the first r coordinates.

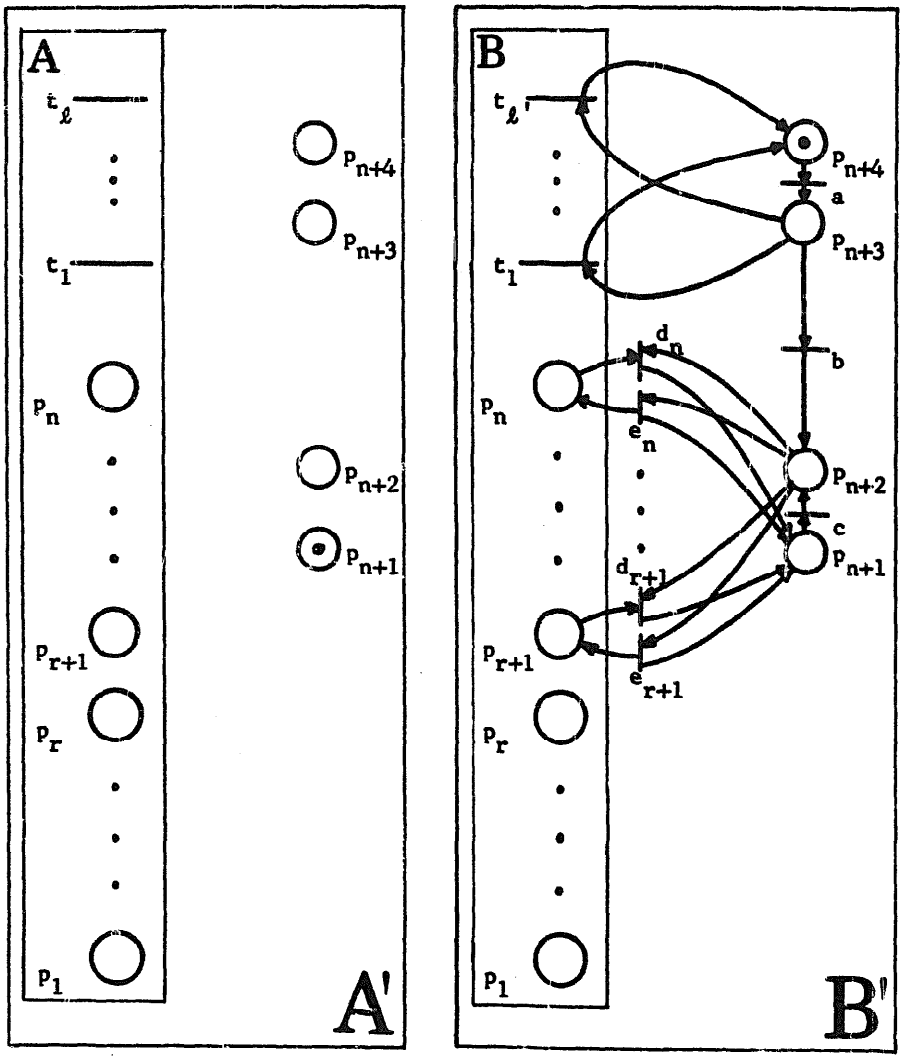
Specifically, the modifications are shown in Fig. 8. Petri net A' differs from A only in the four additional places, which are permanently marked $\langle 1,0,0,0\rangle$. Therefore, we have:

$$R(A') = R(A) \times \{\langle 1,0,0,0\rangle\}.$$

The notation speaks for itself: each marking $M' \in R(A')$ consists of a marking $M \in R(A)$ in places $p_1 \cdots p_n$ and of the marking $\langle 1,0,0,0\rangle$ in places $p_{n+1} \cdots p_{n+4}$; we write this also as $M' = M \cdot \langle 1,0,0,0\rangle$.

The Petri net B', constructed from Petri net B, has, in addition to the four new places $p_{n+1} \cdots p_{n+4}$ which are initially marked $\langle 0,0,0,1\rangle$, several new transitions.

(1) Every old transition, t_1 through t_k, carries one token from p_{n+3} to p_{n+4}. The new transition a carries one token from p_{n+4} to p_{n+3}. This means that to each firing



$$P_r(R(A)) \subseteq P_r(R(B)) \Leftrightarrow R(A') \subseteq R(B')$$

Fig. 8. SIP is reducible to IP.

sequence in B there corresponds a firing sequence in B' obtained by preceding each firing of a t_i, $1 \leq i \leq l$, by a firing of a. The markings generated in B' by these firing sequences are clearly either of the form $M \cdot \langle 0,0,0,1\rangle$ or $M \cdot \langle 0,0,1,0\rangle$, where $M \in R(B)$.

(2) The new transition b carries a token from p_{n+3} to p_{n+2}. After that transition has fired, places p_{n+3} and p_{n+4} will remain empty, and no old transition $t_1 \cdots t_l$ will ever be firable again; in fact, the firing of transition b switches the old net B off.

(3) No new transitions are connected to places $p_1 \cdots p_r$. Thus, after transition b has fired and B has been switched off, the marking of these places remains frozen. Thus, the projections of the reachability sets of B and B' on the first r coordinates are the same: $P_r(R(B')) = P_r(R(B))$.

(4) For each "uninteresting" place, $p_{r+1} \cdots p_n$, we introduce two new transitions d_i and e_i, $r+1 \leq i \leq n$. These transitions carry a token from p_{n+2} to p_{n+1} and, in addition, either decrement (d_i) or increment (e_i) the marking in the "uninteresting"

places. The last new transition, c, carries a token from p_{n+1} back to p_{n+2}. Thus, after the firing of b has put one token on p_{n+2}, that token can shuttle back and forth between p_{n+2} and p_{n+1} while arbitrarily changing the marking in places $p_{r+1}\cdots p_n$, thus effectively "erasing" their marking. Since any new marking $M' \in \mathbf{N}^{n-r}$ can be reached in $p_{r+1}\cdots p_n$ while the marking in $p_1\cdots p_r$ remains frozen at some $M \in P_r(R(B))$, the remaining markings in $R(B')$ are of the form $M \cdot M' \cdot \langle 0,1,0,0\rangle$ or $M \cdot M' \cdot \langle 1,0,0,0\rangle$, where $M \in P_r(R(B))$ and $M' \in \mathbf{N}^{n-r}$.

Therefore,

$$R(B') = R(B) \times \{\langle 0,0,0,1\rangle, \langle 0,0,1,0\rangle\} \cup P_r(R(B)) \times \mathbf{N}^{n-r} \times \{\langle 0,1,0,0\rangle, \langle 1,0,0,0\rangle\}.$$

Recall that

$$R(A') = R(A) \times \{\langle 1,0,0,0\rangle\}.$$

Thus

$$R(A') \subseteq R(B') \Leftrightarrow R(A) \subseteq P_r(R(B)) \times \mathbf{N}^{n-r}$$

$$\Leftrightarrow P_r(R(A)) \subseteq P_r(R(B)).$$

Since we constructed an instance of the IP from a given instance of the SIP, we conclude that the SIP is reducible to the IP. □

Rabin's Theorem. *Given two VASs or Petri nets, it is undecidable whether the reachability set of one is a subset of the reachability set of the other.*

Proof. This follows directly from the undecidability of Hilbert's tenth problem and the cumulative effect of Theorems 2, 3 and 4. All constructions have used Petri nets which can be directly translated into vector addition systems whose coordinates are in fact limited to $\{-1, 0, +1\}$. □

5.4. *The EP*

We are now ready to produce our original contribution to the art.

Theorem 5. *The inclusion problem is recursively reducible to the equality problem.*

Proof. Suppose we are given two n-place Petri nets (n-dimensional VAS) A and B. We wish to test whether $R(A) \subseteq R(B)$. We shall construct from A and B two Petri nets D and E such that

$$R(A) \subseteq R(B) \Leftrightarrow R(D) = R(E).$$

Both nets D and E will be constructed from a common net C which, in a sense, encodes the union $R(A) \cup R(B)$, and we shall use the fact that

$$R(A) \subseteq R(B) \Leftrightarrow R(B) = R(A) \cup R(B).$$

Petri net C is constructed as follows. First, we identify the places of A with the corresponding places of B. This produces the first n places of C. If $a_1\cdots a_l$ are the transitions of A and $b_1\cdots b_k$ are the transitions of B, then place p_i of C is

connected to a_j iff it is so connected in A, and to b_j iff it is so connected in B. In addition, C contains five new places $p_{n+1} \cdots p_{n+5}$ and four new transitions a', a'', b', b''. Places p_{n+1} and p_{n+2} are connected via each transition a_j in one direction, and via a'' in the other. If these places contain no token, then all transitions of A are disabled, otherwise the transitions can fire as they would in A. Places p_{n+3} and p_{n+4}, and transition b'', do the same for the transitions of B. Thus, depending on which pair $\{p_{n+1}, p_{n+2}\}$ or $\{p_{n+3}, p_{n+4}\}$ contains a token, we can simulate either the firing sequences of A or those of B, as well as the corresponding markings, provided we start with the proper initial marking (see Fig. 9).

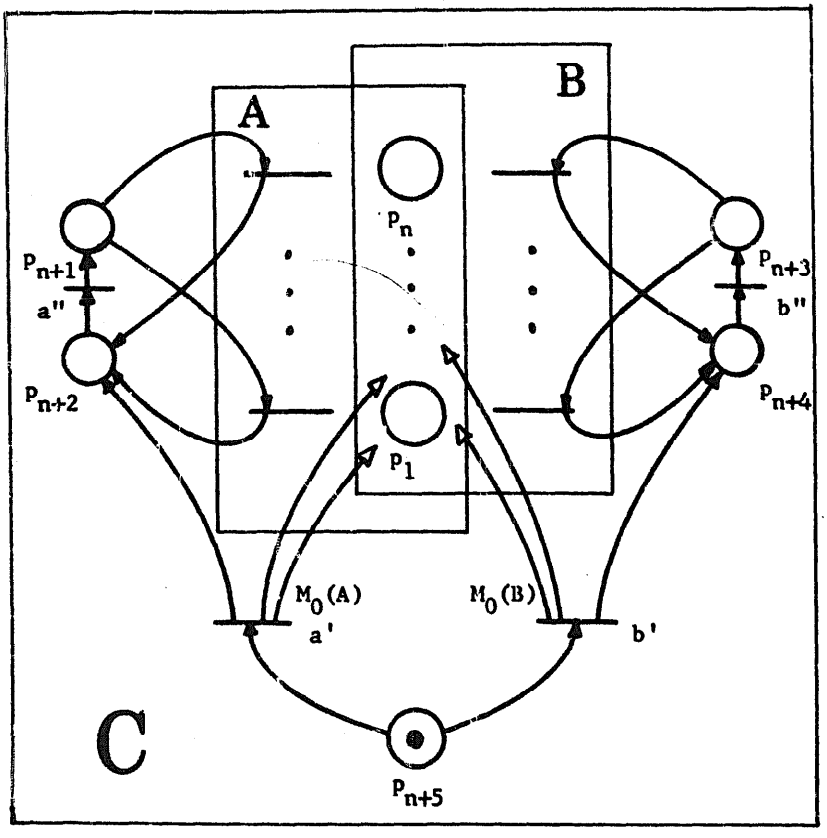


Fig. 9. C can simulate either A or B.

This is enforced by place p_{n+5} and transitions a' and b'. The initial marking of C consists of one token in p_{n+5} and zero tokens everywhere else. Transition a' puts the initial marking $M_0(A)$ of A on places $p_1 \cdots p_n$ and transfers the initial token from p_{n+5} to p_{n+2}; transition b' puts the initial marking $M_0(B)$ of B on $p_1 \cdots p_n$ and transfers the initial token from p_{n+5} to p_{n+4}.

If the initial markings of A and B have at most one token in any one place, then transitions a' and b' can be those of restricted Petri nets (restricted VAS), otherwise we need arc bundles (unrestricted VAS). But we may recall that, in the constructions for Theorems 3 and 4, we have never encountered an initial marking with more than one token per place; in fact, the initial markings for the nets constructed for Theorem 4 from those produced by Theorem 3 consist precisely of

one token in each of two places, as can be checked from Figs. 7 and 8. So, for the purpose of the undecidability proof of EP, we need to consider only restricted Petri nets.

The reachability set of C can now be described as follows:

(i) initial marking: $\langle 0\rangle^n \cdot \langle 0,0,0,0,1\rangle$,

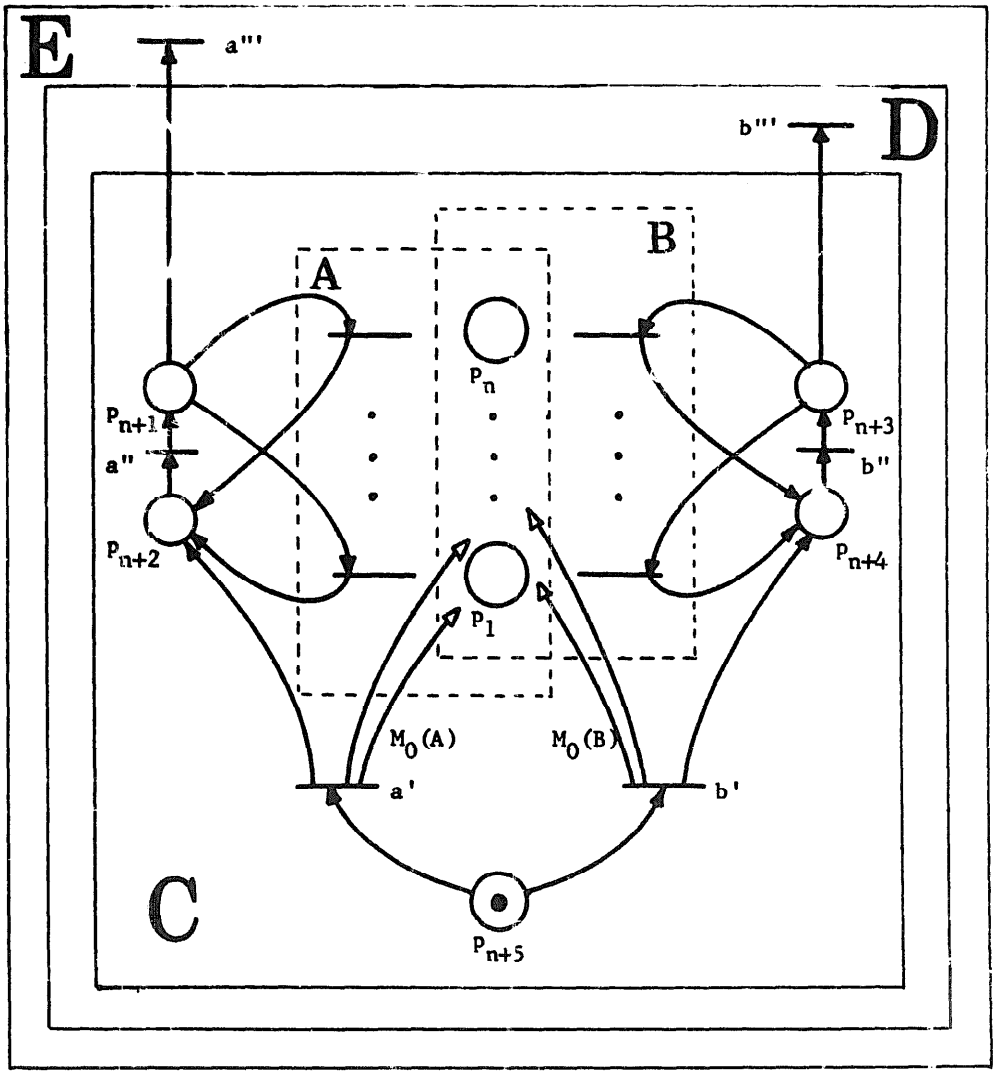
(ii) if the first firing was a': $R(A)\times\{\langle 0,1,0,0,0\rangle, \langle 1,0,0,0,0\rangle\}$,

(iii) if the first firing was b': $R(B)\times\{\langle 0,0,0,1,0\rangle, \langle 0,0,1,0,0\rangle\}$.

Thus

$$R(C) = \langle 0\rangle^n \times \{\langle 0,0,0,0,1\rangle\}$$
$$\cup R(A)\times\{\langle 0,1,0,0,0\rangle, \langle 1,0,0,0,0\rangle\}$$
$$\cup R(B)\times\{\langle 0,0,0,1,0\rangle, \langle 0,0,1,0,0\rangle\}.$$

Now we can construct D and E as shown in Fig. 10. D is obtained from C by adding transition b''', which removes a token from p_{n+3}. This can happen only if the



$$R(A)\subseteq R(B) \Leftrightarrow R(D) = R(E)$$

Fig. 10. IP is reducible to EP.

first firing was b', and thus the only new markings are of the form $M \cdot \langle 0,0,0,0,0\rangle$, where $M \in R(B)$. Thus

$$R(D) = R(C) \cup (R(B) \times \{\langle 0,0,0,0,0\rangle\}).$$

Petri net E is obtained from Petri net D by adding, in addition to b''', the transition a''' which removes a token from p_{n+1}. Thus

$$\begin{aligned} R(E) &= R(D) \cup R(A) \times \{\langle 0,0,0,0,0\rangle\} \\ &= R(C) \cup ((R(A) \cup R(B)) \times \{\langle 0,0,0,0,0\rangle\}). \end{aligned}$$

Since no marking in $R(C)$ ends in $\langle 0,0,0,0,0\rangle$, we conclude that

$$R(D) = R(E) \Leftrightarrow R(A) \subseteq R(B). \ \square$$

This allows us to announce:

Corollary. *The equality problem for vector addition system reachability sets is undecidable.*

In fact, we have proved a much stronger result, since the instance of the EP used in Theorem 5 is quite singular: The two Petri nets whose reachability sets we compare differ only by the presence or absence of a single transition a'''!

Thus we may state:

Theorem 6. *It is undecidable whether the removal of a particular transition in a Petri net or VAS changes the reachability set or not.*

We should point out, however, that this result is not as drastic as it might seem; even though the set of reachable markings may not change, its *connectivity*, as determined by which marking is reachable from which other marking by which firing sequence, is usually quite changed.

References

[1] H. G. Baker, Jr., Rabin's proof of the undecidability of the reachability set inclusion problem of vector addition systems, Computation Structures Group Memo 79, Project MAC, M.I.T., Cambridge, Mass., July 1973.

[2] M. H. Hack, Analysis of production schemata by Petri nets, Technical Report TR-94, Project MAC, M.I.T., Cambridge, Mass., February 1972; Corrections to "Analysis of production schemata by Petri nets", Computation Structures Note 17, Project MAC, M.I.T., Cambridge, Mass., June 1974.

[3] M. H. Hack, Decision problems for Petri nets and vector addition systems, MAC Technical Memorandum 59, Project MAC, M.I.T., Cambridge, Mass., March 1975.

[4] M. H. Hack, The recursive equivalence of the reachability problem and the liveness problem for Petri nets and vector addition systems, in: *Proc. 15th Annual IEEE Symposium on Switching and Automata Theory*, New Orleans, October 1974.

[5] D. Hilbert, Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Kongress zu Paris 1900, *Nachr. Akad. Wiss. Göttingen Math.-Phys.* (1900) 253–297; Translation: *Bull. Am. Math. Soc.* **8** (1901–1902) 437–479.

[6] A. W. Holt et al., Final report of the information systems theory project, Technical Report RADC-TR-68-305, Rome Air Development Center, Griffiss Air Force Base, New York, 1968.

[7] A. W. Holt and F. Commoner, Events and conditions, in: *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation* (ACM, New York, 1970) 3–52.

[8] R. M. Karp and R. E. Miller, Parallel program schemata: A mathematical model for parallel computation, *J. Comput. System Sci.* **3** (2) (1969) 147–195.

[9] R. M. Keller, Vector replacement systems: A formalism for modeling asynchronous systems, TR 117, Computer Science Laboratory, Princeton University, December 1972.

[10] Ju. V. Matijas'evič, Enumerable sets are diophantine, *Soviet Math. Dokl.* **11** (2) (1970) 354–357.

[11] R. E. Miller, Some relationships between various models of parallelism and synchronization, IBM Research report RC-5074, October 1974.

[12] B. O. Nash, Reachability problems in vector addition systems, *Am. Math. Monthly* **80** (1973) 292–295.

[13] C. A. Petri, Kommunikation mit Automaten, University of Bonn, 1962.

[14] M. Rabin, Private communication, Fall 1972.

[15] J. van Leeuwen, A partial solution to the reachability problem for vector addition systems, in: *6th Annual ACM Symposium on Theory of Computing*, May 1974, 303–309.