The Complexity of the Finite Containment Problem for Petri Nets

ERNST W. MAYR AND ALBERT R. MEYER

Massachusetts Institute of Technology, Cambridge, Massachusetts

ABSTRACT. If the reachability set of a Petri net or vector addition system is finite, it can be effectively constructed. Furthermore, this finiteness is decidable The complexity of decision procedures for the containment and equality problem of finite reachability sets is investigated, and it is shown by reducing a bounded version of Hilbert's Tenth Problem to the finite containment problem that these two problems are extremely hard—that, in fact, the complexity of each decision procedure exceeds any primitive recursive function infinitely often The finite containment and equality problems are thus the first uncontrived decidable problems which are not primitive recursive

KEY WORDS AND PHRASES. inclusion problem, reachability set, Petri net, primitive recursive complexity

CR CATEGORIES: 5.21, 5.25, 5.27

1. *Introduction*

The containment problem for Petri nets or vector addition systems [14] is the problem of determining for any two given Petri nets whether the reachability set of the first is contained in the other. (Though vector addition systems and Petri nets are essentially notational variants describing the same mathematical systems, we prefer to use the Petri net terminology because of the convenience of representation.) By reducing Hilbert's Tenth Problem [10] concerning integer solutions of Diophantine equations, which is known to be undecidable [6, 17], to the containment problem, Rabin has shown the unsolvability of the latter problem (see [3]). The situation changes, however, when one considers subclasses of the general problem. A result by Karp and Miller [13] shows that it is decidable whether the reachability set of a given Petri net is finite. They also give an algorithm for generating any finite reachability set. Hence the finite containment problem (FCP) (the finite equality problem (FEP)), that is, the problem of determining for any two given Petri nets whether their reachability sets are each finite and one is contained in (is equal to) the other, is decidable by exhaustion. We investigate the complexity of decision procedures for FCP and prove the following result.

THEOREM. *FCP (FEP) is decidable, but the complexity of each decision procedure for FCP (FEP) exceeds any primitive recursive function infinitely often.*

This paper was prepared with the support of the National Science Foundation under Contract MCS 74-12997-A04 and with the support of the German Academic Exchange Service (DAAD) under Grant 430-402-559-7

Authors' address: Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139

The intrinsic complexity of the decision procedures is not due to the fact that the reachability sets have to be tested for finiteness. Rackoff [19] shows that this can be achieved in exponential space. Thus, even if the decision procedure is supplied with the answer to this subproblem, the complexity still is nonprimitive recursive.

To establish this result, we present in Section 2, following the basic definitions, a bounded version of Hilbert's Tenth Problem with nonprimitive recursive complexity. Section 3 contains results for three constructions of weak Petri net computers (WPNCs) which are carried out in detail in the appendix (see also [18]). We first describe WPNCs for a sequence of functions which are closely related to Ackermann's function [1]. We then discuss a property of WPNCs for multivariate polynomials with nonnegative integer coefficients which makes it possible to reduce the subspace inclusion problem for reachability sets to the inclusion problem while preserving the finiteness of the reachability sets. Theorems 3 and 4 state results about two modifications of polynomial WPNCs which further exploit this property in order to reduce the bounded version of Hilbert's Tenth Problem effectively to FCP. In Section 4 the reduction is carried out, and the main results of the paper are proved. Section 5 demonstrates connections to some related questions and states some open problems.

2. *Preliminaries*

2.1 PETRI NETS, FCP, AND FEP. We assume that the reader is familiar with notions like the free monoid T^* over a finite alphabet T (the empty word is denoted by λ, $T^+ := T^* - \{\lambda\}$ is the set of nonempty words over T, and $|w|$ is the length of a word $w \in T^*$); the concept of the free commutative monoid generated by a finite set S, which we write $C(S)$; and basic algebraic concepts like the semiring $\mathrm{N}[\mathrm{x}_m] := \mathrm{N}[x_1, \ldots, x_m]$ of polynomials with nonnegative integer coefficients in the unknowns $x_1, \ldots, x_m$.

Definition 1

(a) A *Petri net* $\mathscr{P}$ is a 4-tuple $(S, T, pre, post)$ with the properties

- (i) S is a finite ordered set of *places*;
- (ii) T is a finite set of *transitions*, $S \cap T = \emptyset$;
- (iii) *pre* is a multiset over $S \times T$;
- (iv) *post* is a multiset over $T \times S$.

(b) A *marking* of $\mathscr{P}$ is a mapping

$$a: S \to \mathrm{N} \qquad (\mathrm{N} = \text{set of nonnegative integers})$$

(or, equivalently, an element of $C(S)$, $a = \prod_{s \in S} s^{a(s)}$).

In diagrams, places are denoted by circles, transitions by bars, and elements of *pre* ∪ *post* by directed arcs. Numbers attached to arrows give the multiplicity μ of the corresponding element in *pre* ∪ *post* if this multiplicity is greater than one. If $(s, t) \in pre$, then s is called an *input place* of t, and if $(t, s) \in post$, an *output place* of t. A transition t is said to be *controlled* by a place s iff $\mu(s, t) = \mu(t, s) = 1$ (represented in diagrams by a double line between s and t). A transition t with no output place and one input place with $\mu(s, t) = 1$ is called an *erasing* transition (of s).

Definition 2. Let $\mathscr{P} = (S, T, pre, post)$ be a Petri net and a a marking of $\mathscr{P}$.

(a) A transition $t \in T$ is *firable* at a and takes a to the marking b (written $a \rightarrow^{t} b$) iff

- (i) $(\forall s \in S)[a(s) \geq \mu(s, t)]$, and
- (ii) $(\forall s \in S)[b(s) = a(s) - \mu(s, t) + \mu(t, s)]$.

(b) A *firing sequence* $\mathbf{t}$ is an element $\mathbf{t} \in T^+$.

(c) A firing sequence $\mathbf{t}$ is firable at a and takes a to the marking b (written $a \to^{\mathbf{t}} b$) iff

$$(\exists r \geq 1\ \exists t_1, \ldots, t_r \in T)[\mathbf{t} = t_1 t_2 \cdots t_r \text{ and}$$
$$(\exists b_0, b_1, \ldots, b_r)[a = b_0 \wedge b = b_r \wedge (\forall i \in \{1, \ldots, r\})[b_{i-1} \to^{t_i} b_i]]].$$

The sequence $(b_i)_{0 \leq i \leq r}$ is called the *marking sequence* generated by $\mathbf{t}$.

(d) A marking b of $\mathscr{P}$ is said to be reachable from a (written $a \to^* b$) iff $a = b$ or $(\exists \mathbf{t} \in T^+)[a \to^{\mathbf{t}} b]$.

$\to^t$, $\to^{\mathbf{t}}$, and $\to^*$ depend, of course, on $\mathscr{P}$. It will, however, always be clear from the context which Petri net is being considered.

Definition 3.

(a) The *reachability set* of a Petri net $\mathscr{P}$ with initial marking a is

$$R(\mathscr{P}, a) := \{b;\ a \to^* b\}.$$

Let $\mathscr{P}_i = (S_i, T_i, pre_i, post_i)$ be a Petri net with initial marking a_i ($i = 1, 2$), $|S_1| = |S_2|$ (footnote 1), and let $\bar{h}: C(S_1) \to C(S_2)$ be the semigroup-isomorphism generated by the order-preserving bijection $h: S_1 \to S_2$.

(b) The *containment problem* CP is the problem of deciding for two Petri nets $\mathscr{P}_1$ and $\mathscr{P}_2$ with markings a_1 and a_2, respectively, whether the reachability set of the first net is contained in that of the second; that is,

$$\mathrm{CP} := \{\langle(\mathscr{P}_1, a_1), (\mathscr{P}_2, a_2)\rangle;\ \bar{h}(R(\mathscr{P}_1, a_1)) \subseteq R(\mathscr{P}_2, a_2)\}.$$

(c) The *finite containment problem* FCP is

$$\mathrm{FCP} := \{\langle(\mathscr{P}_1, a_1), (\mathscr{P}_2, a_2)\rangle;\ |R(\mathscr{P}_2, a_2)| < \infty \text{ and } \langle(\mathscr{P}_1, a_1), (\mathscr{P}_2, a_2)\rangle \in \mathrm{CP}\}.$$

(d) The *finite equality problem* FEP is

$$\mathrm{FEP} := \{\langle(\mathscr{P}_1, a_1), (\mathscr{P}_2, a_2)\rangle;\ \langle(\mathscr{P}_1, a_1), (\mathscr{P}_2, a_2)\rangle, \langle(\mathscr{P}_2, a_2), (\mathscr{P}_1, a_1)\rangle \in \mathrm{FCP}\}.$$

The proof that FCP and FEP are nonprimitive recursive proceeds by effectively reducing to FCP a special bounded version of Hilbert's Tenth Problem dealing with the ranges of values of polynomials with nonnegative integer coefficients. The complexity is measured in terms of the number of steps any Turing machine for FCP and FEP, respectively, needs on instances of the problems, as a function of the length of the input.

Though the main results of this paper hold for any reasonable encoding of the data involved (i.e., polynomials and Petri nets), we choose for definiteness particular encodings and corresponding notions of the size of encodings.

Petri nets are encoded essentially by writing down a list of the arcs in the net, preceded by the multiplicity if it is greater than 1. By ordering the edges according to the transitions on which they are incident, it suffices to show only the places touched by the edges. A place can be identified by the radix notation for its number in the ordered set of places. The length of this encoding motivates

Definition 4. Let $\mathscr{P} = (S, T, pre, post)$ be a Petri net with marking a. Then

$$size(pre) := \sum_e \lceil \log(\mu(e) + 1) \rceil \qquad \text{(footnote 2)},$$

[1] For a set S, $|S|$ denotes the cardinality of S

[2] All logarithms are to the base 2 throughout the paper.

where the sum is taken over the *different* arcs in the multiset *pre*; *size*(*post*) is defined similarly:

$$size(\mathscr{P}) := (size(pre) + size(post) + 1)\cdot\lceil\log(|S| + 1)\rceil;$$
$$size(\mathscr{P}, a) := size(\mathscr{P}) + |S|\cdot\lceil\log(1 + \max\{a(s);\ s \in S\})\rceil.$$

The code for a polynomial $p \in \mathrm{N}[\mathbf{x}_m]$ consists of the sequence of the codes for its monomial constituents. The code for a monomial is a sequence of integers obtained by first writing down the nonzero integer coefficient, then the nondecreasing sequence of indices in which each $j \in \{1, \ldots, m\}$ occurs just as often as the degree of x_j in the monomial indicates. Numbers are written again in radix notation. The size of $p \in \mathrm{N}[\mathbf{x}_m]$ is the length of its code.

2.2 A Bounded Version of Hilbert's Tenth Problem. While Hilbert's Tenth Problem [10] concerning (nonnegative) integer solutions of Diophantine equations is undecidable [6, 17], we restrict ourselves to finding solutions in some finite initial segment of N, a problem which obviously is decidable by exhaustion. In particular, we consider segments bounded by the function $A : \mathrm{N} \to \mathrm{N}$, defined by

$$\left.\begin{aligned} A_0(x) &:= 2x + 1, \\ A_{n+1}(0) &:= 1, \\ A_{n+1}(x + 1) &:= A_n(A_{n+1}(x)), \\ A(n) &:= A_n(2), \end{aligned}\right\} \quad n, x \in \mathrm{N}.$$

A function similar to A is studied in [7]. The results obtained there imply that A majorizes the primitive recursive functions, that is,

$$(\forall f, f \text{ primitive recursive } \exists n_0 \in \mathrm{N}\ \forall n \geq n_0)[A(n) > f(n)].$$

Definition 5. The *Bounded Polynomial Inequality Problem* BPI is

$$\mathrm{BPI} := \{(p, q, n);\ p, q \in \mathrm{N}[\mathbf{x}_m] \text{ for some } m \in \mathrm{N}, \textit{ and } (\forall \mathbf{y}_m \in \{0, 1, \ldots, A(n)\}^m)[p(\mathbf{y}_m) \leq q(\mathbf{y}_m)]\}.$$

Results by Adleman and Manders [2] imply

Theorem 1. *BPI is nonprimitive recursive; that is, if M is a (multitape) Turing machine for BPI and f is any primitive recursive function, then there are infinitely many instances (p, q, n) such that M needs more than f(size(p) + size(q) + n) steps to determine whether (p, q, n) ∈ BPI.*

Proof. Using straightforward transformations of polynomials with (not necessarily nonnegative) integer coefficients, it follows from [2, Th. 5] that the complexity of BPI is greater than $\log(\log(\log(A(m^{1/4})^{1/5})))$ on some inputs of size m for infinitely many m. This latter function must also majorize the primitive recursive functions, since A does. Q.E.D.

3. *Weak Petri Net Computers* (*WPNCs*)

Rabin has introduced the concept of a number-theoretic function being weakly computed by a Petri net (cf. [8]). To weakly compute a function of m arguments, one takes a net in which m places are designated to be marked initially with the values of the arguments. The value of the function equals the maximum value of the reachable markings on some other designated place. It is also convenient to have a so-called start place which initially determines the firability of firing sequences in the net. We use the names $\mathbf{i}_m$, o, and s, respectively, for these places, and, for $\mathbf{n}_m \in \mathrm{N}^m$, the abbreviation $\mathbf{i}_m^{\mathbf{n}_m}$ for the word $\prod_{j=1}^{m} i_j^{n_j} \in C(S)$.

Definition 6. Let $\mathscr{P} = (S, T, pre, post)$ be a Petri net, and let $s, \mathbf{i}_m, o \in S$ be $m + 2$ designated places such that $s, \mathbf{i}_m$ are not output places and o is not an input place of any transition in T. Let $r \in C(S - \{s, \mathbf{i}_m, o\})$, $D \subseteq \mathbf{N}^m$, and $f: D \to \mathbf{N}$. $\mathscr{P}$ is an *r-weak Petri net computer* (r-WPNC) (footnote 3) for f iff

$$(\forall \mathbf{n}_m \in D)[\{k;\ s\mathbf{i}_m^{\mathbf{n}_m} r \to^* o^k a \text{ for some } a \in C(S - \{s, \mathbf{i}_m, o\})\} = \{0, 1, \ldots, f(\mathbf{n}_m)\}].$$

The following theorems summarize results of constructions carried out in detail in the appendix.

THEOREM 2

(*i*) $(\forall n \in \mathbf{N}\ \exists \mathscr{A}_n)[\mathscr{A}_n$ *is a* λ*-WPNC for* A_n *with three designated places* $s_n, i_n, o_n]$;
(*ii*) $size(\mathscr{A}_n) = O(n \cdot log(n))$;
(*iii*) $R(\mathscr{A}_n, s_n i_n^2)$ *is finite.*

See Section A1 for the proof.

The nets $\mathscr{A}_n$ are used to supply the arguments for WPNCs for polynomials with nonnegative integer coefficients. As far as these polynomial WPNCs are concerned, we are actually only interested in the markings of the designated places, as these reflect all the information about the polynomials regarded as functions. In [9] a construction is given which allows the reduction of the subspace containment problem, that is, the containment problem with respect to designated places in nets, to the containment problem. Unfortunately, however, this construction does not preserve the finiteness of reachability sets. We now introduce two modifications of (polynomial) WPNCs which enable us to circumvent this obstacle, namely, *bounded* and *blurring* WPNCs.

An M-bounded WPNC is one in which all markings reachable from the initial marking are bounded by M on the nondesignated places. More precisely, for $u \in S$, $n \in \mathbf{N}$, let $\langle u \rangle^n$ denote $\{\lambda, u, u^2, \ldots, u^n\} \subset C(S)$.

Definition 7. Let $\mathscr{P} = (S, T, pre, post)$ be a Petri net, $M \in \mathbf{N}$. $\mathscr{P}$ is an *M-bounded r-WPNC* for $f: \mathbf{N}^m \supseteq D \to \mathbf{N}$ with designated places $S_\mathrm{d} := \{s, \mathbf{i}_m, o\} \subseteq S$ iff

(i) $\mathscr{P}$ is an r-WPNC for f with designated places S_d;
(ii) $(\forall \mathbf{n}_m \in D)[R(\mathscr{P}, s\mathbf{i}_m^{\mathbf{n}_m} r) \subseteq \langle s \rangle^1 \prod_{j=1}^m \langle i_j \rangle^{n_j} \prod_{u \in S - S_\mathrm{d}} \langle u \rangle^M \langle o \rangle^{f(\mathbf{n}_m)}]$.

THEOREM 3. *Let* $p \in \mathbf{N}[\mathbf{x}_m]$, $\|p\| :=$ *maximum of the coefficients of* p, *and for* $N \in \mathbf{N}$, *let* $g(N) := N + \|p\|$. *Then there exists a Petri net* $\mathscr{P}_{bd} = (S, T, pre, post)$ *with* $m + 3$ *designated places* $s, \mathbf{i}_m, o$, *and* $b \in S$ *such that*

(*i*) $\mathscr{P}_{bd}$ *is a* $g(N)$*-bounded* $b^{g(N)}$*-WPNC with designated places* $s, \mathbf{i}_m, o$ *for* p *restricted to* $\{0, 1, \ldots, N\}^m$, *for all* $N \in \mathbf{N}$. *In particular,*

$$(\forall N \in \mathbf{N}\ \forall \mathbf{n}_m \in \{0, 1, \ldots, N\}^m)[R(\mathscr{P}_{bd}, s\mathbf{i}_m^{\mathbf{n}_m} b^{g(N)}) \text{ is finite}].$$

(*ii*) $size(\mathscr{P}_{bd}) = O(size(p) \cdot log(size(p)))$.

See Section A2 for the proof.

The second modification of polynomial WPNCs ensures the possibility of "blurring" any telltale information which is represented in the markings of the nondesignated places. This is achieved by constructing the net so that every marking of the nondesignated places up to a prespecified bound is reachable.

Definition 8. A Petri net $\mathscr{P} = (S, T, pre, post)$ is a *blurring WPNC* for $f: \mathbf{N}^m \supseteq D \to \mathbf{N}$ iff $\mathscr{P}$ has $m + 5$ designated places $S_\mathrm{d} := \{s, \mathbf{i}_m, o, c_1, c_2, e\} \subset S$ such that

[3] If we do not want to emphasize the marking r, we also call $\mathscr{P}$ a WPNC

(i) $(\forall N \in \mathbf{N})[\mathscr{P}$ is an e^N-WPNC for f with respect to s, $\mathbf{i}_m$, o];
(ii) $(\forall N \in \mathbf{N}\ \forall \mathbf{n}_m \in D)[\prod_{u \in S - S_d} \langle u \rangle^N \langle o \rangle^{f(\mathbf{n}_m)} \subseteq R(\mathscr{P}, s\mathbf{i}_m^{\mathbf{n}_m} e^N)]$.

(Note that the reachable markings considered in (ii) do not contain c_1, c_2, and e.)

THEOREM 4. *Let $q \in \mathbf{N}[\mathbf{x}_m]$ for some $m \in \mathbf{N}$. Then there exists a Petri net $\mathscr{Q}_{\mathscr{U}} = (S, T, pre, post)$ such that*

(i) $\mathscr{Q}_{\mathscr{U}}$ is a blurring WPNC for q with designated places s, $\mathbf{i}_m$, o, c_1, c_2, $e \in S$;
(ii) $(\forall N \in \mathbf{N}\ \forall \mathbf{n}_m \in \{0, 1, \ldots, N\}^m)[R(\mathscr{Q}_{\mathscr{U}}, s\mathbf{i}_m^{\mathbf{n}_m} e^N)$ is finite];
(iii) $size(\mathscr{Q}_{\mathscr{U}}) = O(size(q) \cdot log(size(q)))$.

See Section A3 for the proof.

4. *Reduction of BPI to FCP*

The results of the previous section now enable us to reduce BPI to FCP efficiently. We prove

THEOREM 5. *BPI is polynomial-time-reducible to FCP.*

PROOF. Given (p, q, n), we first construct the two Petri nets $\mathscr{P}_{\mathscr{A}}$ and $\mathscr{Q}_{\mathscr{A}}$ as indicated in Figure 1. Each net contains a copy of the λ-WPNC $\mathscr{A}_n$ for A_n of Theorem 2. $\mathscr{P}_{\mathscr{A}}$ also contains the bounded version $\mathscr{P}_{\mathscr{B}\mathscr{d}}$ of a WPNC for p of Theorem 3, and $\mathscr{Q}_{\mathscr{A}}$ the blurring WPNC $\mathscr{Q}_{\mathscr{U}}$ for q of Theorem 4. (The start place and the input places of the latter have been primed in order to avoid confusion with the corresponding places of $\mathscr{A}_n$.) $\mathscr{P}_{\mathscr{A}}$ also has unconnected places c_1, c_2, e (with no tokens on them) to match the corresponding places in $\mathscr{Q}_{\mathscr{U}}$ of $\mathscr{Q}_{\mathscr{A}}$ which do not get blurred in $\mathscr{Q}_{\mathscr{U}}$. We may assume that $\mathscr{P}_{\mathscr{A}}$ and $\mathscr{Q}_{\mathscr{A}}$ have the same number of places. (If this is not the case a priori, one can either add further dummy places which are not connected to any transition to $\mathscr{P}_{\mathscr{A}}$ or add further dummy places with erasing transitions attached to them to $\mathscr{Q}_{\mathscr{A}}$ within $\mathscr{Q}_{\mathscr{U}}$; see Section A3.) The count place shown in the figure "remembers" the maximal input to the polynomial WPNC. $\mathscr{A}_n$, $\mathscr{P}_{\mathscr{B}\mathscr{d}}$, $\mathscr{Q}_{\mathscr{U}}$, and thus $\mathscr{P}_{\mathscr{A}}$ and $\mathscr{Q}_{\mathscr{A}}$ can be constructed from (p, q, n) in time bounded by a polynomial in $size(p) + size(q) + n$. To conclude the proof, it suffices to show

LEMMA 1

$$(p, q, n) \in BPI \Leftrightarrow \langle(\mathscr{P}_{\mathscr{A}}, s), (\mathscr{Q}_{\mathscr{A}}, s)\rangle \in FCP.$$

PROOF OF THE LEMMA. We assume that the two sets of places are ordered suitably, for example, first s, then the places in the $\mathscr{A}_n$-copies (in the same order in both nets), then *count*, s', $\mathbf{i}'_m$, and o, followed by e, c_1, c_2, and finally the remaining places in any order.

$\Rightarrow$. Consider some marking $a \in R(\mathscr{P}_{\mathscr{A}}, s)$ which contains c', $\mathbf{n}'_m$, k tokens on the respective places *count*, $\mathbf{i}'_m$, o. From the structure of $\mathscr{P}_{\mathscr{A}}$ it follows (Figure 1) that this marking could have been reached only by a firing sequence which placed c' tokens on each of the places i'_j, of which $c' - n'_j$ tokens then have been used as input by the polynomial WPNC $\mathscr{P}_{\mathscr{B}\mathscr{d}}$. As $\mathscr{P}_{\mathscr{B}\mathscr{d}}$ is a WPNC for the polynomial p, this implies that $k \leq p(c' - n'_1, \ldots, c' - n'_m)$. The marking on the nondesignated places of $\mathscr{P}_{\mathscr{B}\mathscr{d}}$ and on b, e, c_1, and c_2 is bounded by $c' + \max\{\|p\|, \|q\|\}$ (Theorem 3). Clearly, the same marking as a is reachable in $\mathscr{Q}_{\mathscr{A}}$ as far as the places in $\mathscr{A}_n$ and the places s, s', and *count* are concerned. $\mathscr{Q}_{\mathscr{U}}$ may now use $c' - n'_j$ tokens from each of the places i'_j in $\mathscr{Q}_{\mathscr{A}}$ and output $k \leq q(c' - n'_1, \ldots, c' - n'_m)$ tokens on o, since $(p, q, n) \in$ BPI. Then $\mathscr{Q}_{\mathscr{U}}$ can blur all its nondesignated places appropriately and reach a marking with no

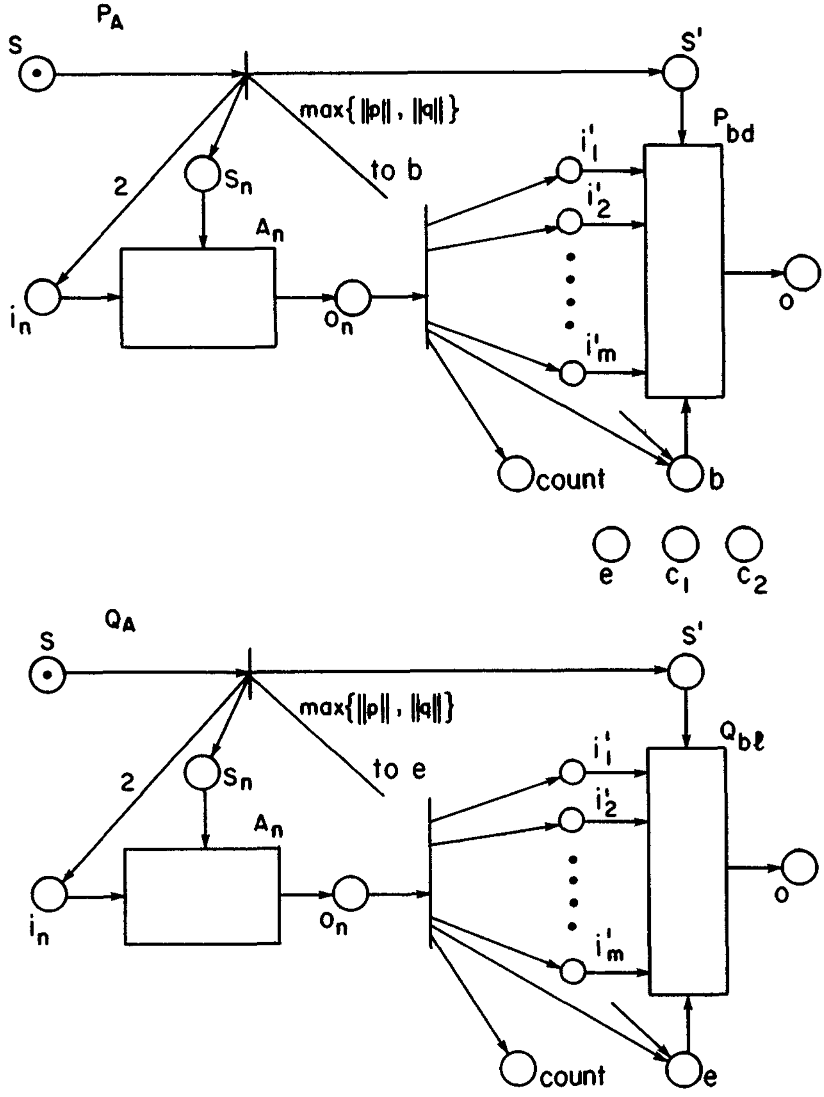


FIG 1. Petri nets $\mathscr{P}_{\mathscr{A}}$ and $\mathscr{Q}_{\mathscr{A}}$ for the reduction of BPI to FCP

tokens on e, c_1, and c_2 (Definition 8(ii)), thus matching a. $R(\mathscr{P}_{\mathscr{A}}, s)$ and $R(\mathscr{Q}_{\mathscr{A}}, s)$ are finite (this comes from Theorems 2(iii), 3(i), 4(ii), and the construction of $\mathscr{P}_{\mathscr{A}}$ and $\mathscr{Q}_{\mathscr{A}}$, respectively). It then follows that $\langle(\mathscr{P}_{\mathscr{A}}, s), (\mathscr{Q}_{\mathscr{A}}, s)\rangle \in$ FCP.

$\Leftarrow$. Since $R(\mathscr{P}_{\mathscr{A}}, s) \subseteq R(\mathscr{Q}_{\mathscr{A}}, s)$, it follows a fortiori that the projection of $R(\mathscr{P}_{\mathscr{A}}, s)$ on the places *count*, $\mathbf{i}'_m$, o is contained in the corresponding projection of $R(\mathscr{Q}_{\mathscr{A}}, s)$. By the definition of a WPNC (Definition 7), this implies that $q(c' - n'_1, \ldots, c' - n'_m) \geq k$ for $0 \leq k \leq p(c' - n'_1, \ldots, c' - n'_m)$, for all $c' \in \{0, 1, \ldots, A(n)\}$ and all $\mathbf{n}'_m$;6 $\{0, 1, \ldots, c'\}^m$. Hence $(p, q, n) \in$ BPI. Q.E.D.

THEOREM 6. *FCP is decidable, but the complexity of each decision procedure for FCP exceeds any primitive recursive function infinitely often.*

PROOF. Each primitive recursive decision method for FCP would yield a primitive recursive algorithm for BPI via the reduction of Theorem 5, and would thus contradict Theorem 1. Q.E.D.

THEOREM 7. *FEP is decidable, but the complexity of each decision procedure for FEP exceeds any primitive recursive function infinitely often.*

PROOF. Hack's reduction of the general inclusion problem to the equality problem [9, p. 122] preserves finiteness and can be effected in polynomial time. Hence the same argument as in the proof of Theorem 6 applies. Q.E.D.

We remark that Theorems 6 and 7 actually do not depend heavily on the encoding used for Petri nets and polynomials so long as the ratio to the particular code chosen in this paper is bounded by a primitive recursive function.

5. *Conclusions and Open Problems*

FCP and FEP are the first uncontrived problems shown to be decidable but not primitive recursive. (We consider BPI as contrived because the nonprimitive recursive complexity is obtained by explicitly building in a nonprimitive recursive function as upper bound for the arguments; such a special device does not appear in FCP or FEP.)

Another subclass of the class of general Petri nets for which the containment and equality problem are known to be solvable is the reversible Petri nets. It is not difficult to see that the reachability set of a reversible Petri net is a semilinear set [11], and the results of Biryukov [4] and Taiclin [20] yield a uniform constructive method for obtaining this semilinear set. As containment and equality of semilinear sets are decidable, so are the corresponding properties for the reachability sets of reversible Petri nets. It is not known, however, whether these problems are also nonprimitive recursive. In [5] it has been shown that the reachability problem for reversible Petri nets is exponential-space-complete under log-space transformability.

Other important classes of Petri nets which have been studied in detail are the persistent nets and, within this class, the proper subclass of conflict-free Petri nets [12, 15]. It is known [15] that the reachability sets of persistent nets are semilinear, but no algorithm has been found so far to obtain these semilinear sets.[4] In [12], among others, the complexity of the reachability problem for the restricted class of 1-conservative Petri nets (which have finite reachability sets) is shown to be polynomial-space-complete. Besides this special case and Lipton's exponential space lower bound [16], no nontrivial bounds are known for the finite reachability problem.

Appendix

This section supplies the constructions and proofs for the theorems in Section 3.

A1 ITERATIVE PETRI NET COMPUTERS. According to the structure of the definition of the functions A_n, $n \in \mathbf{N}$, we construct WPNCs for the A_n recursively. In such a WPNC for $A_{n+1}(m)$, the embedded WPNC for A_n is restarted m times. In general, since after a computation of a WPNC tokens may be left on nondesignated places affecting the subsequent computations, we want to ensure, in order to be able to start a WPNC iteratively, that the successive computations are properly separated and that in a computation which produces the maximum marking on the output place no "garbage"-tokens are left on the nondesignated places (markings are thought here to consist of an appropriate number of *tokens*). As the WPNCs under consideration

[4] Added in proof: Note that an algorithm to decide persistence and construct semilinear representations of persistent reachability sets is given in E.W. Mayr, Persistence of vector replacement systems is decidable, Tech. Memo 189, Lab. for Computer Science, M.I.T., Cambridge, Mass., 1980 (to appear in *Acta Informatica*)

usually are r-WPNCs for some $r \neq \lambda$, this *initialization* r has to satisfy special properties.

Definition 9. Let $\mathscr{P} = (S, T, pre, post)$ be a WPNC with designated places S_d, and for any $S' \subseteq S$ let the projection $j(S'):C(S) \rightarrow C(S')$ be the homomorphism defined by

$$j(S')(u) := \begin{cases} u & \text{if} \quad u \in S'; \\ \lambda & \text{otherwise.} \end{cases}$$

$r \in C(S)$ is *conservative* iff there is a set of "control places" $S_c \subseteq S - S_d$, such that

(i) $r \in C(S_c)$;
(ii) $(\forall a, b \in C(S))[(j(S_c)(a) = r) \wedge (a \rightarrow^* b) \Rightarrow |j(S_c)(b)| = |r|]$;
(iii) $(\forall a, b \in C(S))[(j(S_c)(a) = r) \wedge (a \rightarrow^* b) \Rightarrow (\exists t \in T^*)[j(S_c)(b) \rightarrow^t r]]$.

Definition 10. Let $\mathscr{P} = (S, T, pre, post)$ be a Petri net and $f: \mathbf{N} \rightarrow \mathbf{N}$ a number-theoretic function. $\mathscr{P}$ is an *iterative Petri net computer* (IPNC) for f with designated places $S_d = \{s, i, o\} \subseteq S$ iff the following both hold.

(i) There is a conservative $r \in C(S_c)$ for some set of control places $S_c \subseteq S - S_d$ such that $\mathscr{P}$ is an r-WPNC for f.
(ii) Let $S_o := S - (S_d \cup S_c)$ be the so-called operational places and define

$$RC_r := \{r' \in C(S_c);\ (\exists a, b \in C(S))\{j(S_c)(a) = r \wedge j(S_c)(b) = r' \wedge a \rightarrow^* b]\}.$$

Then

IC1: $(\forall a, b \in C(S_o \cup \{i, o\}), \forall r' \in RC_r)[(sar' \rightarrow^* br) \Rightarrow (|b| \leq f(|a|)]$.
IC2: $(\forall n \in \mathbf{N})[si^n r \rightarrow^* oar$ for some $a \in C(S) \Rightarrow j(\{s\})(a) = \lambda]$ and $(\forall a, b, c \in C(S_o), \forall r', r'' \in RC_r, \forall l, l', l'', k, k', n, n' \in \mathbf{N})[(s^l i^n o^k ar' \rightarrow^* s^{l'} i^n o^{k'} br'' \rightarrow^* s^{l''} i^{n'} o^{k'} cr) \wedge (k' > k) \wedge (n' < n) \Rightarrow (l'' \leq l - 1)]$.

Because of IC1, r is called an *iteratively conservative* initial marking of $\mathscr{P}$. Informally speaking, IC1 ensures that no "garbage" is produced, and IC2 means that no output can be produced without a start token s, and that input and output phases of an IPNC alternate and are controlled by s; that is, to produce any (additional) output at all, a token of s has to be consumed, and if another computation is to follow thereafter, yet another start token s has to be used. IC1 together with the fact that r is conservative ensures not only that the initial marking r of S_c can be restored, but also that there is no gain in not restoring it. We remark that because of IC1, functions for which IPNCs exist are strictly increasing.

We are now going to show that the class of functions computed by IPNCs is essentially closed under iteration. In particular, let $\mathscr{F}$ be an IPNC computing a function $f: \mathbf{N} \rightarrow \mathbf{N}$ with $f(0) > 0$, and let $g: \mathbf{N} \rightarrow \mathbf{N}$ be defined by

(i) $g(0) := 1$;
(ii) $g(n + 1) := f(g(n))\ \forall n \in \mathbf{N}$.

Define the Petri net $\mathscr{G}$ as given in Figure 2. Essentially, a feedback mechanism is added to $\mathscr{F}$ which allows the output of $\mathscr{F}$ to be transferred back to its input place as many times as there are tokens on the input place i_1 of $\mathscr{G}$. The other places constitute the so-called standard structure for IPNCs which ensures IC2. (In this standard structure the block denoted by the dotted line is considered as a black box.)

LEMMA 2. *Let f, g, $\mathscr{F}$, and $\mathscr{G}$ be as above. Then $\mathscr{G}$ is an IPNC for g.*

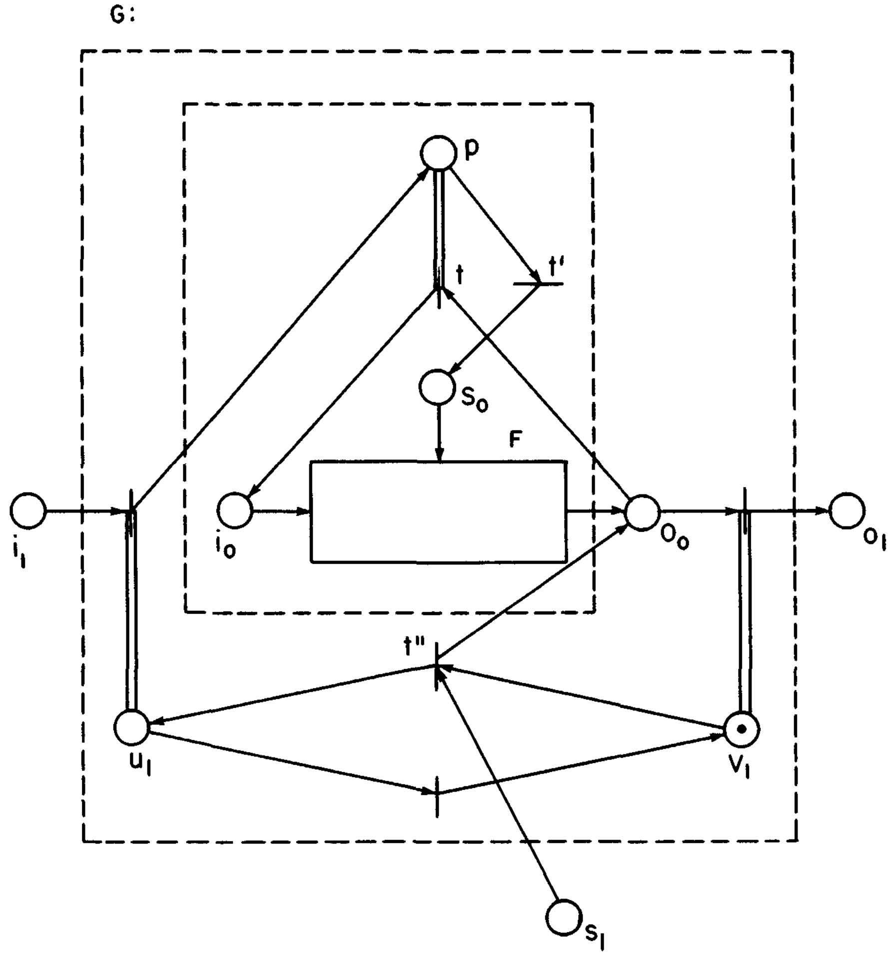


FIG. 2. Petri net computer for $g(n) = f^{(n)}(1)$

PROOF. Let r_0 be an iteratively conservative marking of $\mathcal{F}$ such that $\mathcal{F}$ is an r_0-WPNC for f, and let $S_d^0 = \{s_0, i_0, o_0\}$, S_c^0, and S_o^0 denote the set of designated, control, and operational places, respectively, of $\mathcal{F}$. With $\mathcal{F} = (S^0, T^0, pre^0, post^0)$ and $\mathcal{G} = (S^1, T^1, pre^1, post^1)$,

$$S_d^1 := \{s_1, i_1, o_1\},$$
$$S_c^1 := S_c^0 \cup \{u_1, v_1\},$$
$$S_o^1 := S^1 - (S_d^1 \cup S_c^1).$$

It can easily be seen from Figure 2 that $r_1 := v_1 r_0$ is conservative. Now let $g^* : \mathrm{N} \to \mathrm{N} \cup \{\infty\}$ be the function for which $\mathcal{G}$ is an r_1-WPNC. As property IC2 of Definition 10 is ensured by the standard structure of $\mathcal{G}$, it suffices to show (i) IC1 for g^* and (ii) $g^* = g$.

(i) It again follows from the standard structure that we may assume without loss of generality that $r' \in v_1 RC_{r_0}$ in order to show that

$$(\forall a, b \in C(S_o^1 \cup \{i_1, o_1\}), \forall r' \in RC_{r_1})[s_1 a r' \to^* b r_1 \Rightarrow |b| \le g^* (|a|)].$$

As o_1 is not an input place for any $t \in T^1$, we may also assume that $a \in C(S_o^1 \cup \{i_1\})$. Let $a = cd \neq \lambda$ with $c \in C(\{i_1, p, s_0\})$, $d \in C(S_o^0 \cup \{i_0, o_0\})$ (see Figure 2).

Case 1. $c = \lambda$. As $\mathscr{F}$ is an IPNC for f we have $|b| < 1 + f(|d|) - f(0) \leq f(|a|)$. Note that as no token is present on s_0, $f(0)$ tokens cannot be output by $\mathscr{F}$. Nevertheless, IC1 is applicable because of the monotonicity of the firing rule for Petri nets. Also note that $f(0) > 0$. It can be seen by induction that $f(n) \leq g^*(n)$. Hence, $|b| \leq g^*(|a|)$.

Case 2. $|c| = m > 0$. A firing sequence of $\mathscr{G}$ leading from s_1ar' to br_1 ($r' \in v_1RC_{r_0}$) has without loss of generality the form

$$\begin{aligned} s_1cdr' &\to^* p^m d_1 v_1 r_{0,1} \to^{t'} p^{m-1} s_0 d_1 v_1 r_{0,1} \\ &\to^* p^{m-1} d_2 v_1 r_{0,2} \to^{t'} p^{m-2} s_0 d_2 v_1 r_{0,2} \\ &\vdots \\ &\to^* p d_m v_1 r_{0,m} \to^{t'} s_0 d_m v_1 r_{0,m} \to^* br_1, \end{aligned}$$

with $d_i \in C(S_o^0 \cup \{i_0, o_0\})$, $r_{0,i} \in C(S_c^0)$ for $i = 1, \ldots, m$, or can trivially be simulated by such a sequence if c already contains tokens on s. Now set $a_i := p^{m+1-i} d_i$ for $i = 1, \ldots, m$, $a_{m+1} := b$. It suffices to show that

$$|a_i| \leq g^*(|d| + i - 1) + m + 1 - i \qquad \text{for} \quad i = 1, \ldots, m + 1. \tag{$*$}$$

For $i = 1$, this comes from property IC2 for $\mathscr{F}$, which allows the argument of case 1 to be applied, as no token from the places i_1, p, s_0 has been used so far. Assume that ($*$) is established for all i with $1 \leq i < i_0 \leq m + 1$. We have $r_{0,i_0-1} \in RC_{r_0}$ and, as r_0 is iteratively conservative, we may assume that $r_{0,i_0-1} = r_0$. From IC1 for $\mathscr{F}$ we then obtain $|d_{i_0}| \leq f(|d_{i_0-1}|)$, and hence

$$\begin{aligned} |a_{i_0}| &= |d_{i_0}| + m + 1 - i_0 \\ &\leq f(|a_{i_0-1}| + i_0 - m - 2) + m + 1 - i_0 \\ &\leq f(g^*(|d| + i_0 - 2)) + m + 1 - i_0 \qquad \text{by induction hypothesis} \\ &\leq g^*(|d| + i_0 - 1) + m + 1 - i_0. \end{aligned}$$

The last inequality comes from the fact that with an additional token on s_1 (or p), $\mathscr{F}$ can be applied once more to the tokens so far collected on o_0, and that the transport of tokens from o_0 to o_1 can be postponed, in any case, to the very last. Thus IC1 holds for $\mathscr{G}$.

(ii) We obviously have $g^*(0) = 1 = g(0)$ and $g^*(1) = f(1) = g(1)$, and inspection of the net $\mathscr{G}$ shows that $g^* \geq g$. Let $n > 1$ be minimal such that $s_1 i_1^n r_1 \to^{\mathbf{t}} br_1$ for some $b \in C(S^1)$ with $|b| > g(n)$ and some $\mathbf{t} \in (T^1)^+$. $\mathbf{t}$ is without loss of generality of the form

$$\begin{aligned} s_1 i_1^n r_1 &\to^* p^2 d_1 v_1 r_0' \to^{t'} p s_0 d_1 v_1 r_0' \\ &\to^{\mathbf{t}_1} p d_2 v_1 r_0'' \to^{\mathbf{t}_2} p d_3 v_1 r_0'' \to^{t'} s_0 d_3 v_1 r_0'' \to^* br_1, \end{aligned}$$

with $d_1, d_2, d_3 \in C(S_o^0 \cup \{i_0, o_0\})$; $r_0', r_0'' \in RC_{r_0}$; $\mathbf{t}_1 \in (T^0)^+$ such that the first transition of $\mathbf{t}_1$ removes s_0; and $\mathbf{t}_2 \in \{t\} \cdot (T^0 \cup \{t\})^* \cup \{\lambda\}$. We may also assume that d_1 does not contain o_0. As, reaching $pd_2v_1r_0''$, the last token on p actually was not used, and as n is minimal, we have $|d_2| \leq g(n - 1)$. If d_2 contains tokens on o_0, they were placed there by $\mathbf{t}_1$. Because of IC2 for $\mathscr{F}$, we actually may assume that $\mathbf{t}_2 \in \{t\}^*$, as

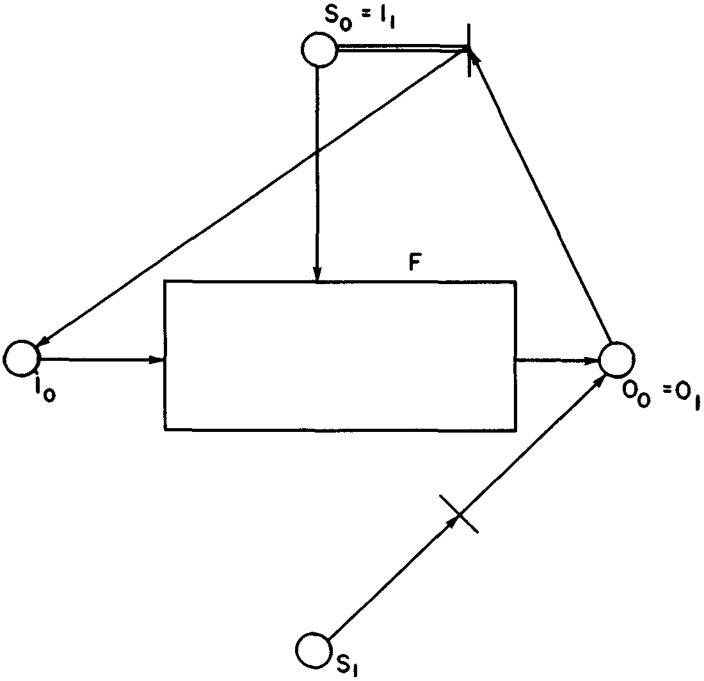


FIG. 3. A smaller Petri net computer for the iteration of f.

no tokens from i_0 could have been used, and therefore we have $|d_3| = |d_2|$; and because of IC1 for $\mathscr{F}$, we have $|b| \leq f(|d_3|) \leq f(g(n-1)) = g(n)$. If d_2 contains no tokens on o_0, we may analogously conclude that $t_2 = \lambda$ and thus $d_2 = d_3$. IC1 for $\mathscr{F}$ again yields $|b| \leq f(|d_3|) \leq g(n)$. This contradicts the choice of n. Thus, $g^* = g$. Q.E.D.

We want to remark that the construction of $\mathscr{G}$ is not optimal with respect to the number of additional places and transitions. Figure 3 shows a smaller solution (without proof) for a WPNC for g, and this construction can also be applied recursively, provided the net at the initial stage is an IPNC. We think, however, that the standard structure facilitates the proof of Lemma 2 and unifies the recursive application of the construction. The nets $\mathscr{A}_n$, $n \in \mathbf{N}$, of Theorem 2 are now easily obtained by starting with an IPNC for $f(m) = 2m + 1$ which can be constructed in a straightforward way using the standard structure (Figure 2) and applying the construction of Lemma 2 n times. At the last stage an additional place and a transition which initializes the marking $r_n := \prod_{i=1}^{n} v_i$ is inserted between s_n and the transition corresponding to t'' in Figure 2 so that a λ-WPNC $\mathscr{A}_n$ for A_n is obtained. Claims (ii) and (iii) of Theorem 2 are immediate from the construction.

A2 BOUNDED WPNCS FOR POLYNOMIALS. Let $p(\mathbf{x}_m) = \sum_{i=1}^{v} a_i \prod_{j=1}^{m} x_j^{e_{ij}}$ be a polynomial with positive integer coefficients a_i, and $e_{ij} \in \mathbf{N}$ for $i = 1, \ldots, v$, $j = 1, \ldots, m$. The net $\mathscr{T}$ of Figure 4a has the property that $n_1 n_2 = \max\{k; i^{n_1} j^{n_2} r \rightarrow^* o^k r a$ for some $a\}$ for all $(n_1, n_2) \in \mathbf{N}^2$. The verification of this fact is left to the reader. These basic multiplier nets have also been introduced in [8]. We connect d instances of $\mathscr{T}$ by identifying the output place of a copy of $\mathscr{T}$ with the input place j of the following copy. Renaming the j-place of the first net by c, this leaves d input places (i.e., the i-places of all the copies), which are renamed $i_1, \ldots, i_d$. This yields a

(a)

(b)

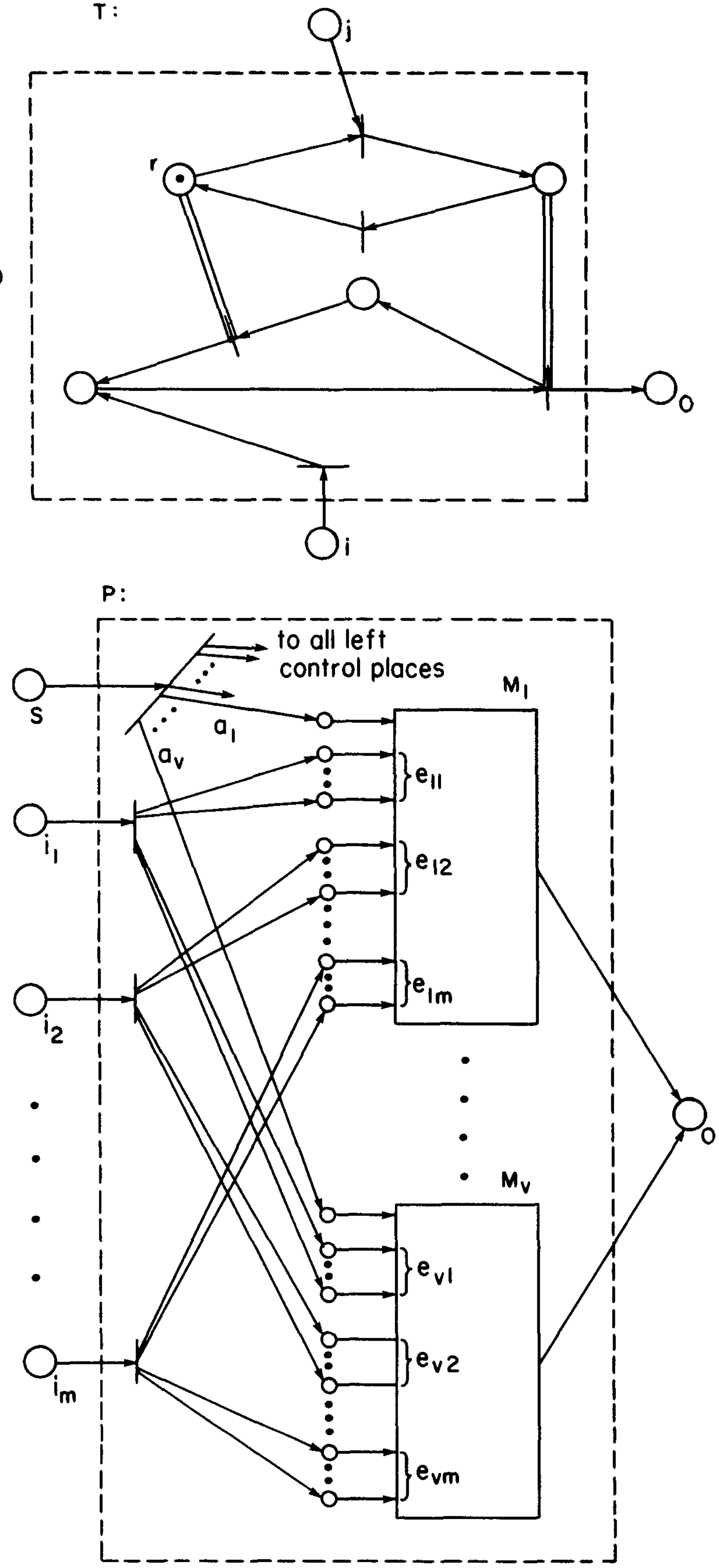


FIG 4. (a) Basic multiplier net (b) λ-WPNC for the polynomial $\sum_{i=1}^{v} a_i \prod_{j=1}^{m} x_j^{e_{ij}}$.

net for the homogeneous monomial $a\prod_{j=1}^{d} x_j$, namely, $a\prod_{j=1}^{d} n_j = \max\{k; c^a \mathbf{i}_d^{\mathbf{n}_d} r \to^* o^k ra$ for some $a\}$ for all $\mathbf{n}_d \in \mathbf{N}^d$ (r is the product of all the r's in the $\mathcal{T}$-copies). In order to get a WPNC $\mathcal{P}$ for the polynomial p, monomial nets $\mathcal{M}_1, \ldots, \mathcal{M}_v$ with suitable degrees are combined as in Figure 4b. The input places $\mathbf{i}_m$ may be connected to several input places of each subnet for a homogeneous monomial, so that in effect we get a subnet computing an arbitrary monomial of p. The "left control places" are those corresponding to the place r of Figure 4a. The construction of monomial nets for constant monomials is left to the reader. It is easy to see that $\mathcal{P}$ is a λ-WPNC for p and that each marking $s\mathbf{i}_m^{\mathbf{n}_m}$ with $\mathbf{n}_m \in \mathbf{N}^m$ permits only finite firing sequences. The reader may observe about the multiplier net $\mathcal{T}$ of Figure 4a that during a computation none of the "internal" places ever contains more than $\max\{1, n_1\}$ tokens. Thus, by firing in a monomial net a cycle in the $\mathcal{T}$-component closest to the output place which has a token on the input place corresponding to j, $\mathcal{P}$ can compute $p(\mathbf{n}_m)$ by a firing sequence whose marking sequence is bounded on all nondesignated places by $g(N) = N + \|p\|$ for all $\mathbf{n}_m \in \{0, 1, \ldots, N\}^m$. Let $\mathcal{P} = (S, T, pre, post)$ be the net constructed above, set $O := S - \{s, \mathbf{i}_m, o\}$, and let O^c be a copy of O (disjoint from S), with $u^c \in O^c$ corresponding to $u \in O$. Now define the Petri net $\mathcal{P}'_{bd} = (S', T', pre', post')$ as follows:

$$\begin{aligned} S' &:= S \cup O^c, \\ T' &:= T, \\ pre' &:= pre \cup \{(u^c, t);\, u \in O,\, (t, u) \in post\}, \\ post' &:= post \cup \{(t, u^c);\, u \in O,\, (u, t) \in pre\}, \end{aligned} \left.\vphantom{\begin{aligned} a \\ a \end{aligned}}\right\} \text{in the multiset sense.}$$

Further, let $r_N := \prod_{u \in O^c} u^{g(N)}$.

LEMMA 3. *$\mathcal{P}'_{bd}$ is a $g(N)$-bounded r_N-WPNC for p restricted to* $\{0, 1, \ldots, N\}^m$.

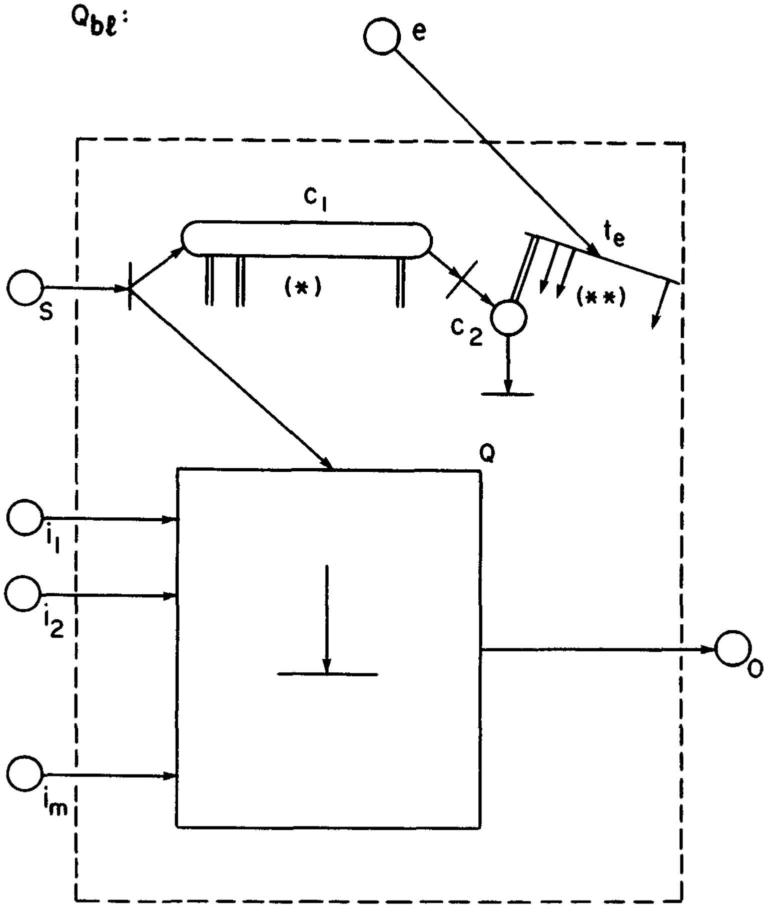
PROOF. The definition of pre' and $post'$ implies that for each $u \in O$, the sum of the tokens on u and u^c is constant (and equals $g(N)$) for each marking sequence of $\mathcal{P}'_{bd}$ starting from a marking in $r_N C(\{s, \mathbf{i}_m, o\})$. Further, each firing sequence of $\mathcal{P}$ starting at any $a \in C(\{s, \mathbf{i}_m, o\})$ whose marking sequence is $g(N)$-bounded on O is also firable in $\mathcal{P}'_{bd}$ starting at ar_N, and conversely. Thus the lemma follows from our previous observation that $p(\mathbf{n}_m)$ tokens on o can be obtained by firing sequences which are $g(N)$-bounded on the nondesignated places whenever $\mathbf{n}_m \in \{0, 1, \ldots, N\}^m$. Q.E.D.

The net $\mathcal{P}_{bd}$ of Theorem 3 is now easily obtained from $\mathcal{P}'_{bd}$ by adding to $\mathcal{P}'_{bd}$ a place b and a transition, with b as input place and all $u \in O^c$ as output places, which uses the tokens on b to initialize the marking r_N. Part (i) of Theorem 3 is then immediate, and (ii) follows from the observation that both the number of arcs of multiplicity one and the number of places in $\mathcal{P}_{bd}$ are bounded by the sum of the degrees of the monomials of p times a constant and that the code for multiple arcs in $\mathcal{P}_{bd}$ uses space proportional to the code for the coefficients of p.

A3 BLURRING WPNCs FOR POLYNOMIALS. In order to obtain a blurring WPNC $\mathcal{Q}_{bl}$ for a polynomial $q \in \mathbf{N}[\mathbf{x}_m]$ (see Definition 8), construct a λ-WPNC $\mathcal{Q}$ for q, as in A2, with designated places s, $\mathbf{i}_m$, o, and extend it as follows (see Figure 5).

(a) Attach an erasing transition to each nondesignated place u of $\mathcal{Q}$, that is, a transition with input place u and no output place (indicated in the diagram by a transition with an entering arc in the box for $\mathcal{Q}$).

(b) Add the places c_1, c_2, and e, and the transitions shown in the diagram. When the net is started with one token on s, this token enables $\mathcal{Q}$ to output tokens on o as



(*) controlling all transitions in Q (not the erasing transitions)
(**) to all nondesignated places of Q

FIG. 5. Blurring WPNC.

long as the one token received on c_1 from s remains there. When it is transported to c_2, the subnet $\mathcal{Q}$ is frozen. Now t_e may fire up to N times if there are initially N tokens on e, thus gathering at least N tokens on all nondesignated places of $\mathcal{Q}_{\mathcal{U}}$. Then, finally, the erasing transitions can generate any number of tokens between zero and N on each of the nondesignated places. Obviously, the erasing transitions do not affect the WPNC-property of $\mathcal{Q}$. By the construction of $\mathcal{Q}_{\mathcal{U}}$, if t_e ever is enabled, the output on o is frozen; so $\mathcal{Q}_{\mathcal{U}}$ is an e^N-WPNC for q for all $N \in \mathbf{N}$, and it generates any number of tokens up to at least N on the nondesignated places. Hence $\mathcal{Q}_{\mathcal{U}}$ is a blurring WPNC for q, and Theorem 4 (i) is established. The verification of parts (ii) and (iii) is left to the reader.

REFERENCES

1. ACKERMANN, W. Zum Hilbertschen Aufbau der reellen Zahlen. *Math. Ann.* 99 (1928), 118–133.
2 ADLEMAN, L , AND MANDERS, K. Computational complexity of decision procedures for polynomials. Conf. Rec 16th Ann. IEEE Symp on Foundations of Computer Science, Berkeley, Calif., 1975, pp. 169–177

3. BAKER, H.G. 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.
4. BIRYUKOV, A.P. Some algorithmic problems for finitely defined commutative semigroups. *Siberian Math. J. 8* (1967), 384–391.
5. CARDOZA, E., LIPTON, R.J., AND MEYER, A.R. Exponential space complete problems for Petri nets and commutative semigroups. 8th Ann. ACM Symp. on Theory of Computing, Hershey, Pa., May 1976, pp. 50–54
6. DAVIS, M. Hilbert's Tenth Problem is unsolvable. *Am. Math Monthly 80* (1973), 233–269.
7. ENGELER, E. *Introduction to the Theory of Computation.* Academic Press, New York, London, 1973.
8. HACK, M. Decision problems for Petri nets and vector addition systems Tech Memo. 59, Project MAC, M.I.T , Cambridge, Mass., 1975
9. HACK, M. Decidability questions for Petri nets. Ph D. Dissertation, Tech. Rep. 161, Laboratory for Computer Science, M.I.T., Cambridge, Mass., June 1976.
10. HILBERT, D. Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Kongress zu Paris 1900 *Nach. K. Ges. Wiss. Göttingen, Math.-Phys. Kl* (1900), 253–297 [English translation: *Bull Am. Math Soc. 8* (1901–1902), 437–479].
11. JAFFE, J. Semilinear sets and applications. Master's Thesis, Tech. Rep. 182, Laboratory for Computer Science, M.I.T., Cambridge, Mass , June 1977
12 JONES, N.D , LANDWEBER, L.H., AND LIEN, Y.E. Complexity of some problems in Petri nets *Theor. Comput Sci. 4* (June 1977), 277–299
13. KARP, R , AND MILLER, R. Parallel program schemata. *J. Comput Syst Sci. 3* (1969), 147–195.
14. KELLER, R.M Vector replacement systems: A formalism for modelling asynchronous systems Computer Science Laboratory Tech. Rep. 117, Princeton Univ., Princeton, N.J., December 1972.
15. LANDWEBER, L.H., AND ROBERTSON, E L. Properties of conflict-free and persistent Petri nets. Computer Science Dep. Tech Rep. 264, Univ of Wisconsin, Madison, Wis., 1975.
16. LIPTON, R.J. The reachability problem is exponential-space-hard. Yale Computer Science Rep. 62, Yale Univ , New Haven, Conn., Jan 1976.
17. MATIJASEVIC, J.V. Enumerable sets are Diophantine. *Sov. Math. Dokl. 11* (1970), 354–357.
18 MAYR, E.W. The complexity of the finite containment problem for Petri nets. Master's Thesis, Tech. Rep. 181, Laboratory for Computer Science, M.I T., Cambridge, Mass., May 1977
19. RACKOFF, C The covering and boundedness problems for vector addition systems. *Theor Comput. Sci. 6* (1978), 223–231.
20 TAICLIN, M A. Algorithmic problems for commutative semigroups. *Sov Math. Dokl 9* (1968), 201–204.

RECEIVED JUNE 1977; REVISED MAY 1980; ACCEPTED JUNE 1980