MIT/LCS/TR-181

THE COMPLEXITY OF THE FINITE CONTAINMENT PROBLEM FOR PETRI NETS

Ernst Wilhelm Mayr

June 1977

This report was prepared with the support of the National Science Foundation research grant MCS74-12997-A04 and DAAD (German Academic Exchange Service) grant no. 430-402-559-7.

Massachusetts Institute of Technology

Laboratory for Computer Science

(formerly Project MAC)

Cambridge Massachusetts 02139

THE COMPLEXITY OF THE FINITE CONTAINMENT PROBLEM FOR PETRI NETS

by

Ernst Wilhelm Mayr

Submitted to the Department of Electrical Engineering and Computer Science on May 12, 1977 in partial fulfillment of the requirements for the Degree of Master of Science.

ABSTRACT

If the reachability set of a Petri net (or, equivalently, vector addition system) is finite it can be effectively constructed. Furthermore, the finiteness is decidable. Thus, the containment and equality problem for finite reachability sets become solvable. We investigate the complexity of decision procedures for these problems and show 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 problem are thus the first uncontrived, decidable problems with provably non-primitive recursive complexity.

THESIS SUPERVISOR: Albert R. Meyer

TITLE: Professor of Computer Science and Engineering

Acknowledgements

I wish to thank my thesis supervisor, Professor Albert R. Meyer, for his helpful suggestions during the preparation of this thesis.

This research was supported by DAAD (German Academic Exchange Service) grant No. 430-402-559-7, and by the National Science Foundation research grant MCS74-12997-A04.

Table of Contents

I. Introduction

The containment problem for Petri nets is the problem to determine of any two given Petri nets whether one reachability set is contained in the other. By reducing Hilbert's Tenth Problem concerning integer solutions of diophantine equations, which is known to be undecidable [13], to the containment problem, Rabin has shown the unsolvability of the latter (see [3]). The situation changes, however, when one considers subclasses of the general problem. A result by Karp and Miller [11] yields the decidability of the problem whether the reachability set of a given Petri net is finite. It also gives an algorithm to enumerate finite reachability sets. Hence, the finite containment problem (FCP), i.e. the problem to determine of any two given Petri nets whether their reachability sets are each finite and one is contained in the other, is decidable by exhaustion. This thesis deals with the complexity of decision procedures for FCP. We show that all those procedures are necessarily enormously complex, specifically, that they are non-primitive recursive.

The intrinsic complexity of the decision procedures is not due to the fact that the reachability sets have to be tested for finiteness. Even if the decision procedure is supplied with the answer to this subproblem the remaining complexity still is non-primitive recursive.

To establish this result we will present in section II a bounded version of Hilbert's Tenth Problem whose complexity we know is non-primitive recursive. To reduce it effectively to FCP two versions of Petri net computers are introduced in section III, weak computers for polynomials and a more restricted class of 'iterative' computers for functions defined by primitive recursion. Section IV contains the recursive construction of such computers for a sequence of functions closely related to Ackermann's function [1]. Section V and VI then discuss a property of the polynomial computers introduced before 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. Two modifications of the polynomial computers exploit this property and serve to reduce the bounded version of Hilbert's Tenth Problem effectively to FCP. In section VII the reduction is carried out and the main results of this thesis are proven.

II. Basic definitions and properties

In this section we shall give precise definitions of Petri nets and related concepts like marking of a Petri net, firability, firing sequence, and reachability set. We are then going to formally state the problems whose complexity we want to examine, and we shall also give the definition of that bounded version of Hilbert's Tenth Problem which will be reduced to FCP.

We assume that the reader is familiar with the notions like the free monoid Σ^* over a finite alphabet Σ, the set Σ^+ of all non-empty words over Σ (the empty word will be denoted by λ, the length of a word $\alpha \in \Sigma^*$ by $|\alpha|$), the concept of the free commutative monoid generated by Σ which we will write $C(\Sigma)$, and basic concepts of algebra like the semiring $\mathbb{N}[x_1, \ldots, x_m]$ of polynomials with nonnegative integer coefficients in the unknowns $x_1, \ldots, x_m$.

<u>Definition 1:</u>

a) A <u>Petri</u> <u>net</u> $\mathcal{P}$ is a 4-tupel (S, T, <u>in</u>, <u>out</u>) with the properties

i) S is a finite ordered set;

ii) T is a finite set, $S \cap T = \emptyset$;

iii) <u>in</u> is a multiset over $S \times T$;

iv) <u>out</u> is a multiset over $T \times S$.

b) A <u>marking</u> of $\mathcal{P}$ is a mapping

$$\alpha : S \longrightarrow \mathbb{N} \quad (\mathbb{N} = \text{set of nonnegative integers})$$

The elements of S are called the <u>places</u> of $\mathcal{P}$, the elements of T are called <u>transitions</u>. In diagrams, places are drawn as small circles, transitions as bars, and elements of <u>in</u> or <u>out</u> are denoted by directed arrows. If the multiplicity of elements in <u>in</u> or <u>out</u> is greater than 1 this is indicated in the diagram by the corresponding number attached to the arrow.
If $(s,t) \in$ <u>in</u>, s is called an <u>input-place</u> of t, and if $(t,s) \in$ <u>out</u>, an <u>output-place</u> of t. A transition t is said to be <u>controlled</u> by a place s if s is both an input and output-place of t, connected by an arc in each direction of multiplicity one. In order to simplify the pictures this will be represented by a double line connecting s and t.

Let $s_1,\ldots,s_m$ be the elements of S. Sometimes it will be convenient to write a marking α of $\mathcal{P}$ as

$$\alpha = \prod_{i=1}^{m} s_i^{\alpha(s_i)}$$

and consider it as an element of the free commutative monoid C(S) generated by S.
A Petri net $\mathcal{P}$ together with a marking α of $\mathcal{P}$ will be denoted by the pair $(\mathcal{P},\alpha)$.

Definition 2:

Let $\mathcal{P} = (S, T, \underline{in}, \underline{out})$ be a Petri net, and let $\nu_{\underline{in}}(s,t)$ denote the multiplicity of $(s,t) \in S \times T$ in $\underline{in}$, $\nu_{\underline{out}}(t,s)$ that of $(t,s) \in T \times S$ in $\underline{out}$.

a) A transition $t \in T$ is *firable at a marking α* of $\mathcal{P}$ and takes α to the marking β (written $\alpha \xrightarrow{t} \beta$) iff

   i) $(\forall s \in S)[\alpha(s) \geq \nu_{\underline{in}}(s,t)]$, and

   ii) $(\forall s \in S)[\beta(s) = \alpha(s) - \nu_{\underline{in}}(s,t) + \nu_{\underline{out}}(t,s)]$.

b) A *firing sequence* τ is an element $\tau \in T^+$.

c) A firing sequence $\tau \in T^+$ is *firable at a marking α* of $\mathcal{P}$ and takes α to the marking β (written $\alpha \xrightarrow{\tau} \beta$) iff

   $(\exists r \geq 1\ \exists t_1, \ldots, t_r \in T)[\tau = t_1 t_2 \ldots t_r$ $\underline{\text{and}}$

   $$(\exists \beta_0, \beta_1, \ldots, \beta_r)[\alpha = \beta_0 \wedge \beta = \beta_r \wedge (\forall 1 \leq i \leq r)[\beta_{i-1} \xrightarrow{t_i} \beta_i]]].$$

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

d) A marking β of $\mathcal{P}$ is said to be *reachable* from a marking α of $\mathcal{P}$ (written $\alpha \xrightarrow{*} \beta$) if $\alpha = \beta$ $\underline{\text{or}}$ $(\exists \tau \in T^+)[\alpha \xrightarrow{\tau} \beta]$.

Of course, the relations $\xrightarrow{*}$ as well as $\xrightarrow{\tau}$ and $\xrightarrow{t}$ depend on the Petri net $\mathcal{P}$. It will however, always be clear from the context which Petri net is being considered.

Definition 3:

The <u>reachability set</u> of a Petri net $\mathcal{P}$ with initial marking α is the set of all markings reachable from α:

$$R(\mathcal{P},\alpha) := \{\beta;\alpha \xrightarrow{*} \beta\}.$$

If we are given two Petri nets $\mathcal{P}$ and $\mathcal{P}'$ with initial markings α and α', resp., we may ask questions about relationships between the two reachability sets, e.g. whether they are equal or one is contained in the other. To be able to do this we have, of course, to set up a 1-1 correspondence between the places of the first and the second net.

Definition 4:

Let $\mathcal{P}_i = (S_i, T_i, \underline{in}_i, \underline{out}_i)$ be a Petri net with initial marking α_i $(i=1, 2)$, $|S_1| = |S_2|$, and let $\bar{h}$: $C(S_1) \longrightarrow C(S_2)$ be the semigroup-isomorphism generated by the order-preserving bijection h: $S_1 \longrightarrow S_2$.

a) $R(\mathcal{P}_1,\alpha_1)$ is <u>contained</u> in $R(\mathcal{P}_2,\alpha_2)$ (written $R(\mathcal{P}_1,\alpha_1) \subseteq_h R(\mathcal{P}_2,\alpha_2)$) if

$$\bar{h}(R(\mathcal{P}_1,\alpha_1)) \subseteq R(\mathcal{P}_2,\alpha_2).$$

b) $R(\mathcal{P}_1,\alpha_1) =_h R(\mathcal{P}_2,\alpha_2) \Leftrightarrow_{def} \bar{h}(R(\mathcal{P}_1,\alpha_1)) \subseteq R(\mathcal{P}_2,\alpha_2)$ <u>and</u>

$$R(\mathcal{P}_1,\alpha_1) \subseteq \bar{h}^{-1}(R(\mathcal{P}_2,\alpha_2)).$$

To simplify the notation, we will omit the subscript for the 1-1 correspondence h between the places of the two nets.

Definition 5:

a) The **containment problem** CP is the problem to decide for two Petri nets $\mathcal{P}_1$ and $\mathcal{P}_2$ with markings α_1 and α_2, resp., whether the reachability set of the first net is contained in that of the second:

$$CP := \{\langle(\mathcal{P}_1,\alpha_1),(\mathcal{P}_2,\alpha_2)\rangle;\ R(\mathcal{P}_1,\alpha_1) \subseteq R(\mathcal{P}_2,\alpha_2)\}.$$

b) The **finite containment problem** FCP is

$$FCP := \{\langle(\mathcal{P}_1,\alpha_1),(\mathcal{P}_2,\alpha_2)\rangle;\ |R(\mathcal{P}_2,\alpha_2)|<\infty \ \underline{\text{and}}\ \langle(\mathcal{P}_1,\alpha_1),(\mathcal{P}_2,\alpha_2)\rangle \in CP\}.$$

c) The **finite equality problem** FEP is

$$FEP := \{\langle(\mathcal{P}_1,\alpha_1),(\mathcal{P}_2,\alpha_2)\rangle;\ |R(\mathcal{P}_2,\alpha_2)|<\infty \ \underline{\text{and}}\ R(\mathcal{P}_1,\alpha_1) = R(\mathcal{P}_2,\alpha_2)\}.$$

The proof that FCP and FEP are non-primitive 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. Though the main results of this thesis 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.

Thus, we want to encode Petri nets by first writing down the number of places, and then for each transition a pair of sequences of places (designated by their number in the ordered set of places) which contains in the first component all input

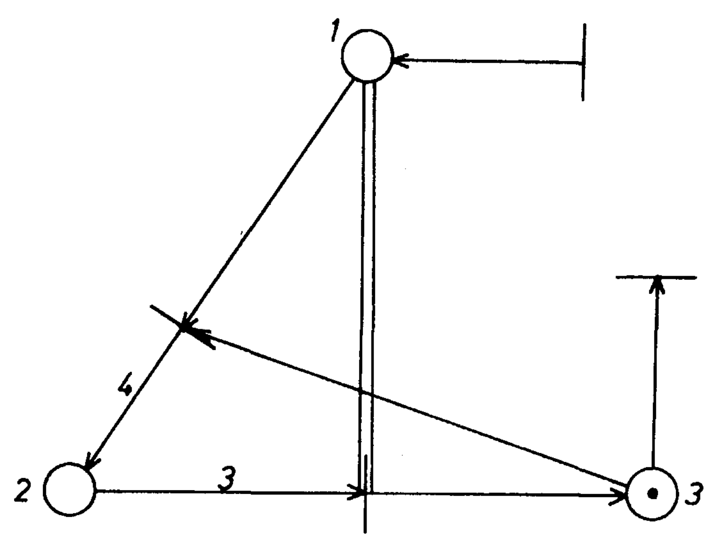
and in the second component all output-places of this transition, enclosed in brackets and preceded by the multiplicity of the connecting arc if the latter is greater than 1. Transitions which are not connected to any place are disregarded.
Markings will be encoded by writing down their values on the places in order. We assume that all numbers are written in binary. It is easy to see how a code over the alphabet $\{0, 1\}$ alone could be obtained by encoding the symbols of our code by short words over $\{0, 1\}$.

Example: The encoding for the Petri net of figure 1 together with the marking (0, 0, 1) indicated by dots in the places may, therefore, look like (numbers written in decimal):

3,()(1),(1,3)(4(2)),(1,3(2))(1,3),(3)(),0,0,1

(The code for the Petri net is followed immediately by the code for the marking).

Figure 1:



Considering, in general, the length of this encoding, the following definition is motivated.

<u>Definition 6:</u>

Let $\mathcal{P}=(S, T, \underline{in}, \underline{out})$ be a Petri net, α a marking of $\mathcal{P}$. Then

size($\underline{in}$) $:= \sum_e \lceil \log(\nu_{\underline{in}}(e)+1)\rceil$, where the sum is taken over the different arcs in the multiset $\underline{in}$. Similarly size($\underline{out}$).

size($\mathcal{P}$) $:=$ (size($\underline{in}$) + size($\underline{out}$) + 1)$\cdot\log(|S|+1)$;

size(α) $:= |S|\cdot\max\{\log(1+\alpha(i));\ i=1,\dots,|S|\}$;

size($\mathcal{P},\alpha$) $:=$ size($\mathcal{P}$) + size(α).

The length of the encoding discussed above is bounded by a constant times size($\mathcal{P},\alpha$), as easily can be seen.

Likewise, we are going to describe an encoding for multivariable polynomials with nonnegative integer coefficients. The code for such a polynomial will be a sequence of codes for its monomial constituents, separated by special delimiters. We may assume that the variables of the polynomial are $x_1,\dots,x_m$ for some $m\in\mathbb{N}$. Then the code for a monomial is the sequence of integers obtained by writing down first the nonzero integer coefficient of the monomial, then the nondecreasing sequence of integers from $\{1,\dots,m\}$ in which each $j\in\{1,\dots,m\}$ occurs just as often as the degree of x_j in the monomial indicates. Again, delimiters are used to separate the numbers. If, for example,

| denotes the delimiter for separating monomials and $\cdot$ that for separating numbers within monomials then the code for the polynomial

$$4x_1x_3 + x_1^2x_2 + 3 \in \mathbb{N}[x_1,x_2,x_3]$$

looks like (numbers written in decimal)

$$|4\cdot1\cdot3|1\cdot1\cdot1\cdot2|3| \; .$$

By writing the numbers in binary and then encoding each of the four symbols $|,\cdot,0,1$ by a combination of two symbols from $\{0, 1\}$, a binary code for multivariable polynomials with nonnegative integer coefficients is obtained. Let

$$pc : \bigcup_{m \geq 0} \mathbb{N}[x_1,\ldots,x_m] \longrightarrow \{0, 1\}^*$$

denote this encoding.

<u>Definition 7:</u>

Let $p \in \mathbb{N}[x_1,\ldots,x_m]$ for some $m \in \mathbb{N}$. Then

$$\text{size}(p) := |pc(p)| .$$

Hilbert's Tenth Problem is the problem to decide whether a multivariable polynomial $p \in \mathbb{Z}[x_1,\ldots,x_m]$ has a zero $(a_1,\ldots,a_m) \in \mathbb{Z}^m$ ($\mathbb{Z}$ is the set of integers). It is not difficult to see (and we won't prove it here) that this problem is equivalent to asking whether a polynomial has a nonnegative integer solution. Matijasevic [13] has shown that for each recursively enumerable set $M \subseteq \mathbb{N}$ there exists a polynomial $p \in \mathbb{Z}[x_1,\ldots,x_m]$ for some $m \in \mathbb{N}$, such that

$$a \in M \Leftrightarrow (\exists b_2,\dots,b_m \in \mathbb{N})[p(a,b_2,\dots,b_m) = 0].$$

As there are r.e. sets in $\mathbb{N}$ which are not recursive Hilbert's Tenth Problem is undecidable.

On the other hand, if we fix $N \in \mathbb{N}$ and restrict ourselves to asking whether any given polynomial $p \in \mathbb{Z}[x_1,\dots,x_m]$ has a zero $(a_1,\dots,a_m) \in \{0, 1,\dots, N\}^m$, this problem becomes decidable by exhaustion. More generally, we might make N a function of $n \in \mathbb{N}$ and investigate the complexity of finding zeroes of polynomials as above, bounded by $N(n)$.

Adleman and Manders [2] have proven results which imply

<u>Lemma 1:</u>

Let $g: \mathbb{N} \longrightarrow \mathbb{N}$ be a (monotone) recursive function which majorizes the primitive recursive functions. Then the problem to decide whether a $p \in \mathbb{Z}[x_1,\dots,x_m]$ (for some $m \in \mathbb{N}$) has a zero $(a_1,\dots,a_m) \in \{0, 1,\dots,g(\mathrm{size}(p))\}^m$ requires, for any primitive recursive function pr, more than pr(size(p)) steps on a multitape Turing machine, for infinitely many p.

<u>Remark:</u> Let $\mathcal{F}$ be a class of functions from $\mathbb{N}$ to $\mathbb{N}$. A function $g : \mathbb{N} \longrightarrow \mathbb{N}$ is said to majorize $\mathcal{F}$ iff

$$(\forall f \in \mathcal{F}\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0)[\,g(n) > f(n)\,].$$

<u>Proof of the lemma:</u> See theorem 5 of [2] and note that if a function h(n) doesn't majorize the primitive recursive func-

tions then the function $2^{2^{2^{h(n^2)^5}}}$ doesn't, either.

We are now going to define a special fast growing function which satisfies the conditions of the lemma, but which nonetheless can be computed by reasonably small Petri nets, in a sense which will be made precise in the next section.

<u>Definition 8:</u>

Let $A : \mathbb{N} \longrightarrow \mathbb{N}$ be the function defined by

$$\begin{aligned} A_o(x) &:= 2x + 1, \\ A_{n+1}(0) &:= 1, \\ A_{n+1}(x+1) &:= A_n(A_{n+1}(x)), \\ A(n) &:= A_n(2) . \qquad\qquad (n, x \in \mathbb{N}). \end{aligned}$$

A function similar to A is studied in [6]. A result about that function which immediately applies to A is

<u>Lemma 2:</u>

A majorizes the primitive recursive functions.

<u>Proof:</u> See [6], pp. 144-149.

We are now able to define a problem which is related to the bounded version of Hilbert's Tenth Problem described in lemma 1.

Definition 9:

The Bounded Polynomial Inequality Problem BPI is

$$\text{BPI} := \{(p,q,n);\ p,\ q \in \mathbb{N}[x_1,\dots,x_m] \text{ for some } m \in \mathbb{N}, \text{ and } (\forall (y_1,\dots,y_m) \in \{0,1,\dots,A(n)\}^m)[p(y_1,\dots,y_m) \le q(y_1,\dots,y_m)]\} .$$

From lemmas 1 and 2 we obtain the result that BPI is extremely complex, for any reasonable complexity measure for the triples (p,q,n). Nonetheless, for definiteness, we set

Definition 10:

$$\text{size}(p,q,n) := \text{size}(p) + \text{size}(q) + n.$$

Theorem 1:

BPI is non-primitive recursive.

Proof: It suffices to prove

Lemma 3:

The bounded version of Hilbert's Tenth Problem of lemma 1 with A as upper bound is polynomial-time-reducible to BPI.

Proof of the lemma: Let's assume we are given $n \in \mathbb{N}$ and a polynomial $p \in \mathbb{Z}[x_1,\dots,x_m]$ for some $m \in \mathbb{N}$. Then p has a zero $(a_1,\dots,a_m) \in \{0,1,\dots,A(n)\}^m$ iff $p^2(x_1,\dots,x_m) - 1 \ge 0$ does not hold for all $(x_1,\dots,x_m) \in \{0,1,\dots,A(n)\}^m$. The latter, however, is equivalent to $(q^-,q^+,n) \notin$ BPI, where q^+ (resp. $-q^-$) is the sum of the monomials of $p^2(x_1,\dots,x_m)$

$- 1$ with positive (negative) coefficients, i.e.

$p^2(x_1,\ldots,x_m) - 1 = q^+(x_1,\ldots,x_m) - q^-(x_1,\ldots,x_m)$ and

$q^+, q^- \in \mathbb{N}[x_1,\ldots,x_m]$. Obviously, (q^-,q^+,n) can be obtained

in polynomial time from (p,n). q.e.d.

If we had a primitive recursive decision procedure for BPI we could by means of the polynomial reduction in the proof of the lemma construct a primitive recursive decision procedure for the A-bounded version of Hilbert's Tenth Problem, in contradiction to lemma 1 (The relevant properties of polynomial-time-reducibility used here are discussed in [15]). This proves theorem 1.

q.e.d.

III. Two concepts for Petri net computers

Several ways have been studied, e.g. in [7, 8], to use Petri nets to compute number theoretic functions. A straightforward approach is to designate some places to contain numbers of tokens representing the arguments of the function and obtain the function value by counting how often a transition can fire or by the length of the longest firing sequence possible at the given initial marking. As firable transitions may fire or not, we can in general not expect that each firing sequence represents the computation of a function value. It turns out, however, that the following concept which Rabin called "weak computation" works for quite a large class of functions.

Definition 11:

Let $\mathcal{P}=(S, T, \underline{in}, \underline{out})$ be a Petri net, and let $s, i_1, \ldots, i_m, o \in S$ be m+2 designated places (also called the set S_{ip} of <u>i</u>nterconnecting <u>p</u>laces) such that $s, i_1, \ldots, i_m$ are not output-places and o is not an input-place of any transition in T. Let $\varrho \in C(S \setminus \{s, i_1, \ldots, i_m, o\})$, $D \subseteq \mathbb{N}^m$, and $f : D \longrightarrow \mathbb{N} \cup \{\infty\}$.

$\mathcal{P}$ is a <u>ϱ-weak-Petri-net-computer</u> (ϱ-WPNC) for f iff

$(\forall (n_1, \ldots, n_m) \in D \, \forall k \in \mathbb{N})[0 \leq k \leq f(n_1, \ldots, n_m) \Leftrightarrow$

$$s \prod_{j=1}^{m} i_j^{\,n_j} \varrho \xrightarrow{*} o^k \alpha \text{ for some } \alpha \in C(S \setminus \{i_1, \ldots, i_m, o\}) \quad .$$

($f(n_1, \ldots, n_m) = \infty$ is interpreted as $(\forall k \in \mathbb{N})[f(n_1, \ldots, n_m) \geq k]$).

If we do not want to emphasize the marking ρ, we also call $\mathcal{P}$ simply a WPNC.

WPNC's essentially as defined above have been investigated in [7] and [8]. It is easy to see that the functions computed by WPNC's are closed under addition and composition, and multiplication, as a WPNC for the product $f(n_1,n_2)= n_1 n_2$ can be constructed (see [8], and section V of this thesis).

We want to construct WPNC's for the functions A_n in definition 8. The structure of this definition suggests doing this recursively, i.e. obtain a WPNC for A_{n+1} from one for A_n. In such a WPNC for $A_{n+1}(m)$, the embedded WPNC for A_n would be restarted m times, since by definition $A_{n+1}(m) = A_n^{(m)}(1)$ (= the m-th iteration of A_n). In general, after a computation of a WPNC some tokens may be left on non-designated places. Those remaining tokens can affect the subsequent computations if the WPNC is restarted, so we have to refine the concept of a WPNC as stated in definition 11. In order to be able to start a WPNC iteratively, we basically make sure that the successive computations are properly separated and that in a computation which produces the maximal number of tokens on the output-place no 'garbage'-tokens are left on the non-designated places. This does not mean that there are no tokens at all left on the non-designated places. Rather, the WPNC under consideration usually is a ϱ-WPNC for some $\varrho \neq \lambda$, and we want to ensure that after each computation ϱ is conserved or can easily be restored. We, therefore, introduce the following concept of a conservative marking.

Definition 12:

Let $\mathcal{P} = (S, T, \underline{in}, \underline{out})$ be a WPNC with designated places S_{ip}, and for any $S' \subseteq S$ let the projection $j(S') : C(S) \longrightarrow C(S')$ be the homomorphism defined by

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

$\varrho \in C(S)$ is <u>conservative</u> iff there is a set of "<u>c</u>ontrol <u>p</u>laces", $S_{cp} \subseteq S \setminus S_{ip}$, such that

i) $\varrho \in C(S_{cp})$;

ii) $(\forall \alpha, \beta \in C(S))[(j(S_{cp})(\alpha) = \varrho) \wedge (\alpha \xrightarrow{*} \beta) \Rightarrow |j(S_{cp})(\beta)| = |\varrho|]$;

iii) $(\forall \alpha, \beta \in C(S))[(j(S_{cp})(\alpha) = \varrho) \wedge (\alpha \xrightarrow{*} \beta) \Rightarrow$

$$(\exists \tau \in T^*)[j(S_{cp})(\beta) \xrightarrow{\tau} \varrho]] .$$

For a given ϱ the set S_{cp} of control places needn't be uniquely determined. Condition ii) states that the sum of the tokens on the control places is constant for any firing sequence starting at a marking which agrees with ϱ on the control places. It could be replaced by the slightly stronger condition

ii') $(\forall \alpha, \beta \in C(S))[(\alpha \xrightarrow{*} \beta) \Rightarrow (|j(S_{cp})(\alpha)| = |j(S_{cp})(\beta)|)]$

without restricting the class of WPNC's we have in mind. Condition iii) finally says that from any marking ϱ' on the control places which had been produced starting from a marking extending ϱ, ϱ can be restored by using only transitions in the subnet defined by the set of control places. This follows at once from the observation that each firing of a transition in T which produces tokens on control places, also has to consume the same number of tokens on control places because of ii).

We can now proceed to refine the concept of WPNC to what we call "iterative-Petri-net-computer". We will first state the technical definition and explain it afterwards.

<u>Definition 13:</u>

Let $\mathcal{P}=(S, T, \underline{in}, \underline{out})$ be a Petri net, $f : \mathbb{N} \longrightarrow \mathbb{N}$ a number theoretic function. $\mathcal{P}$ is an <u>iterative-Petri-net-computer</u> (IPNC) for f iff

i) there is a set $S_{ip}=\{s,i,o\} \subseteq S$ of interconnecting places and a conservative $\varrho \in C(S_{cp})$ for some set of control places $S_{cp} \subseteq S \setminus S_{ip}$ such that $\mathcal{P}$ is a ϱ-WPNC for f, and

ii) let $S_{op} := S \setminus (S_{ip} \cup S_{cp})$ be the so-called <u>operational places</u> and define

$$RC_{\varrho} := \{ \varrho' \in C(S_{cp}); \ (\exists \alpha, \beta \in C(S))[j(S_{cp}(\alpha) = \varrho \wedge j(S_{cp})(\beta) = \varrho' \wedge \alpha \xrightarrow{*} \beta]\}.$$

Then

IC1: $(\forall \alpha, \beta \in C(S_{op} \cup \{i, o\}), \forall \varrho' \in RC_\varrho)[(s\alpha\varrho' \xrightarrow{*} \beta\varrho) \Rightarrow |\beta| \leq f(|\alpha|)]$.

IC2: $(\forall n \in \mathbb{N})[si^n\varrho \xrightarrow{*} o\alpha\varrho$ for some $\alpha \in C(S) \Rightarrow \alpha$ doesn't contain s$]$ <u>and</u>

$$(\forall \alpha, \beta, \gamma \in C(S_{op}), \forall \varrho', \varrho'' \in RC_\varrho, \forall l_1, l_2, l_3, k, k', n, n' \in \mathbb{N})$$
$$[(s^{l_1} i^n o^k \alpha\varrho' \xrightarrow{*} s^{l_2} i^n o^{k'} \beta\varrho'' \xrightarrow{*} s^{l_3} i^{n'} o^{k'} \gamma\varrho) \wedge (k' > k)$$
$$\wedge (n' < n) \Rightarrow (l_3 \leq l_1 - 1)].$$

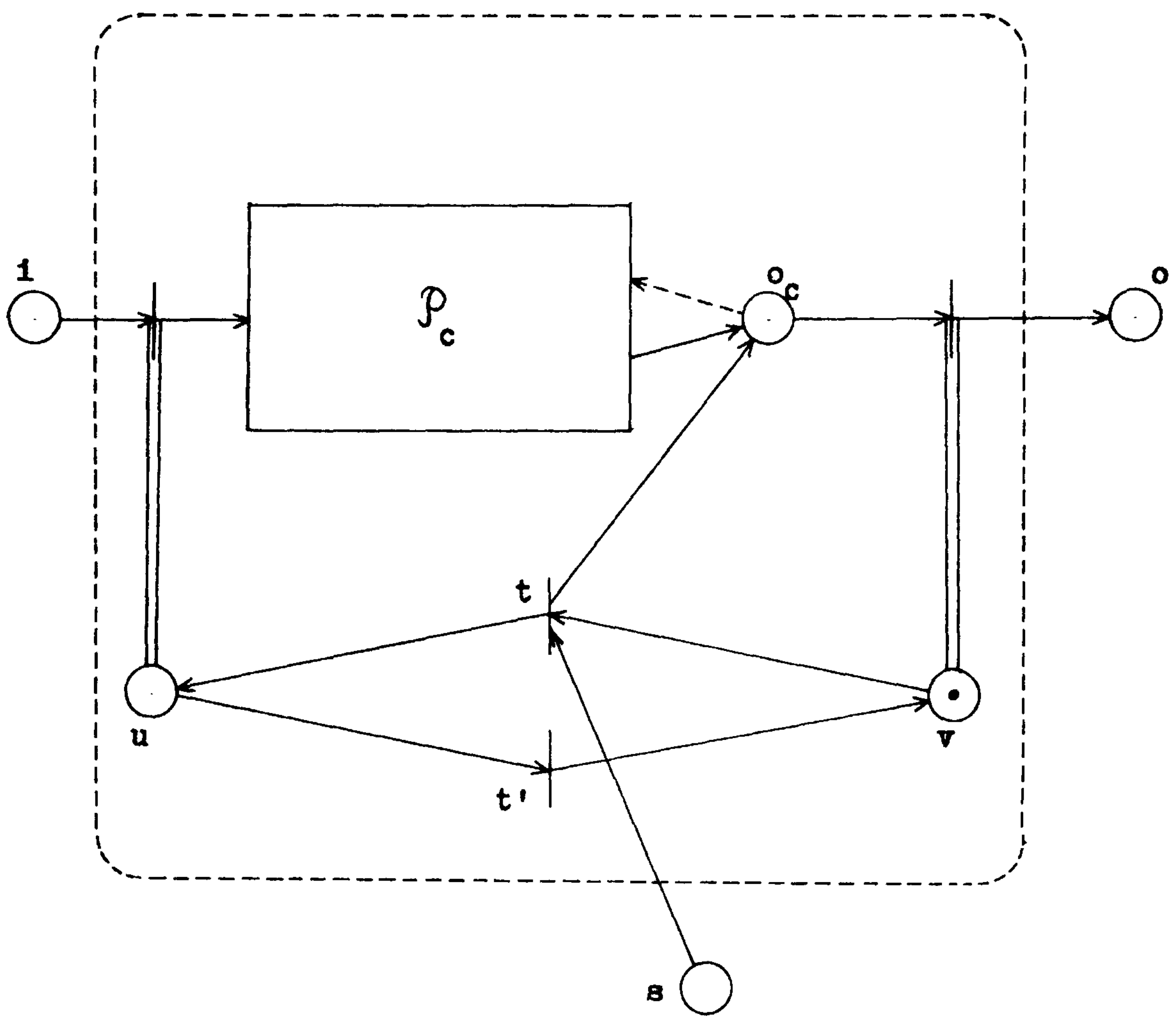
Because of IC1, ϱ is called an <u>iteratively conservative</u> initial marking of $\mathcal{P}$.

Informally speaking, IC1 ensures that no more tokens than necessary for the output are produced during a computation of an IPNC, 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, i.e. 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 ϱ is iteratively conservative ensures not only that the initial marking ϱ of the control places can be restored, but also that there is no gain in not restoring it.

The IPNC's constructed in this and the next section will have the standard structure of figure 2. The places u and v are used to establish IC2. Choosing $\varrho = v$ and $S_{cp} = \{u, v\}$ it can be seen

Figure 2:

$\mathcal{P}$:



that ρ is conservative: The token on v can only be transported to u and back to v, and no additional tokens are added by any transition to the places in S_{cp}. Further, if the token on v is transported to u it can be restored on v by firing t', with a marking of zero tokens on all the other places of the net. Also, IC2 holds if we assume that the subnet $\mathcal{P}_c$ in figure 2, which is called the <u>core</u> of $\mathcal{P}$, cannot produce tokens on o_c without using 'input-tokens' from i. Under this condition, t has to be fired first, thus consuming the token on s. In a phase in which tokens are produced on the 'output-place' o, a token has to be present on v. If such a phase is to be followed by transitions consuming tokens from i, u first must receive the token from v by a firing of t, which uses a token from s. Thus, the two conditions of IC2 are satisfied.

The place o_c in figure 2 may or may not have an arrow pointing into $\mathcal{P}_c$.

We want to remark that functions $f : \mathbb{N} \longrightarrow \mathbb{N}$ for which an IPNC $\mathcal{P} = (S, T, \underline{in}, \underline{out})$ with designated places s,i,o exists are strictly increasing, i.e. $(\forall n \in \mathbb{N})[f(n+1) > f(n)]$. Otherwise assume $n_o \in \mathbb{N}$ is minimal with the property that $f(n_o+1) \leq f(n_o)$. But then

$$si^{n_o+1}\rho \xrightarrow{*} io^{f(n_o)}\rho\alpha$$

for some $\alpha \in C(S)$ and an iteratively conservative submarking ρ, as $\mathcal{P}$ is a ϱ-WPNC for f and, therefore, can produce $f(n_o)$ tokens on o by using up n_o tokens on i (note that i is not an output-place for any transition in T). Applying IC1, we obtain

$$(*) \qquad |io^{f(n_o)}\alpha| = |\alpha| + f(n_o) + 1 \leq f(|i^{n_o+1}|) = f(n_o+1),$$

and, hence, $f(n_o) < f(n_o+1)$, contradicting the choice of n_o. By the same argument, we get for all $n \in \mathbb{N}$ that

$$si^n\varrho \longrightarrow o^{f(n)}\varrho\alpha \quad \text{for some } \alpha \in C(S)$$

implies that $\alpha = \lambda$.

Figure 3 shows two examples for cores of the net in figure 2 and gives the corresponding functions computed by the IPNC of figure 2 where the core is plugged in for $\mathcal{P}_c$. As we shall use the first example later on we state

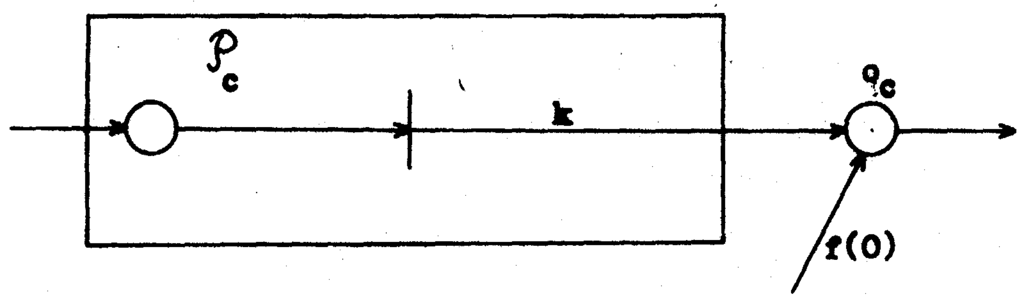
__Lemma 4:__

The Petri net of figure 2 with the net of figure 3a) replacing $\mathcal{P}_c$ is an IPNC for $f : \mathbb{N} \longrightarrow \mathbb{N}$ with $f(n) = kn + f(0)$ $(k \in \mathbb{N}^+,\ f(0) \in \mathbb{N})$.

__Proof:__ The resulting net is clearly a v-WPNC for f, and so it only remains to show that IC1 is satisfied. IC1 follows from the observation that any m tokens distributed among the places i, o, and the operational places of the net can obviously produce the maximal number of tokens on o if all m tokens are initially on place i; moreover, there cannot be produced more to-

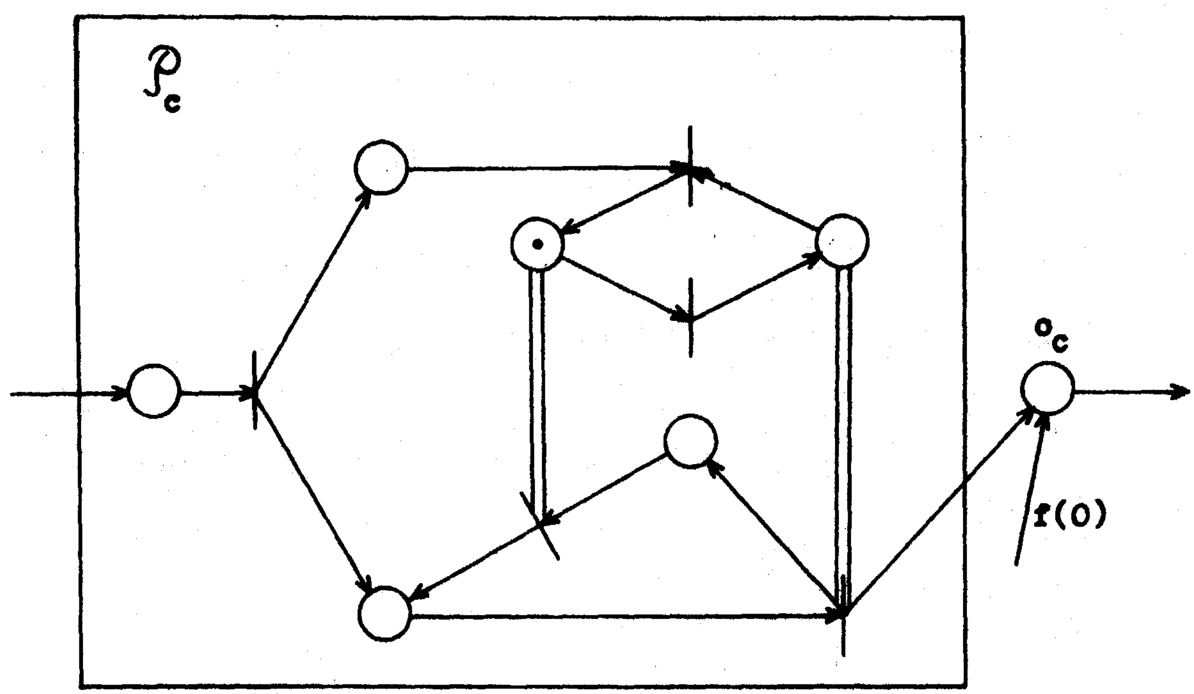
Figure 3:

Example 1:



$f(n) = kn + f(0)$ $(k \in \mathbb{N}^+,\ f(0) \in \mathbb{N})$

Example 2:



$f(n) = n^2 + f(0)$ $(f(0) \in \mathbb{N})$

kens on o if the initial marking of the control places is $\rho' = u$ and not $\rho = v$ (note that $RC_{\rho} = \{\rho, \rho'\}$) as there is no feedback from o_c into the core.

q.e.d.

IV. Recursive construction of an IPNC for A_n

In this section we are going to show that the class of functions which are computed by IPNC's is essentially closed under iteration. Exploiting this fact, we are able to construct small WPNC's for the functions A_n, $n \in N$. In particular, let $\mathcal{F}$ be an IPNC computing a function

$$f : N \longrightarrow N$$

with $f(0) > 0$, and let $g : N \longrightarrow N$ be defined by

$$\text{i)} \quad g(0) = 1,$$

$$\text{ii)} \quad g(n+1) = f(g(n)) \qquad \forall\, n \in N,$$

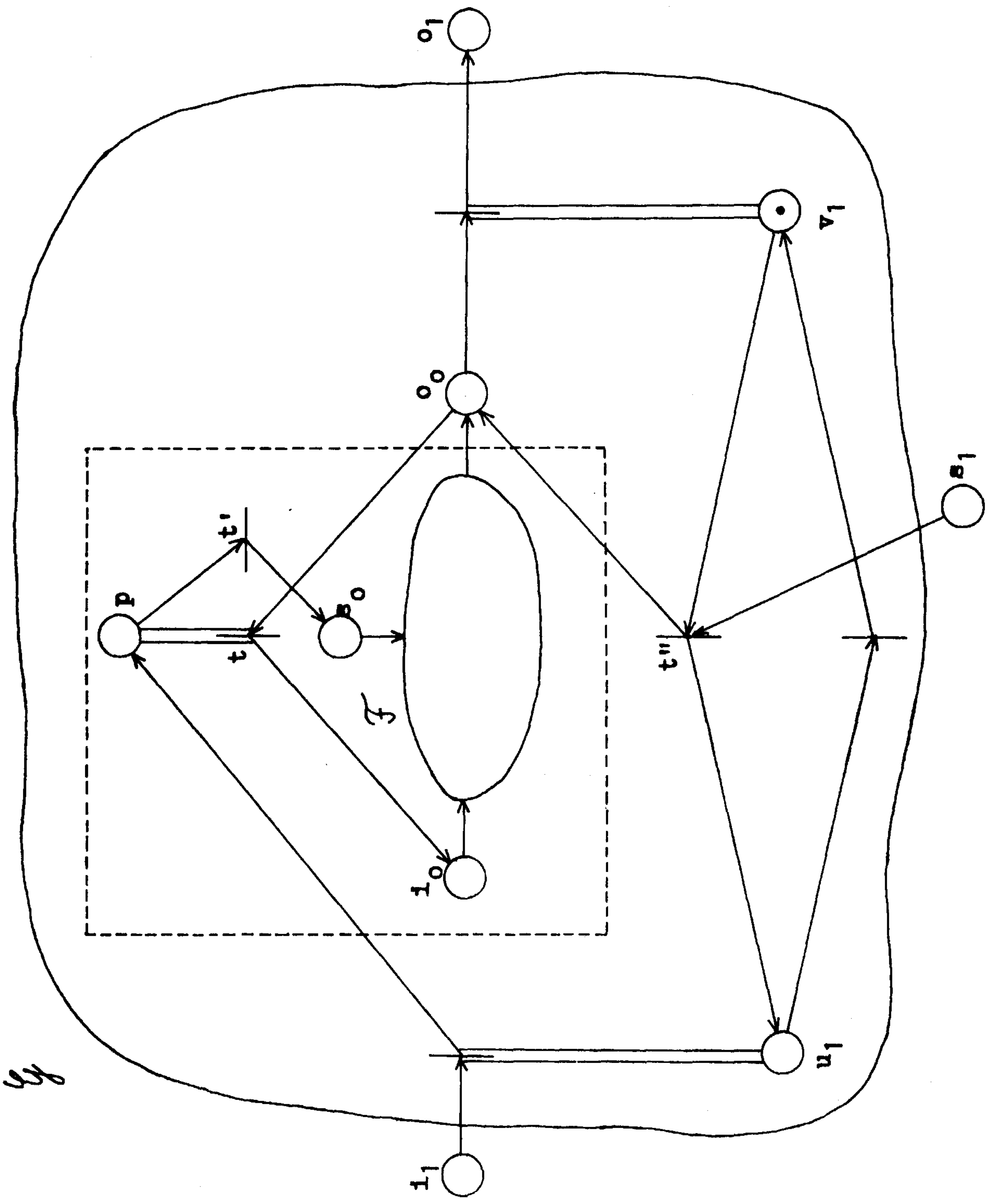
i.e. $g(n) = f^{(n)}(1)$ is the n-th iterate of f at 1.

Now define the Petri net $\mathcal{G}$ as given in figure 4. Essentially, a feedback mechanism is added to $\mathcal{F}$ which allows to transfer the output of $\mathcal{F}$ back to its input-place as many times as there are tokens on the input-place i_1 of $\mathcal{G}$. The other additional places (u_1 and v_1) are part of the standard structure and ensure IC2 for $\mathcal{G}$. The dotted line indicates the core of $\mathcal{G}$, corresponding to figure 2. To denote corresponding places in $\mathcal{G}$ and its subnet $\mathcal{F}$, we use the same letter and index 0 for $\mathcal{F}$, 1 for the additional places in $\mathcal{G}$. As in figure 4 for $\mathcal{F}$, we will in simplified diagrams only draw the interconnecting places and indicate the rest of the net by an oval-shaped line.

Lemma 5:

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

<u>**Figure 4:**</u>



<u>Proof:</u> Let ρ_o be an iteratively conservative marking of $\mathcal{F}$ such that $\mathcal{F}$ is a ρ_o-WPNC for f, and let $S^o_{ip} = \{s_o, i_o, o_o\}$, S^o_{cp}, and S^o_{op} denote the set of interconnecting, control and operational places, resp., of $\mathcal{F}$. With $\mathcal{F} = (S^o, T^o, \underline{in}^o, \underline{out}^o)$ and $\mathcal{G} = (S^1, T^1, \underline{in}^1, \underline{out}^1)$ set

$$S^1_{ip} := \{s_1, i_1, o_1\} ,$$
$$S^1_{cp} := S^o_{cp} \cup \{u_1, v_1\} ,$$
$$S^1_{op} := S^1 \setminus (S^1_{ip} \cup S^1_{cp})$$

(i.e. the operational places of $\mathcal{G}$ are all places of $\mathcal{F}$ except the control places, together with an additional place p).
It can easily be seen that $\rho_1 := v_1 \rho_o \varepsilon C(S^1_{cp})$ is conservative: when the token on v_1 is transferred to u_1, it can always be restored to v_1. The sum of the tokens on the two places u_1 and v_1 is constant as no token can be deposited on any of them without removing at the same time a token from the other place, and vice versa. As $\rho_1 = v_1 \rho_o$, and $\rho_o \varepsilon C(S^o_{cp})$ is itself conservative, so is ρ_1.
Now let $g^* : N \longrightarrow N \cup \{\infty\}$ be the function for which $\mathcal{G}$ is a ρ_1-WPNC. As property IC2 of definition 13 is ensured by the standard structure of $\mathcal{G}$, it suffices to show (i) IC1 for g^* in place of f, and (ii) $g^* = g$.
(i) If $\rho' \varepsilon RC_\rho$ (= set of all submarkings on S^1_{cp} reachable in $\mathcal{G}$

from any marking that agrees with ρ on S^1_{cp}) contains u_1 instead of v_1, first some or all tokens on i_1 may be transported into the core, then the u_1-token is transferred to v_1 and t" can fire, restoring the token on u_1. As each firing sequence which is firable at some marking is also firable at any bigger marking, the maximal number of tokens obtained on o_o or o_1 does not depend on whether ρ' contains u_1 or v_1. Hence, we may assume w.l.g. that $\rho' = v_1 \rho_o'$ for some $\rho_o' \in RC_{\rho o}$. It follows immediately from the structure of $\mathcal{G}$ that RC_{ρ_o} is independent from whether it is computed w.r.t. $\mathcal{G}$ or w.r.t. $\mathcal{F}$, i.e. independent from whether reachability is considered in $\mathcal{G}$ or its subnet $\mathcal{F}$. Thus, we have to show

$$(\forall \alpha, \beta \in C(S^1_{op} \cup \{i_1, o_1\}), \forall \rho' \in v_1 RC_{\rho_o})[s\alpha\rho' \xrightarrow{*} \beta\rho_1 \Rightarrow |\beta| \le g^*(|\alpha|)] .$$

As o_1 is not an input-place for any transition in T^1 we may assume w.l.g. $\alpha \in C(S^1_{op} \cup \{i_1\})$. Let $\alpha = \gamma\delta \ne \lambda$ with $\gamma \in C(\{i_1, p, s_o\})$, $\delta \in C(S^o_{op} \cup \{i_o, o_o\})$ (see figure 4).

<u>Case 1:</u> $\gamma = \lambda$

As $\mathcal{F}$ is an IPNC for f we have for $\beta \in C(S^1_{op} \cup \{i_1, o_1\})$ and $\rho' \in v_1 RC_{\rho_o}$ with $s_1\delta\rho' \xrightarrow{*} \beta\rho_1$, that

$|\beta| \le 1 + f(|\delta|) - f(0) \le 1 + f(|\delta|) - 1 \le f(|\alpha|)$, as there is no token on s_o. We now show by induction for $n \ge 1$:

$$f(n) \le g^*(n).$$

$n=1$: $g^*(1) \geq f(1)$ as t'' produces a token on o_0 which can be transferred to i_0. The token on i_1 is first used to enable this transport and then to start $\mathcal{F}$ on input i_0.

$n-1 \to n$: $g^*(n) \geq f(g^*(n-1))$ as the n-th input-token on i_1 may be used to start $\mathcal{F}$ once more on the output so far accumulated on o_0, which is at most $g^*(n-1)$. As f is strictly monotone and has no fixed points ($f(0)>0$) we obtain from the induction hypothesis: $f(g^*(n-1)) \geq f(f(n-1)) \geq f(n)$.

Hence, $|\beta| \leq g^*(|\delta|) = g^*(|\alpha|)$.

<u>Case 2:</u> $|\gamma| = m > 0$

A firing sequence of $\mathcal{G}$ leading from $s_1\alpha\varrho'$ to $\beta\varrho_1$ ($\varrho' \varepsilon\, v_1 RC_{\varrho_0}$) has w.l.g. the form:

$$s_1\gamma\delta\varrho' \xrightarrow{*} \underbrace{p^m\delta_1}_{=:\alpha_1}\varrho_{0,1}v_1 \xrightarrow{t'} p^{m-1}s_0\delta_1\varrho_{0,1}v_1 \xrightarrow{*}$$

$$\xrightarrow{*} \underbrace{p^{m-1}\delta_2}_{=:\alpha_2}\varrho_{0,2}v_1 \xrightarrow{t'} p^{m-2}s_0\delta_2\varrho_{0,2}v_1 \xrightarrow{*}$$

$$\vdots$$

$$\xrightarrow{*} \underbrace{p\delta_m}_{=:\alpha_m}\varrho_{0,m}v_1 \xrightarrow{t'} s_0\delta_m\varrho_{0,m}v_1 \xrightarrow{*} \beta\varrho_1$$

with $\delta_i \varepsilon\, C(S^o_{op} \cup \{i_o, o_o\})$, $\varrho_{o,i} \varepsilon\, C(S^o_{cp})$ for $i = 1,\ldots,m$, or can trivially be simulated by such a sequence if γ already contains tokens on s_0. Informally speaking, this decomposition is obtained by breaking any firing sequence $s_1\alpha\varrho' \xrightarrow{*} \beta\varrho_1$ whenever t' is fired.

Now set $\alpha_i := p^{m+1-i}\delta_i$ for $i = 1,\dots,m$, $\alpha_{m+1} := \beta$.

It suffices to show

(*) $|\alpha_i| \le g^*(|\delta|+i-1) + m + 1 - i$ for $i = 1,\dots,m+1$.

For $i = 1$ this comes from case 1 and property IC2 for $\mathcal{F}$, as no token of γ is consumed and, therefore, no additional tokens on o_o can be generated. Shuffling tokens from o_o to i_o which is made possible by tokens on p does not affect the argument in case 1. Thus

$$|\alpha_1| \le g^*(|\delta|) + m.$$

Assume that (*) is established for all i with $1 \le i < i_o \le m+1$.

Consider the subsequence

$$\alpha_{i_o-1}\varrho_{o,i_o-1}v_1 = p^{m+1-(i_o-1)}\delta_{i_o-1}\varrho_{o,i_o-1}v_1 \xrightarrow{t'}$$
$$p^{m+1-i_o}\delta_{i_o-1}\varrho_{o,i_o-1}v_1 \xrightarrow{*} p^{m+1-i_o}\delta_{i_o}\varrho_{o,i_o}v_1 = \alpha_{i_o}\varrho_{o,i_o}v_1 .$$

We have $\varrho_{o,i_o-1} \in RC\varrho_o$ and, as ϱ_o is conservative, we may assume that $\varrho_{o,i_o} = \varrho_o$.

From IC1 for $\mathcal{F}$ we obtain, then,

$$|\delta_{i_o}| \le f(|\delta_{i_o-1}|)$$

and hence

$$\begin{aligned} |\alpha_{i_o}| &= |\delta_{i_o}| + m + 1 - i_o \\ &\le f(|\delta_{i_o-1}|) + m + 1 - i_o \\ &\le f(|\alpha_{i_o-1}|+i_o-m-2) + m + 1 - i_o \end{aligned}$$

$$\le f(g^*(|\delta|+i_o-2)) + m + 1 - i_o \qquad \text{(ind.hyp.)}$$

$$\le g^*(|\delta|+i_o-1) + m + 1 - i_o \,.$$

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

For $i_o = m+1$ we obtain, therefore,

$$|\alpha_{m+1}| = |\beta| \le g^*(|\delta|+m) = g^*(|\alpha|) \,.$$

Hence, IC1 holds for $\mathcal{G}$.

(ii) We obviously have

$$g^*(0) = 1 = g(0) \text{ and } g^*(1) = f(1) = g(1).$$

Assume that $g^* \neq g$. Inspection of the net $\mathcal{G}$ shows that clearly $g^* \ge g$ as we might first fire t'', transport all tokens of i_1 to p and then, as long as there are tokens on p, shuffle all tokens from o_o back to i_o and apply $\mathcal{F}$, consuming one of the tokens on p. In this way, we can iterate $\mathcal{F}$ as often as n times if i_1 initially had n tokens, and as $\mathcal{F}$ iteratively computes f and we start off with one token on o_o (after firing t'') we obtain by this firing sequence at least $f^{(n)}(1) = g(n)$ tokens on o_o (and thus also on o_1) after the last iteration of $\mathcal{F}$. Let, therefore, $n > 1$ be minimal such that

$s_1 i_1^n \varrho_1 \xrightarrow{\tau} \beta\varrho_1$ for some $\beta \in C(S^1)$ with $|\beta| > g(n)$ and some $\tau \in (T^1)^+$.

τ is w.l.g. of the form

$$s_1 i_1^n \varrho_1 \xrightarrow{*} p^2 \delta_1 v_1 \varrho_o' \xrightarrow{t'} p s_o \delta_1 v_1 \varrho_o' \xrightarrow{\tau_1} p\delta_2 v_1 \varrho_o'' \xrightarrow{\tau_2} p\delta_3 v_1 \varrho_o'' \xrightarrow{t'}$$

$\xrightarrow{t'} s_o\delta_3 v_1 \varrho_o'' \xrightarrow{n} \beta\varrho_1$,

with $\delta_1, \delta_2, \delta_3 \in C(S^o_{op} \cup \{i_o, o_o\})$, ϱ_o', $\varrho_o'' \in RC_{\varrho_o}$, $\tau_1 \in (T^o)^+$ such that the first transition of τ_1 removes s_o, and $\tau_2 \in \{t\}^*$, as it suffices to have one token on s_o at a time and as, because of IC2 for $\mathcal{F}$, the firings of t can be collected in τ_2 after τ_1. We may also assume that δ_1 does not contain o_o because shuffling them to i_o (enabled by p) would certainly yield a bigger output. As, reaching $p\delta_2 v_1 \varrho_o''$, the last token on p actually was not used, and as n is minimal, we have

$$|\delta_2| \leq g(n-1).$$

<u>Case 1:</u> δ_2 contains tokens on o_o.

These tokens were placed on o_o by τ_1. Because of IC2 for $\mathcal{F}$ we, therefore, have $|\delta_3| = |\delta_2|$, and because of IC1 for $\mathcal{F}$

$$|\beta| \leq f(|\delta_3|) \leq f(g(n-1)) = g(n).$$

<u>Case 2:</u> δ_2 contains no tokens on o_o.

Then $\tau_2 = \lambda$ and $\delta_2 = \delta_3$. IC1 for $\mathcal{F}$ yields again

$$|\beta| \leq f(|\delta_3|) \leq g(n).$$

Together with the remark at the beginning of (ii) we thus have shown
$$g^* = g.$$

q.e.d.

We want to remark that the construction of $\mathcal{G}$ from $\mathcal{F}$ is not optimal concerning the number of additional places and transitions. One might observe that the places u_1 and v_1 are not necessary, thus obtaining the net of figure 5a) which is, of course, no longer an IPNC. Using transformations discussed in [14] this net can be simplified even further (figure 5b)). We note that the net of figure 5b) has only one more place than $\mathcal{F}$. Without proof we state that both nets of figure 5 are WPNC's for g (with the modification that i_1 and o_1 are no longer only input resp. output-places of transitions of the net), and that the construction by which they are obtained from $\mathcal{F}$ can be applied recursively, yielding WPNC's for the iterate of g, its iterate etc.. We think, however, that the standard structure facilitates the proof of lemma 5 and unifies the recursive application of the construction.

We summarize the results of this section in

<u>Theorem 2:</u>

i) $(\forall n \in \mathbb{N} \exists \mathcal{A}_n)[\mathcal{A}_n$ is a λ-WPNC for A_n with designated places

$$s_n, i_n, o_n];$$

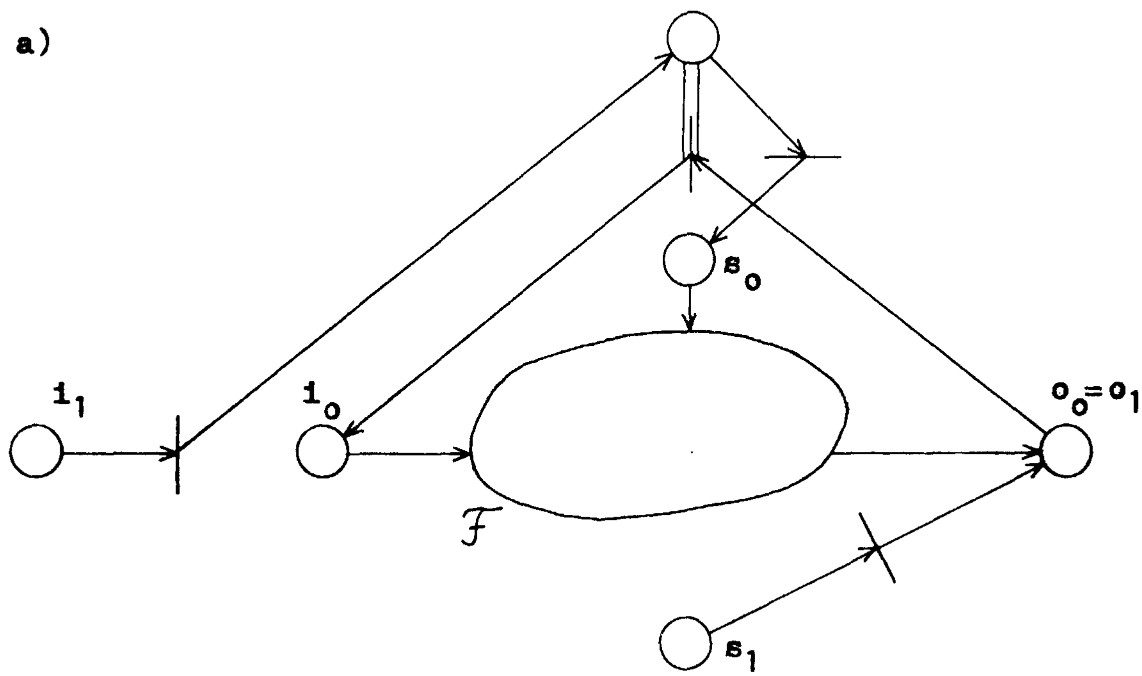
ii) $\text{size}(\mathcal{A}_n) = O(n \cdot \log(n))$;

iii) $R(\mathcal{A}_n, s_n i_n^2)$ is finite.

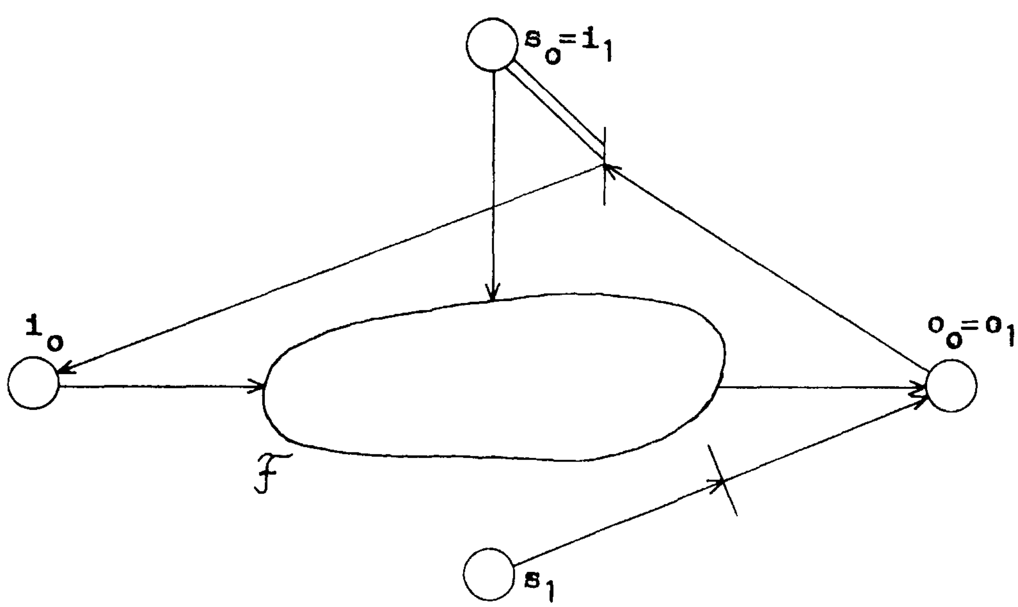
<u>Proof:</u> (i) Using the IPNC of lemma 4 for $f(x) = 2x + 1$ and applying the construction of lemma 5 recursively n times we obtain by lemma 5 an IPNC for A_n (with iteratively conservative

<u>Figure 5:</u>

a)



b)



marking $\varrho_n = \prod_{i=1}^{n} v_i$). Inserting between s_n and the transition corresponding to t" in figure 4 an additional place and a transition which initializes the marking ϱ_n we then get a λ-WPNC $\mathcal{A}_n$ for A_n.

(ii) is immediate from definition 6 as in each step of the recursive construction a constant number of places and arcs is added.

(iii) Each iteration of $\mathcal{F}$ in $\mathcal{G}$ consumes a token from s_o, and thus properly decreases the number of tokens on i_1, p, and s_o. We may assume inductively that $\mathcal{F}$ permits only finite firing sequences (the IPNC of lemma 4 for $f(x) = 2x + 1$ certainly does so). But as the loop between u_1 and v_1 in $\mathcal{G}$ consumes tokens from s_1 we conclude that the reachability set of $\mathcal{G}$, and hence recursively, that of $\mathcal{A}_n$ is finite for the given initial marking.

q.e.d.

V. Boundable WPNC's for polynomials

In this section we are going to construct WPNC's for multivariable polynomials with nonnegative integer coefficients with the special property that they are boundable, i.e. the number of tokens on any place in the markings of marking sequences of computations can essentially be bounded by the size of the input. The basic multiplier nets have also been introduced in [7].

<u>Lemma 6:</u>

Let $p(x_1,\dots,x_m) = \sum_{i=1}^{v} a_i \prod_{j=1}^{m} x_j^{e_{ij}}$ be a polynomial with positive integer coefficients a_i, and $e_{ij} \varepsilon \mathbb{N}$ for $i = 1,\dots,v$, $j = 1,\dots,m$. There exists a λ-WPNC $\mathcal{P}$ for p.

<u>Proof:</u> We shall build up $\mathcal{P}$ in two steps from basic units which serve as multipliers and which can be connected to form weak computers for monomials. Several of these then constitute $\mathcal{P}$.

(i) The net $\mathcal{T}$ of figure 6a) has the property that $n_1 n_2 = \max\{k;\ i^{n_1} j^{n_2} r \xrightarrow{*} o^k r\alpha \text{ for some } \alpha\}$ for all $(n_1,n_2)\,\varepsilon\,\mathbb{N}^2$. Obviously, $n_1 n_2$ tokens on o can be achieved by transporting all n_1 tokens from u to o and u', and from u' back to u, as often as a token on the control places r' and then r enables the firing sequence t^{n_1} followed by $(t')^{n_1}$. This can happen exactly n_2 times which also shows that $n_1 n_2$ is the maximal number of tokens reachable on o. As the number of tokens on i, u, and u' cannot increase, and as each 'cycle' (firing sequence) $\varepsilon \{t\}^{+}\{v\}^{+}$ consumes a token from j, an initial marking of $\mathcal{T}$ of the form $i^{n_1}j^{n_2}r$ permits no infinite firing sequences.

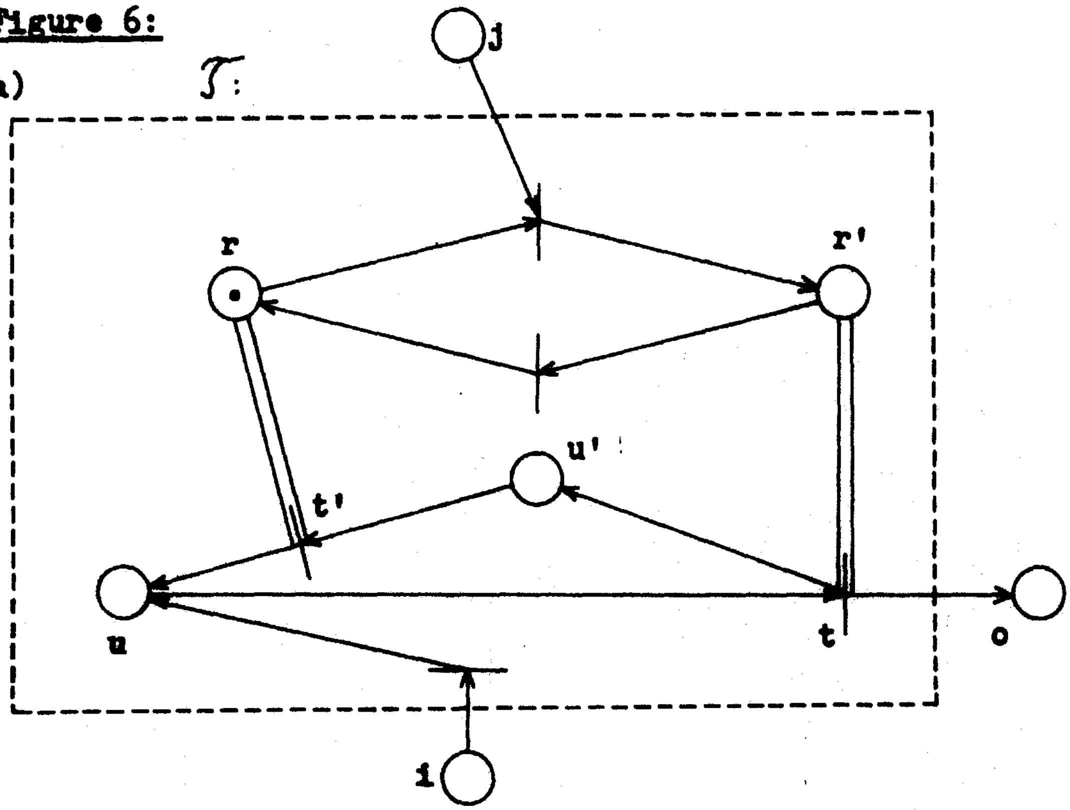
(ii) Now we connect d instances of $\mathcal{T}$ to form the net $\mathcal{M}$ in the way shown in figure 6b) where each box stands for the part of $\mathcal{T}$ surrounded by a dotted line in figure 6a) (note that the input-places are connected to the previous output-places). Let ρ denote the product of the places corresponding to r of figure 6a) in the instances of $\mathcal{T}$. Then repeated application of (i) yields for $\mathcal{M}$:

$a\prod_{j=1}^{d} n_j = \max\{k;\ c^{a}\prod_{j=1}^{d} i_j^{n_j}\rho \xrightarrow{\alpha} o^{k}\rho\alpha \text{ for some } \alpha\}$ for all $(n_1,\dots,n_d) \varepsilon\, \mathbb{N}^{d}$, i.e. $\mathcal{M}$ weakly computes the homogeneous monomial $a\prod_{j=1}^{d} x_j$ of degree d. Likewise, initial markings of the form $c^{a}\prod_{j=1}^{d} i_j^{n_j}\rho$ have no infinite firing sequences.

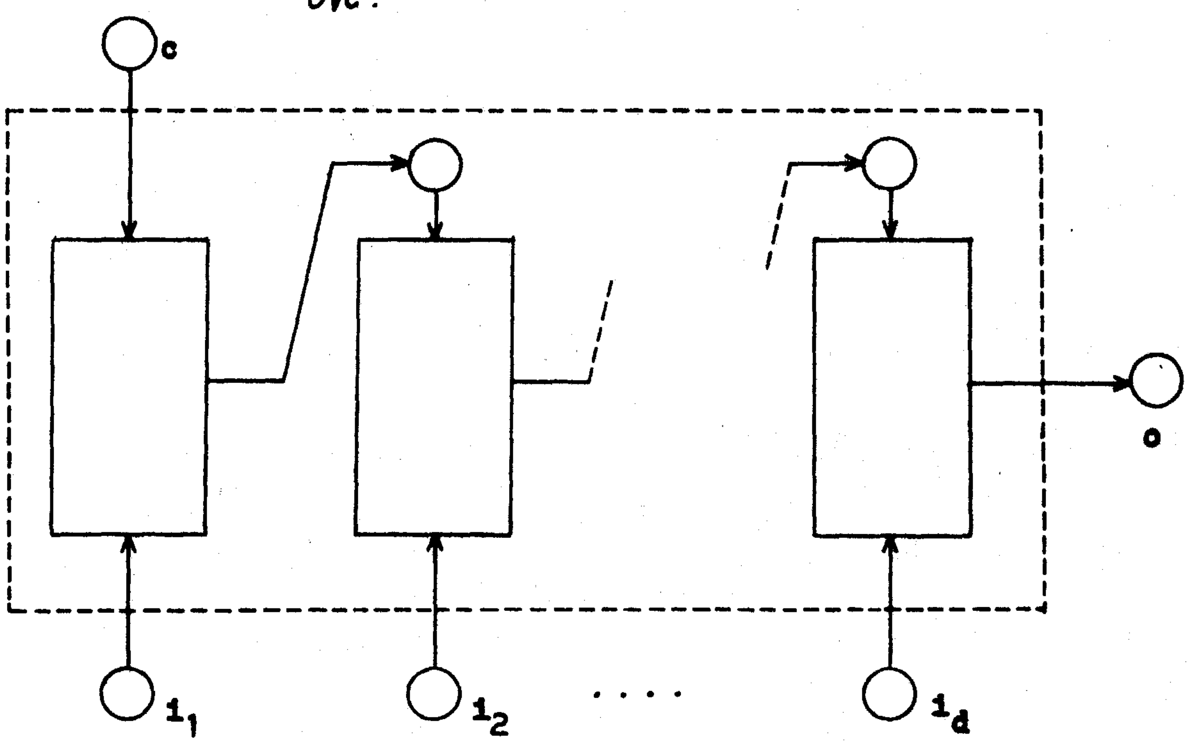
(iii) In order to obtain a WPNC $\mathcal{P}$ for the polynomial p, v monomial nets $\mathcal{M}_1,\dots,\mathcal{M}_v$ (see figure 6c)) with suitable degrees are combined sharing a common output-place o. The net $\mathcal{P}$ has input-places $i_1,\dots,i_m$ (to avoid ambiguity, we assume that the places in the monomial nets are given distinct names), which deliver their tokens to as many input-places in each monomial net as the degree of the corresponding variable in the monomial indicates. A monomial net for a constant monomial a_j $(a_j > 0)$ consists just of one place c_j and a transition connecting it

Figure 6:

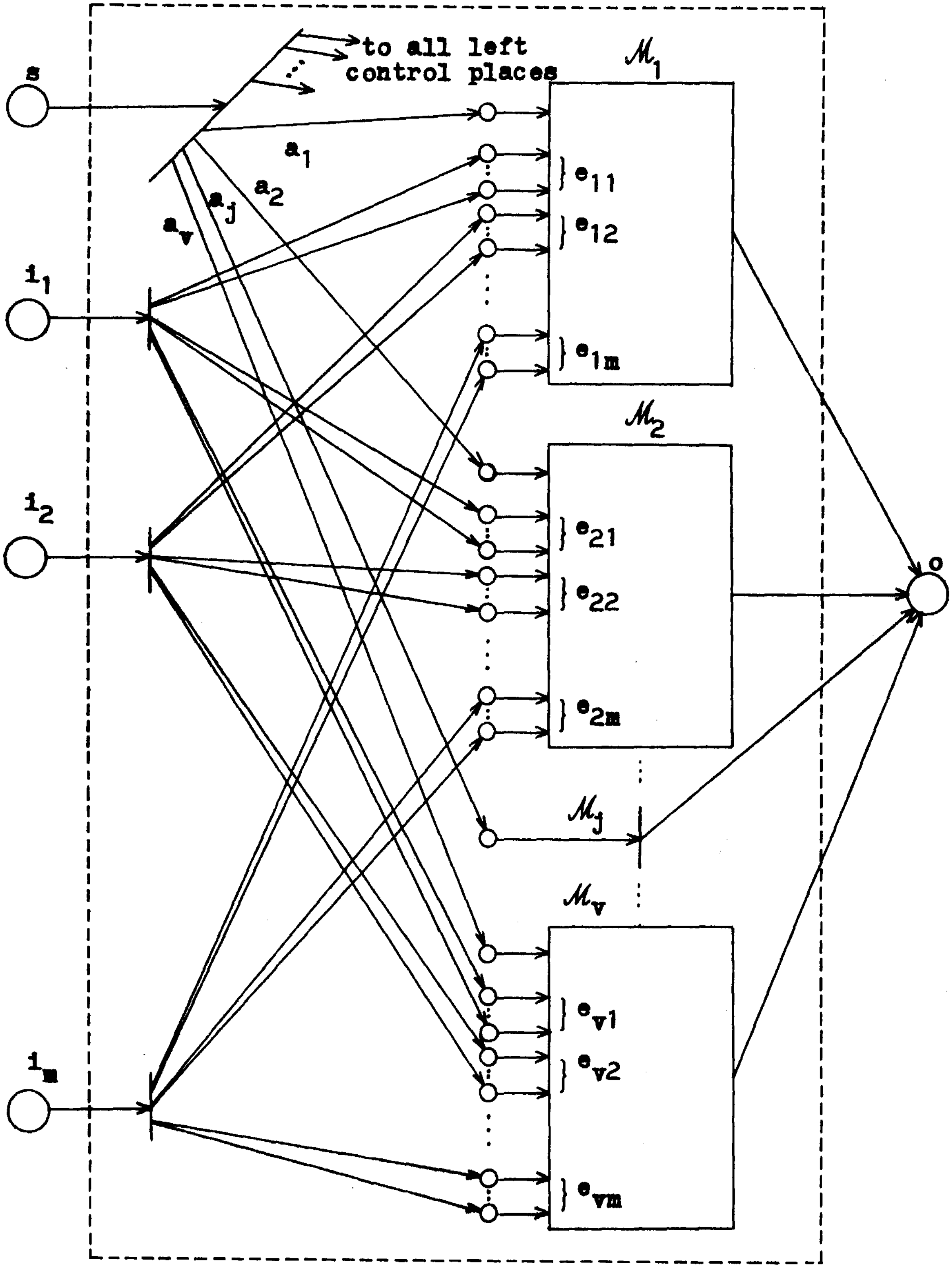
a) $\mathcal{T}$:



$\mathcal{M}$:



c) $\mathcal{P}$:


s
to all left
control places
$\mathcal{M}_1$
a_1
a_2
a_j
a_v
i_1
i_2
i_m
e_{11}
e_{12}
e_{1m}
$\mathcal{M}_2$
e_{21}
e_{22}
e_{2m}
$\mathcal{M}_j$
$\mathcal{M}_v$
e_{v1}
e_{v2}
e_{vm}
o

to o. The places c_i (figure 6c)) are those corresponding to c in figure 6b), and the places referred to as 'left control places' are those corresponding to r in figure 6a). It is easy to see that $\mathcal{P}$ is a λ-WPNC for p and that each marking $s\prod_{j=1}^{m} i_j^{n_j}$ with $(n_1,\dots,n_m) \in \mathbb{N}^m$ permits only finite firing sequences.

q.e.d.

A very important observation about the multiplier net $\mathcal{T}$ of figure 6a) is that during a computation none of the 'internal' places u,u',r,r' ever contains more than $\max\{1, n_1\}$ tokens where n_1 is the number of tokens initially on place i. Also, one cycle which outputs another n_1 tokens on o can be performed if just one token is available on j.

Let $\mathcal{N} = (S, T, \underline{in}, \underline{out})$ be a Petri net, $S' \subseteq S$, α a marking of $\mathcal{N}$, and τ a firing sequence of $\mathcal{N}$ starting at α. We say that τ is <u>bounded on S' by $N \in \mathbb{N}$</u> iff all markings of the marking sequence generated by τ contain at most N tokens on any of the places in S'.

Considering the above remark about the multiplier nets we are going to define a special class of functions. These functions have the property that they are computable by WPNC's by firing sequences which are bounded on the non-designated places by a (simple) function of the arguments.

<u>Definition 14:</u>

Let $f : N^m \longrightarrow N$, $g : N \longrightarrow N$ be functions. f is <u>g-boundable</u> iff there exists a ρ-WPNC $\mathcal{F}=(S, T, \underline{in}, \underline{out})$ with designated places $s, i_1, \ldots, i_m, o$ for f such that

$(\forall N \in N, \forall (n_1, \ldots, n_m) \in \{0,1,\ldots,N\}^m, \forall k \in \{0,1,\ldots,f(n_1,\ldots,n_m)\}$

$\exists \tau \in T^+)$

$[s \prod_j^m i_j^{n_j} \rho \xrightarrow{\tau} o^k \alpha$ for some $\alpha \in C(S \setminus \{o\})$ <u>and</u> τ is bounded by $g(N)$ on $S \setminus \{s, i_1, \ldots, i_m, o\}]$.

<u>Theorem 3:</u>

Let $p \in N[x_1, \ldots, x_m]$ be a polynomial with nonnegative integer coefficients, $\|p\|$:= maximum of the coefficients of p, and

$$\forall N \in N: \quad g(N) := N + \|p\|.$$

Then p is g-boundable.

<u>Proof:</u> Consider the WPNC $\mathcal{P}$ of figure 6c). The input-places of the monomial components of $\mathcal{P}$ receive tokens only from the places s and $i_1, \ldots, i_m$. Because of the structure of $\mathcal{P}$, it suffices to show that for the monomial net $\mathcal{M}$ of figure 6b) with initialized left control places, up to N tokens on each of the places $i_1, \ldots, i_d$, and at most $\|p\|$ tokens on c, there is a firing sequence of $\mathcal{M}$ producing the desired output which is bounded by g(N) on all places but o. But, referring to the observation two paragraphs above about the multiplier nets, this can be achieved

by, recursively, firing a complete cycle in the rightmost $\mathcal{J}$-component of $\mathcal{M}$ which has at least one token on its upper input-place.

q.e.d.

VI. Two modifications of polynomial WPNC's

When reducing BPI to the containment problem for reachability sets of Petri nets only the markings of the designated places are of interest. In [8], a method is discussed which enables us to forget about the non-designated places in the general case where the reachability sets needn't be finite. This construction modifies the Petri net computers in such a way that in an additional stage after a computation transitions are enabled which can feed in or consume arbitrarily many tokens on each non-designated place. However, this construction is not applicable in our case as it would produce Petri nets with infinite reachability sets. But the result of the preceding section makes it possible to introduce two modifications of polynomial WPNC's which are also guided by the idea of factoring out the marking on the non-designated places and which do preserve finiteness of the reachability sets.

Informally speaking, in order to test whether $(p,q,n) \varepsilon$ BPI we construct appropriate WPNC's $\mathcal{P}$ and $\mathcal{Q}$ for p and q such that in $\mathcal{P}$ all non-designated places are suitably bounded whereas in $\mathcal{Q}$ the marking on those places may take on any value up to this bound, and only finitely many values above it.

Let $\mathcal{P}=(S, T, \underline{in}, \underline{out})$ be a λ-WPNC for the polynomial $p \varepsilon \mathbb{N}[x_1, \ldots, x_n]$, as constructed in lemma 6, with designated places s,

$i_1,\dots,i_m,o$, set $O := S \setminus \{s, i_1,\dots,i_m,o\}$, and let O^c be a copy of O (disjoint from S). The unique element in O^c corresponding to $u \in O$ will be denoted u^c.

Now define the Petri net $\mathcal{P}' = (S', T', \underline{in}', \underline{out}')$ as follows:

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

<u>Lemma 7:</u>

Let p, $\mathcal{P}$, and $\mathcal{P}'$ be as above, set for $N \in \mathbb{N}$ $g(N) := N + |p|$ and $\varphi_N := \prod_{u \in O^c} u^{g(N)}$.

Then $\mathcal{P}'$ is a φ_N-WPNC for p restricted to $\{0,1,\dots,N\}^m$.

<u>Proof:</u> Because of the definition of $\underline{in}'$ and $\underline{out}'$, the firing of any transition in T' which removes tokens from a place $u \in O$ adds just as many tokens on u^c, and a transition which adds tokens to $u \in O$, removes the same number from u^c. This is also true with O and O^c interchanged. Thus, the sum of the tokens on u and u^c always equals $g(N)$, for all $u \in O$. Further, each firing sequence of $\mathcal{P}$ starting at $\alpha \in C(S \setminus O)$ which is $g(N)$-bounded on O, is also firable in $\mathcal{P}'$, starting at $\alpha\varphi_N$, and conversely. But from theorem 3 we know that p is g-boundable, and that, in fact, for each input tupel $(n_1,\dots,n_m) \in \{0,1,\dots,N\}^m$ and each k with $0 \le k \le p(n_1,\dots,n_m)$, $\mathcal{P}$ allows a firing sequence τ with $s\prod_{j=1}^{m} i_j^{n_j} \xrightarrow{\tau} o^k\alpha$

for some $\alpha \in C(S \setminus \{o\})$ which is bounded by $g(N)$ on O. Hence, $s\prod_{j=1}^{m} i_j^{n_j} \varrho_N \xrightarrow{\tau} o^k \beta$ for some $\beta \in C(S' \setminus \{o\})$ also holds. Together with the fact that each such firing sequence of $\mathcal{P}_p'$ can also be executed on $\mathcal{P}$, starting at $s\prod_{j=1}^{m} i_j^{n_j}$, the claim follows.

q.e.d.

The following theorem summarizes the results.

<u>Theorem 4:</u>

Let $p \in \mathbb{N}[x_1,\dots,x_m]$ be a polynomial with nonnegative integer coefficients, and set for all $N \in \mathbb{N}$ $g(N) := N + \|p\|$. Then there exists a Petri net $\mathcal{P}_p = (S,T,\underline{\text{in}},\underline{\text{out}})$ with m+3 designated places $s,i_1,\dots,i_m,o$, and $b \in S$ such that

(i) $\mathcal{P}_p$ is a $b^{g(N)}$-WPNC with designated places $S_{ip} := \{s,i_1,\dots,i_m, o\}$ for p restricted to $\{0,1,\dots,N\}^m$, for all $N \in \mathbb{N}$.

(ii) Let, for $u \in S$, $n \in \mathbb{N}$, $\langle u\rangle^n$ denote the set $\{u, \lambda\}^n$. Then

$$(\forall N \in \mathbb{N}, \forall (n_1,\dots,n_m) \in \mathbb{N}^m)\left[R(\mathcal{P}_p,\ s\prod_{j=1}^{m} i_j^{n_j} b^{g(N)}) \subseteq \right.$$

$$\left. \langle s\rangle^1 \prod_{j=1}^{m} \langle i_j\rangle^{n_j} \prod_{u \in S \setminus S_{ip}} \langle u\rangle^{g(N)} \langle o\rangle^{p(n_1,\dots,n_m)}\right].$$

In particular,

$$(\forall N \in \mathbb{N}, \forall (n_1,\dots,n_m) \in \{0,1,\dots,N\}^m)\left[R(\mathcal{P}_p, s\prod_{j=1}^{m} i_j^{n_j} b^{g(N)}) \text{ is finite}\right].$$

(iii) $\mathrm{size}(\mathcal{P}_p) = O(\mathrm{size}(p) \cdot \log(\mathrm{size}(p)))$.

Proof: Take the net $\mathcal{P}_p'$ of lemma 7 and add a place b and a transition which has b as input-place and all $u \in O^c$ as output-places. Call the new net $\mathcal{P}_p$. Together with lemma 7 this implies (i). The number of tokens on o is bounded by the WPNC-property of $\mathcal{P}_p$, and the number of tokens on each non-designated place $(= O \cup O^c$ of $\mathcal{P}_p')$ is bounded by g(N) by the construction of $\mathcal{P}_p'$ as noted in the proof of lemma 7. As the designated places s, i_1, ..., i_m and b are only input-places (ii) holds.
Condition (iii) follows from the observation that both the number of arcs of multiplicity one and the number of places in $\mathcal{P}_p$ are bounded by the sum of the degrees of the monomials of p times a constant and that the code for multiple arcs in $\mathcal{P}_p$ uses space proportional to the code for the coefficients of p.

q.e.d.

The second modification we are going to introduce has the purpose to 'blur' the marking on the unimportant places of a polynomial WPNC sufficiently, preserving at the same time, however, the finiteness of the reachability set.

Definition 15:
Let $f : N^m \longrightarrow N$ be a number theoretic function, $\mathcal{F} = (S, T, \underline{in}, \underline{out})$ a Petri net. $\mathcal{F}$ is a blurring WPNC for f iff $\mathcal{F}$ has m+5 designated places $s, i_1, \ldots, i_m, o, c_1, c_2, e \in S$ such that

i) $(\forall N \in \mathbb{N})[\mathcal{F}$ is a e^N-WPNC for f w.r.t. $s, i_1, \ldots, i_m, o]$;

ii) $(\forall N \in \mathbb{N}, \forall (n_1, \ldots, n_m) \in \mathbb{N}^m)[\prod_{u \in O} \langle u \rangle^N \langle o \rangle^{f(n_1, \ldots, n_m)} \subseteq$

$$R(\mathcal{F}, s \prod_{j=1}^{m} i_j^{n_j} e^N)] \text{ , where}$$

$O \subseteq S$ is the set of non-designated places

(Note that the reachable markings considered in ii) don't contain tokens on c_1, c_2, and e).

Theorem 5:

Let $q \in \mathbb{N}[x_1, \ldots, x_m]$ be a polynomial with nonnegative integer coefficients. Then there exists a Petri net $\mathcal{Q}_\varepsilon = (S, T, \underline{in}, \underline{out})$ such that

(i) $\mathcal{Q}_\varepsilon$ is a blurring WPNC for q with designated places $s, i_1, \ldots$

$\ldots, i_m, o, c_1, c_2, e \in S$;

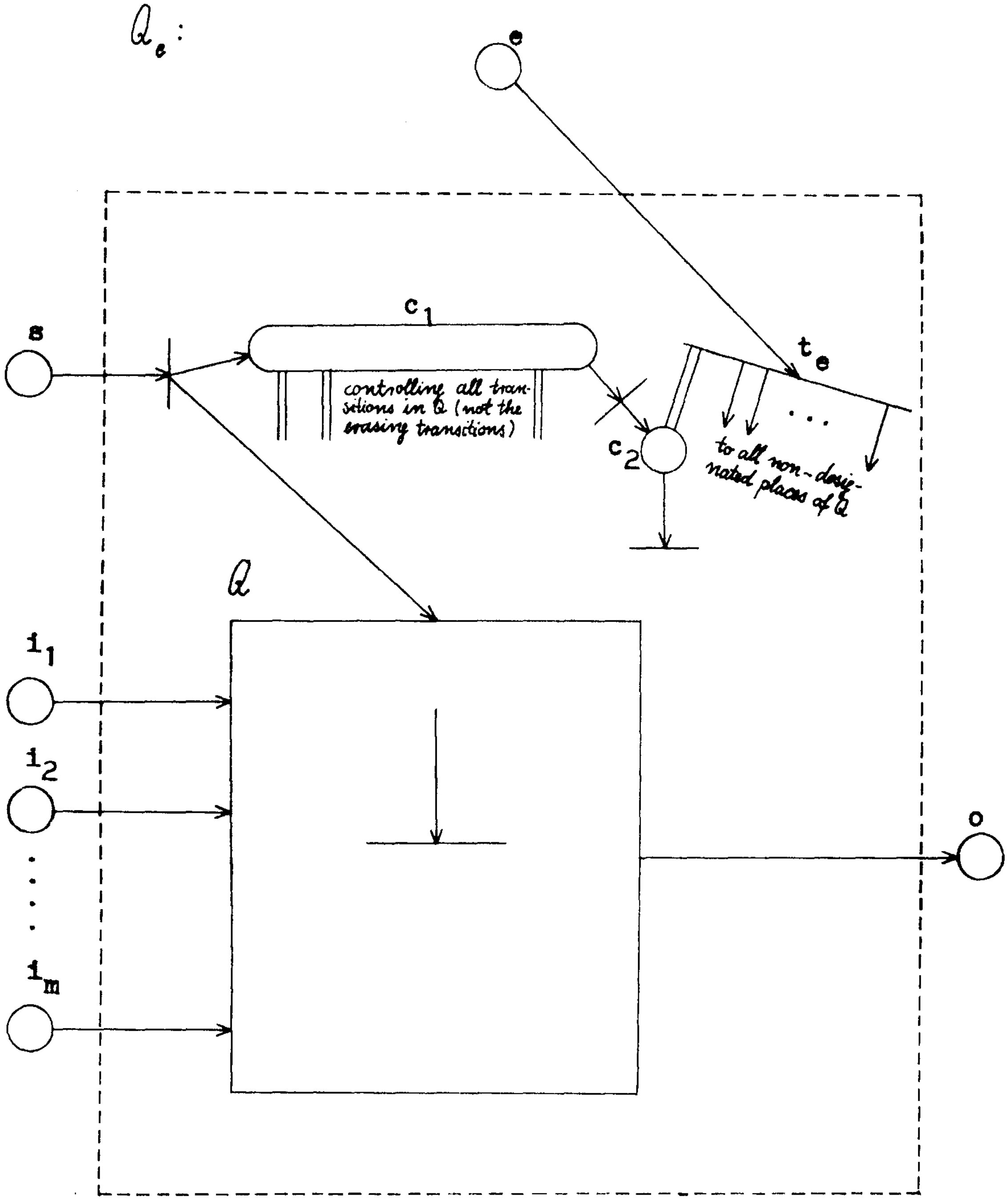
(ii) $(\forall N \in \mathbb{N}, \forall (n_1, \ldots, n_m) \in \{0, 1, \ldots, N\}^m)[R(\mathcal{Q}_\varepsilon, s \prod_{j=1}^{m} i_j^{n_j} e^N)$ is finite$]$;

(iii) $\text{size}(\mathcal{Q}_\varepsilon) = O(\text{size}(q) \cdot \log(\text{size}(q)))$.

Proof: (i) Construct a λ-WPNC $\mathcal{Q}$ for q, as in lemma 6, with designated places $s, i_1, \ldots, i_m, o$. To obtain $\mathcal{Q}_\varepsilon$, $\mathcal{Q}$ is extended as follows (figure 7):

a) an erasing transition is attached to each non-designated place u of $\mathcal{Q}$, i.e. a transition with input-place u and no output-place. This is indicated in the diagram by a transition

Figure 7:


Q_e:
e
c_1
s
controlling all transitions in Q (not the erasing transitions)
t_e
. . .
c_2
to all non-designated places of Q
Q
i_1
i_2
i_m
o

in the box for $\mathcal{Q}$ which has only an entering arc.

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 long as the one token received on c_1 from s remains there. As soon as it is transported to c_2 $\mathcal{Q}$ is frozen and cannot produce any more output. Now t_e may fire up to N times if there are initially N tokens on e, thus gathering at least N tokens an all non-designated places of $\mathcal{Q}_e$. Then, finally, erasing transitions can generate any number of tokens between zero and N on each of the non-designated places. Obviously, the erasing transitions don't affect the WPNC-property of $\mathcal{Q}$ as they only decrease the markings. By the construction of $\mathcal{Q}_e$, if t_e ever is enabled, the output on o is frozen, so $\mathcal{Q}_e$ is an e^N-WPNC for q for all $N \in \mathbb{N}$, and it generates any number of tokens up to at least N on the non-designated places. Hence, $\mathcal{Q}_e$ is a blurring WPNC for q.

(ii) For $(n_1,\dots,n_m) \in \{0,1,\dots,N\}^m$, let $M > 0$ be a bound on the markings of the non-designated places of $\mathcal{Q}$, reachable from $s\prod_{j=1}^{m} i_j^{n_j}$. Such a bound exists as the reachability set of $\mathcal{Q}$ at the given initial marking is finite. As $\mathcal{Q}$ is frozen when t_e is enabled, M + N is an upper bound on the markings of the non-designated places of $\mathcal{Q}_e$ which are reachable from $s\prod_{j=1}^{m} i_j^{n_j} e^N$. But this implies (ii).

(iii) follows in the same way as (iii) in theorem 4.

q.e.d.

VII. Reduction of BPI to FCP

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

Theorem 6:

BPI is polynomial-time-reducible to FCP.

Proof: Given a triple (p,q,n) with $n \in \mathbb{N}$ and $p,\ q \in \mathbb{N}[x_1,\ldots,x_m]$ for some $m \in \mathbb{N}$, we first construct the two Petri nets $\mathcal{P}_{\#}$ and $\mathcal{Q}_{\#}$ as indicated in figure 8a) and b). Each net contains a copy of the λ-WPNC $\mathcal{A}_n$ for A_n of theorem 2 and the bounded version $\mathcal{P}_p$ of a WPNC for p of theorem 4 resp. the blurring WPNC $\mathcal{Q}_q$ for q of theorem 5 (The start-place and the input-places of the latter have been primed in order to avoid confusion with the corresponding places of $\mathcal{A}_n$). The places e, c_1, and c_2 of $\mathcal{P}_{\#}$ are needed to match the corresponding places in $\mathcal{Q}_{\varepsilon}$ of $\mathcal{Q}_{\#}$ which don't get blurred in $\mathcal{Q}_{\varepsilon}$. W.l.g. we may assume that $\mathcal{P}_{\#}$ and $\mathcal{Q}_{\#}$ have the same number of places. If this is not the case <u>a priori</u> one can add further dummy places either to $\mathcal{P}_{\#}$ which aren't connected to any transition, or to $\mathcal{Q}_{\#}$ within $\mathcal{Q}_{\varepsilon}$, i.e. with erasing transitions attached to them and connected to the transition t_e of $\mathcal{Q}_{\varepsilon}$ (figure 7). The count-places in the two nets 'remember' the maximal input to the polynomial WPNC's. Clearly,

$\mathcal{P}_{\mathcal{A}}$ and $\mathcal{Q}_{\mathcal{A}}$ can be constructed from (p,q,n) in polynomial time (note that in definition 10 the unary length of n was used to define size(p,q,n)). To conclude the proof it suffices to show

<u>Lemma 8:</u>

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

<u>Proof of the lemma:</u> We assume that the two sets of places are ordered suitably, e.g. as follows: first s, then the places of the $\mathcal{A}_n$-copies (in the same order in both nets), then count, s', $i_1',\ldots,i_m'$, and o, followed by e, c_1, and c_2, and finally the remaining places in the polynomial WPNC's, in any order and independent from each other, including place b of $\mathcal{P}_{\mathcal{A}}$.

Assume first that $(p,q,n) \in BPI$ and consider some marking α of $\mathcal{P}_{\mathcal{A}}$ reachable from s which contains c', $n_1',\ldots,n_m'$, k tokens on the places count, $i_1',\ldots,i_m'$, o, respectively. As the places i_1', $\ldots,i_m'$ received c' tokens each from $\mathcal{A}_n$ and as those places cannot receive tokens from other places of $\mathcal{P}_p$, $\mathcal{P}_p$ used up $c' - n_j'$ tokens from the place i_j' (for $j=1,\ldots,m$). As $\mathcal{P}_p$ is a WPNC for p this implies $k \le p(c'-n_1',\ldots,c'-n_m')$. The marking on the non-designated places of $\mathcal{P}_p$ and on b, e, c_1, and c_2 is bounded by $c' + \max\{\|p\|, \|q\|\}$ because of the properties of $\mathcal{P}_p$ (theorem 4). Clearly, as the $\mathcal{A}_n$-components agree in $\mathcal{P}_{\mathcal{A}}$ and $\mathcal{Q}_{\mathcal{A}}$, the same marking is reachable in $\mathcal{Q}_{\mathcal{A}}$ as far as the places in $\mathcal{A}_n$ and s, s', and count are concerned. $\mathcal{Q}_e$ may now use $c' - n'_j$ tokens from each of the places $i'_1,\ldots,i'_m$ in $\mathcal{Q}_{\mathcal{A}}$, and output any number of tokens up to $q(c'-n'_1,\ldots,c'-n'_m)$ on o. But as $(p,q,n) \in \mathrm{BPI}$, $k \leq q(c'-n'_1,\ldots,c'-n'_m)$. In the final stage of $\mathcal{Q}_{\mathcal{A}}$'s computation $\mathcal{Q}_e$ can blur all its non-designated places up to at least $c' + \max\{\|p\|, \|q\|\}$ and then reach a marking with no tokens on the places e, c_1, and c_2 of $\mathcal{Q}_{\mathcal{A}}$, thus matching the given marking of $\mathcal{P}_{\mathcal{A}}$. As the reachability sets of $\mathcal{A}_n$ and $\mathcal{Q}_e$ are finite and there is no feedback from $\mathcal{Q}_e$ to $\mathcal{A}_n$, $\langle(\mathcal{P}_{\mathcal{A}}, s),(\mathcal{Q}_{\mathcal{A}}, s)\rangle \in \mathrm{FCP}$.

Now assume conversely that $\langle(\mathcal{P}_{\mathcal{A}}, s),(\mathcal{Q}_{\mathcal{A}}, s)\rangle \in \mathrm{FCP}$. Then <u>a fortiori</u> the projection of $R(\mathcal{P}_{\mathcal{A}}, s)$ on the places count, $i'_1,\ldots, i'_m$, o is contained in the corresponding projection of $R(\mathcal{Q}_{\mathcal{A}}, s)$. Consider such a submarking of $\mathcal{P}_{\mathcal{A}}$ with c', $n'_1,\ldots,n'_m$, k tokens, resp.. As $\mathcal{P}_b$ is a WPNC for p we have $0 \leq k \leq p(c'-n'_1,\ldots,c'-n'_m)$, and each of these values for k is possible. But, as the same submarking is reachable in $\mathcal{Q}_{\mathcal{A}}$, and as $\mathcal{Q}_e$ is a WPNC for q this implies that $q(c'-n'_1,\ldots,c'-n'_m) \geq k$ for each k with $0 \leq k \leq p(c'-n'_1,\ldots,c'-n'_m)$. As count may receive any number of tokens between 0 and A(n) we obtain, therefore, $p(c'-n'_1,\ldots,c'-n'_m) \leq q(c'-n'_1,\ldots,c'-n'_m)$ for all $c' \in \{0,1,\ldots,A(n)\}$ and all $(n'_1,\ldots, n'_m) \in \{0,1,\ldots,c'\}$. Hence, $(p,q,n) \in \mathrm{BPI}$.

q.e.d.

We can now at once derive our main result:

Theorem 7:

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

Proof: Each fast decision method for FCP would yield a fast algorithm for BPI via the reduction of theorem 6, and would thus contradict theorem 1.

q.e.d.

Corollary: The finite equality problem 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 for reachability sets to the equality problem [8, p. 122] preserves finiteness if the reachability sets of the two original Petri nets are finite. The reduction can be effected in polynomial time. Hence, the same argument as in the proof of the theorem applies.

q.e.d.

We remark that theorem 7 and its corollary actually do not depend heavily on the encoding used for Petri nets and polynomials as long as the ratio to the particular code chosen in this thesis is bounded by a primitive recursive function. In particular, we might use log(n) instead of n in definition 10 for size(p,q,n).

VIII. Conclusion and Open Problems

The Petri nets that were constructed in the course of this thesis to demonstrate the computational complexity of FCP and FEP had *a priori* finite reachability sets. Rackoff's upper bounds for the boundedness problem [16] show that the complexity of the containment decision procedures does not increase substantially in the general case when this information is not given. Thus, the non-primitive recursive lower bound for FCP and FEP is intrinsically due to the containment property for reachability sets which - as stated in the introduction - becomes undecidable when we consider the class of general Petri nets.

FCP and FEP are the first decision problems that are uncontrived and whose decision procedures are known to be non-primitive recursive (as far as one accepts Petri nets and vector addition systems as 'natural' concepts)(we consider BPI as contrived because the non-primitive recursive complexity is obtained by explicitly building in a non-primitive 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 are the reversible Petri nets. It is not difficult to see that the reachability set of a reversible Petri net is a semilinear set [9], and the results of Biryukov [4] and Taiclin [18] yield a constructive uniform method to obtain 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 non-primitive recursive. In [5], it has been shown that the reachability problem for reversible Petri nets is exponential space complete under log-space transformability.

The concepts used in this thesis do not apply to the reachability problem because WPNC's are not forced to produce some number of output-tokens, and no way is known to build 'strong' Petri net computers for polynomials restricted to a finite domain. In fact, the existence of unrestricted 'strong' Petri net computers for polynomials (or even only for the squaring function) would imply the undecidability of the general reachability problem, contradicting the recent results of Sacerdote and Tenney [17].

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 [10, 12]. It is known [12] that the reachability sets of persistent nets are semilinear, but no algorithm has been found so far to obtain these semilinear sets. In [10], 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, no nontrivial bounds are known for the finite reachability problem.

IX. References

[1] Ackermann, W.: Zum Hilbertschen Aufbau der reellen Zahlen. Math. Annalen, 99 (1928), pp. 118-133

[2] Adleman, L., Manders, K.: Computational Complexity of Decision Procedures for Polynomials. 16th IEEE Ann. Symp. on Foundations of Computer Science, 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., July 1973

[4] Biryukov, A. P.: Some Algorithmic Problems for Finitely Defined Commutative Semigroups. Siberian Mathematics Journal, Vol. 8, 1967, pp. 384-391

[5] Cardoza, E., Lipton, R. J., Meyer, A. R.: Exponential Space Complete Problems for Petri Nets and Commutative Semigroups. 8th Ann. ACM Symp. on Theory of Computing, May 1976, pp. 50-54

[6] Engeler, E.: Introduction to the Theory of Computation. Academic Press, New York, London, 1973

[7] Hack, M.: Decision Problems for Petri Nets and Vector Addition Systems. MAC-TM 59, Project MAC, M.I.T., 1975

[8] Hack, M.: Decidability Questions for Petri Nets. Ph.D. Thesis. TR 161, Laboratory for Computer Science, M.I.T., June 1976

[9] Jaffe, J.: Semilinear Sets and Applications. Master's Thesis. Department of Electrical Engineering and Computer Science, M.I.T., May 1977

[10] Jones, N. D., Landweber, L. H., Lien, Y.E.: Complexity of Some Problems in Petri Nets. To appear in Theoretical Computer Science

[11] Karp, R., Miller, R.: Parallel Program Schemata. JCSS, Vol. 3, 1969, pp. 147-195

[12] Landweber, L. H., Robertson, E. L.: Properties of Conflict Free and Persistent Petri Nets. TR 264, Computer Science Department, University of Wisconsin, 1975

[13] Matijasévič, Ju. V.: Enumerable Sets are Diophantine. Soviet Math. Dokl., Vol. 11, 1970, pp. 354-357

[14] Mayr, E. W.: Einige Sätze über Umformungen und verklemmungsfreie Führbarkeit bei bewerteten Petrinetzen. Diplomarbeit. Institut für Informatik, Technische Universität München, West Germany (in German)

[15] Meyer, A. R., Stockmeyer, L.: The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space. 13th IEEE Symp. on Switching and

Automata Theory, 1972, pp. 125-129

[16] Rackoff, Ch.: The Covering and Boundedness Problems for Vector Addition Systems. To appear in Theoretical Computer Science.

[17] Sacerdote, G. S., Tenney, R. L.: The Decidability of the Reachability Problem for Vector Addition Systems. 9th Ann. ACM Symp. on Theory of Computing, May 1977

[18] Taiclin, M. A.: Algorithmic Problems for Commutative Semigroups. Soviet Math. Dokl., Vol. 9, 1968, pp. 201-204

CS-TR Scanning Project
Document Control Form

Date : 11 / 3 / 95

Report # LCS-TR-181

Each of the following should be identified by a checkmark:
Originating Department:

- ☐ Artificial Intellegence Laboratory (AI)
- ☒ Laboratory for Computer Science (LCS)

Document Type:

- ☒ Technical Report (TR)
- ☐ Technical Memo (TM)
- ☐ Other: ____________

Document Information

Number of pages: 65 (70-images)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
- ☒ Single-sided or
- ☐ Double-sided

Intended to be printed as :
- ☐ Single-sided or
- ☒ Double-sided

Print type:
- ☐ Typewriter
- ☐ Offset Press
- ☐ Laser Print
- ☐ InkJet Printer
- ☒ Unknown
- ☐ Other: ____________

Check each if included with document:
- ☐ DOD Form
- ☐ Funding Agent Form
- ☐ Cover Page
- ☒ Spine
- ☐ Printers Notes
- ☐ Photo negatives
- ☐ Other: ____________

Page Data:

Blank Pages(by page number): ____________

Photographs/Tonal Material (by page number): ____________

Other (note description/page number):

Description :	Page Number:
IMAGE MAP:	(1-65) UNH'ED TITLE PAGE, 2-65
	(66-70) SCANCONTROL, SPINE, TRGT'S (3)

Scanning Agent Signoff:

Date Received: 11 / 3 / 95 Date Scanned: 11 / 17 / 95 Date Returned: 11 / 22 / 95

Scanning Agent Signature: Michael W. Cook