According theorem 3 we can construct an optimal solution of the problem (P), by constructing successively the columns of the optimal matrix.

Let us remark that in the case $m_{ik} = m_k$, $\forall i \in S_k$, we have $M_k = |S_k| / m_k$ and consequently in (35), (36) we get

$$x_{ik} = \frac{v(S_k)}{|S_k|}, \quad (i \in S_k), \quad C_k(X^k) = \frac{m_k\, v(S_k)}{|S_k|}. \tag{38}$$

Of course, to find an optimal solution of the problem (P) in more general cases is an open question.

REFERENCES

1. Dragan I. — *A network flow model for solving a cooperative game*. Techn. Rep. RWTH Aachen, 17/74. Symposia Mathematica, Proceedings of the Conference on Mathematical programming held in Rome, April 1974, vol. XIX, p. 215-227, 1976.
2. Dragan I. — *A two-dimensional Shapley value for cooperative n-person games*, to be published in Proceedings of the Institute of Mathematics Iasi.
3. Zangwill W. — *Nonlinear programming: a unified approach*. Prentice-Hall, Inc., Englewood Cliffs, N. J., 1969.

Received 4. V. 1977

Faculty of Mathematics
University of Iasi
Romania

Analele ştiinţifice ale Universităţii „Al. I. Cuza" din Iaşi
Tomul XXIII, S. I a. f. 2, 1977

GENERATION POLYNOMIAL RANGES BY ITERATIVE ONE-DIMENSIONAL ARRAYS

BY

ADRIAN SOIL

In this Note we present a way to construct an iterative one-dimensional array of finite-state automata which generates the power k of all positive integers, for a given positive integer k. Using this array, we construct another one which generates the range of a polynomial with given non-negative integers as coeficients for positive integer values of the argument.

1. Preliminaries. Let $N = \{1, 2, 3, \ldots, n, n+1 \ldots\}$ and $N^* = N \cup \{0\}$. We will denote an automaton (see [1]) by $A = (I, S, O, f, g)$. For $n \in N^*$, the set $\{j^n \mid j \in N\}$ will be called the set of perfect powers n.

1.1. Definition ([2]). *An infinite iterative one-dimensional array (IIAI) is a set of finite-state automata* $\{A_n \mid n \in N^*\}$ *with the following properties*:

i) for every n, $n \in N$, the automata are interconnected in such a way that the state of the automaton A_n at time t depends only on the states of the automata A_{n-1}, A_n, A_{n+1} at time $t-1$;

ii) all the automata of the array are identical with the exception of A_0 and, at time $t = 0$, they are all in the same initial state:

iii) the inputs and the outputs of the array are connected to A_0 and the state of A_0 at time t depends only on the input and on the states of A_0 and A_1 at time $t-1$.

1.2. Let $n \in N$: in section 2 we shall construct, with the finite-state automata A_0^n, A_1^n, ..., an *IIAI* which will have the oputput 1 at time t, if t is a perfect power n and 0, for others t.

1.3. A_0^n may easily be transformed, so that it recognizes the perfect powers n, in the following way; the output at time t is 1 if the input at time $t-1$ was 1 and t is a perfect power n and zero otherwise. Excepting this transformation the array works like the *IIAI* that generates the perfect powers n.

1.4. We can interconnect the automata of the array so that we realise 1.1 *i*) and *iii*):

— each A_k, $k > 0$, has an input line with two terminals and an output line with two terminals;

— the left input terminal and the left oputut terminal of A_k are connected respectively to the right output terminal and to the right input terminal of A_{k-1};

— the right input terminal and the right output terminal of A_k are connected respectively to the left output terminal and to the left input terminal of A_{k+1}, a.s.o.

The inputs and the outputs are ordered couples of signals. The output function of every A_k is defined such that it shows the state transitions at the respective time.

A_0 has the same properties for his right terminals while those of left are the input terminal and the output terminal of the array.

2. The generation of the perfect powers k. The construction of the array relies on the recurrence law between the power n of two succesive positive integers

$$(k+1)^n = k^n + \sum_{i=1}^{n} C_n^i k^{n-i}. \tag{2.1}$$

The array that generates the perfect powers n will be constructed using the generation way of the powers $n-1$, $n-2$,..., 2, for every $n \in N$, $n \geq 3$.

2.2. The construction of the array for $n = 1$ is trivial and we need A_0^1 only.

2.3. We shall construct now the array for $n = 2$. In this case the recurrence relation is

$$(k+1)^2 = k^2 + 2k + 1.$$

In fig. 1 the automata of the array are represented on the A_n^k-axis, the time on the t-axis and in the location (t, A_n^2) is represented the state of A_n^2 at time t. The operations performed by the array per unit time will be called step.

It results from fig. 1 how the array starts to work. Let's suppose that at time t_*, the output of the array was 1, hence t_0 was a perfect square. Then a „pulse" starts from A_0^2 and is propagated to the array's automata by the states A at the speed of one automaton by unit time. When the pulse arrives to the marked automaton (that will be described below) it is reflected and, with the same velocity, it is transmitted to A_0^2 through the states D. In the step following the reflection of the pluse, the next automaton in the array (in the sense of ascending n from 1.1) is marked and the mark of the automaton that has reflected the pulse is erased.

The mark is realised by the states B and is transmitted by the states C. If A_k^2 is the marked automaton by which the pulse was reflected, then this pulse will arrive to A_0^2 $2(k-1)$ times after it started from A_0^2. In fig. 1 it is shown how to mark the automata in the first steps so that, if t a perfect square, than the marked automaton (by which will be reflected the pulse started from A_0^2 at time t_*) will be $A^2_{\sqrt{t_*}+1}$. In this case the pulse started from A_0^2 will arrive back to A_0^2 after $2(\sqrt{t_*}+1-1) = 2\sqrt{t_*}$ steps, when the state of A_0^2 will be D; in the next step, that is $t_0 + 2\sqrt{t_*}+1$, the state of A_0^2 will be P and the array will have 1 at his output.

I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	B	B	B	B	B	
I	I	I	I	I	I	I	I	I	I	I	I	I	I	B	B	B	B	B	B	B	C	T	T	T	T	T	
I	I	I	I	I	I	I	I	B	B	B	B	B	C	T	T	T	T	T	T	D	T	T	T	T	T	T	
I	I	I	B	B	B	B	C	T	T	T	T	D	T	T	T	T	T	T	A	T	D	T	T	T	T	T	
I	I	C	T	T	T	D	T	T	T	T	A	T	D	T	T	T	T	A	T	T	T	D	T	T	T	T	
I	H	T	T	T	A	T	D	T	T	A	T	T	T	D	T	T	A	T	T	T	T	T	D	T	T	A	
I	Q	Q	D	P	T	T	T	D	P	T	T	T	T	T	D	P	T	T	T	T	T	T	T	D	P	T	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	t

Fig. 1

The automata of the array are given by:

$M_0^2 = (I_0^2, S_0^2, O_0^2, f_0^2, g_0^2)$ where $I_0^2 = \{(0, i), (0, p), (0, q), (0, t), (0, d)\}$, $S_0^2 = \{I, P, Q, T, D\}$, $O_0^2 = \{(0, i), (0, p), (0, q), (0, t), (1, d)\}$ (the symbols i, p, q, t, d correspond to the states I, P, Q, T, D) and f_0^2 and g_0^2 are defined as it results from fig. 1 and 1.4 (for example $f_0^2(I, (0, i)) = Q$, $f_0^2(T, (0, d)) = T$, $g_0^2(D, (0, t)) = (1, d)$, $g_0^2(P, (0, t)) = (0, p)$);

$M_k^2 = (I_k^2, S_k^2, O_k^2, f_k^2, g_k^2)$ where $I_k^2 = O_k^2 = \{p, i, h, t, d, a, b, c, q\}^2$, $S_k^2 = \{I, H, T, D, A, B, C\}$ and f_k^2 and g_k^2 are defined according to fig. 1 and 1.4, this for every $k > 0$, $k \in N$.

2.4. Theorem. *For every $n \in N$, we can construct an IIAI which generates the set of perfect powers n.*

Proof. The case $n = 1$ was presented at 2.2 and the case $n = 2$ at 2.3.

Let's suppose that we have constructed the arrays which generate the perfect powers m, $0 < m \leq n$, $m \in N$. We shall construct the array which generates the perfect power $n+1$. We shall use (2.1). The construction of the array is made taking into account the rules which follows.

— At time $t = 1$ a perfect power $n+1$ is generated and A_1^{n+1} is marked.

— A step after the generation of a perfect power $n+1$ of i, A_0^{n+1} "orders" the successive generation of C_{n+1}^0 perfect powers n of i, C_{n+1}^2 perfect powers $n-1$ of i,..., C_{n+1}^{n-h+1} perfect powers h of i,..., C_{n+1}^{n-1} perfect squares of i and afterwards it emits a pulse which arrives again to

A_0^{n+1} after C_{n+1}^n steps; in the next step we shall have 1 at the output of the array since we have obtained the perfect power $n+1$ for $i+1$.

— For the generation of i^{n+1}, the marked automaton will be A_{i-1}^{n+1}.

— If A_h^{n+1} is marked, then, if t is the time at which the pulse that counts the C_{n+1}^n steps arrives "completely" at A_h^{n+1} (we shall see in the sequel what this signifies), at the next time $t+1$ this pulse is reflected and at time $t+2$, A_{h+1}^{n+1} is marked and the mark of A_h^{n+1} is erased. We shall generate i^h, C_{n+1}^{n-h+1} times, using C_{n+1}^{n-h-1} distinct groups of states, similar to those used for the generation of i^h, $h < n+1$. Each group releases, when it is ended, the generation of the next group, and when the last group is ended, the generation of i^{h+1}, C_{n+1}^{n-h} times, is released. We remark that the generation of $i^n, i^{n-1}, \ldots, i^3$ is made using the generation way of i^2 and the marked automaton A_i^{n+1}. For the generation of C_{n+1}^n $i = (n+1)\,i$ we use an "incident wave", started from A_0^{n+1} when the „calculation" of $C_{n+1}^{n-1}\, i^2$ is finished. This wave "waits" n steps at each automaton (i.e. the wave is arrived "completely" at the automaton), is reflected by the marked automaton A_{i-1}^{n+2} (but it is not transmited to this) and it becomes a wave that goes to A_0^{n+1} with the speed of one automaton by unit time.

Therefore the wave "waits" at each automaton $A_0^{n+1}, A_1^{n+1}, \ldots, A_{i-1}^{n+1}$ exactly n steps, hence from the moment of his generation by A_0^{n+1} until his return to A_0^{n+1} there are $(n+1)\,i$ steps.

In the next step the output of the array will be 1. If for the construction of A_i^h, $i > 0$, $h \geqslant 2$, we used S_h states. then we need S_{n+1} states for the construction of A_i^{n+1}, where

$$S_{n+1} = \sum_{k=1}^{n} C_{n+1}^k \, S_k \, .$$

If for the construction of A_0^h, $h \geqslant 2$ we used S_h^0 states, then for the construction of A_0^{n+1} we need S_{n+1}^0 states, where

$$S_{n+1}^0 = \sum_{k=1}^{n} C_{n+1}^k \, S_k^0 \, .$$

3. The generation of the range of polynomials with coeficients from N, for natural positive values of the argument. The generation of monomials ck^n, where $c, n, k \in N$, c, n fixed, can be realised using the construction of the array for k^n and repeting this construction c times (in the sense of the proof of 2.4 — similarly to the case $C_{n+1}^{n-i+1} k^i$).

Let $P(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + a_n$ with $a_i \in N^*$, $i = \overline{0, n}$ $a_0 \neq 0$, $x \in N^*$, a given polynomial. Using the methods given in 2.4 and above, we may construct $n+1$ $HA1$-s $R_0, R_1, \ldots, R_n$ which generate respectively $a_n, a_{n-1} x, \ldots, a_0 x^n$. We interconnect this arrays so that the input terminal of R_0 is the input terminal for the new array. The output terminal of R_i is connected to the input terminal of R_{i+1}, $i = \overline{0, n-1}$, and the output terminal of R_n is the output terminal of the new array. When R_j ends the generation of $a_{n-j} x^j$, it releases the generation of $a_{n-j+1} x^{j+1}$ by

R_{j+1}, therefore the arrays do not work simultaneously for the calculation of the same value $P(a)$.

To the array constructed here will be attached an automaton M_0, to which are connected the terminals of the array composed by $R_0, R_1, \ldots, R_n$ and which realises the "communications" of the array. This array generates the values of $P(x)$, for $x \in N^*$.

REFERENCES

1. Creangă, I., Reischer, C., Simovici, D. — *Teoria automatelor*. Ed. Junimea Iaşi, 1973.
2. Fischer, P. C. — *The Generation of primes by an iterative one-dimensional real-time array*. J. ACM 12 (1965), 388—394.

Received 12. XII. 1976

Computer Center,
University of I a ş i
Romania