

# Algorytmy cz. II. Program który nie liczy

dr Andrzej SKOWRON

Stwierdziłmy poprzednio, że do sprecyzowania, czym jest algorytm, niezbędne jest określenie jego sterowania. Sterowanie algorytmu scharakteryzowane jest przez zbiór czynności elementarnych (nazywanych rozkazami), które to sterowanie „umie” wykonać, oraz przez zdolność do wykonywania tych czynności w kolejności określonej przez sieć działań.

Urządzenie lub człowiek pełniący rolę sterowania w przypadku konkretnego algorytmu, często jest w stanie pełnić rolę sterowania dla wielu innych algorytmów. Mając informacje o tym, jakie czynności elementarne umie wykonać obiekt pełniący rolę sterowania oraz jak należy budować sieci działań dla tego obiektu, stawiamy pytanie: jak konstruować sieci działań wyznaczające algorytmy o zadanych własnościach?

Nie dysponujemy niestety metodami matematycznymi, pozwalającymi w pełni rozwiązywać tego typu zadania. Dlatego umiejętność konstruowania sieci działań wyznaczających, przy określonych własnościach sterowania, algorytmy o żądanych własnościach, nazywa się czasem sztuką programowania. Rozważmy przykład algorytmu rozpoznającego, czy podany napis jest poprawnym zapisem w układzie rzymskim liczby całkowitej dodatniej, nie większej od 20. Załóżmy, że dysponujemy miejscami o nazwach  $j, z$ , których zawartościami mogą być liczby 1, ..., 5, miejscem o nazwie  $y$ , którego zawartością mogą być liczby 0, 1 oraz miejscami o nazwach  $x[1], \dots, x[5]$ , których zawartościami mogą być symbole  $I, V, X, *$ . Zgodnie z umową przyjętą poprzednio  $j, z, y, x[1], \dots, x[5]$  oznaczają odpowiednio zawartości miejsc  $j, z, y, x[1], \dots, x[5]$ .

Jako stany pamięci przyjmujemy ciągi  $(j, z, y, x[1], \dots, x[5])$ .

Założmy, że dysponujemy sterowaniem, które „umie” wykonać tylko następujące czynności elementarne (rozказы):

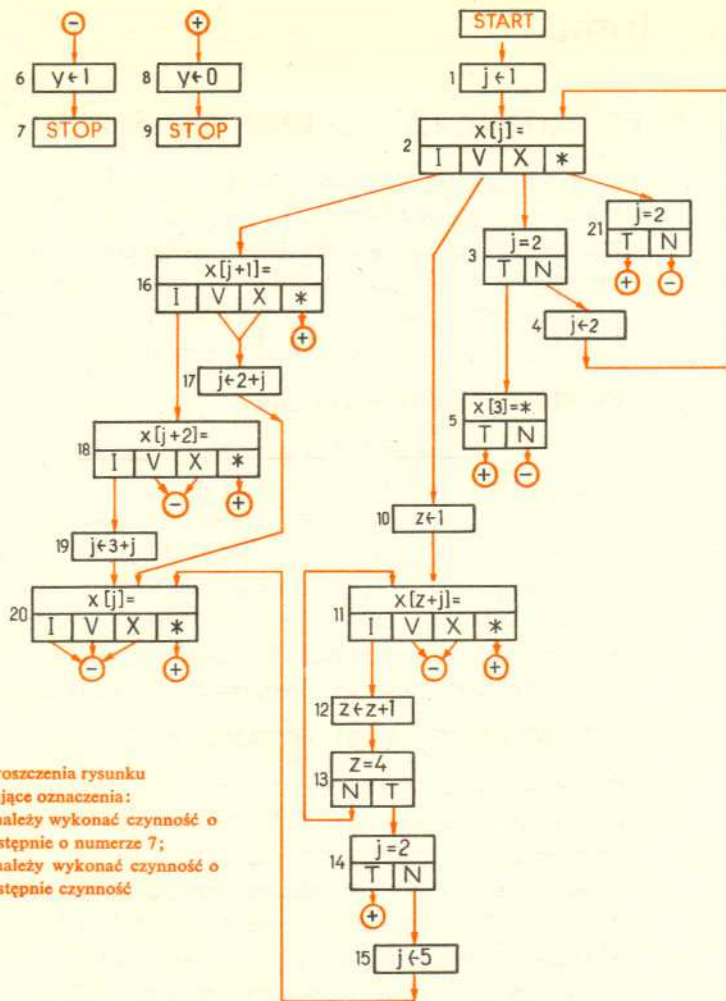
1. Jeśli  $p$  jest liczbą całkowitą i  $0 \leq p \leq 5$ , to do zbioru rozkazów należy czynność polegająca na zapisaniu liczby  $p$  w miejscu o nazwie  $j$ .
2. Czynność polegająca na sprawdzeniu, który z symboli  $I, V, X, *$  jest zapisany w miejscu o nazwie  $x[k]$ , gdzie  $k$  jest równe sumie zawartości miejsc  $j, z$ , gdy  $j+z \leq 5$  oraz 1 w pozostałych przypadkach.
3. Jeśli  $p$  jest liczbą całkowitą i  $0 \leq p \leq 2$ , to do zbioru rozkazów należy czynność polegająca na sprawdzeniu, który z symboli  $I, V, X, *$  jest zapisany w miejscu o nazwie  $x[k]$ , gdzie  $k$  jest równe sumie zawartości miejsca  $j$  oraz liczby  $p$ , gdy  $j+p \leq 5$  oraz 1 w pozostałych przypadkach.
4. Czynność polegająca na sprawdzeniu, czy zawartość miejsca  $j$  jest równa 2.
5. Czynność polegająca na sprawdzeniu, czy zawartość miejsca  $z$  jest równa 4.
6. Czynność polegająca na sprawdzeniu, czy zawartość miejsca  $x[3]$  jest równa  $*$ .
7. Czynność polegająca na zapisaniu 0 w miejscu  $y$ .
8. Czynność polegająca na zapisaniu 1 w miejscu  $y$ .
9. Czynność polegająca na zapisaniu 1 w miejscu  $z$ .
10. Czynność polegająca na zapisaniu liczby równej zawartości  $z$  zwiększonej o 1 w miejscu  $z$ .
11. Jeśli  $p = 2$  lub  $p = 3$ , to do zbioru rozkazów należy czynność polegająca na zapisaniu liczby równej sumie zawartości  $j$  i liczby  $p$  w miejscu  $j$ .
12. Czynność nie powodująca zmiany zawartości żadnego miejsca. (Po jej wykonaniu sterowanie nie wyznacza żadnej następnej czynności do wykonania).

Przyjmujemy, że obiekt, który pełni rolę sterowania „umie” wykonać czynności elementarne w kolejności określonej przez sieci działań.

Określmy teraz dokładniej zadanie, które chcemy rozwiązać. Niech  $c = (j, z, y, x[1], \dots, x[5])$  będzie dowolnym stanem pamięci. Ciąg powstały z ciągu  $(x[1], \dots, x[5])$ , po odrzuceniu wszystkich wyrazów, poczynając od pierwszego wyrazu równego  $*$ , oznaczmy przez  $\zeta$ . Należy podać sieć działań, taką, aby w przypadku, gdy  $\zeta$  jest poprawnym zapisem w układzie rzymskim liczby całkowitej dodatniej nie większej od 20, zawartość miejsca  $y$ , po zakończeniu obliczenia o stanie początkowym  $c$ , była równa 0; w przeciwnym wypadku aby była ona równa 1. Przykład takiej sieci działań podano na rysunku.

Oznaczenia czynności elementarnych

$j \leftarrow p$
$(0 \leq p \leq 5)$
$x[j+z] =$
$I \quad V \quad X \quad *$
$x[j+p] =$
$I \quad V \quad X \quad *$
$0 \leq p \leq 2$
$j = 2$
$T \quad N$
$z = 4$
$T \quad N$
$x[3] = *$
$T \quad N$
$y \leftarrow 0$
$y \leftarrow 1$
$z \leftarrow 1$
$z \leftarrow z+1$
$j \leftarrow j+p$
$(p = 2, 3)$
STOP



Uwaga: Dla uproszczenia rysunku przyjęto następujące oznaczenia:  
 — oznacza, że należy wykonać czynność o numerze 6, a następnie o numerze 7;  
 + oznacza, że należy wykonać czynność o numerze 8, a następnie czynność o numerze 9.

Sieć działań algorytmu rozpoznającego, czy podany napis jest poprawnym zapisem w układzie rzymskim liczby całkowitej, dodatniej, nie większej od 20.

W tabelach 1 i 2 podano, jak będą zmieniały się zawartości miejsc pamięci w trakcie realizowania przez sterowanie czynności zgodnie z siecią działań podaną na rysunku, gdy początkowy stan pamięci jest równy odpowiednio (1 1 1 X V I I \*) i (1 4 0 V I I I I)

Tabela 1

Numer czynności wykonywanej	Zawartości miejsc po wykonaniu kolejnej czynności							Numer następnej czynności	
	j	z	y	x [1]	x [2]	x [3]	x [4]		x [5]
1	1	1	1	X	V	I	I	*	2
2	1	1	1	X	V	I	I	*	3
3	1	1	1	X	V	I	I	*	4
4	2	1	1	X	V	I	I	*	2
2	2	1	1	X	V	I	I	*	10
10	2	1	1	X	V	I	I	*	11
11	2	1	1	X	V	I	I	*	12
12	2	2	1	X	V	I	I	*	13
13	2	2	1	X	V	I	I	*	11
11	2	2	1	X	V	I	I	*	12
12	2	3	1	X	V	I	I	*	13
13	2	3	1	X	V	I	I	*	11
11	2	3	1	X	V	I	I	*	8
8	2	3	0	X	V	I	I	*	9
9	2	3	0	X	V	I	I	*	—

Tabela 2

Numer czynności wykonywanej	Zawartości miejsc po wykonaniu kolejnej czynności							Numer następnej czynności	
	j	z	y	x [1]	x [2]	x [3]	x [4]		x [5]
1	1	4	0	V	I	I	I	I	2
2	1	4	0	V	I	I	I	I	10
10	1	1	0	V	I	I	I	I	11
11	1	1	0	V	I	I	I	I	12
12	1	2	0	V	I	I	I	I	13
13	1	2	0	V	I	I	I	I	11
11	1	2	0	V	I	I	I	I	12
12	1	3	0	V	I	I	I	I	13
13	1	3	0	V	I	I	I	I	11
11	1	3	0	V	I	I	I	I	12
12	1	4	0	V	I	I	I	I	13
13	1	4	0	V	I	I	I	I	14
14	1	4	0	V	I	I	I	I	15
15	5	4	0	V	I	I	I	I	20
20	5	4	0	V	I	I	I	I	6
6	5	4	1	V	I	I	I	I	7
7	5	4	1	V	I	I	I	I	—