ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1974

Seria: Automatyka, z. 27 Nr kol. 396

Alicja Wakulicz-Deja

Instytut Kompleksowych Systemów Sterowania

PEWIEN OPIS FORMALNY MASZYNY TURINGA I AUTOMATU SKOŃCZONEGO

Streszczenie. W artykule przedstawiony jest opis formalny Maszyny Turinga i automatu skończonego. Posłużono się tutaj teorio-mnogościowym ujęciem maszyny cyfrowej. Opis ogólny tych typów maszyn jak i przedstawione szczegółowe przykłady pokazują, że teoria zbiorów pozwala stosunkowo łatwo opisać pewne typy maszyn cyfrowych - jak również, że opis ten prowadzi do powstania pewnego, wydaje się dość efektywnego, języka sformalizowanego.

Wstęp

W ostatnich latach coraz częściej odczuwa się konieczność ogólnego, ścisłego matematycznie zapisu pewnych zjawisk, układów automatyki, a w końcu i opisu maszyn cyfrowych. Coraz częściej do budowy konkretnego układu, do budowy konkretnej maszyny, do analizy tych układów potrzebny jest model matematyczny. Aby przedstawić dany model w sposób ścisły potrzebny jest pewien zbiór symboli - poszukujemy dziedzin matematyki, w których opis danego układu jest najbardziej przejrzysty.

Język opisujący model matematyczny, dający możliwość opisu pewnej klasy układów, jest już pewnym językiem sformalizowanym, dobrym narzędziem syntezy i analizy układu.

W roku 1964 Elgot zwrócił uwagę na fakt łatwego zapisu maszyny cyfrowej w teorii zbiorów. Po dokładnej analizie okazuje się, że model matematyczny maszyny cyfrowej, przedstawiony w teorii zbiorów, jest analogiczny z modelem matematycznym maszyny cyfrowej stworzonym w oparciu o teorię grafów funkcyjnych. Obie formy opisu prowadzą do powstania dość przejrzystego języka sformalizowanego. Ponieważ pewne rodzaje

maszyn cyfrowych daje się prosto i przejrzyście opisać w teorii zbiorów, bez wprowadzania grafów, przedstawmy modele dwóch rodzajów maszyn cyfrowych: Maszyny Turinga i Automatu Skończonego.

Pojęcia wprowadzające

Zakładając, że pojęcie zbioru i ogólne działania na zbiorach są znane, przypomnijmy tylko pewne oznaczenia i pojęcia, które będą używane w dalszej części pracy.

1) $a \in A$ - oznacza, że element a należy do zbioru A.

2) $A \subset B$ - oznacza, że zbiór A, jest podzbiorem zbioru B.
Zbiór A jest podzbiorem zbioru B, jeżeli każdy element zbioru A jest elementem zbioru B.

3) $\langle a,b \rangle$ - oznacza parę uporządkowaną. Jest to para uporządkowana o poprzedniku a i następniku b.
Parę $\langle a, b \rangle$ uważamy za różną od pary $\langle b, a \rangle$ (chyba, że $a = b$).

4) $X \times Y$ - oznacza iloczyn kartezjański dwóch zbiorów X i Y.
Iloczynem kartezjańskim dwóch zbiorów X i Y nazywamy zbiór wszystkich par uporządkowanych $\langle x, y \rangle$, gdzie $x \in X$ i $y \in Y$, zatem

$$\left[\langle x,y \rangle \in X \times Y \right] = (x \in X)(y \in Y).$$

Przykładem iloczynu kartezjańskiego jest płaszczyzna liczb zespolonych $\mathcal{E} \times \mathcal{E}$, gdzie $\mathcal{E}$ jest liczbą rzeczywistą. (Liczba zespolona, to para uporządkowana złożona z dwóch liczb rzeczywistych). Innym przykładem iloczynu kartezjańskiego jest powierzchnia walca - jest to iloczyn kartezjański okręgu (obwodu koła), przez odcinek będący wysokością walca.

5) $A^{(m)}$ - oznacza $\overbrace{A \times A \times \ldots\ldots \times A}^{m \text{ razy}}$.
Jest to iloczyn kartezjański m-wymiarowy.

6) $C = B^A$ - oznacza zbiór wszystkich funkcji przekształcających elementy zbioru A, w elementy zbioru B. Jest to zbiór funkcji,

których argumenty są elementami zbióru A, wartości zaś elementami zbioru B.

Przez funkcję, której argumenty przebiegają zbiór A, wartości zaś należą do zbioru B, rozumiemy każdy podzbiór iloczynu kartezjańskiego A x B, o tej własności, że dla każdego $a \in A$ istnieje jedno i tylko jedno $b \in B$, takie, że para uporządkowana $\langle a,b \rangle$ jest elementem tego podzbioru. Jeżeli wartości wypełniają cały zbiór B, to funkcję taką nazywamy przekształceniem zbioru A na zbiór B.

I. Model matematyczny Maszyny Turinga

1. Krótki opis ogólny M.T. i sposoby przedstawiania na maszynach realnych

Maszyna Turinga składa się z nieskończonej taśmy i głowicy. Na taśmie mogą być dane i program. Taką maszynę nazywamy maszyną uniwersalną.

Program może być również umieszczony oddzielnie, na taśmie stanowiącej pamięć równoległą maszyny - zewnętrznie względem maszyny (maszyny specjalizowane).

Głowica porusza się nad taśmą i zmienia swój stan przy zapisie i odczycie. W zależności od konstrukcji głowica może przyjmować pewną ilość stanów q. Najwłaściwszym przedstawieniem Maszyny Turinga jest maszyna uniwersalna, która z kolei ma długi czas działania ze względu na konieczność wykonywania przez głowicę ruchów oscylacyjnych między programem a danymi.

Ogólnie działanie Maszyny Turinga polega na zapisie symbolu, przeglądzie zapisanych symboli i zmianie stanu wewnętrznego urządzenia realizującego obliczenia. Nieskończona taśma maszyny Turinga oznacza nieskończoną pamięć maszyny. Nie potrafimy jednak zbudować żadnej maszyny realnej posiadającej nieskończoną pamięć. W związku z tym zachodzi konieczność przybliżania pamięci nieskończonej, jakimś rodzajem pamięci skończonej. Możemy uważać, że pamięć wewnętrzną maszyny przedłuża pamięć zewnętrzna, która składa się z nieskończonej ilości taśm wymien-

nych, lub dysków. Taką pamięć można uważać za nieskończoną. Podobnie mogę np. taśmę skończoną dzielić na segmenty przyjmując, że ilość tych segmentów może być nieskończona. Taka taśma będzie również odzwierciedlać dobrze nieskończoną taśmę maszyny Turinga.

2. Opis formalny Maszyny Turinga

W zależności od sposobu przedstawienia taśmy nieskończonej wystąpią dwa rodzaje interpretacji symboli. Sposób I określa przedstawienie taśmy nieskończonej przez podział taśmy skończonej na nieskończoną ilość segmentów, sposób II zakłada nieskończony zbiór taśm skończonych.

Symbol	Znaczenie w sposobie I	Znaczenie w sposobie II
A	Zbiór segmentów taśmy	Zbiór taśm
B	Zbiór symboli (danych) w segmencie	Zbiór symboli (danych) na taśmie
C	Zbiór segmentów zawierających instrukcje	Zbiór taśm zawierających instrukcje
I	Zbiór instrukcji	Zbiór instrukcji
K	Zbiór funkcji przyporządkowujących segmentom dane	Zbiór funkcji przyporządkowujących taśmom dane

$A = \{a_0, a_1 \ldots\ldots a_n, \ldots\}$ - zbiór nieskończony.

$B = \{b_0, b_1 \ldots\ldots b_n, \ldots\}$ - zbiór nieskończony.

$C = \{c_0, c_1 \ldots\ldots c_n, \ldots\}$ - zbiór nieskończony.

$C = \{0,1\}^z$ - przy określeniu alfabetu dwusymbolowego

lub

$C = Z$ - zbiór liczb naturalnych.

$I = \{i\}^z$ - stan taśm zawierających instrukcje.

$K = B^A$ - stan segmentów lub taśm zawierających dane.

$\langle k,z \rangle \in \Sigma$ - stan ze zbioru stanów maszyny.

$\sigma_{k+1}(a,z) = F(K_k, z_{k+1})$ - stan maszyny w chwili $(k+1)$ $|\sigma \in \Sigma$

C - moglibyśmy określić jako podzbiór A, wtedy jednak opis działania maszyny jest utrudniony.

Zakładamy zatem C jako zbiór wyróżnionych segmentów lub taśm rozdzielnych ze zbiorem A.

Dla taśm nieskończonych w jedną stronę możemy określić początek taśmy

$K(a,b)_o = 0$ - znak wyróżniony

i

$f(z)_o = 0$ - znak wyróżniony

lub dla jednej taśmy

$F(a,b,z)_o = 0.$

Zapiszmy teraz sposób wywoływania informacji i ruch maszyny

$$f(a,b,z)_1 = z_1,$$

$$f_{k+1}(a,b,z) = F(f_k, a_{k+1}),$$

w chwili $k+1 = n$

$$f_n(a,b,z) = z_2,$$

$$f_{n+1}(a,b,z) = F(f_n, a_{n+1}),$$

w chwili $n+1 = m$

$$f_m(a,b,z) = z_3$$

$$\vdots \qquad \vdots$$

itd.

w chwili $m = \xi + 1$

$$f_{\xi+1}(a,b,z) = F(f_\xi, a_{\xi+1}) = f_\xi(a,b,z) \text{ - stop.}$$

W chwili 1 jest wywoływana 1 taśma instrukcji. W każdej chwili stan maszyny określony jest przez stan w chwili poprzedniej i aktualny segment (taśmę) danych. W chwili k+1 równej n zostaje wywołana następna taśma instrukcji, dalej znów obliczenia w obrębie danych itd. Jest to ruch postępujący lub malejący w zależności od uporządkowania k, n, m i z_1, z_2, z_3 ..... W chwili $\xi+1$ stan maszyny nie ulega zmianie jest określony przez funkcję f_ξ - maszyna stoi.

Wywoływanie instrukcji jest liniowe. Można tak poumieszczać taśmy danych i instrukcji, aby obliczenia były wykonywane liniowo, dane są odpowiednie do wywołanej instrukcji. Jest to szczególnie przydatne dla taśm oddzielnych, bo zawsze możemy założyć taśmę z aktualnie potrzebnymi danymi.

W przypadku większego ciągu danych (niż potrzebny do jednej instrukcji), można przyjąć zapis

$$f_{k+1}(a,b,z) = F(f_k, a_{k+1}),$$

aż do chwili k+1 równej m

$$f_{m+1}(a,b,z) = f_m(a,b,z)$$

$$f_{m+2}(a,b,z) = z_2 \text{ (itd.)},$$

zatem po wykorzystaniu przez instrukcję danych jej potrzebnych maszyna staje, w chwili następnej zostaje wywołana nowa instrukcja i znów operacja wykonuje się na części danych potrzebnych do danej instrukcji i następuje stop maszyny. Jest to dogodne, gdy mamy oddzielną taśmę danych, bez wyodrębnionych segmentów dla kolejnych instrukcji.

Gdy chcemy zmienić kolejność wywoływania danych, to przyjmujemy zapis

$$\begin{aligned} f_{n+1}(a,b,z) &= \varphi(a,b,z), \\ \varphi_{k+1}(a,b,z) &= F(\varphi_k, z_{k+1}). \end{aligned} \tag{1}$$

Funkcja φ działa tak jak funkcja f, ale dla innego przyporządkowania $A \rightarrow B$

$$\varphi(a_i) = b_{n-i}$$

- gdy i rośnie mamy przyporządkowanie odwrotne.
Można założyć przyporządkowanie dowolne

$$\varphi(a_\xi) = b_\gamma,$$

ξ, γ - dowolne liczby naturalne.

Zapis (1) oznacza, że w chwili n+1 funkcja f przechodzi w funkcję φ, która oznacza pracę na danych o zmienionym uporządkowaniu. Funkcja φ oznacza, że mamy teraz inną numerację miejsc danych (segmentów lub taśm) i wywołujemy np. piątą taśmę, potem czwartą, trzecią, drugą itd. lub dowolnie: piątą, pierwszą, trzecią..... w zależności od funkcji φ. Gdy chcemy wrócić do przyporządkowania kolejnego, to wracamy do funkcji f.

$$f(a_i) = b_i$$

W chwili k+1 = m

$$\varphi_m(a) = f(a)$$

i dalej

$$f_{k+1}(a,b,z) = F(f_k, z_{k+1}).$$

W maszynie realnej taka zmiana przyporządkowania może być dokonana podprogramem. W taśmach możemy uważać, że taśma oznaczona numerem np. 5, ma teraz numer 1, a w segmentach odpowiadałoby to przejściu w tył lub w przód o skończoną ilość segmentów i odczytaniu danych np. przed piątym segmentem instrukcji.

3. Szczegółowe przykłady działania maszyny Turinga

3.1. Przykłady dla szczególnego przedstawienia Maszyny Turinga na maszynie realnej

Przykład I. Maszyna dająca słowo podwójne

$M(S) = SS$ np. abc - abc-abc

$A = \{a_0, a_1, a_2, \ldots\}$ - zbiór segmentów taśmy zawierających dane, o numerach 1.3.5.....

$B = \{b_0, b_1, b_2, \ldots\}$ - zbiór danych $S_1, S_2, \ldots S_n$ - słowo na taśmie.

$C = Z$ - zbiór segmentów wyróżnionych zawierających instrukcje o numerach 2,4,6,.....

$I = \{i\}^z$ - zbiór instrukcji.

$K = B^A$ - stan segmentów z danymi.

$\langle k,z \rangle \in \Sigma$ - stan ze zbioru stanów maszyny.

$\sigma_{k+1}(a,z) = F(K_k, z_{k+1})$ - stan w chwili k+1 ($\sigma \in \Sigma$).

$F(a,b,z)_0 = 0$ - początek taśmy.

$f(a_1) = S_1$,

$f(a_2) = z_1$,

$f(a_3) = S_2$,

$f(a_4) = z_2$,

. .

. .

. .

. .

$f(a_{2n}) = S_n$.

Chcemy otrzymać

$f(a_{2n+1}) = S_1$

$f(a_{3n+1}) = S_1$

$f(a_{2n+2}) = S_2$

$\vdots \qquad \vdots$

Instrukcje zatem powinny być typu, z_i: wpisz w miejsce

$$\begin{matrix} a_{2n+i} \searrow \\ \qquad Si, \\ a_{3n+i} \nearrow \end{matrix}$$

czyli

$$Z_i = S_i^{\begin{matrix} a_{2n+i} \\ a_{3n+i} \end{matrix}}$$

$i = 1,2,\ldots\ldots n+1$ - parametr zmieniający się od 1 do n+1

$$f_i(a,b,z) = F(S_i, z_i, a_{m+i})$$

gdzie

$$m = 2n, 3n,$$

ogólnie

$$f(a) = S, \qquad K = S^A.$$

Dla $i = n+1$

$f_{n+i}(a,b,z) = F(S_{n+1}, z_{n+1}, a_{m+n+1}) = f_n(a,b,z)$ - stop maszyny.

Przykład 2. Maszyna dająca słowo odwrócone

$M(S) = S^{-1}$ np. abc $\longrightarrow$ cba

$f(a_1) = S_1$,

$f(a_2) = z_1$,

$\vdots$

$f(a_{2n}) = S_n$.

Chcemy otrzymać

$f(a_{2n+1}) = S_n$,

$f(a_{2n+2}) = S_{n-1}$,

$\vdots$

$f(a_{3n}) = S_1$.

Instrukcje zatem muszą być postaci

$$z_i: \text{ wpis w miejsce } a_{3n+1-i} \longrightarrow S_i$$

czyli

$$Z_i = S_i^{a_{3n+1-i}}$$

dla i równego 1,2,3,..... n,n+1.

Ogólnie

$$f_i(a,b,z) = F(S_i, z_i, a_{3n+1-i}).$$

W chwili

$$i = n+1$$

$$f_{n+1}(a,b,z) = F(S_{n+1}, z_{n+1}, a_{3n}) = f_n(a,b,z) - \text{stop}.$$

Przykład 3. Maszyna podwajająca litery słowa

$M(S) = 2S$ np. abc $\longrightarrow$ aabbcc

$f(a_1)\ S_1,$

$f(a_2) = z_1;$

$\vdots$

$f(a_{2n}) = S_n.$

Chcemy otrzymać

$f(a_{2n+1}) = S_1;$

$f(a_{2n+2}) = S_1,$

$f(a_{2n+3}) = S_2;$

$f(a_{2n+4}) = S_2,$

$\vdots$

$f(a_{4n-1}) = S_n;$

$f(a_{4n}) = S_n.$

Instrukcja powinna mieć postać

$$z_i: \text{ wpisz w miejsce } a_{2n-1+2i}, a_{2n+2i} \longrightarrow S_i$$

czyli

$$Z_i = S_i^{\,a_{2n-1+2i}\,a_{2n+2i}}$$

dla i równego 1,2,.... n+1.

Ogólnie

$$f_i(a,b,z) = F(S_i, z_i, a_{m+2i})$$

gdzie

$$m = 2n, \quad 2n-1$$

dla i=n+1

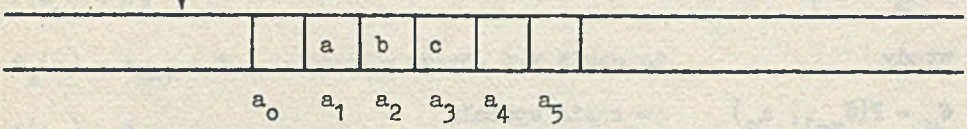
$$f_{n+1}(a,b,z) = F(S_{n+1}, z_{n+1}, a_{m+2(n+1)}) = f_n(a,b,z) - \text{stop.}$$

3.2. Przykłady ogólnego działania Maszyny Turinga

Przykłady 3.1 dotyczyły tylko szczególnej maszyny przepisującej słowa o długości n i tylko wykonującej operacje nad takimi słowami. Maszyna tego typu ma wiele narzuconych ograniczeń, trzeba zauważyć, że instrukcje, ich postacie wymagają w każdym przykładzie innej, dość skomplikowanej budowy głowicy Maszyny Turinga. Postać instrukcji, dość prosta matematycznie, może nasuwać wiele problemów przy zadawaniu jej maszynie. Spróbujmy teraz te same przykłady opisać przyjmując ogólny zapis Maszyny Turinga w ujęciu teorii zbiorów, nie narzucając żadnych ograniczeń na maszynę, ani na słowa, nad którymi dokonujemy operacji.

Przykład 1

M(S) = SS GM np. abc → abcabc



$A = \{a_0, a_1 \ldots\ldots\}$ - zbiór miejsc na taśmie.

$B = \{b_0, b_1 \ldots\ldots\}$ - zbiór danych na taśmie.

$\emptyset \in B$ - słowo puste należy do zbioru B.

$\sigma_{k+1} = F(\sigma_k, a_{k+1})$ - stan maszyny w chwili k+1.

$f(a_0) = \emptyset$,

$f(a_1) = b_1$,

$\sigma_1 = F(\sigma_0, a_1)$ - odczyt 1 litery zapis $\emptyset$ (zmiana stanu)

$f_k(a) = a_{k-1} + 1$ - ruch w prawo, tak długo, aż

$f(a_k) = \emptyset$,

wtedy

$\sigma_k = F(\sigma_{k-1}, a_k)$ - zapis symbolu na miejscu a_k

$f_{k+1}(a) = a_{k-1}$ - ruch w lewo tak długo, aż

$f(a_n) = \emptyset$,

wtedy

$\sigma_n = F(\sigma_{n-1}, a_n)$ - odczyt $\emptyset$, zapis symbolu

$f_{n+1}(a) = a_{n+1}$

i $\sigma_{n+1} = F(\sigma_n, a_{n+1})$ - odczyt 2 litery zapis $\emptyset$.

Dalej

$f_{n+2}(a) = a_{n+1} + 1$ - tak długo, aż

$f(a_m) = \emptyset$,

wtedy

$\sigma_m = F(\sigma_{m-1}, a_m)$ - zapis symbolu

i

$f_{m+1}(a) = a_m - 1$

$f(a_p) = \emptyset$.

Znów

$\sigma_p = F(\sigma_{p-1}, a_p)$ - zapis symbolu

$f_{p+1}(a) = a_p + 1$

i

$\sigma_{p+1} = F(\sigma_p, a_{p+1})$ - odczyt symbolu, zapis $\emptyset$

itd.

Stop maszyny nastąpi gdy odczyta ostatnią literę słowa i wtedy

$\sigma_\xi = \sigma_{\xi-1} = \sigma_k$ - stan stopu.

Jeżeli po odczytaniu ostatniej litery słowa nie nastąpiłby stop, to maszyna nie zapisywałaby słowa podwójnie, ale wielokrotnie. Praca maszyny ogólnie polega na odczytaniu przez głowicę symbolu i w jego miejscu zapisaniu symbolu pustego. Dalej głowica przesuwa się w prawo, aż do miejsca pustego wpisuje zapamiętany symbol, wraca w lewo, aż do najbliższego miejsca pustego wpisuje symbol i odczytuje następną literę, zapisując symbol pusty.

Przykład 2

$M(S) = S^{-1}$ np. abc $\longrightarrow$ cba

Znaczenie symboli jak w przykładzie 1.

$f(a_o) = \emptyset,$

$f_k(a) = a_{k-1} + 1$ - ruch w prawo, tak długo aż

$f(a_n) = \emptyset,$

wtedy

$f_{n+1}(a) = a_n - 1$

$\sigma_{n+1} = F(\sigma_n, a_{n+1})$ - odczyt symbolu i zapis $\emptyset,$

$f_{n+2}(a) = a_{n+1} + 1,$

$\sigma_{n+2} = F(\sigma_{n+1}, a_{n+2})$ - zapis symbolu

$f_{n+3}(a) = a_{n+2} - 1$ - tak długo, aż

$f(a_m) = \emptyset$

i

$f(a_{m+1}) = b$

$\sigma_{m+1} = F(\sigma_m, a_{m+1})$ - odczyt i zapis $\emptyset$

$f_{m+2}(a) = a_{m+1} - 1$ - tak długo, aż

$f(a_p) = b$

i

$f(a_{p+1}) = \emptyset,$

wtedy

$\sigma_{p+1} = F(\sigma_p, a_{p+1})$ - zapis symbolu.

Dalej

$f_{p+2}(a) = a_{p+1} - 1$ - tak długo, aż

$f(a_r) = \emptyset$

i

$f(a_{r+1}) = b$

-itd.

Aż do momentu

$\sigma_{\xi+1} = \sigma_{\xi} = \sigma_k$ - stan stopu.

Męszyna staje w momencie, gdy

$f_{r+1}(a) = a_r - 1$ i stale $\sigma_{r+1} = \sigma_r$ - nie ma już odczytu.

Możemy wprowadzić

$f(a_{-1}) = 1$

i

$\sigma(1) = \text{stop}$.

Wtedy, gdy nie ma już odczytu, głowica przebiega do miejsca a_{-1} (poruszając się w lewo) i staje.

Działanie maszyny polega na przesunięciu głowicy w prawo, aż do ostatniej litery słowa, odczycie ostatniej litery, zapisie symbolu pustego, przesunięciu w prawo i zapisie odczytanej litery.

Dalej powrót w lewo, aż do drugiej od końca litery w słowie, odczyt itd.....

Przykład 3

$M(S) = 2S$ np.: abc $\longrightarrow$ aabbcc

Oznaczenie symboli jak w przykładzie 1.

$f(a_o) = \emptyset$,

$f(a_1) = b$,

$\sigma_1 = F(\sigma_0, a_1)$ - odczyt pierwszej litery zapis $\emptyset$

$f_2(a) = a_1 + 1$

i

$f_k(a) = a_{k-1} + 1$ - tak długo, aż

$f(a_k) = \emptyset$,

wtedy

$\sigma_k = F(\sigma_{k-1}, a_k)$ - zapis pierwszej litery na pierwszym miejscu.

$f_{k+1}(a) = a_k + 1$

i

$\sigma_{k+1} = F(\sigma_k, a_{k+1})$ - zapis pierwszej litery na drugim miejscu.

Dalej

$f_{k+2}(a) = a_{k+1} - 1$ - ruch w lewo, tak długo aż

$f(a_n) = \emptyset$,

wtedy

$f(a_{n+1}) = a_n + 1$

i

$\sigma_{n+1} = F(\sigma_n, a_{n+1})$ - odczyt drugiego symbolu zapis $\emptyset$.

$f_{n+2}(a) = a_{n+1} + 1$ - tak długo, aż

$f = (a_m) = \emptyset$,

wtedy

$\sigma_m = F(\sigma_{m-1}, a_m)$ - zapis drugiego symbolu na pierwszym miejscu.

$$\left.\begin{array}{l} f_{m+1}(a) = a_m + 1 \\ \text{i} \\ \sigma_{m+1} = F(\sigma_m, a_{m+1}) \end{array}\right\} \begin{array}{l} \text{zapis drugiego symbolu} \\ \\ \text{na drugim miejscu.} \end{array}$$

Dalej, ruch w lewo.

Ruch trwa tak długo, aż odczytamy wszystkie symbole słowa, wtedy

$\sigma_{\xi+1} = \sigma_{\xi} = \sigma_k$ - stop maszyny.

Gdyby nie było stopu maszyny przed odczytem pierwszej litery następnego słowa, to otrzymalibyśmy słowa o literach poczwórnych, 2^3, 2^4 itd.

Ogólnie działanie maszyny polega na odczycie pierwszej litery, przesunięciu w prawo i zapisaniu tej litery na dwóch kolejnych miejscach pustych. Dalej odczyt drugiej litery itd.

II. Model matematyczny automatu skończonego

1. Ogólny opis działania automatu skończonego

Automatem skończonym nazywamy maszynę Turinga o następujących własnościach:

1) skończona taśma,
2) ruch jednostronny (tylko w lewo lub tylko w prawo),
3) maszyna akceptuje słowo.

Praca automatu skończonego polega na kolejnym badaniu miejsc taśmy i zmianie stanu w zależności od tego, czy symbol zapisany w danym miejscu należy do zadanego zbioru symboli (alfabetu), czy nie. Po sprawdzeniu całego słowa maszyna przechodzi do stanu określającego, że całe słowo zostało zaakceptowane (np. stan stopu maszyny) lub do określonego stanu odpowiadającego niezaakceptowaniu słowa. Dokładnie działanie automatu skończonego zilustruje prosty przykład.

Przykład

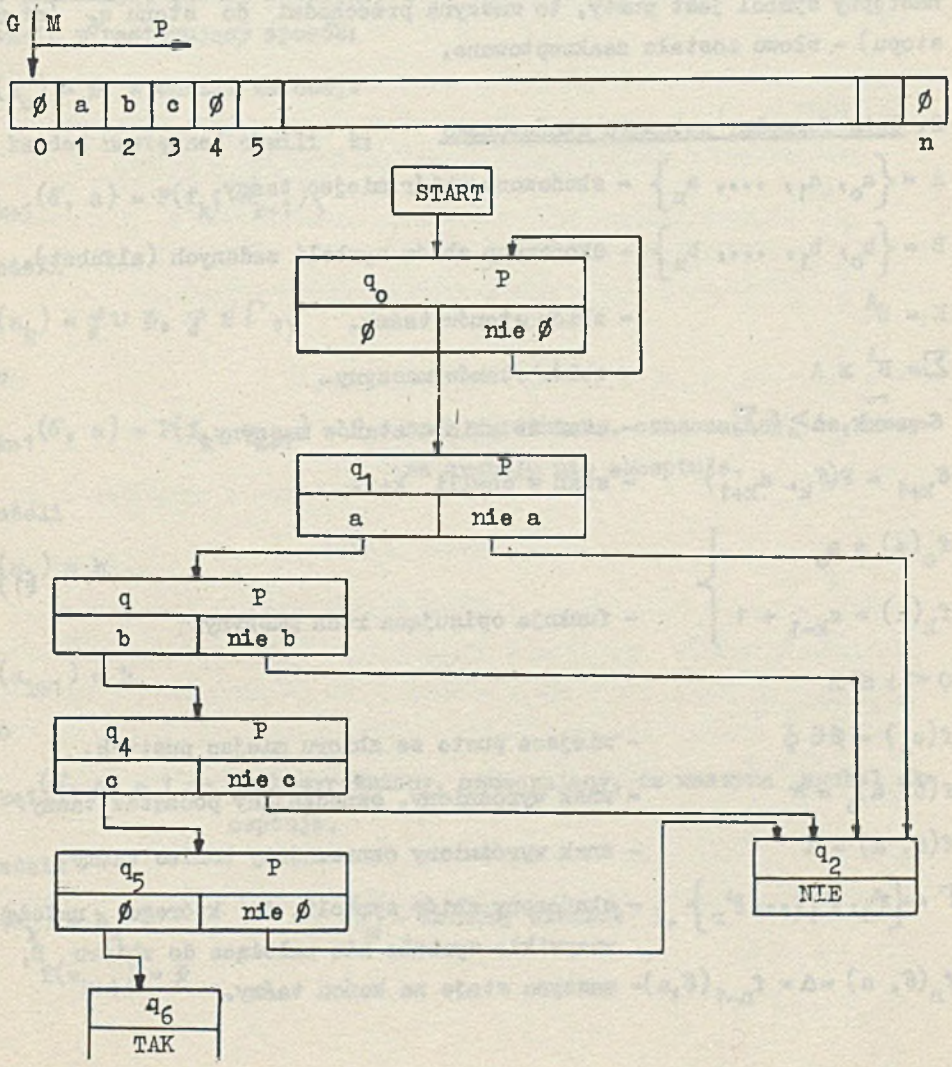
Automat skończony ma zbadać słowo zapisane na taśmie i zaakceptować, jeśli jest to słowo abc lub przejść do pewnego stanu wyróżnionego (nieakceptować), jeśli to nie jest słowo abc. Przyjmujemy ruch w prawo.

Ø - symbol pusty.

P - symbol określający przesunięcie w prawo.

A = {0, 1, 2, ..., n} - zbiór miejsc na taśmie.

B = abc - zbiór symboli (alfabet).



q - stan głowicy.

q_6 - może być stanem stopu.

q_2 - pewien wyróżniony stan np. 0.

Głowica rozpoczyna pracę od pozycji a_0, kiedy znajduje się w stanie q_0, dalej przesunięcie w prawo i odczytanie zawartości a_1, jeżeli należy do alfabetu (B), to następuje zmiana stanu, przesunięcie w prawo i badanie następnego symbolu; jeżeli nie należy do B, to następuje przejście do stanu q_2, który świadczy o niezaakceptowaniu słowa (np. 0) - tak badamy każdy symbol. Gdy całe słowo zostało zaakceptowane i następny symbol jest pusty, to maszyna przechodzi do stanu q_6 (stan stopu) - słowo zostało zaakceptowane.

2. Opis formalny automatu skończonego

$A = \{a_0, a_1, \ldots, a_n\}$ - skończony zbiór miejsc taśmy.

$B = \{b_0, b_1, \ldots, b_n\}$ - skończony zbiór symboli zadanych (alfabet).

$K = B^A$ - zbiór stanów taśmy.

$\Sigma = B^A \times A$ - zbiór stanów maszyny.

$\sigma = \overleftarrow{<k, a>} \in \Sigma$ - stan ze zbioru stanów maszyny.

$\sigma_{k+1} = F(\sigma_k, a_{k+1})$ - stan w chwili k+1.

$$\left.\begin{aligned} f_0(a) &= a_0 \\ f_k(a) &= a_{k-1} + 1 \end{aligned}\right\} \quad (1)$$

- funkcja opisująca ruch maszyny.

$0 < k \leq n$

$f(a_0) = \emptyset \in \Phi$ - miejsce puste ze zbioru miejsc pustych.

$f(\sigma, a)_0 = *$ - znak wyróżniony, oznaczający początek taśmy.

$f(\sigma, a) = \Delta$ - znak wyróżniony oznaczający koniec taśmy.

$\Gamma = \{\gamma_0, \gamma_1, \ldots \gamma_r\}$ - skończony zbiór symboli, do którego należą wszystkie symbole nie należące do zbioru B,

$f_n(\sigma, a) = \Delta = f_{n-1}(\sigma, a)$ - maszyna staje na końcu taśmy.

W opisie tym nie ma sensu pojęcie instrukcji. Jest natomiast wprowadzona pewna funkcja (1), opisująca ruch maszyny w prawo o jedno miejsce. Gdybyśmy poruszali się nie o jedno miejsce, ale o m to $f_k(a)$ równa się $a_{k-1} + m$, $1 < m < n$. Przyjęliśmy, założenie, że a_o jest miejscem pustym. Wyróżniony został początek i koniec taśmy zakładając, że na końcu taśmy maszyna zawsze staje. Wprowadziliśmy dodatkowy zbiór zawierający wszystkie możliwe symbole nie należące do zbioru B.

Zbiór Γ jest zatem rozłączny ze zbiorem B. Na zbiorach A, B, z dodatkiem symbolu pustego, możemy opisać działanie automatu skończonego zakładając zasadę śledzenia (1). Działanie automatu można przedstawić w następujący sposób:

$f(a_o) = \emptyset$ w chwili zerowej.

W każdej następnej chwili k:

$f_{k+1}(\sigma, a) = F(f_k, a_{k+1})$,

jeżeli

$f(a_k) = \gamma \cup \emptyset$, $\gamma \in \Gamma$,

to

$f_{k+1}(\sigma, a) = F(f_k, a_{k+1}) = 0$ - znak wyróżniony, oznaczający, że maszyna symbolu nie akceptuje.

Jeżeli

$f(a_k) = b$

i

$f(a_{k+1}) = b$

to

$f_{k+1}(\sigma, a) = 1$ - znak wyróżniony, oznaczający, że maszyna symbol akceptuje.

Jeżeli

$f(a_k) = b_m$ b_m - końcowy element B,

i $f(a_{k+1}) = \emptyset$

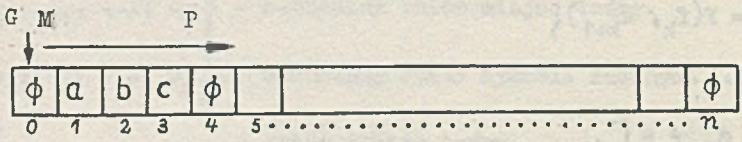
to

$f_{k+1}(\sigma, a) = f_k(\sigma, a) = 0$ - stan stopu - Maszyna zaakceptowała całe słowo.

Ogólnie maszyna akceptuje każdy symbol b, jeżeli za nim również jest b. Maszyna nie akceptuje żadnego symbolu ze zbioru Γ niezależnie od tego co następuje przed nim lub po nim. Jeżeli po końcowym elemencie zbioru B następuje symbol pusty, to maszyna akceptuje całe słowo. Śledząc działanie automatu skończonego mamy zatem zera i jedynki, w zależności od akceptacji symboli i stop po zaakceptowaniu całego słowa.

Przykład

Przedstawmy w naszym ujęciu formalnym przykład z punktu 1.



$A = \{0, 1, 2, 3, 4, \ldots, n\}$ - miejsca taśmy.

$B = a, b, c$ - alfabet.

Γ - wszystkie symbole z wyjątkiem a, b, c.

$\emptyset$ - symbol pusty.

$K = B^A$ - zbiór stanów taśmy.

$\Sigma = B^A \times A$ - zbiór stanów maszyny.

Ruch maszyny

$$\begin{cases} f_o(a) = a_o \\ f_k(a) = a_{k-1} + 1 \end{cases}$$

$f(a_o) = \emptyset$,

$f(a_1) = a \in B$,

$f(a_2) = b \in B$,

stąd

$f_2(\sigma, a) = 1$ - akceptacja litery.

$f(a_3) = c \in B_\lambda$

czyli

$f_3(\sigma, a) = 1$ - akceptacja.

Dalej

$f(a_4) = \emptyset_\lambda$

przy czym

$f(a_3) = b_3$ - symbol końcowy $\in B_\lambda$

zatem

$f_4(\sigma, a) = f_3(\sigma, a) = 0$ - stop.

Słowo zostało zaakceptowane.

Ogólnie np. w chwili trzeciej stan maszyny opisuje funkcja

$$f_3(\sigma, a) = \langle k_3, a_4 \rangle ,$$

gdzie

$$k_3 \Longrightarrow f(a_3) = b_3.$$

Zakończenie

Zapis maszyn cyfrowych przedstawiony w pracy jest zapisem prawie formalnym. Aby przejść do zapisu czysto formalnego (język sformalizowany), należałoby przeprowadzić analizę grafu funkcyjnego przedstawionych modeli i zapisać ten graf określonymi zbiorami symboli lub uściślić przedstawiony powyżej zapis tak, aby nie wymagał on uzupełnień słownych.

Dalsza praca może pójść w kierunku zapisania rzeczywistej maszyny cyfrowej i jej działania za pomocą pewnego ściśle określonego zbioru

symboli. Dopiero po takich próbach będzie można dojść do wprowadzenia pewnych funkcji bądź relacji, które w sposób bardziej przejrzysty i ogólniejszy przedstawią model dowolnej maszyny cyfrowej.

LITERATURA

1. Kuratowski K.: Wstęp do teorii mnogości i topologii, PWN, Warszawa 1971.
2. Mostowski A.: Elementy logiki, Monografie matematyczne, Warszawa 1948.
3. Elgot C.C.: Machines species and thier computation languages, Formal language description languages, North Holland Publisching Company - Amsterdam 1966, 160-179.
4. Elgot C.C., Robinson A.: Random - acces, stored-program, an approach to programming languages, Journal of the ACM 11 (1964), 365-399.
5. Wang H.: A wariant do Turings theory of computing machines, Journal of the ACM 4 (1957), 63-92.
6. Blikle A.: Matematyczne aspekty organizacji maszyn bezadresowych, Praca doktorska, Uniwersytet Warszawski 1967.

НЕКОТОРОЕ ФОРМАЛЬНОЕ ОПИСАНИЕ
МАШИНЫ ТЬЮРИНГА И КОНЕЧНОГО АВТОМАТА

Р е з ю м е

В работе приводится формальное описание машины Тьюринга и конечного автомата. Здесь использован теоретико - множественный подход к описанию вычислительной машины. Общее описание этих типов машин, так как и примеры, указывают, что теория множеств позваляет довольно легко описывать некоторые из вычислительных машин, и такой образ описывания ведёт к некоторому, кажется достаточно эффективному, формализированному языку.

A FORMAL DESCRIPTION OF THE TURING'S MACHINE AND THE FINITE AUTOMATA

S u m m a r y

The present paper concerns formal description of the Turing's machine and the finite automata. The description of a digital computer is based on the Set Theory. The notion of a digital computer can be relatively easy formalized on the base of the Set Theory for some classes of those machines. The description leads to rather effective formalized language for particular classes of digital computers.