R O Z D Z I A Ł   V I I I

Maszyna Turinga

1. Definicja maszyny Turinga

Najprostszym narzędziem do rozpoznawania języków jest automat skończony. Nie ma on pamięci zewnętrznej, a jedynie wewnętrzną. Informacją pamiętaną z przeszłości jest stan automatu. Jak wiadomo, umożliwia to rozróżnianie skończonej ilości (równej ilości stanów) informacji i akceptację języków regularnych. Automat ze stosem posiada już pamięć zewnętrzną pozwalającą na zapisywanie i rozróżnianie nieskończonej ilości informacji. Sposób korzystania z niej poddany jest jednak daleko idącym ograniczeniom, a mianowicie:

a) informację odczytuje się z wierzchu stosu,

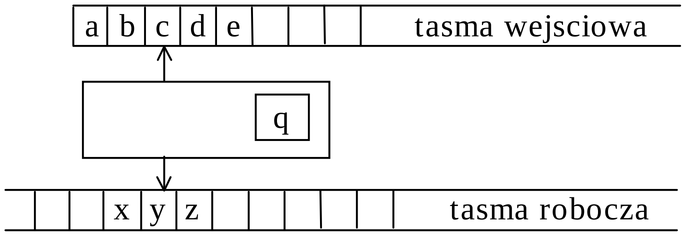
b) po odczycie informacja jest wymazywana, czyli niszczona.

Chcąc więc dotrzeć do informacji zapisanej we wnętrzu stosu trzeba odczytać (a więc zniszczyć) wszystko to, co było zapisane później. Ograniczenia a) i b) w korzystaniu z pamięci zostały tak dobrane, aby klasa języków akceptowanych przez automaty ze stosem była klasą języków bezkontekstowych.

Spróbujmy teraz usunąć ograniczenia a) i b). Zachowajmy (tak jak w automacie ze stosem) taśmę wejściową z prawej strony nieograniczoną, odczytywaną od lewego krańca do prawego, oraz taśmę roboczą, obustronnie nieskończoną, po której głowica może poruszać się w obu kierunkach. Automat pracuje w kolejności: odczyt, zapis, ruch taśmy, co składa się na jeden c y k l p r a c y .

Przypuśćmy, że mamy sytuację, jak na rysunku 8.1, tj. automat będąc w stanie q czyta:

- z taśmy wejściowej literę c słowa abcde,
- z taśmy roboczej (będącej tu odpowiednikiem stosu z automatu ze stosem) literę y.

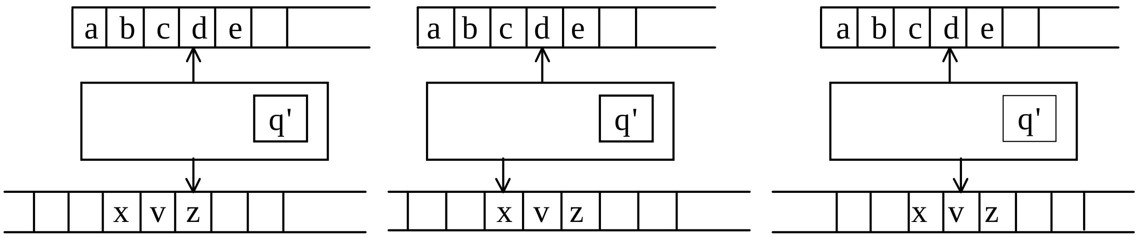


Rys. 8.1.

Instrukcja automatu, podobnie jak w automacie ze stosem, będzie miała poprzednik yqc. Następnik będzie się składał z:

- symbolu v zapisanego na taśmie roboczej w tej klatce, z której czytany był symbol y,
- nowego stanu q' przyjętego przez automat,
- znacznika wyboru klatki taśmy roboczej, z której czytany będzie symbol w następnym cyklu pracy, przy czym przyjmuje się tu tylko trzy możliwości do wyboru:
  - R: następny odczyt z taśmy roboczej z sąsiedniej klatki w prawo (R - od ang. „right”),
  - L: następny odczyt z taśmy roboczej z sąsiedniej klatki w lewo (L - od ang. „left”),
  - S: następny odczyt z taśmy roboczej z tej samej klatki, co poprzednio (S - od ang. „the same”).

Instrukcje automatu mogą więc mieć postać jak poniżej (nad każdą z nich na rys. 8.2 przedstawiono wynik jej wykonania).

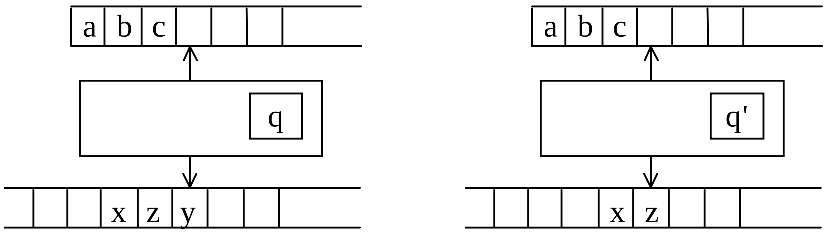


yqc $\rightarrow$ vq'R　　　　yqc $\rightarrow$ vq'L　　　　yqc $\rightarrow$ vq'S

Rys. 8.2.

Następny zapis nie musi więc wcale odbywać się na prawo od ostatnio zapisanego symbolu. Jak to widać na środkowym rysunku - do odczytu informacji x zapisanej przed v można dojść nie niszcząc informacji v zapisanej ostatnio.

Zauważmy jeszcze, że alfabety używane na taśmie wejściowej i taśmie roboczej mogą (lecz nie muszą) być różne. Przyjmujemy jednak, że oba te alfabety posiadają wspólny symbol B (od ang. „blank” = pusty, niezapisany) pustej klatki. Przeczytanie symbolu pustej klatki odpowiada przeczytaniu symbolu B blanki. Napisanie symbolu B odpowiada z kolei wymazaniu poprzedniego znaku i pozostawieniu klatki pustej (a więc zniszczeniu informacji). Sytuację tę przedstawia rysunek 8.3.



sytuacja przed wykonaniem instrukcji yqB $\to$ Bq'L

sytuacja po jej wykonaniu

Rys. 8.3.

Zauważmy, że nie przyjmowaliśmy tu żadnych założeń dotyczących zupełności, czy też determinizmu tego automatu. Jeżeli w danej konfiguracji nie ma instrukcji, którą można zastosować, to praca automatu przerywa się (automat wyłącza się). Jeżeli z kolei jest kilka instrukcji dających się zastosować, to można posłużyć się dowolną z nich. W związku z tym, opisane tu urządzenie nosi nazwę *niedeterministycznej dwutaśmowej maszyny Turinga*.

2. Akceptowalność języków przez maszynę Turinga

Mówimy, że słowo w jest *akceptowane* przez dwutaśmową maszynę Turinga ze stanem początkowym q_0 i zbiorem stanów końcowych H, jeżeli dla konfiguracji początkowej:

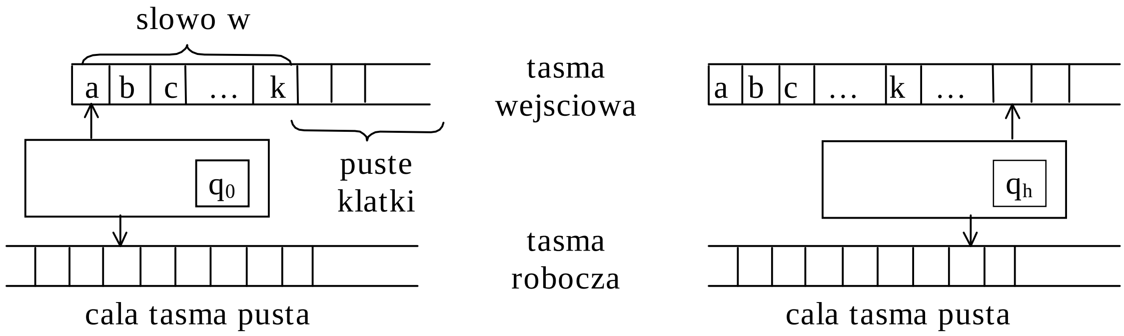
– słowo w jest wypisane na początku taśmy wejściowej, a reszta tej taśmy jest pusta,
– automat jest ustawiony na pierwszym symbolu słowa w w stanie q_0 i dowolnej klatce pustej taśmy roboczej,

istnieje przebieg doprowadzający do konfiguracji:

– słowo w, jak również dowolna ilość pustych klatek za nim, jest przeczytana,
– taśma robocza jest pusta,
– zaś automat znajduje się w jakimś stanie q_h należącym do zbioru stanów końcowych.

Sytuacja taka pokazana została na rys. 8.4.

Oczywiście, dla danej dwutaśmowej maszyny Turinga, zbiór słów akceptowalnych przez nią nazywamy *językiem akceptowalnym przez maszynę Turinga*.



– konfiguracja poczstkowa konfiguracja akceptujaca ($q_h \in H$)

Rys. 8.4.

Rozważmy to na konkretnym przykładzie.

Przykład 8.1.

Niech:

- alfabet taśmy wejściowej T = {a, b, c, B}, gdzie B jest symbolem blanki,
- alfabet taśmy roboczej Z = {0, 1, B}, gdzie B znów jest symbolem blanki,
- zbiór stanów automatu K = {q_0, q_a, q_b, q_c, q_h},
- q_0 jest stanem początkowym,
- zbiór stanów końcowych H = {q_h},
- zaś lista instrukcji M będzie następująca:

1) $Bq_0a \to Bq_aR$, 2) $Bq_aa \to 1q_aR$, 4) $1q_bb \to 0q_bL$, 6) $0q_cc \to Bq_cR$,
3) $Bq_ab \to 1q_bS$, 5) $1q_bc \to 0q_cS$, 7) $0q_cB \to Bq_hS$

Zobaczmy, jak automat ten analizuje słowo $a^nb^nc^n$ (gdzie $n \geq 1$):

– najpierw czytając a - jeden raz wykonuje instrukcję 1), w wyniku czego na taśmie roboczej zapisuje B, a sam przechodzi w stan q_a; głowica taśmy roboczej została przesunięta o jedno miejsce w prawo,

– następnie czytając $a^{n-1}b$ - n-1 razy wykonuje instrukcję 2) i 1 raz instrukcję 3), w wyniku czego na taśmie roboczej zapisuje n jedynek (czyli słowo 1^n) za każdym razem (oprócz ostatniego) przesuwając głowicę taśmy roboczej o jedno miejsce w prawo, a sam przechodzi w stan q_b,

– następnie czytając $b^{n-1}c$ - n-1 razy wykonuje instrukcję 4) i jeden raz instrukcję 5), w wyniku czego na taśmie roboczej zastępuje n jedynek zerami za każdym razem (oprócz ostatniego) przesuwając głowicę taśmy roboczej o jedno miejsce w lewo, a sam przechodzi w stan q_c (w tym momencie na taśmie roboczej jest więc zapisane słowo 0^n),

– następnie czytając c^{n-1} B - n-1 razy wykonuje instrukcję 6) i 1 raz instrukcję 7), w wyniku czego na taśmie roboczej zastępuje n zer blankami (czyli całkiem wymazuje taśmę) za każdym razem (oprócz ostatniego) przesuwając głowicę taśmy roboczej o jedno miejsce w prawo, a sam przechodzi w stan q_h.

W ten sposób analizując słowo $a^n b^n c^n$ przeszliśmy od stanu początkowego q_0 do stanu końcowego q_h, głowica taśmy wejściowej znajduje się na blance, a cała taśma robocza jest pusta. Automat ten akceptuje więc słowo $a^n b^n c^n$.

Zauważmy, że przy każdym słowie w różnym od $a^n b^n c^n$ automat ten wyłączy się przed przeczytaniem do końca tego słowa (z taśmą robocza pustą lub nie), czy też wyłączy się po przeczytaniu całego słowa w, ale taśma roboczą nie będzie pusta.

Ponieważ automat ten najpierw notuje liczbę przeczytanych a, potem sprawdza, czy liczba przeczytanych b jest taka sama (nie gubiąc informacji, ile a zostało przeczytanych), by w końcu sprawdzić, czy ilość przeczytanych c też jest taka sama i czy nastąpił wówczas koniec słowa - zatem automat ten rozpoznaje jedynie wszystkie słowa języka (lub krócej: język) $L = \{a^n b^n c^n : n \geq 1\}$. □

Jak już wiemy (z wniosku z twierdzenia 2.6), nie jest to język kontekstowy, a zatem (z transpozycji twierdzenia 7.4) - nie jest on rozpoznawalny przez automat ze stosem. Łatwo jest pokazać z kolei, że każdy automat ze stosem jest szczególnym przypadkiem dwutaśmowej maszyny Turinga. Tak więc maszyny Turinga rozpoznają szerszą klasę języków niż automaty ze stosem, czyli akceptują one szerszą klasę języków od klasy języków bezkontekstowych. Dokładnie ujmuje to następujące twierdzenie:

Twierdzenie 8.1.

L jest językiem typu 0 witw, gdy jest akceptowany przez dwutaśmową maszynę Turinga. □

Zadanie 8.1. Dla automatu z przykładu 8.1 podaj przykłady takich słów, by w wyniku ich analizy automat ten:

- wyłączył się przed przeczytaniem całego słowa (z taśmą robocza pustą lub nie),
- wyłączył się po przeczytaniu całego słowa, ale aby taśma robocza nie była wtedy pusta. □

Zadanie 8.2. Zbuduj dwutaśmową maszynę Turinga akceptującą:

a) język V^* określony nad alfabetem V,

b) język pusty $\emptyset$,

c) język $\{\lambda\}$ złożony ze słowa pustego λ. □

Zadanie 8.3. Sformułuj i podaj formalną definicję maszyny Turinga. □

Zadanie 8.4. Wykaż, że rzeczywiście automat ze stosem jest szczególnym przypadkiem maszyny Turinga. □

3. Funkcje sygnalizujące i złożoność problemu akceptacji języków

Pozostaje nam jeszcze do wyjaśnienia kwestia, jaka klasa automatów akceptuje języki kontekstowe. Zagadnieniem tym zajmiemy się właśnie w niniejszym paragrafie. Przyjrzyjmy się w tym celu poniższemu przykładowi.

Przykład 8.2.

Rozpatrzmy maszynę Turinga z przykładu 8.1, akceptującą język $L = \{a^n b^n c^n : n \geq 1\}$. Do zbioru jej stanów dodajmy nowy stan q_w, będący tzw. stanem wyłączenia, i każdą z instrukcji postaci $yqx \rightarrow yq_wS$ (gdzie $x \in \{a,b,c,B\}$, $q \in \{q_0, q_a, q_b, q_c\}$, a $y \in \{0,1,B\}$), jeśli tylko trójka yqx nie jest poprzednikiem żadnej z instrukcji 1) - 7) z przykładu 8.1. Zauważmy, że stan końcowy q_h oraz stan wyłączenia q_w występują wyłącznie po prawej stronie instrukcji. Niemożliwe jest więc w momencie zaistnienia jednego z nich dalsze działanie maszyny. Dlatego też stany te (końcowy q_h i wyłączenia q_w) nazywamy s t a n a m i s t o p u j ą c y m i, a konfiguracje zawierające je - k o n f i g u r a c j a m i s t o p u j ą c y m i. Ponieważ dla każdej konfiguracji niestopującej mamy określony następny stan, i to w sposób jednoznaczny, więc tak otrzymana maszyna jest zupełna i deterministyczna. Oczywiście, ma ona 48 instrukcji, bo $4 \cdot 4 \cdot 3 = 48$, gdzie kolejne czynniki są mocami odpowiednio zbiorów $\{a, b, c, B\}$, $\{q_0, q_a, q_b, q_c\}$ i $\{0, 1, B\}$, tj. zbiorów symboli, które w instrukcji mogą stać odpowiednio na pierwszej, drugiej i trzeciej pozycji.

Przypatrzmy się jeszcze, w jaki sposób maszyna ta akceptuje słowa języka L spośród wszystkich słów $w \in \{a,b,c\}^+$. Startując ze stanu początkowego q_0, każde z takich słów doprowadza ona do konfiguracji stopującej. Wówczas słowo jest akceptowane witw, gdy zawiera ona stan końcowy q_h, a w dodatku taśma robocza jest pusta. □

Złożoność procesu od słowa w do konfiguracji stopującej jest określona poprzez:

– ilość kroków (a więc tym samym czas, zakładając że każda instrukcja jest wykonywana w tym samym czasie) $T(w)$ prowadzących od konfiguracji początkowej (ze słowem w) do konfiguracji stopującej,
– tzw. ilość potrzebnej przestrzeni (a konkretnie: klatek) taśmy roboczej $S(w)$.

Przykład 8.2 – c.d.

Zważywszy na fakt, jak są akceptowane słowa przez tę maszynę, otrzymujemy:

dla w =	T(w) =	S(w) =
$a^k\ b^k\ c^k$	3k + 1	k + 1
ba^{n-1}	1	2
a^n	n + 1	n + 1
λ	1	1

Tab. 8.1. □

Ogólnie, dla wszystkich słów długości n, przyjmujemy i mamy więc odpowiednio:

$$T'(n) = \max T(w) = n + 1,$$

$$S'(n) = \max S(w) = n + 1.$$

Tak więc dla przebadania (i akceptacji lub nieakceptacji) wszystkich słów o długości n potrzeba (i wystarcza) zarówno T'(n) = n + 1 jednostek czasu (czyli cykli pracy maszyny), jak i S'(n) = n + 1 komórek (czyli innymi słowy: klatek) pamięci.

Funkcje te dla rozpatrywanej maszyny Turinga nazywamy odpowiednio funkcjami sygnalizującymi czas i przestrzeń.

Są one jednak określone tylko dla określonej maszyny Turinga, a nie dla języka przez nią akceptowanego. Widać to bardzo wyraźnie chociażby na poniższym przykładzie.

Przykład 8.3.

Skonstruujemy dwie maszyny Turinga, w których:

– alfabet zarówno taśmy wejściowej, jak i taśmy roboczej składa się z symbolu dolara $ oraz z symbolu blanki B,

– q_0 jest stanem początkowym, a q_h stanen stopującym (są to jedyne stany tych maszyn),

– występują następujące instrukcje:

– w pierwszej maszynie:	- w drugiej maszynie:
1) $Bq_0\$ \rightarrow \$q_0S$	
2) $\$q_0\$ \rightarrow Bq_0S$	identycznie
3) $\$q_0B \rightarrow Bq_hS$	
4) $Bq_0B \rightarrow Bq_0S$	4') $Bq_0B \rightarrow Bq_0R$.

Obydwie te maszyny:

– są deterministyczne i zupełne,
– ich instrukcje 1) i 2) pozwalają na analizę badanego słowa, a instrukcje 3) i 4) odpowiednio na jego akceptację czy nieakceptację,
– akceptują ten sam język:
  – słowa postaci w' = $\$^{2k+1}$ (zawierające nieparzystą liczbę symboli dolara) są akceptowane w czasie T(w') = T'(w') = 2k + 2 (= |w'| + 1),
  – słowa postaci w'' = $\$^{2k}$ (zawierające parzystą liczbę symboli dolara) nie są akceptowane, a (ponieważ po analizie całego słowa, ostatnia instrukcja będzie wykonywana nieskończenie wiele razy, więc) T(w'') = T'(w'') = ∞.

Tak więc

$$T'(n) = T(n) = \begin{cases} n+1 & \text{dla n nieparzystych,} \\ \infty & \text{dla n parzystych.} \end{cases}$$

Jednak dla pierwszej maszyny S(n) = 1 (gdyż w każdym przypadku tylko pierwsza klatka jest rozpatrywana), a dla drugiej maszyny

$$S'(n) = S(n) = \begin{cases} 1 & \text{dla n nieparzystych,} \\ \infty & \text{dla n parzystych.} \end{cases}$$ □

Mając to na uwadze, wprowadzamy następującą definicję:

Mówimy, że dwutaśmowa maszyna Turinga *akceptuje język L z funkcją sygnalizującą przestrzeń* S'(n) = f(n), jeżeli:

1) maszyna ta akceptuje język L,
2) dla każdego słowa w takiego, że |w| = n (a więc dowolnie, czy należącego, czy też nie należącego do L), istnieje przebieg maszyny, dla którego S(w) ≤ f(n) (gdzie f(n) < ∞).

Według tej definicji, pierwsza z maszyn z przykładu 8.3 akceptuje język L z funkcją sygnalizującą przestrzeń S'(n) = f(n) = 1, a druga - nie.

Obecnie możemy już podać naczelne w tym paragrafie twierdzenie.

Twierdzenie 8.2.

L jest językiem kontekstowym (czyli typu 1) witw, gdy jest akceptowany przez dwutaśmową maszynę Turinga z funkcją sygnalizującą przestrzeń S(n) = n + 1. □

Wnioski.

1) Język z przykładu 8.2 jest kontekstowy.

2) Kontekstowy jest także język z przykładu 8.3, ze względu na akceptującą go pierwszą maszynę z insrukcjami 1) - 4) (bo skoro jest akceptowalny przez maszynę Turinga z funkcją

sygnalizującą przestrzeń S'(n) = f(n) = 1, to i jest on akceptowalny przez maszynę Turinga z funkcją sygnalizującą przestrzeń S'(n) = f(n) = n + 1). □

Opisany w twierdzeniu sposób akceptacji, nosi nazwę a k c e p t a c j i w s p o s ó b l i n i o w y. Widzimy, że przy akceptacji słowa o długości n wystarczy rozpatrywać tu maszynę Turinga, której taśma robocza ma jedynie n + 1 klatek.

Zadanie 8.5. Niech T = {a, b, c}. Zbuduj maszynę Turinga akceptująca tylko i wyłącznie wszystkie słowa postaci:

a) $a^n b^{2n} c^n$ (gdzie $n \in N$),

b) $a^n b^{2n} c^{3n}$ (gdzie $n \in N$),

c) $a^{2n} b^{3n} c^2$ (gdzie $n \in N$),

d) $a^n b^{n+1} c^{n+2}$ (gdzie $n \in N$).

Dla słów tych języków określ funkcje sygnalizujące czas i przestrzeń. □

Zadanie 8.6. Zbuduj dla języka akceptowanego przez maszynę z przykładu 8.3 automat skończony akceptujący ten język. □

Zadanie 8.7. Pokaż, że język L jest regularny witw, gdy istnieje taka dwutaśmowa maszyna Turinga akceptująca L, dla której T(n) = n + 1, a S(n) = 1. □

Zadanie 8.8. Pokaż, że dla języka $L = aV^*b$ złożonego ze słów rozpoczynających się od ustalonego symbolu $a \in V$ i kończących się ustalonym symbolem $b \in V$, można zbudować taką maszynę Turinga, dla której czas akceptacji słowa o długości n wynosi T(n) = n + 1, zaś przestrzeń S(n) = 1. Wskazówka: skorzystaj z poprzedniego zadania. □

4. Rozstrzygalne i nierozstrzygalne problemy lingwistyki

Już w przedmowie zapoznaliśmy się z pojęciami „problemy rozstrzygalne” i „problemy efektywnie rozstrzygalne”, przyjmując że przez problemy rozstrzygalne rozumiemy te z nich, które jesteśmy w stanie rozstrzygnąć w przeliczalnej liczbie kroków, a przez problemy efektywnie rozstrzygalne te, które jesteśmy w stanie rozstrzygnąć w skończonej liczbie kroków. Obecnie podejdziemy do tego zagadnienia z innej strony, wykorzystując pojęcie maszyny Turinga. Rozpatrzymy to na przykładzie problemu „$x \in L$”, tj. przynależności danego słowa x (określonego nad pewnym alfabetem V) do danego języka L. Mianowicie:

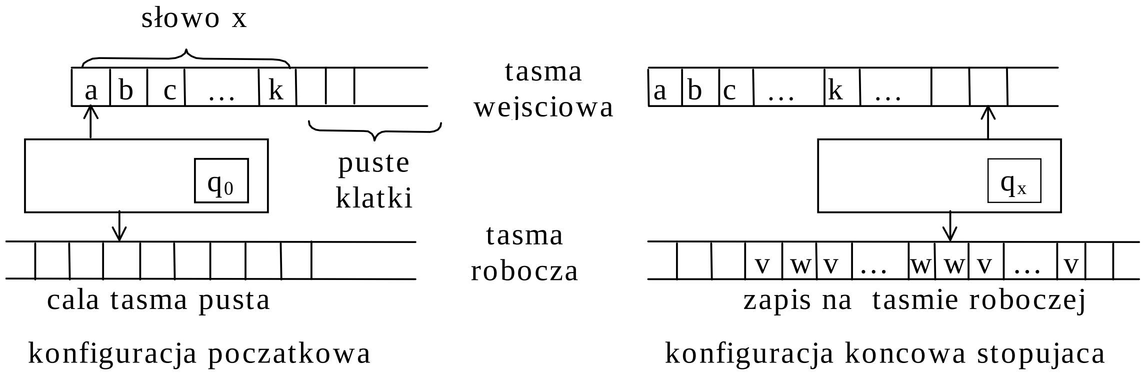
Problem „$x \in L$” nazywamy r o z s t r z y g a l n y m (lub równoważnie: o b l i c z a l n y m) witw, gdy istnieje deterministyczna dwutaśmową maszyna Turinga o takim podziale stanów stopujących na dwa rozłączne zbiory TAK i NIE, że dla konfiguracji początkowej:

- słowo x wpisane na taśmie wejściowej,
- maszyna ustawiona na pierwszej literze słowa x, (∗)
- taśma robocza pusta,

przyjmuje:

- dla $x \in L$ stan końcowy q_x należący do zbioru TAK,
- zaś dla $x \notin L$ stan końcowy q_x należący do zbioru NIE.

Sytuacja taka pokazana została na rys. 8.5.



(dla $x \in L$ - $q_x \in$ TAK
dla $x \notin L$ - $q_x \in$ NIE).

Rys. 8.5.

Porównajmy tę definicję z definicją akceptacji słowa przez maszynę Turinga.

1) Dla rozstrzygalności <u>nie żądamy</u>, by w maszynie po przyjęciu przez nią stanu ze zbioru stanów końcowych (czyli tu: zbioru stanów TAK) <u>taśma robocza była pusta</u>, jak to ma miejsce przy akceptacji. Nie jest to jednak istotna różnica, gdyż jak się okazuje:

Jeżeli problem „$x \in L$" jest rozstrzygalny przez pewną dwutaśmową maszynę Turinga, to język L jest typu 0, tj. jest on akceptowalny przez pewną maszynę Turinga (a więc dochodzi wówczas do wyczyszczenia taśmy roboczej).

2) Podobnie <u>nieistotną</u> różnicą jest żądanie <u>determinizmu</u> maszyny (przyjmuje się je jedynie dla prostoty sformułowań i dowodów).
3) <u>Istotne</u> jest więc jedynie to, że dla słowa x zapisanego na taśmie wejściowej w konfiguracji początkowej, przebieg <u>doprowadza do konfiguracji stopującej</u>.

Widzimy nadto, że postępowanie to jest efektywne. Pozwala ono bowiem po <u>skończonej</u> liczbie kroków odpowiedzieć na pytanie czy $x \in L$, czy też $x \notin L$. Po uruchomieniu maszyny w konfiguracji początkowej (∗) ze słowem x, czekamy aż dojdzie ona do konfiguracji stopującej (co zgodnie z definicją problemu rozstrzygalnego zawsze nastąpi w skończonym czasie), a następnie sprawdzamy, czy osiągnięty stan należy do zbioru TAK (wówczas i $x \in L$), czy też należy on do zbioru NIE (wówczas to $x \notin L$).

Zauważmy, że z kolei w definicji akceptowalności słów języka L przez maszynę Turinga - jedynie dla przebiegów akceptujących żądamy, aby osiągano konfiguracje stopujące (patrz diagram 8.1).

konfiguracja:	akceptująca	nieakceptująca	
Osiągalność stopu:	tak	tak	nie

Diagram 8.1.

W takim razie dla pewnych konfiguracji nieakceptujących możemy nie osiągnąć stopu. Jeśli bowiem postąpimy tak jak poprzednio i uruchomimy maszynę dla konfiguracji początkowej (∗) ze słowem x, to nie dla każdego słowa x będziemy w stanie efektywnie orzec, czy $x \in L$, czy też $x \notin L$. Możemy bowiem bardzo długo przeprowadzać analizę tego słowa i wciąż nie osiągać konfiguracji stopującej (która umożliwiłaby nam określenie akceptowalności tego słowa przez ten automat). Jeżeli słowo x cały czas nie doprowadza nas do konfiguracji stopującej, to wcale nie upoważnia nas to jeszcze do stwierdzenia, że jest ono nieakceptowane przez ten automat. Kto wie, czy jeżeli przeprowadziliśmy już 10^{20} cykli i nie osiągnęliśmy jeszcze stopu, czy oby w następnym kroku nie osiągniemy go i być może okaże się, że właśnie $x \in L$. Jak się okazuje, żadnej roli w tym nie odgrywa tu nawet niedeterminizm maszyny Turinga (patrz: następny paragraf). Istotny jest tu natomiast fakt, który niesie z sobą poniższe twierdzenie.

Twierdzenie 8.3 (o nierozstrzygalności problemu stopu dla maszyny Turinga).

Istnieje taka dwutaśmową maszyna Turinga (o alfabecie taśmy roboczej V), że dla zbioru L_{STOP} słów $x \in V^*$ doprowadzanych przez nią od konfiguracji początkowej (∗) do konfiguracji stopującej, problem czy $x \in L_{STOP}$ czy też $x \notin L_{STOP}$ jest nierozstrzygalny. □

Jest ono równoważne następującemu twierdzeniu:

Twierdzenie 8.4.

Istnieje język L typu 0, dla którego problem „$x \in L$” należenia słowa do języka jest nierozstrzygalny. □

Ponieważ wiemy już (patrz: twierdzenie 4.2), że problem ten jest rozstrzygalny dla języków kontekstowych, więc wnioskujemy stąd, że:

1) jest on nierozstrzygalny tylko dla tych języków typu 0, które nie są kontekstowe (a więc jedynie dla „właściwych” języków typu 0),

2) a zatem (ponieważ jest on rozstrzygalny dla wszystkich języków kontekstowych) - jest on również rozstrzygalny dla języków bezkontekstowych i regularnych (jako jego szczególnych typów).

Na koniec tego paragrafu zbierzmy jeszcze w tab. 8.2. wyniki kwestii rozstrzygalności zarówno tego, jak i niektórych innych badanych przez nas problemów (gdzie „+” oznacza rozstrzygalność, a „-” - nierozstrzygalność).

język (typ gramatyki)	język akceptowany przez	rozstrzygalność problemu			
		$x \in L$	$L(G)= \emptyset$	jednozn. G	$G \equiv G'$
regularny	automat Rabina-Scotta	+	+	+	+
bezkontekstowy	automat ze stosem	+	+	-	-
kontekstowy	dwutaśmową maszynę Turinga w sposób liniowo ograniczony	+	-	-	-
struktur frazowych	dwutaśmową maszynę Turinga	-	-	-	-

Tab. 8.2.

5. Dodatkowe uwagi i spostrzeżenia

Oprócz zdefiniowanej na początku tego rozdziału dwutaśmowej maszyny Turinga, jak się łatwo domyślić, istnieją również więcej-taśmowe maszyny Turinga, posiadające więcej taśm roboczych. Nazywamy je *wielotaśmowymi maszynami Turinga* lub też k-taśmowymi maszynami Turinga, gdzie $k \geq 3$ oznacza liczbę jej taśm roboczych. Jak się jednak okazuje, zachodzi poniższe twierdzenie:

Twierdzenie 8.5.

Dla każdej wielotaśmowej maszyny Turinga istnieje dwutaśmowa maszyna Turinga taka, że języki przez nie rozpoznawane są identyczne. □

Zajmijmy się jeszcze na koniec determinizmem maszyny Turinga. Mówi nam o nim (ostatnie już w naszym kursie) poniższe twierdzenie:

Twierdzenie 8.6.

Dla każdej niedeterministycznej maszyny Turinga, istnieje deterministyczna maszyna Turinga taka, że języki przez nie rozpoznawane są identyczne. □

Tak więc ani kwestia determinizmu, ani kwestia ilości taśm roboczych maszyny Turinga nie wpływają na zakres akceptowanych przez nie języków.