R O Z D Z I A Ł V I

Automaty skończenie-stanowe

1. Podstawowe definicje

Dotychczas mówiliśmy jedynie o gramatykach jako o generatorach języków. Obecnie zajmiemy się akceptorami języków, jakimi są automaty. Nasze rozważania rozpoczniemy od najprostszych z nich, jakimi są automaty skończenie-stanowe, zwane inaczej automatami Rabina - Scotta.

Formalnie, a u t o m a t e m s k o ń c z o n y m (lub równoważnie skończenie-stanowym) nazywamy uporządkowaną piątkę $\mathcal{A} = <K, T, M, K_0, H>$, w której:

- K jest niepustym skończonym zbiorem stanów (odpowiednikiem w gramatykach jest V_N),
- T jest niepustym skończonym zbiorem, zwanym alfabetem (odpowiednik V_T),
- M (odpowiednik F) - to tzw. relacja przejścia, M: K x T $\rightarrow$ K; przyjmujemy, że jest to relacją całkowita, tj. że dla każdej pary argumentów (a,b) $\in$ K x T wyznacza ona pewną wartość z K,
- $K_0 \subseteq K$ nazywamy zbiorem stanów początkowych automatu (odpowiednik S),
- $H \subseteq K$ - to zbiór stanów akceptowalnych (odpowiednik zbioru reguł końcowych).

W przypadku, gdy $M \subseteq K \times T \times K$ będzie funkcją (tj. zawsze jednoznacznie będzie wyznaczać następny stan automatu), a zamiast zbioru stanów początkowych K_0 będziemy mieć jeden stan początkowy $q_0 \in K$, to o takim automacie $\mathcal{A} = <K, T, M, q_0, H>$ powiemy, że jest d e t e r m i n i s t y c z n y. Automat skończony, który nie jest deterministyczny nazywamy n i e d e t e r m i n i s t y c z n y m.

Istnieje różnica między gramatykami (produkującymi słowa), a automatami (akceptującymi je). Automat bowiem nigdy się nie zapętli, czego nie możemy przecież powiedzieć o gramatykach. Automat zawsze przeanalizuje całe dowolne słowo nad danym alfabetem (gwarantuje nam to całkowitość relacji przejścia M, co czyni wszystkie automaty zupełnymi), a dopiero potem „powie”, czy je akceptuje, czy też nie. W przypadku zaś niezupełnych gramatyk generatywnych, jako że nie każde słowo nad danym alfabetem musi się dać w nich wygenerować, w związku z tym niektóre derywacje mogą się zatrzymać „w środku”, nie dając żadnego słowa. Tak dobre, jak automaty, są tylko gramatyki deterministyczne i zupełne. Gwarantują one bowiem wygenerowanie dowolnego słowa nad danym alfabetem. Dopiero wówczas to

sprawdza on, czy stan, w którym się znalazł jest końcowy, czy też nie (jest to odpowiednik sprawdzania w automatach, czy aktualny stan jest akceptowalny, czy też nie).

Prześledźmy pracę tych automatów (a tym samym zasadę ich działania) na konkretnych przykładach.

Przykład 6.1.

Automat deterministyczny $\mathcal{A}$ obserwuje pewne słowo, powiedzmy $A_1A_2A_3A_4$, a sam jest w stanie q_0. Będąc w tym stanie i czytając A_1 - przechodzi np. w stan q_1, wyznaczony przez funkcje M (tu: $M(q_0, A_1) = q_1$). Ponieważ M jest funkcją - więc wyznacza ona zawsze następny stan w sposób jednoznaczny, a ponieważ nadto jest całkowita - więc automat zawsze przechodzi do następnego stanu (nigdy nie „utknie" on wewnątrz analizowanego słowa). Niech więc dalej będzie np.:

$M(q_1, A_2) = q_2$,

$M(q_2, A_3) = q_3$,

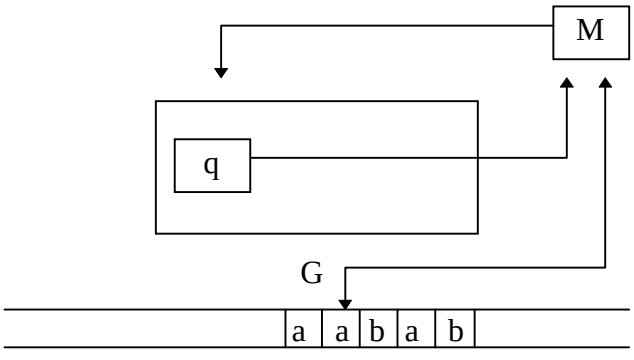
$M(q_3, A_4) = q_4$.

Tak więc, automat czytając po kolei litery: A_1 A_2 A_3 A_4,

zmienia następująco swe stany: q_0 q_1 q_2 q_3 q_4.

Jeśli teraz $q_4 \in H$ (tj. jest stanem akceptowalnym), to mówimy, że automat ten a k c e p t u j e to słowo; w przeciwnym razie słowo to nie jest akceptowane przez ten automat. □

Prace automatu można zilustrować również następującym rysunkiem:



Rys. 6.1.

Przesuwająca się taśma zawiera analizowane słowo. Czytając z niej (za pośrednictwem głowicy G) literę po literze - automat przesuwa ją za każdym razem o jedną kratkę w lewo (czyli przechodzi do następnej litery, bo w każdej kratce znajduje się jedna litera). Czytając daną jego literę - automat za pomocą funkcji (relacji) M zmienia swój stan z jednego na drugi (tu znajduje się w stanie q). Zbiór wszystkich stanów (K), alfabet (T), stan początkowy (q_0) i zbiór stanów końcowych

nie zostały tu uwidocznione, jako zbędne (nieistotne przy analizie danej litery badanego słowa).

Uwaga.

Oczywiście, zamiast mówić o przesuwającej się taśmie, możemy równoważnie mówić o przesuwającej się w drugą stronę głowicy. □

Przykład 6.2.

Niech $\mathcal{A}$ = <K, T, M, q_0, H>, gdzie K = {q_0,..., q_3}, T = {a, b}, H = {q_0}, a funkcja M jest określona w następujący sposób:

M:	a	b
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Tab. 6.1.

Rozpatrzmy, czy słowo bbabab jest akceptowalne przez powyższy automat.

Mamy: q_0 q_1 q_0 q_2 q_3 q_1 q_0 - kolejne stany

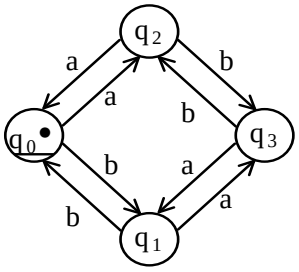
b b a b a b - litery czytanego słowa

Ponieważ zaś $q_0 \in H$, więc odpowiedź na powyższe pytanie jest twierdząca. □

Analogicznie jak dla gramatyk, również dla automatów możemy sporządzić wykres - tzw. wykres relacji (funkcji) przejścia. Wierzchołki opisują w nim stany automatu, a krawędzie oznaczamy symbolami alfabetu. Odpowiednikami reguł produkcji będą tu więc reguły relacji (funkcji) przejścia.

Przykład 6.2 - c.d.

Wykres powyższego automatu wygląda następująco:



Kropą oznaczono tu stan początkowy automatu, a podkreśleniem - każdy z jego stanów końcowych.

Rys. 6.2. □

Zadanie 6.1. Rozstrzygnij, jakie słowa akceptuje automat z przykładu 6.2. □

Przypomnijmy, że w automacie następuje przejście ze stanu q_i do stanu q_{i+1} witw, gdy zachodzi równość: M (q_i, czytany symbol) = q_{i+1}.

P r z e b i e g i e m automatu $\mathcal{A}$ nad wyrażeniem $a_1 \ldots a_n \in T^*$ $(n \geq 0)$ nazywamy ciąg stanów $q_0, \ldots, q_n$, taki że $M(q_i, a_{i+1}) = q_{i+1}$ dla $0 \leq i < n$.

Uwaga.

Zamiast pisać $M(q_i, a_{i+1}) = q_{i+1}$, piszemy też $q_i\, a_{i+1} \rightarrow q_{i+1}$, tj. wszystkie reguły funkcji przejścia mają postać $qa \rightarrow p$, gdzie $p,q \in K$, zaś $a \in T$.

Niech $\mathcal{A} = \langle K, T, M, K_0, H \rangle$, a X i Y niech będą słowami nad alfabetem KT^* (tj. $X, Y \in KT^*$). O słowach takich mówimy, że są k o n f i g u r a c j a m i a u t o m a t u $\mathcal{A}$. Z kolei mówimy, że automat $\mathcal{A}$ r e d u k u j e s ł o w o X d o s ł o w a Y w j e d n y m k r o k u (co oznaczamy: $X \underset{\mathcal{A}}{\rightarrow} Y$) witw, gdy istnieje trójka $(p,a,q) \in M$ oraz słowo $P \in T^*$ takie, że $X = paP$, a $Y = qP$. Mówimy, że $\mathcal{A}$ r e d u k u j e X do Y (co zapisujemy: $X \overset{*}{\underset{\mathcal{A}}{\rightarrow}} Y$) witw, gdy w automacie $\mathcal{A}$ istnieje taki ciąg redukcji, który doprowadza od słowa X do słowa Y (jest to oczywiście definicja indukcyjna względem długości słowa Y).

J ę z y k i e m r o z p o z n a w a n y m (a k c e p t o w a n y m) p r z e z a u t o m a t $\mathcal{A}$ nazywamy zbiór $L(\mathcal{A}) = \{P \in T^*: \exists_{q_0 \in K_0} \exists_{p \in H}: q_0 P \overset{*}{\underset{\mathcal{A}}{\rightarrow}} p\}$.

Jest to więc zbiór wszystkich słów akceptowalnych przez automat $\mathcal{A}$ (jako że opisany jest on przez te wszystkie słowa P, z których stworzone konfiguracje początkowe q_0P automat ten jest w stanie zredukować do któregokolwiek ze stanów końcowych $q_0 \in K_0$).

Zastanówmy się, kiedy automat akceptuje słowo puste λ, tj. kiedy mamy $p\lambda \rightarrow q$, gdzie $p \in K_0$, a $q \in H$. Korzystając z faktu, że w dowolnym automacie $\mathcal{A}$, $p \overset{*}{\underset{\mathcal{A}}{\rightarrow}} p$, oraz z tego, że $p\lambda = p$, otrzymujemy, że zachodzi to oczywiście tylko wtedy, gdy $p\lambda \overset{*}{\underset{\mathcal{A}}{\rightarrow}} p$, gdziep $\in K_0 \cap H$ (tj. gdy $\exists_{p \in K_0} \exists_{q \in H}: p = q$).

Mamy więc: $\lambda \in L(\mathcal{A}) \Leftrightarrow K_0 \cap H \neq \emptyset$.

Oczywiście, gdy $\mathcal{A}$ jest automatem deterministycznym, to definicja języka rozpoznawanego przez niego będzie wyglądała następująco:

$$L(\mathcal{A}) = \{P \in T^*: \exists_{p \in H}: q_0 P \overset{*}{\rightarrow} p\}$$

(definicja ta nie zawiera członu określającego istnienie pewnego stanu początkowego, gdyż jego kształt jest tu jednoznacznie zdeterminowany).

Język akceptowany przez automat skończony można zdefiniować również w inny, poniżej przedstawiony sposób.

Niech $r = (q_0, ..., q_n)$ będzie przebiegiem automatu $\mathcal{A}$ nad wyrażeniem $P = a_1 a_2 ... a_n \in T^*$. Oznaczmy:

- $p(r) = q_0$ („p" - od „początek"),
- $k(r) = q_n$ („k" - od „koniec"),
- przez $RUN_{\mathcal{A}}(P)$ - zbiór wszystkich przebiegów $\mathcal{A}$ nad $P \in T^*$.

Mówimy, że automat $\mathcal{A}$ akceptuje wyrażenie $P \in T^*$ witw, gdy

$$\exists_{r \in RUN_{\mathcal{A}}(P)}: p(r) \in K_0, \text{ a } k(r) \in H.$$

Przez $\overline{RUN}_{\mathcal{A}}(P)$ oznaczmy zbiór wszystkich przebiegów właściwych automatu $\mathcal{A}$ nad słowem P, zdefiniowany w następujący sposób: $\overline{RUN}_{\mathcal{A}}(P) = \{r \in RUN_{\mathcal{A}}(P): p(r) \in K_0\}$.

Wówczas $L(\mathcal{A})$ możemy zdefiniować następująco:

$$L(\mathcal{A}) = \{P \in T^*: \exists_{r \in} \overline{RUN}_{\mathcal{A}}(P): k(r) \in H\}.$$

Gdy $\mathcal{A}$ jest automatem deterministycznym, to dla każdego $P \in T^*$ istnieje oczywiście dokładnie jeden przebieg właściwy $\mathcal{A}$ nad P.

Przykład 6.3.

Skonstruujmy automat (i narysujmy jego wykres), który akceptuje dowolne wyrażenie $P \in \{0, 1\}^*$ takie, że $10 \overset{\subseteq}{e} P$ (czyli P zaczyna się od 10, tj. $P = 10P_1$, gdzie $P_1 \in \{0, 1\}^*$).

$T = \{0,1\}$,

$K = \{q_0, q_1, q_2, q_3\}$,

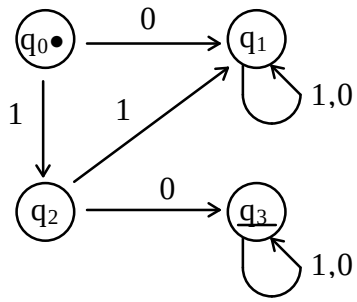
$K_0 = \{q_0\}$,

$H = \{q_3\}$,

$M = \{(q_0, 0, q_1), (q_0, 1, q_2), (q_2, 0, q_3), (q_2, 1, q_1), (q_3, 0, q_3), (q_3, 1, q_3),$
$(q_1, 0, q_1), (q_1, 1, q_1)\}$

(zapis (q_i, a_k, q_j) czytamy: automat będąc w stanie q_i, po przeczytaniu a_k przechodzi w stan q_j).

Jego wykres wygląda następująco:



Kropa przy symbolu q_0 oznacza, że jest on stanem początkowym, a podkreślenie symbolu q_3 oznacza, że jest on stanem akceptowalnym.

Z wykresu widzimy, że q_1 jest „stanem ucieczki", a nadto, że nie jest on akceptowalny (tj. że $q_1 \notin H$). □

Rys. 6.3.

Zadanie 6.2. Zbuduj automat akceptujący jedynie twoje nazwisko. □

2. Dwa twierdzenia Kleene'go

Uważni czytelnicy spostrzegli zapewne podobieństwo wykresów gramatyk regularnych z wykresami automatów skończonych. Nie jest ono przypadkowe. Mówią o tym dwa następujące twierdzenia, stwierdzające wzajemnie jednoznaczną odpowiedniość pomiędzy gramatykami regularnymi, a automatami skończonymi.

Twierdzenie 6.1 - I twierdzenie Kleene'go (czyt.: Kliniego).

Dla każdego niedeterministycznego automatu $\mathcal{A}$ istnieje gramatyka regularna G taka, że $L(\mathcal{A}) = L(G)$.

Dowód.

Niech $\mathcal{A} = <K, T, M, K_0, H>$ będzie automatem niedeterministycznym.

Gramatykę $G = <V_N, V_T, S, F>$ definiujemy w następujący sposób:

- $V_N = K \cup \{S\}$,
- $V_T = T$,
- S jest symbolem początkowym,

a w F wyróżniamy cztery następujące rodzaje reguł produkcji:

1) $\lceil q_0\, a \rightarrow p \rceil \in M$ (gdzie $q_0 \in K_0$, $a \in T$, a $p \in K$) $\Leftrightarrow \lceil p \rightarrow a \rceil \in F$
(gdzie $p \in V_N$, zaś $a \in V_T$),

2) $\lceil qa \rightarrow p \rceil \in M$ (gdzie $q \in K \setminus K_0$, $a \in T$, a $p \in K$) $\Leftrightarrow \lceil p \rightarrow qa \rceil \in F$
(gdzie $p, q \in V_N$, zaś $a \in V_T$),

3) $p \in H \Leftrightarrow \ulcorner S \to p \urcorner \in F$,

4) $K_0 \cap H \neq \varnothing$ (jest to warunek konieczny i dostateczny, aby $\lambda \in L(\mathcal{A}) \Leftrightarrow \ulcorner S \to \lambda \urcorner \in F$).

Z punktu 4) wnosimy, że $\lambda \in L(\mathcal{A}) \Leftrightarrow \lambda \in L(G)$. W ten sposób sprawę rozpatrywania słowa pustego mamy już zakończoną.

Weźmy więc $P \in L(\mathcal{A})$, gdzie $P \neq \lambda$. Zatem $\exists_{q_0 \in K_0} \exists_{p \in H}: q_0 P \overset{*}{\underset{\mathcal{A}}{\to}} p$.

Wówczas automat $\mathcal{A}$ analizuje słowo $P = a_1 a_2 \ldots a_n$ np. w następujący sposób:

$$q_0 a_1 a_2 \ldots a_{n-1} a_n \to q_1 a_2 \ldots a_{n-1} a_n \to \ldots \to q_{n-2} a_{n-1} a_n \to q_{n-1} a_n \to q_n = p \in H$$

(w pierwszym kroku zastosowano tu regułę 1), a we wszystkich pozostałych - regułę 2)).

W G mamy więc derywację:

$S \to q_n \to q_{n-1} a_n \to q_{n-2} a_{n-1} a_n \to \ldots \to q_1 a_2 \ldots a_{n-1} a_n \to a_1 a_2 \ldots a_n$, gdzie:

- pierwsza reguła $\ulcorner S \to q_n \urcorner$ należy do F na mocy „$q_n = p \in H$” (reguła 3)),
- ostatnia stosowana reguła $\ulcorner q_1 \to a_1 \urcorner$ należy do F na mocy „$\ulcorner q_0 a_1 \to q_1 \urcorner \in M$” (reguła 1)),
- pozostałe reguły (kształtu $q_k \to q_{k-1} a_k$) należą do F na mocy tego, że reguły kształtu $q_{k-1} a_k \to q_k$ należą do M (reguły 2)).

Tak więc derywacja w gramatyce G musi zastartować od reguły typu 3), następnie mamy ciąg reguł typu 2), a na koniec regułę typu 1). Mamy więc:

$$S \overset{3}{\to} q \overset{2}{\to} \ldots \overset{2}{\to} p a_2 \ldots a_{n-1} a_n \overset{1}{\to} a_1 a_2 \ldots a_n, \text{ gdzie } q \in H.$$

W ten sposób pokazaliśmy, że każde słowo języka $L(\mathcal{A})$ jest słowem języka L(G), tj. że $L(\mathcal{A}) \subseteq L(G)$. Inkluzję w drugą stronę pokazuje się w analogiczny sposób.

Pozostaje nam jeszcze udowodnić, że tak skonstruowana gramatyka jest regularna. Wszystkie reguły typu 3) można prosto usunąć z niej, zastępując je regułami typu 2) (mianowicie, zastępując każdą z reguł postaci $S \to q$ zbiorem reguł postaci $S \to pa$, o ile tylko $\ulcorner q \to pa \urcorner \in F$). Zauważmy, że reguły typu 1), 2) i 4) (z odpowiednika twierdzenia 3.1 dla gramatyk lewostronnie liniowych) są lewostronnie liniowe, a więc gramatyka przez nie wyznaczona jest lewostronnie liniowa, zatem w oparciu o ostatni z wniosków z twierdzenia 3.4 istnieje równoważna jej gramatyka prawostronnie liniową, czyli regularna. □

Twierdzenie 6.2 - II twierdzenie Kleene’go.

Dla każdej gramatyki regularnej G istnieje skończony automat $\mathcal{A}$ taki, że $L(G) = L(\mathcal{A})$.

Dowód.

Niech $G = <V_N, V_T, S, F>$ będzie gramatyką regularną. O ile G nie jest w postaci normalnej, to sprowadzamy ją do niej (możemy tego dokonać na podstawie twierdzenia 3.1, według procedury podanej w jego dowodzie).

Zatem reguły produkcji w G są jedynie kształtu $X \to aY$ i $X \to \lambda$, gdzie $X, Y \in V_N$, zaś $a \in V_T$.

Konstruujemy automat $\mathcal{A}$ postaci $\mathcal{A} = <K, T, M, q_0, H>$, w którym obieramy:

- $T = V_T$,
- $K = V_N$,
- $q_0 = S$,
- $H = \{Z \in V_N (=K): \ulcorner Z \to \lambda \urcorner \in F\}$,

a $\ulcorner Xa \to Y \urcorner \in M \Leftrightarrow \ulcorner X \to aY \urcorner \in F$ (ponieważ w F mogą być zarazem reguły $X \to aY_1$ i $X \to aY_2$, bo G może być niedeterministyczna, więc w M mogą być odpowiadające im reguły $Xa \to Y_1$ i $Xa \to Y_2$, czyli automat $\mathcal{A}$ może być niedeterministyczny, a za M musimy przyjąć ogólnie, że jest relacją, a nie koniecznie funkcją sensu stricto).

Każda derywacja w G przebiega według schematu:

$$S \to a_1 X_1 \to a_1 a_2 X_2 \to a_1 a_2 a_3 X_3 \to \ldots \to a_1 a_2 a_3 \ldots a_n X_n \to a_1 a_2 a_3 \ldots a_n.$$

Wyżej zdefiniowany automat $\mathcal{A}$ rozpoznaje to wyrażenie $a_1a_2a_3...a_n$ w następujący sposób:

$$Sa_1 a_2 a_3 \ldots a_n \to X_1 a_2 a_3 \ldots a_n \to X_2 a_3 \ldots a_n \to \ldots \to X_{n+1} a_n \to X_n.$$

Stosowaliśmy tu po kolei reguły z M, odpowiadające stosowanym wyżej regułom z F. I tak:

– w pierwszym kroku stosowaliśmy regułę $\ulcorner Sa_1 \to X_1 \urcorner \in M$, odpowiadającą stosowanej w powyższej derywacji w pierwszym kroku regule $\ulcorner S \to a_1 X_1 \urcorner \in F$,
– w drugim kroku stosowaliśmy regułę $\ulcorner X_1 a_2 \to X_2 \urcorner \in M$, odpowiadającą stosowanej w powyższej derywacji w drugim kroku regule $\ulcorner X_1 \to a_2 X_2 \urcorner \in F$,... itd.

Oczywiście $X_n \in H$, bo do F należy reguła $X_n \to \lambda$, o czym świadczy ostatni krok w powyższej derywacji. Zatem słowo $a_1a_2a_3...a_n$ jest akceptowalne przez automat $\mathcal{A}$.

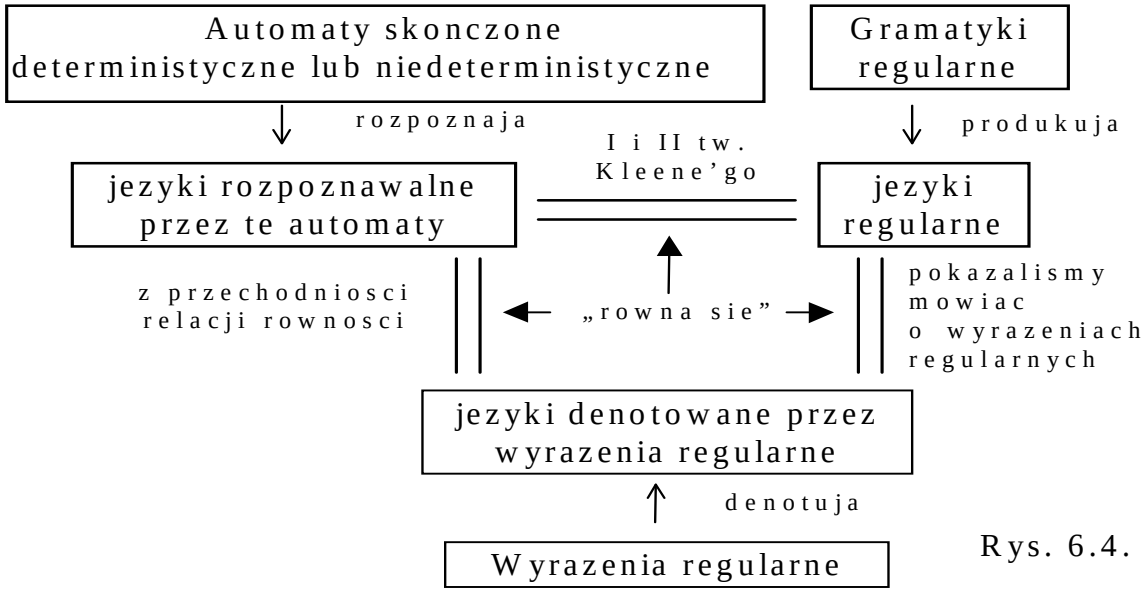
W ten sposób pokazaliśmy, że każde słowo języka L(G) jest słowem języka $L(\mathcal{A})$, tj. że $L(G) \subseteq L(\mathcal{A})$. Inkluzję w drugą stronę pokazuje się w analogiczny sposób. □

Wniosek (z I i II tw. Kleene'go).

Klasa języków rozpoznawalnych przez automaty niedeterministyczne jest identyczna z klasa języków regularnych ($\mathcal{L}_3$). □

Załóżmy, że pokazaliśmy, że klasa języków rozpoznawalnych przez automaty niedeterministyczne jest identyczna z klasa języków rozpoznawalnych przez automaty deterministyczne (mówi o tym twierdzenie Scotta - tw. 6.3).

Mamy wówczas:



Rys. 6.4.

Tak więc dodatkowo, klasa języków rozpoznawalnych przez automaty skończone pokrywa się z klasa języków denotowanych przez wyrażenia regularne.

Na koniec tego paragrafu zauważmy jeszcze, że tak analizy, jak i syntezy automatu (tj. odpowiednio określenia, jakie słowa automat akceptuje i zbudowania gramatyki generującej dany język) dokonuje się identycznie, jak to miało miejsce z analizą i syntezą gramatyk (patrz trzeci paragraf trzeciego rozdziału).

3. Twierdzenie Scotta i jego konsekwencje

Twierdzenie Scotta jest dla automatów skończonych odpowiednikiem stosowanego dla gramatyk regularnych twierdzenia Myhilla. Zanim jednak je podamy, wprowadźmy niezbędną nam definicję występującego w nim pojęcia:

Mówimy, że automaty $\mathcal{A}$ i $\mathcal{B}$ są *równoważne* witw, gdy $L(\mathcal{A}) = L(\mathcal{B})$.

Twierdzenie 6.3 (Scotta).

Dla dowolnego skończonego niedeterministycznego automatu $\mathcal{A}$ można skonstruować deterministyczny automat $\mathcal{B}$ taki, że $\mathcal{A}$ i $\mathcal{B}$ są równoważne i $T_{\mathcal{A}} = T_{\mathcal{B}}$ (gdzie $T_{\mathcal{A}}$ i $T_{\mathcal{B}}$ są ich alfabetami). □

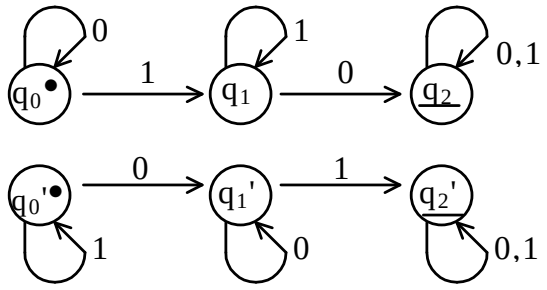
Zamiast dowodu tego twierdzenia, na konkretnym przykładzie prześledzimy metodę konstrukcji takiego automatu.

Przykład 6.4.

Niech T = {0, 1}. Język L($\mathcal{A}$) = {P ∈ T^*: N_0 (P) > 1 i N_1 (P) > 1} (tj. taki, że w każdym jego słowie występuje co najmniej jedno zero i co najmniej jedna jedynka) akceptuje automat o $T_{\mathcal{A}}$=T j.w., z K = {q_0, q_1, q_2, q_0', q_1', q_2'},

M = {(q_0, 0, q_0), (q_0, 1, q_1), (q_0', 0, q_1'), (q_0', 1, q_0'),
(q_1, 0, q_2), (q_1, 1, q_1), (q_1', 0, q_1'), (q_1', 1, q_2'),
(q_2, 0, q_2), (q_2, 1, q_2), (q_2', 0, q_2'), (q_2', 1, q_2')},

$K_0^{\mathcal{A}}$ = {q_0, q_0'} i H = {q_2, q_2'}.



Rys. 6.5.

Jest to oczywiście automat niedeterministyczny. Bardzo dobrze to widać chociażby po jego wykresie (rys. 6.5).

Zgodnie z powyższym twierdzeniem, możemy skonstruować równoważny mu deterministyczny automat $\mathcal{B}$ = <$K_{\mathcal{B}}$, $T_{\mathcal{B}}$, $M_{\mathcal{B}}$, $Q_0^{\mathcal{B}}$, $H_{\mathcal{B}}$> taki, że $T_{\mathcal{B}}$ = $T_{\mathcal{A}}$. Z założenia obieramy więc $T_{\mathcal{A}}$ = $T_{\mathcal{B}}$, a następnie za $Q_0^{\mathcal{B}}$ przyjmujemy zbiór stanów początkowych automatu $\mathcal{A}$: $Q_0^{\mathcal{B}}$ = {q_0, q_0'}. W następnym (najdłuższym) kroku, podobnie jak to miało miejsce w przypadku gramatyk, znajdując kolejne wartości funkcji przejścia M', definiujemy nowe stany automatu $\mathcal{B}$ oraz określamy, które z nich są akceptowalne (podobnie jednak jak tam, zamiast M' pisać będziemy „S” od „Scotta”). Najpierw otrzymujemy więc:

S($Q_0^{\mathcal{B}}$, 0) = {q_0, q_1'} = $Q_1^{\mathcal{B}}$.

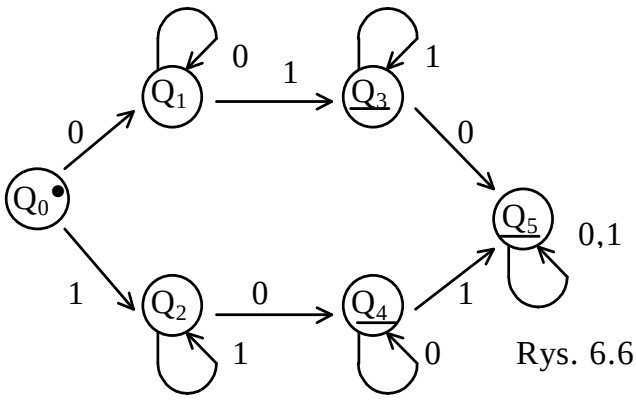
Pierwszą równość znajdujemy tu rozumując następująco: „automat $\mathcal{A}$ będąc w jednym ze stanów $Q_0^{\mathcal{B}}$, tj. w stanie q_0 lub w stanie q_0', po przeczytaniu 0 przechodzi odpowiednio w stan q_0 i q_1', tj. w sumie w jeden ze stanów ze zbioru {q_0, q_1'}”. Drugą równość otrzymujemy rozumując następująco: „ponieważ stan {q_0, q_1'} automatu $\mathcal{A}$

nie był dotychczas zdefiniowany, więc nadajemy mu nową nazwę $Q_1^{\mathcal{B}}$". Ze względu na to, że żaden ze stanów ze zbioru $\{q_0, q_1'\}$ nie jest stanem końcowym automatu $\mathcal{A}$, zatem i stan $Q_1^{\mathcal{B}}$ nie jest stanem końcowym automatu $\mathcal{B}$, a więc go nie podkreślamy (w przeciwnym razie - gdyby choć jeden ze stanów ze zbioru $\{q_0, q_1'\}$ był stanem końcowym automatu $\mathcal{A}$ - wówczas i stan $Q_1^{\mathcal{B}}$ byłby stanem końcowym automatu $\mathcal{B}$ i podkreślilibyśmy go). Idąc dalej tą samą drogą - kolejno otrzymujemy (czytaj kolumnami):

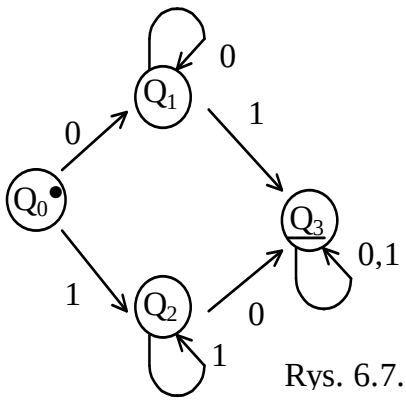
$S(Q_0^{\mathcal{B}}, 1) = \{q_1, q_0'\} = Q_2^{\mathcal{B}}$,
$S(Q_1^{\mathcal{B}}, 0) = \{q_0, q_1'\} = Q_1^{\mathcal{B}}$,
$S(Q_1^{\mathcal{B}}, 1) = \{q_1, \underline{q_2'}\} = \underline{Q_3^{\mathcal{B}}}$,
$S(Q_2^{\mathcal{B}}, 0) = \{\underline{q_2}, q_1'\} = \underline{Q_4^{\mathcal{B}}}$,
$S(Q_2^{\mathcal{B}}, 1) = \{q_1, q_0'\} = Q_2^{\mathcal{B}}$,
$S(Q_3^{\mathcal{B}}, 0) = \{\underline{q_2}, \underline{q_2'}\} = \underline{Q_5^{\mathcal{B}}}$,
$S(Q_3^{\mathcal{B}}, 1) = \{q_1, \underline{q_2'}\} = \underline{Q_3^{\mathcal{B}}}$,
$S(Q_4^{\mathcal{B}}, 0) = \{\underline{q_2}, q_1'\} = \underline{Q_4^{\mathcal{B}}}$,
$S(Q_4^{\mathcal{B}}, 1) = \{\underline{q_2}, \underline{q_2'}\} = \underline{Q_5^{\mathcal{B}}}$,
$S(Q_5^{\mathcal{B}}, 0) = \{\underline{q_2}, \underline{q_2'}\} = \underline{Q_5^{\mathcal{B}}}$,
$S(Q_5^{\mathcal{B}}, 1) = \{\underline{q_2}, \underline{q_2'}\} = \underline{Q_5^{\mathcal{B}}}$.

Procedurę tę kontynuowaliśmy tak długo, jak długo to tylko było możliwe.
W ten sposób otrzymaliśmy zbiór stanów $K_{\mathcal{B}} = \{Q_0^{\mathcal{B}}, Q_1^{\mathcal{B}}, Q_2^{\mathcal{B}}, Q_3^{\mathcal{B}}, Q_4^{\mathcal{B}}, Q_5^{\mathcal{B}}\}$ oraz zbiór stanów końcowych $H_{\mathcal{B}} = \{\{q_1, q_2'\}, \{q_2, q_1'\}, \{q_2, q_2'\}\} = \{Q_3^{\mathcal{B}}, Q_4^{\mathcal{B}}, Q_5^{\mathcal{B}}\}$ (składa się on z tych wszystkich stanów zbioru $K_{\mathcal{B}}$, które zawierają co najmniej jeden ze stanów końcowych q_2, q_2' automatu $\mathcal{A}$).

Wykres tak otrzymanego automatu przedstawia rys. 6.6 (dla prostoty opuściliśmy w nim górne indeksy przy Q, co zresztą można już było zrobić wcześniej).



Rys. 6.6.



Rys. 6.7.

Automat ten właściwie można dalej zredukować do równoważnego mu automatu, którego to wykres przedstawiony jest na rys. 6.7. □

Z przykładu tego widać, że automat deterministyczny otrzymuje się w sposób podobny, jak deterministyczne gramatyki regularne. Tu jednak w realizacji funkcji M, nigdy nie otrzymujemy zbioru pustego, a więc nie będzie kwestii zajmowania się symbolem ucieczki. Wynika to z faktu, że już bazowy automat (jak to każdy automat) jest zupełny.

Twierdzenie Scotta daje nam w sumie równoważność klasy automatów deterministycznych z klasą automatów skończenie-stanowych. Wynika z tego, że automaty niedeterministyczne niczego nowego nie wnoszą (jeśli chodzi o „siłę akceptacyjną") do klasy automatów skończonych.

Rozpatrzmy, jakie inne konsekwencje wypływają jeszcze z twierdzenia Scotta.

Twierdzenie 6.4.

Zbiór wszystkich języków regularnych nad tym samym alfabetem jest zamknięty na operację dopełnienia.

Dowód.

Niech $L \in \mathcal{L}_3$. Zatem z twierdzenia Kleene'go $L = L(\mathcal{A})$, gdzie $\mathcal{A}$ jest automatem deterministycznym. Język $\overline{L} = T^* \setminus L$ jest językiem akceptowalnym przez automat skończony $\overline{\mathcal{A}} = \langle K, T, M, q_0, K \setminus H \rangle$, a zatem (z twierdzenia Kleene'go) jest językiem regularnym. □

Wniosek 1.

Zbiór wszystkich języków regularnych nad tym samym alfabetem z operacjami dopełnienia ($\overline{\ }$), przekroju ($\cap$) i sumy ($\cup$) tworzy algebrę Boole'a.

Dowód.

Niech $V_T = \{a_1, a_2, ..., a_n\}$.

Jedynką tworzonej algebry nad alfabetem V niech będzie V^* (zbiór wszystkich słów nad alfabetem V) generowany przez gramatykę $G = \langle V_N, V_T, S, F \rangle$, gdzie $V_N = \{S\}$, a $F = \{S \to Sa_i : 1 \le i \le n\} \cup \{S \to \lambda\}$. Ponieważ jest to gramatyka regularna, więc V^* jest językiem regularnym.

Jej zerem będzie język pusty ($\varnothing$), generowany przez gramatykę regularną $G = \langle V_N, V_T, S, F \rangle$, gdzie $V_N = S$, a $F = \{S \to S\}$ (będzie więc to również język regularny).

$\mathcal{L}_3$ jest zamknięta na sumę (z twierdzenia 1.1). Stąd oraz z powyższego twierdzenia otrzymujemy, że $\overline{\overline{L_1} \cup \overline{L_2}}$ jest językiem regularnym (bo skoro L_1 i L_2 są językami re-

gularnymi, więc $\overline{L_1}$ i $\overline{L_2}$ są językami regularnymi, zatem $\overline{L_1} \cup \overline{L_2}$ jest językiem regularnym, a więc i $\overline{\overline{L_1} \cup \overline{L_2}}$ jest językiem regularnym). Ponieważ z kolei $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$, więc $L_1 \cap L_2$ jest językiem regularnym, a zatem zbiór wszystkich języków regularnych nad tym samym alfabetem jest zamknięty na operację przecięcia. Pozostaje nam jeszcze sprawdzenie 12 aksjomatów algebry Boole'a (patrz [1], str. 83), co pozostawiam już czytelnikowi. □

Wniosek 2.

Niech $R \in \mathcal{L}_3$, a $L \in \mathcal{L}_2$. Wówczas problem „$L \subseteq R$" jest rozstrzygalny.

Dowód.

Problem „$L \subseteq R$" jest równoważny problemowi „$L \cap \overline{R} = \varnothing$". Ponieważ R jest językiem regularnym, zatem i $\overline{R}$ jest językiem regularnym (z twierdzenia). Wiemy nadto, że klasa języków bezkontekstowych $\mathcal{L}_2$ jest zamknięta na przecięcie z elementami klasy $\mathcal{L}_3$ (z twierdzenia 3.5). Zatem $L \cap \overline{R}$ jest językiem bezkontekstowym. Jednak problem pustości języka bezkontekstowego ($\varnothing = L \cap \overline{R} \in \mathcal{L}_2$) jest rozstrzygalny (patrz twierdzenie 2.4). □

Wniosek 3.

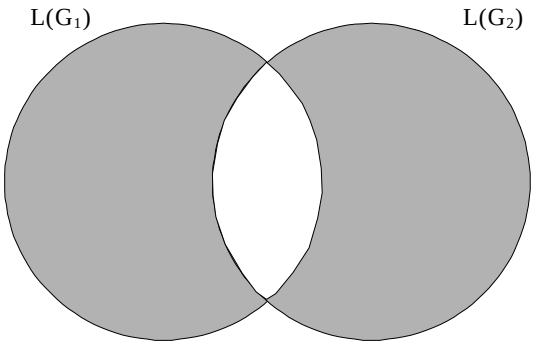
Niech G_1 i G_2 będą gramatykami regularnymi. Problem równości języków generowanych przez te gramatyki jest rozstrzygalny.

Dowód.

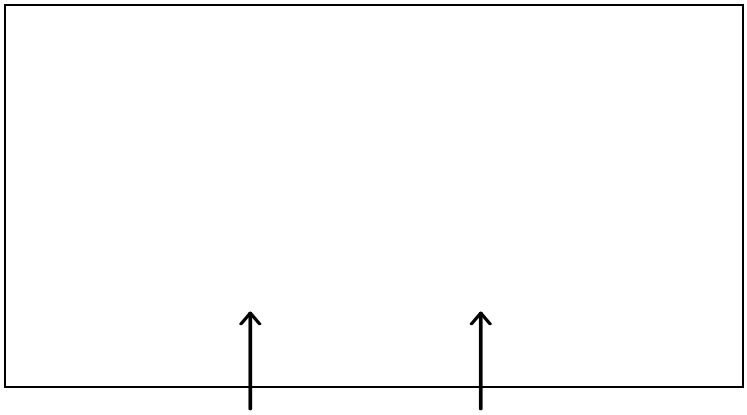
Utwórzmy język $L_3 = (L(G_1) \cap \overline{L(G_2)}) \cup (\overline{L(G_1)} \cap L(G_2))$.

Jest to język regularny, bo:

- skoro G_1 i G_2 są gramatykami regularnymi, więc $L(G_1)$ i $L(G_2)$ są językami regularnymi, a zatem i również $\overline{L(G_1)}$ i $\overline{L(G_2)}$ są językami regularnymi (z twierdzenia),
- zatem również $L(G_1) \cap \overline{L(G_2)}$ i $\overline{L(G_1)} \cap L(G_2)$ są językami regularnymi (z pierwszego wniosku), a stąd właśnie (na mocy tw. 1.1) także



$L_3 = (L(G_1) \cap \overline{L(G_2)}) \cup (\overline{L(G_1)} \cap L(G_2))$ jest językiem regularnym.



Rys. 6.8.

Język L_3 (z definicji) jest zbiorem tych słów języka $L(G_1)$, które nie są słowami języka $L(G_2)$, powiększonym o zbiór tych słów języka $L(G_2)$, które nie są słowami języka $L(G_1)$ (patrz rys. 6.8.). O ile tylko każdy z tych zbiorów będzie pusty (a więc i suma tych zbiorów, będąca zbiorem L, będzie zbiorem pustym) - to języki $L(G_1)$ i $L(G_2)$ będą się pokrywały. Tak więc problem „$L(G_1) = L(G_2)$” jest równoważny problemowi „$L_3 = \varnothing$”. Ponieważ jednak problem ten jest rozstrzygalny (bo z twierdzenia 2.4 wiemy, że jest to problem rozstrzygalny dla języków bezkontekstowych, a każdy język regularny jest bezkontekstowy) - zatem kończy to dowód wniosku. □

Zadanie 6.4. Niech T = {a,b}. Przez blok rozumiemy maksymalny ciąg sąsiednich identycznych liter w słowie (tak np. mamy 4 bloki w słowie aababbb). Zbuduj automat akceptujący tylko i wyłącznie wszystkie słowa:

a) o parzystej liczbie liter,

b) o parzystej liczbie liter, zaczynające się od „ab”,

c) zaczynające i kończące się na „a”,

d) o nieparzystej liczbie bloków, zaczynające się od „a”,

e) o nieparzystej liczbie bloków,

f) o parzystej liczbie bloków,

g) o mocy każdego z bloków równej co najwyżej 3,

h) o długości co najmniej 3,

i) o długości drugiego bloku równej dokładnie 2,

j) o dokładnie 3 blokach,

k) zawierających frazę „aa” lub „bb”,

l) zawierających frazę „aa” lub „ababa”.

O ile skonstruowany przez Ciebie automat nie jest deterministyczny - znajdź również odpowiadający mu automat deterministyczny. Uprość maksymalnie tak otrzymany automat (patrz: przykład 6.4). □

Zadanie 6.5. Niech T będzie zbiorem cyfr, powiększonym o przecinek (','): T = {0,1,2,3,4,5,6,7,8,9} ∪ {','}. Skonstruuj automat nad tym alfabetem, akceptujący tylko i wyłącznie wszystkie słowa będące:

a) liczbami naturalnymi,
b) dodatnimi liczbami wymiernymi o skończonym zapisie dziesiętnym,
c) dodatnimi ułamkami właściwymi,
d) liczbami naturalnymi zbudowanymi jedynie z cyfr parzystych,
e) liczbami dodatnimi o części ułamkowej złożonej z co najwyżej 3 cyfr,
f) liczbami dodatnimi o części ułamkowej złożonej z co najwyżej 3 cyfr znaczących,
g) liczbami dodatnimi o części ułamkowej nie zaczynającej się od zera (tu wliczają się również wszystkie liczby dodatnie bez części ułamkowej, tj. wszystkie liczby naturalne),
h) liczbami dodatnimi o części całkowitej mniejszej niż tysiąc.

Zastosuj się do polecenia zamieszczonego na końcu poprzedniego zadania. □

Zadanie 6.6. Niech T = {0,1,2,3,4,5,6,7,8,9}. Skonstruuj automat nad tym alfabetem, akceptujący tylko i wyłącznie wszystkie słowa będące liczbami ze zbioru N_0 podzielnymi przez:

a) 2, d) 5 g) 8
b) 3, e) 6 h) 9
c) 4 f) 7 (to najtrudniejsze!) i) 10

Zastosuj się do polecenia zamieszczonego na końcu zadania 6.4.

Zadanie 6.7. Niech T = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. Skonstruuj automat nad tym alfabetem, akceptujący tylko i wyłącznie wszystkie słowa będące liczbami – nazwani lat przestępnych naszej ery (uwzględnij zmianę kalendarza w XVI wieku z juliańskiego na gregoriański). Zastosuj się do polecenia zamieszczonego na końcu zadanie 6.4.

Wskazówka: Wykorzystaj automat z zadania 6.6 c).

Zadanie 6.8. Niech T = {a, b}. Skonstruuj automat nad tym alfabetem, akceptujący tylko i wyłącznie wszystkie słowa, w których:

a) tylko jeden blok może mieć długość większą niż 2,
b) tylko jeden blok może mieć długość większą niż 2,
c) istnieje co najmniej jeden blok długości dokładnie 1,
d) jest parzysta liczba bloków o parzystej długości.

Zastosuj się do polecenia zamieszczonego na końcu zadanie 6.4.