Teoretyczne podstawy informatyki

Aleksander Wojdyga

26 listopada 2010

Zalecana literatura: Wprowadzenie do teorii automatów, języków i obliczeń (*Introduction to Automata Theory, Languages and Computation*) John E. Hopcroft, Jeffrey D. Ullman, Wydawnictwo Naukowe PWN 2003 r.

1 Wprowadzenie do języków i automatów

1. Co to jest alfabet, łańcuch / słowo (słowo puste ϵ), przyrostek, przedrostek. Przykłady – $\{0, 1\}$, $\{0, 1, .., 9\}$, $\{a, b\}$, $\{a, , b, c, ..., z, , \}$, ...

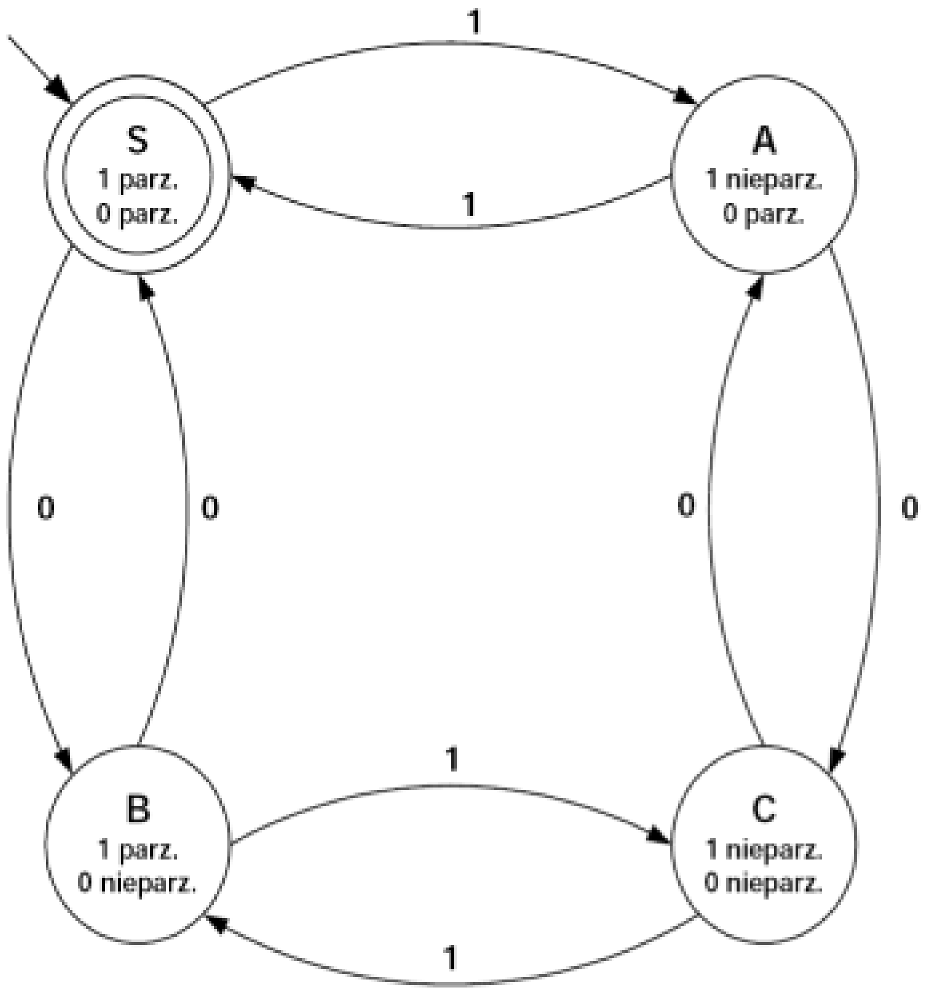
2. Język nad alfabetem, operacje na językach (gwiazdka Kleene'go).

3. Przykłady języków: język polski, liczby binarne, palindromy w języku polskim, palindromy nad alfabetem $\{a, b\}$ – kilka sposobów.

4. Definicja automatu skończonego (deterministycznego), objaśnienie symboli, zależności. Przedstawienie jako grafu oraz jako głowicy i taśmy.

5. Przykład - automat (`http://pluton.pol.lublin.pl/~awojdyga/dydaktyka/tpi-z/automat1.png`) akceptujący słowo binarne z parzystą ilością zer i jedynek (dokładne omówienie).
   Konwencje stosowane w rysunku automatu skończonego:

   - stan startowy oznaczany jest za pomocą strzałki wejściowej
   - strzałki są etykietowane symbolami alfabetu
   - stany akceptujące rysowane są grubą lub podwójną kreską



Zadania

(a) Podać automat skończony akceptujący ciągi 0/1 zawierające podciąg 11.

(b) Przedstawić automat skończony, który akceptuje rzeczownik "kot" we wszystkich przypadkach.

(c) Przedstawić automat skończony, akceptujący poprawne (bez zbędnego zera na początku) liczby binarne.

(d) Podać automat akceptujący ruchy skoczka (figury szachowej).

(e) Podać automat skończony akceptujący ciągi 0/1 zawierające podciąg 11 co najwyżej jeden raz.

(f) Podaj automat skończony akceptujący liczbę podzielną przez

  i. 2 (prosty)

ii. 3 (stan będzie przechowywał resztę)
iii. 4 (stan przechowuje resztę oraz drugą cyfrę z pary) (*)

2 Minimalizacja AS

Minimalizację automatu skończonego przeprowadzamy w celu zmniejszenia liczby stanów (a przez to i funkcji przejścia) usuwając stany nieosiągalne i nierozróżnialne.

Definicja 1 *Stan q jest nieosiągalny, jeśli nie da się do niego przejść ze stanu początkowego q_0 za pomocą żadnego słowa* ($\forall w \in \Sigma^* \delta^*(q_0, w) \neq q$).

Definicja 2 *Stany q_1 i q_2 są nierozróżnialne, jeśli dla każdego słowa $w \in \Sigma^*$ ze stanu q_1 da się przejść do stanu q_1', a z q_2 do q_2' ($q_1' = \delta^*(q_1, w)$, $q_2' = \delta^*(q_2, w)$) i stany te spełniają warunek* $(q_1' \in F \wedge q_2' \in Q - F) \vee (q_1' \in Q - F \wedge q_2' \in F)$.

Dwa stany są nierozróżnialne jeśli da się dojść za pomocą każdego słowa do dwu stanów końcowych albo do dwu stanów niekońcowych. Jeśli dwa stany są rozróżnialne, to są rozróżnialne za pomocą słowa długości co najwyżej $|Q| - 1$.

2.1 Algorytm minimalizacji AS

Wejście: $A = (Q, \Sigma, \delta, q_0, F)$
Wyjście: $A' = (Q', \Sigma, \delta', q_0', F')$

Wersja o złożoności wykładniczej $O(|\Sigma|^{|Q|})$

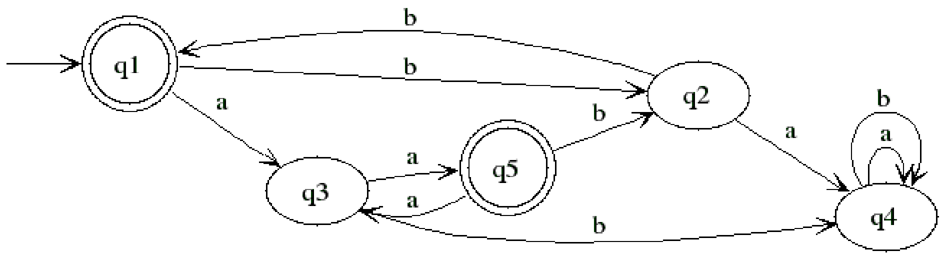
1. Usuń stany nieosiągalne.
2. Dla słów w o długości zero przyjmij za nierozróżnialne dwa podzbiory stanów: wszystkie stany z F oraz wszystkie stany z $Q - F$.
3. Analizuj w sposób systematyczny słowa o wzrastającej długości, w ten sposób uzyskasz kolejne podzbiory stanów nierozróżnialnych (klasy równoważności stanów), w każdym kroku niemniej niż w poprzednim.
4. Jeśli liczba podzbiorów uzyskana w pewnym kroku jest równa liczbie podzbiorów uzyskanych w kroku poprzednim - zakończ analizowanie słów.
5. Utwórz zbiór stanów Q' jako zbiór klas równoważności stanów nierozróżnialnych, zbiór stanów końcowych F', funkcję przejścia δ' – odpowiednio do Q'.

Wersja o złożoności wielomianowej $O(|\Sigma|^2|Q|)$

1. Usuń stany nieosiągalne.
2. Zaznacz wszystkie pary stanów, które składają się z jednego stanu akceptującego i jedno nieakceptującego.
3. dla każdej niezaznaczonej pary stanów $\{p, q\}$ jeśli dla jakiejkolwiek litery s para $\{\delta(p, s), \delta(q, s)\}$ jest zaznaczona, to zaznacz parę $\{p, q\}$
4. Dwa stany są nierozróżnialne jeśli są nie zaznaczone (ulegają "sklejeniu")
5. Odtwórz nową funkcję przejścia.

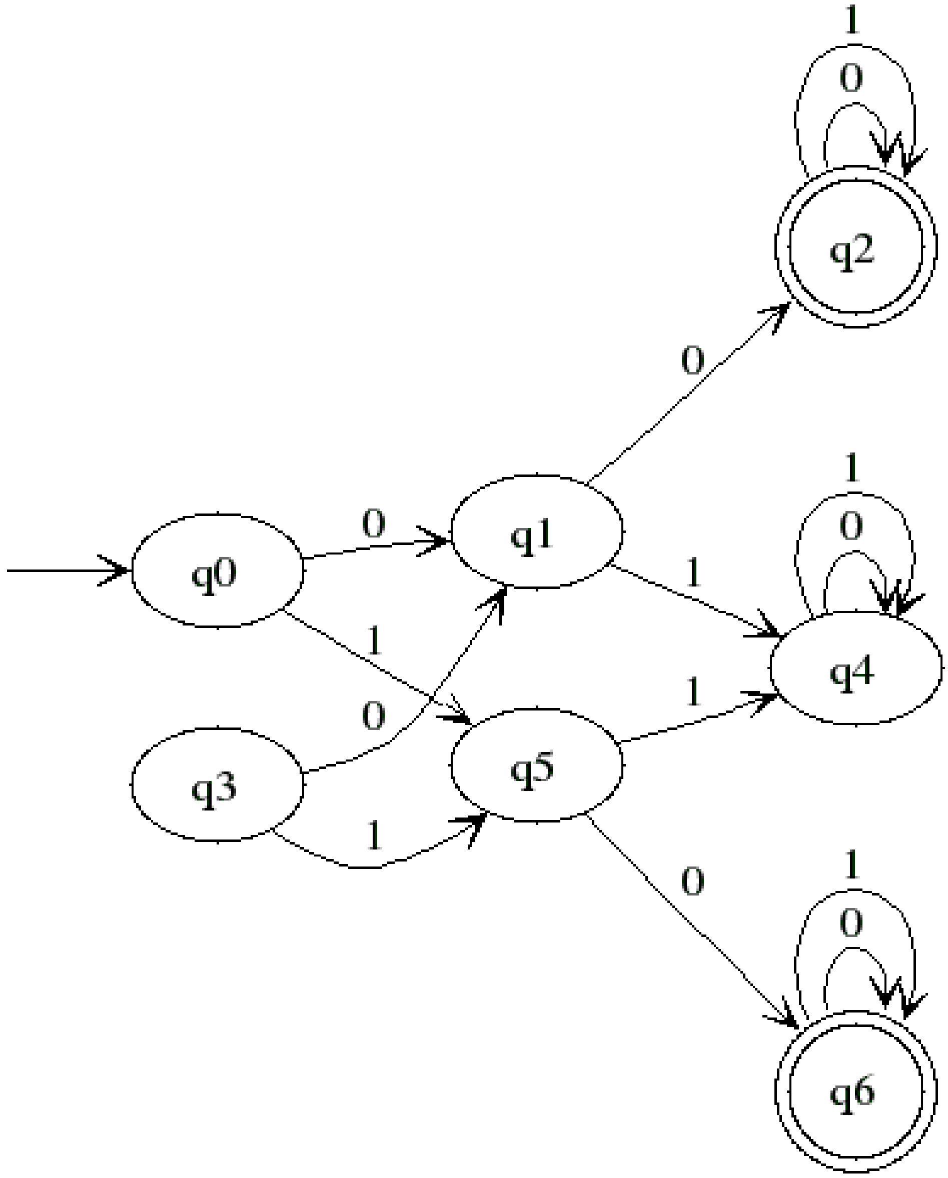
Zadania

1. Zminimalizuj automat skończony A.

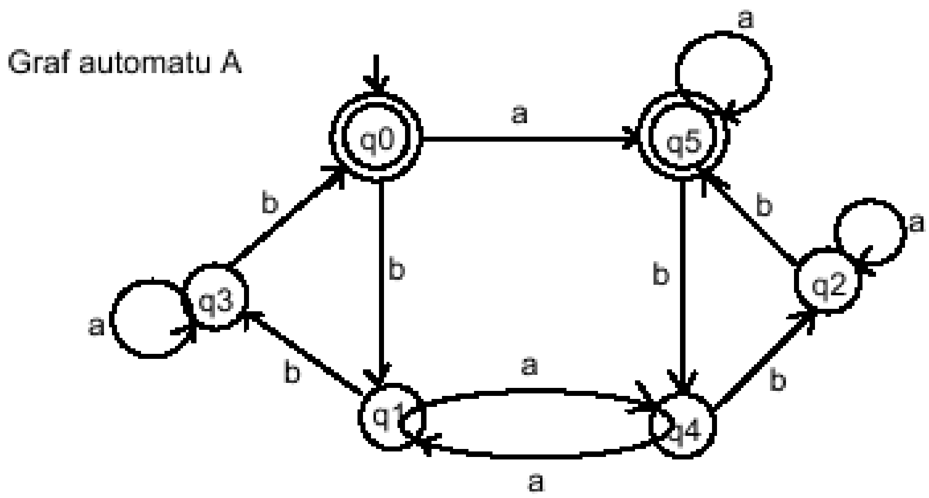


Rysunek 1: Automat skończony A

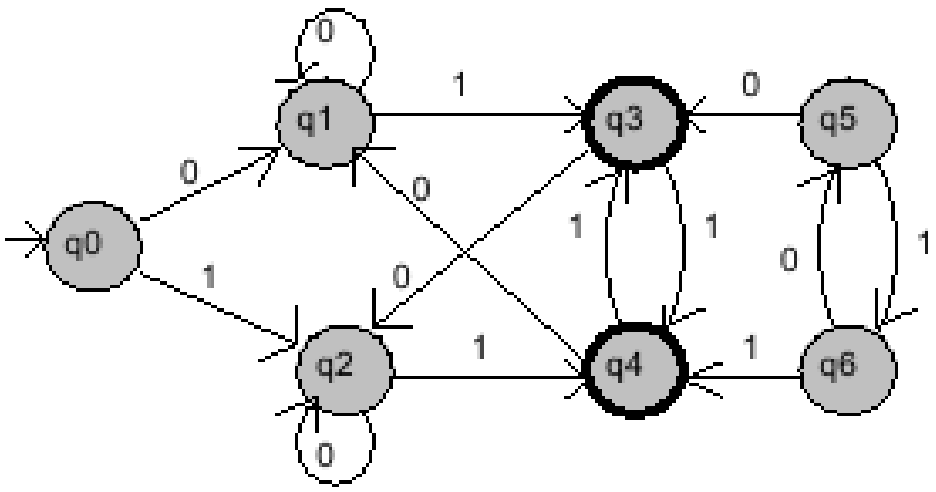
2. Zminimalizuj automat skończony B.
3. Zminimalizuj automat skończony AS1.
4. Zminimalizuj automat skończony AS2.
5. Zminimalizuj automat skończony AS3.



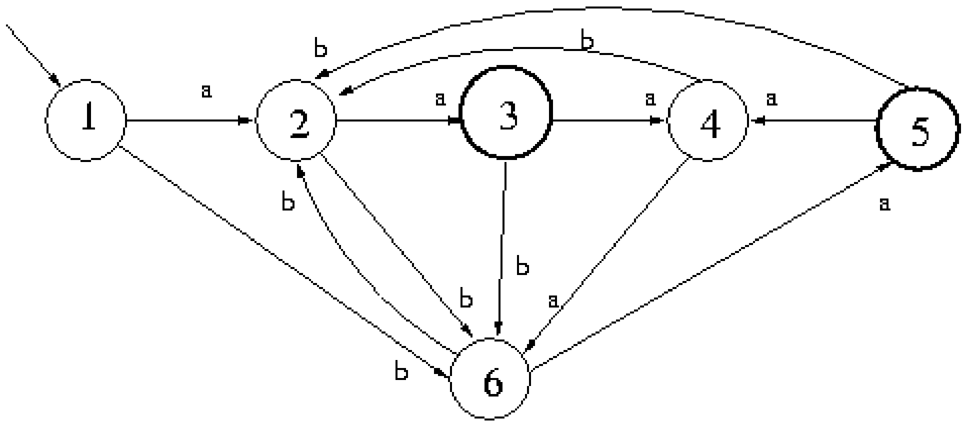
Rysunek 2: Automat skończony B



Rysunek 3: Automat skończony AS1



Rysunek 4: Automat skończony AS2



Rysunek 5: Automat skończony AS3

3 Automaty z funkcją wyjścia

3.1 Automat Moore'a

Definicja 3 *Automatem Moore'a nazywamy uporządkowaną szóstkę* $A = (Q, \Sigma, Y, \delta, \lambda, q_0)$ *gdzie:*

- Q – *skończony zbiór stanów*
- Σ – *skończony zbiór symboli wejściowych (alfabet)*
- Y – *skończony zbiór symboli wyjściowych*
- δ – *funkcja przejścia,* $\delta : Q \times \Sigma \rightarrow Q$
- λ – *funkcja wyjścia,* $\lambda : Q \rightarrow Y$
- q_0 – *stan początkowy*

Wyjście automatu Moore'a zależy tylko od bieżącego stanu.

3.2 Automat Mealy'ego

Definicja 4 *Automatem Mealy'ego nazywamy uporządkowaną szóstkę* $A = (Q, \Sigma, Y, \delta, \lambda, q_0)$ *gdzie:*

- Q – *skończony zbiór stanów*
- Σ – *skończony zbiór symboli wejściowych (alfabet)*
- Y – *skończony zbiór symboli wyjściowych*
- δ – *funkcja przejścia,* $\delta : Q \times \Sigma \rightarrow Q$
- λ – *funkcja wyjścia,* $\lambda : Q \times \Sigma \rightarrow Y$
- q_0 – *stan początkowy*

Wyjście automatu Mealy'ego zależy od bieżącego stanu i symbolu z wejścia. Można nieformalnie powiedzieć, że wyjście automatu zależy od przejścia – funkcje δ i λ mają tą samą dziedzinę.

3.3 Zadania

Podaj automat Moore'a i automat Mealy'ego, które realizują następujące zadania

1. usunięcie jedynek z dowolnego ciągu binarnego

2. wykrywanie w ciągu binarnym sekwencji `001` – automat wypisuje na wyjście znak `1` jeśli wejście zawiera podaną sekwencję, `0` w przeciwnym razie

3. dla danego słowa zbudowanego z liter a, b, c automat ma dodać jedną dodatkową literę c po każdej sekwencji liter c w ciągu wejściowym. Przykład: INPUT=*abcaccb*, OUTPUT=*abccacccb*.

4 Automaty ze stosem

Definicja 5 *Automatem ze stosem nazywamy:* $A_s = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, *gdzie:*

- Q *- skończony zbiór stanów*
- Σ *- skończony zbiór symboli wejściowych (alfabet),*
- Γ *- skończony zbiór symboli na stosie (alfabet stosowy)*
- δ *- funkcja przejścia, funkcja ta odwzorowuje trójkę (stan, symbol wejściowy, symbol na stosie) w parę (stan, słowo nad alfabetem* Γ*):* $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to Q \times \Gamma^*$
- $q_0 \in Q$ *- stan początkowy*
- $Z_0 \in \Gamma$ *- symbol początkowy na stosie (wartownik dna)*
- $F \in Q$ *- zbiór stanów końcowych.*

Działanie automatu stosowego Automat ze stosem to rozwinięcię automatu skończonego. Jako argument funkcja przejścia otrzymuje "szczyt stosu", a w wyniku daje słowo, które zapisać na stosie. Zapisany na stosie może też być symbol kasowania $\Box$, który powoduje usunięcie ostatniego symbolu ze stosu i przesunięcie "szczytu stosu" na kolejny symbol (tuż "pod" bieżącym). Skasować można tylko jeden, ostatni symbol. Jeżeli funkcja przejścia otrzyma jako argument wartownik dna (Z_0), nie może go usunąć (inaczej: funkcja staje się nieokreślona przy usuwaniu symbolu Z_0), wartownika dna nie można również zapisać na stosie.

Wyróżnia się akceptację przy stosie pustym, można wykazać że jest to równoważne akceptacji przez stany akceptujące.

Przykładowe instrukcje:

- $\delta(q_2, s, A) = (q_3, AB)$ – dodanie symbolu B na stosem
- $\delta(q_1, a, A) = (q_3, ABC)$ – dodanie symboli BC na stos, szczytem stosu jest C

- $\delta(q_3, t, B) = (q_3, \Box)$ – usunięcie symbolu B ze szczytu stosu
- $\delta(q_3, z, \_) = (q_3, \Box)$ – skrótowy zapis na usunięcie dowolnego symbolu

Zadania

1. Zaprojektuj automat ze stosem akceptujący język $L = \{0^n 1^n : n \geqslant 0\}$.
2. Zaprojektuj automat ze stosem akceptujący słowa postaci wcw^{-1} nad alfabetem $\{a, b, c\}$, gdzie w^{-1} oznacza odwrócenie słowa w, $w \in \{a, b\}^*$.
3. Podać automat ze stosem akceptujący słowa postaci $a^n b^{2n}$. Czy da się zbudować ogólny automat dla języka $a^n b^{kn}$, k ustalone?
4. Podać automat ze stosem akceptujący słowa postaci $a^{2n} b^n$. Czy da się zbudować ogólny automat dla języka $a^{kn} b^n$, k ustalone?
5. **Praca domowa *** Jak na podstawie poprzednich zadań zbudować automat akceptujący język $a^{kn} b^{ln}$, k, l ustalone? Podaj ten automat.
6. Podaj automat który zaakceptuje poprawne wyrażenie arytmetyczne złożone z 1, +, - oraz nawiasów. Np. 1+(1-1), 1+1+1-1, (1+1)-(1-1+1). *Wskazówka* Narysuj automat skończony, który będzie akceptował kolejność symboli, stos przydaje się w zasadzie tylko do obsługi nawiasów.
7. **Praca domowa *** Podać automat stosowy akceptujący wyrażenia arytmetyczne złożone z jednocyfrowych liczb, znaków $+, -, *, /$ oraz nawiasów.
8. **Praca domowa *** Udowodnić, że automat stosowy akceptujący przy stosie pustym jest równoważny automatowi akceptującemu przez stanu końcowe (a stosie niekoniecznie pustym).

5 Maszyna Turinga

Deterministyczna maszyna Turinga $MT = (Q, q_0, F, \Sigma, \Gamma, b, \delta)$, gdzie:

- Q – skończony zbiór stanów, $q_0 \in Q$ – stan początkowy, $F \subseteq Q$ – zbiór stanów końcowych
- Γ – skończony alfabet taśmy, $\Sigma \subseteq \Gamma$ – alfabet wejściowy, $b \in \Gamma$ – symbol pusty
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, 1\}$ – funkcja przejścia, gdzie -1, 0, 1 oznaczają przesunięcie głowicy.

MT operuje na taśmie (dwustronnie lub jednostronnie nieskończonej); odczytuje symbol z taśmy pod głowicą i według funkcji przejścia zapisuje nowy symbol, zmienia stan i przesuwa (bądź nie) głowicę w dowolnym kierunku. Konfiguracja MT to stan maszyny i zawartość taśmy.
Maszyna niedeterministyczna to taka, gdzie zamiast funkcji przejścia rozważa się relację przejścia, tj. $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{-1,0,1\}}$.

Zadania

1. Zaprojektować maszynę Turinga wyliczającą unarny następnik liczby.
2. Zaprojektować maszynę Turinga obliczającą sumę liczb w postaci unarnej.
3. Zaprojektować maszynę Turinga wyznaczającą liczbę dwa razy większą.
4. Zaprojektować maszynę Turinga obliczającą iloczyn liczb w postaci unarnej.
5. Podać maszynę Turinga wyznaczającą w^{-1} – odwrócone słowo w nad alfabetem $\{a, b\}$. Na taśmie powinny znajdować się słowa w oraz w^{-1}.
6. Podać maszynę Turinga rozpoznającą język $L = \{ww^{-1} : w \in \{a, b\}^*\}$.

6 Zbiory regularne

Σ – dowolny alfabet. Zbiorem regularnym nad tym alfabetem nazywamy zbiór, spełniający jeden z poniższych warunków:

1. zbiór pusty $\emptyset$ jest zbiorem regularnym
2. zbiór $\{\epsilon\}$ złożony z pustego słowa jest zbiorem regularnym
3. zbiór $\{a : a \in \Sigma\}$ jednoliterowych słów jest zbiorem regularnym
4. jeśli P, Q są zbiorami regularnymi, to zbiorami regularnymi są też zbiory:
   (a) $P \cup Q$ – suma teoriomnogościowa zbiorów P, Q
   (b) PQ – złożenie słów ze zbiorów P, Q
   (c) $P\star$ – domknięcie Kleenego zbioru P

7 Wyrażenia regularne

Wyrażenia regularne służą do opisywania zbiorów regularnych, jak powyżej Σ to dowolny alfabet. Wyrażenia regularne tworzy się następująco:

1. $\emptyset$ jest wyrażeniem regularnym do którego pasuje zbiór pusty słów
2. ϵ jest wyrażeniem regularnym do którego pasuje słowo puste
3. **a** jest wyrażeniem regularnym do którego pasuje jednoliterowe słowo a, $a \in \Sigma$
4. jeśli **p**, **q** są wyrażeniami regularnymi to pasującymi do zbiorów P, Q, to wyrażeniami regularnymi są również:
   (a) $(\mathbf{p})$
   (b) $\mathbf{p} + \mathbf{q}$, odpowiada $P \cup Q$
   (c) $\mathbf{p}\ \mathbf{q}$, odpowiada PQ
   (d) $\mathbf{p}\star$, odpowiada $P\star$

7.1 Zadania

1. Opisać zbiór regularny:
   (a) składający się ze słowa "kot"
   (b) składający się ze słów "kot" i "kotek"
   (c) składający się ze słów "kot" i "pies"
   (d) składający się ze słowa "ma" powtórzonego 3 razy
   (e) składający się ze słowa "ma" powtórzonego nieskończenie wiele razy
   (f) binarnych liczb parzystych
   (g) dziesiętnych liczb parzystych
2. Napisać wyrażenia regularne dla zbiorów z zadania 1.
3. Udowodnić tożsamości dla dowolnych wyrażeń regularnych **p**, **q**, **r**:
   $\epsilon\, \mathbf{p} = \mathbf{p}\, \epsilon = \mathbf{p}$ $\mathbf{p}+\mathbf{q} = \mathbf{q}+\mathbf{p}$
   $\epsilon\star = \epsilon$ $\mathbf{p}(\mathbf{qr}) = (\mathbf{pq})\mathbf{r}$
   $\mathbf{p}\star = \mathbf{p} + \mathbf{p}\star = (\mathbf{p} + \epsilon)\star$ $(\mathbf{p}\star)^* = \mathbf{p}\star$
   $\mathbf{p} + \mathbf{p} = \mathbf{p}$ $\mathbf{p}\star + \epsilon = \mathbf{p}\star$
   $\mathbf{p} + \emptyset = \mathbf{p}$ $\epsilon + \mathbf{pp}\star = \mathbf{p}\star$
   $(\mathbf{p}+\mathbf{q})\star = (\mathbf{p}\star + \mathbf{q}\star)\star = (\mathbf{p}\star\mathbf{q}\star)\star$
   $\mathbf{p}\ \mathbf{q} + \mathbf{p}\ \mathbf{r} = \mathbf{p}\ (\mathbf{q} + \mathbf{r})$
   $\mathbf{p}\ \mathbf{r} + \mathbf{q}\ \mathbf{r} = (\mathbf{p} + \mathbf{q})\ \mathbf{r}$

4. Napisać wyrażenie regularne dla następujących zbiorów
   (a) zbiór słów języka polskiego będącego odmianami przez przypadki rzeczownika "kot".
   (b) zbiór słów 0/1 w których podciąg 11 występuje co najwyżej jeden raz.
   (c) zbiór ciągów binarnych o długości wielokrotności 2 lub 3.
   (d) zbiór ciągów binarnych składający się z podciągów zer i podciągów jedynek wielokrotnie powtórzonych
   (e) zbiór ciągów binarnych niezawierających podciągu 100.
   (f) zbiór liczb dziesiętnych podzielnych przez 3.
   (g) zbiór liczb dziesiętnych podzielnych przez 4.
   (h) zbiór liczb dziesiętnych podzielnych przez 40

8 Gramatyki

Gramatyka G to uporządkowana czwórka $G = \langle V, T, s_0, P\rangle$, gdzie

- V – zbiór symboli nieterminalnych
- T – zbiór symboli terminalnych
- $s_0 \in V$ – wyróżniony nieterminalny symbol startowy
- P – zbiór produkcji. $P = \{\alpha \to \beta : \alpha \in (V \cup T)^+, \beta \in (V \cup T)^*\}$.

Produkcje ogólnych gramatyk przekształcają ciągi symboli (nieterminalnych bądź terminalnych) na inne ciągi symboli. Gdy nie ma żadnych ograniczeń na budowę produkcji, to gramatykę zawierającą takie produkcje nazywamy *kombinatoryczną*.

8.1 Gramatyki liniowe

Gramatykę $G = \langle V, T, s_0, P\rangle$ nazywamy *prawostronnie liniową*, jeżeli każda produkcja ma postać $A \to xB$ lub $A \to x$, A, B to symbole nieterminalne, $x \in T^*$ – dowolny łańcuch symboli terminalnych.

Gramatykę $G = \langle V, T, s_0, P\rangle$ nazywamy *lewostronnie liniową*, jeżeli każda produkcja ma postać $A \to Bx$ lub $A \to x$, A, B to symbole nieterminalne, $x \in T^*$ – dowolny łańcuch symboli terminalnych.

8.2 Gramatyki bezkontekstowe

Gramatykę $G = \langle V, T, s_0, P\rangle$ nazywamy *bezkontekstową*, jeżeli każda produkcja ma postać $A \to \beta$, gdzie A jest symbolem nieterminalnym, a β jest dowolnym łańcuchem symboli.

Zadania

1. Podać gramatykę dla
   (a) słowa "kot" odmienionego przez przypadki
   (b) zdania w języku polskim (uproszczonego, o ustalonej kolejności części zdania)
   (c) palindromów nad alfabetem $\{a, b\}$
2. Przedstawić w gramatyce liniowej
   (a) Ciągi binarne zawierające jako podciągi 011 lub 010.
   (b) liczby (całkowite i zmiennoprzecinkowe) w języku Pascal
   (c) identyfikatory (nazwy zmiennych, typów, funkcji, procedur) języka Pascal
3. Przedstawić w gramatyce bezkontekstowej
   (a) deklarację typu rekordowego języka Pascal
   (b) deklarację funkcji w języku Pascal
   (c) instrukcje języka Pascal