DFA=$(Q, \Sigma, \delta, q_0, F)$

Q skończony zbiór stanów

Σ skończony alfabet wejściowy

δ funkcja przejścia postaci $Q \times \Sigma \to Q$

q_0 stan początkowy

$F \subseteq Q$ zbiór stanów akceptujących

Minimalizacja DFA

1. forall p końcowy, q niekońcowy, oznacz (p,q)
2. forall $(p,q) \in (F \times F) \cup (Q \setminus F \times Q \setminus F), p \neq q$ if $\exists_{a \in \Sigma}(\delta(p,a), \delta(q,a))$ jest oznaczona, oznacz (p,q) (rekurencyjnie).
3. nieoznaczone scalamy.

PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Q skończony zbiór stanów

Σ alfabet wejściowy

Γ alfabet stosowy

$q_0 \in Q$ stan początkowy

$Z_0 \in \Gamma$ symbol początkowy na stosie

$F \subset Q$ zbiór stanów akceptujących (jeśli $F = \emptyset$ to akceptujemy przez pusty stos)

δ funkcja przejścia postaci $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$

LOP Zał., że L regularny. Wtedy istnieje stała n, że jeśli $z \in L$ oraz $|z| \geqslant n$, to można podzielić z na $z = uvw$ takie, że:

1. $|v| \geqslant 1$
2. $|uv| \leqslant n$
3. $\forall_{i \in \mathbb{N}} z' = uv^i w \in L$

Podział $\alpha = uvw$, $|uv| \leqslant n$ oraz $|v| \geqslant 1$. Wybieramy i dla którego $|uv^i w| \notin L$ a powinien.
LOP bezk. Zał., że L bezkontekstowy.Wtedy istnieje stała n, że jeśli $z \in L$ oraz $|z| \geqslant n$, to można podzielić z na $z = uvwxy$, takie, że:

1. $|vx| \geqslant 1$
2. $|vwx| \leqslant n$
3. $\forall_{i \in \mathbb{N}} z' = uv^i wx^i y \in L$

Lemat Ogdena Niech L język bezkontekstowy. Wtedy istnieje stała n taka, że jeśli $z \in L$ oraz $|z| >= n$ i oznaczymy w z n lub więcej pozycji jako wyróżnione, to można podzielić z na $z = uvwxy$ takie, że:

1. v i x zawierają łącznie co najmniej jedną wyróżnioną pozycję
2. vwx zawiera co najwyżej n wyróżnionych pozycji
3. $\forall i \in \mathbb{N}\ z' = uv^i wx^i y \in L$

Klasa języków regularnych jest domknięta na operację sumy, dopełnienia, przecięcia, złożenia i domknięcia Kleene'ego. **Gramatyka bezkontekstowa** G=(N,T,P,S)

N - skończony zbiór zmiennych (nieterminale)

T - skończony zbiór zmiennych końcowych(terminale, alfabet)

P - skończony zbiór produkcji postaci A $\to \alpha$ gdzie A $\in$ N i $\alpha \in (\text{N} \cup \text{T})^*$

S $\in$ N - symbol początkowy

Postać normalna Chomsky'ego postaci:
$A \to BC$ albo $A \to a$
Konstrukcje:

1. If po prawej terminal a to zastępujemy go C_a i dopisujemy $C_a \to a$
2. If prawa strona dłuższa niz 1 to zastępujemy $A \to B_1 \dots B_n$ przez
   $A \to B_1 D_1, D_1 \to B_2 D_2, \dots, D_{n-2} \to B_{n-1} B_n$

FIRST(X) - dla symboli

1. X-terminal, to FIRST(X)=X
2. X$\to \varepsilon$ to do FIRST(X) dodajemy ε
3. X - nieterminal i $X \to Y_1 Y_2 ... Y_k$ to dodajemy a do $FIRST(X)$ jeśli istnieje i takie, że $a \in FIRST(Y_i)$ oraz $\varepsilon \in FIRST(Y_j)$ dla każdego $j < i$.
   $\varepsilon \in FIRST(X)$ jeśli należy do wszystkich $FIRST(Y_i)$.
4. $FIRST(X\alpha) = FIRST(X)$ gdy $\varepsilon \notin FIRST(X)$
5. $FIRST(X\alpha) = FIRST(X) \cup FIRST(\alpha)$ gdy $\varepsilon \in FIRST(X)$

FOLLOW(A) - dla nieterminali

1. Dla początkowego S do $FOLLOW(S)$ dodajemy \$
2. Jeśli mamy produkcję $A \to \alpha B \beta$ to do $FOLLOW(B)$ dodajemy wszystkie symbole z $FIRST(\beta)$ poza ε
3. Jeśli $A \to \alpha B$ lub $A \to \alpha B \beta$, gdzie $\varepsilon \in FIRST(\beta)$ to do $FOLLOW(B)$ dodajemy wszystkie symbole z $FOLLOW(A)$

LL(1) - $A \to \alpha$
Tabela: nazwy kolumn terminale i \$!!!⇳
nazwy wierszy nieterminale ⇔

1. $\forall$ produkcji $A \to \alpha$ z gramatyki wykonaj 2 i 3
2. foreach $a \in T$ if $a \in FIRST(\alpha)$ to wpisz $A \to \alpha$ do $M[A, a]$
3. if $\varepsilon \in FIRST(\alpha)$ to dla każdego $b \in FOLLOW(A)$ wpisz $A \to \alpha$ do $M[A, b]$ Jeżeli $\varepsilon \in FIRST(\alpha)$ oraz \$ $\in FOLLOW(A)$, dodaj $A \to \alpha$ do $M[A, \$]$
4. PROTIP: nie ma w tabeli ε!

SLR Tabela:
nazwy kolumn AKCJE (terminale i \$!!!)
i PRZEJŚCIA (nieterminale)
nazwy wierszy stany

1. zbiory sytuacji $C = I_0, \dots, I_n$
   Zaczynamy od $I_0 = domkniecie([S' \to \bullet S])$
2. tabelka + redukcje (zaznaczyć ew. konflikty)
   konstrukcja tabelki: w częsci akcji s_x (shift) i r_x (reduce), a w części przejść (nieterminale) x (liczba)
   ACC dla $S' \to S\bullet$
3. redukcja do FOLLOW(A) (if redukcja była z $A \to \beta\bullet$) tzn. jak jest kropka na końcu to do tabeli dodajemy r_x, gdzie x to numer produkcji

LR(1)

1. zbiory sytuacji z PODGLĄDEM
2. podgląd początkowy \$
3. podgląd przy domknięciu: mamy $[A \to \alpha \bullet B\beta, a] \in I$ dla każdej produkcji z $B \to \gamma$ dodaj $[B \to \bullet\gamma, FIRST(\beta a)]$
4. tabelka jak SLR ale zamiast redukcja do FOLLOW(A) (if redukcja była z $A \to \beta\bullet$) to redukcja do elementów z podglądu

LALR

1. generujemy rodzinę $C = I_0, \dots, I_n$ jak w LR(1)
2. sklejamy jądra nie patrząc na podgląd, a podglądy łączymy - tabelka analogicznie do LR(1)

LEADING(A)-pierwsze term. z A

1. $a \in LEADING(A)$ jeśli mamy produkcję $A \to Ba\beta$ lub $A \to a\beta$
2. if exists prod. $A \to B\alpha$ i $a \in LEADING(B)$ to $a \in LEADING(A)$
3. foreach nieterminali liczymy 1 i powtarzamy 2 aż nic się nie zmienia

TRAILING(A)-ostatnie term. z A

1. $a \in TRAILING(A)$ jeśli mamy produkcję $A \to \beta aB$ lub $A \to \beta a$
2. if exists prod. $A \to \alpha B$ i $a \in TRAILING(B)$ to $a \in TRAILING(A)$
3. foreach nieterminali liczymy 1 i powtarzamy 2 aż nic się nie zmienia

Tab. priorytetów $\doteq \lessdot \gtrdot$

TT $T \doteq T$

TNT $T \doteq T$

TN foreach $a \in LEADING(N)$ do $T \lessdot a$ (wiersze) ⇔

NT foreach $a \in TRAILING(N)$ do $a \gtrdot T$ (kolumny) ⇳

\$ zawsze gorszy

Zbiory sytuacji

1. Wzbogacenie $S' \to S$
2. Ponumerować produkcje (do redukcji!!!).
3. Idziemy od góry z wygenerowanych, jeśli mamy jakąś sytuację $A \to \alpha \cdot S\beta$ (kropka nie na końcu) to generujemy d(I_{teraz}, S)

$E \to \varepsilon \Rightarrow E \to .$

4. W ostateczności musimy dojść z każdą produkcją z kropką na koniec

Rekurencja

1. $A \to A\alpha|\beta$
2. $A \to \beta A'$

   $A' \to \alpha A'|\varepsilon$

Faktoryzacja

1. $A \to \alpha\beta_1|...|\alpha\beta_k$
2. $A \to \alpha A'$

   $A' \to \beta_1|...|\beta_k$

język	lem	slowo	notes
$L = \{uvv^R w : u,v,w \in \{0,1,2\}^* v \neq \varepsilon\}$	reg	dwa te same sym obok	
$L = \{wwx : w \in \{0,1,2\}^* x \in \{0,1\}^* \lvert w\rvert > 0\}$	Ogd	$21^n\underline{0^n 221^n}0^n 2$	
$L = \{wxw^R : w \in \{0,1,2\}^* x \in \{0,1\}^* \lvert w\rvert > 0\}$	reg	konczy sie tym czym zaczyna	
$L = \{a^n b^k a^n, n \neq k\}$	Ogd	dwa warunki to za mało	
$L = \{a^i b^j c^k, i < j < k\}$	Ogd	nie bezkontekstowy	
$L = \{a^i b^j c^k, k = max(i,j)\}$	lop/ogd	nie bezkontekstowy	
$L = \{a^i b^j c^k, i + j = k\}$	lop	bezkontekstowy	
$L = \{a^i b^j c^k, j = i + k\}$		bezkontekstowy	
$L = \{a^n b^m c^n, m \neq n\}$		nie jest bezkontekstowy	
$L = \{ww, w \in \{0,1\}^*\}$		Nie bezkontekstowe, ale dopełnienie bezkontekstowe	
$L = \{w \in \{0,1,2\}^* w = palindrom\ i\ \lvert w\rvert_0 = \lvert w\rvert_2 mod 13\}$		bezkontekstowy, hybryda	
$L = \{w \in \{a,b,c\}^* w = palindrom\ i\ \lvert w\rvert_a = \lvert w\rvert_b i \lvert w\rvert_c > 0\}$	Ogden	$a^n b^n c b^n a^n$	
$\omega = xxy \wedge x \neq \varepsilon$	LOP	$ab^n ab^n$	$i = 0$
$\omega = xyyz \wedge y \neq \varepsilon$	reg	$len \geqslant 4$	dobrać krótsze
$\omega\omega^R \wedge \lvert\omega\rvert_a \equiv \lvert\omega\rvert_b \equiv 0 (mod 13)$	LOP	$a^{13n} b^{13n} b^{13n} a^{13n}$	ozn.
$\omega : \lvert\omega\rvert_a \equiv \lvert\omega\rvert_b (mod 3)$	reg	mini	
$\omega = xyy^R \wedge y \neq \varepsilon$	reg	2 obok	
$\omega : palindrom \wedge \lvert\omega\rvert_a = \lvert\omega\rvert_c$	LOP	$a^n c^n c^n a^n$	
$\omega = xcycz \wedge$ xy i yz $\in \{a,b\}^*$palindromy	Ogd	$a^m bca^m cba^m$	śr. ozn.
$\lvert\omega\rvert_a = \lvert\omega\rvert_b$	bezk.		
$\lvert\omega\rvert_a = \lvert\omega\rvert_b = \lvert\omega\rvert_c$	LOP	$a^n b^n c^n$	
$\omega : \lvert\omega\rvert_a \neq \lvert\omega\rvert_b \neq \lvert\omega\rvert_c$	Ogd	$a^{m+m!} b^m a^{m+m!}$	ozn b.
$\omega : \lvert\omega\rvert_a = \lvert\omega\rvert_b = \lvert\omega\rvert_c$	LOP	$a^n b^n c^n$	i=0
$\omega : \lvert\omega\rvert_a = \lvert\omega\rvert_c > \lvert\omega\rvert_b$	LOP	$a^{n+1} b^n c^{n+1}$	
$\omega\omega\omega$	LOP	$0^n 1^n 0^n 1^n 0^n 1^n$	i=0
$\omega\omega^R\omega$	LOP	$0^n 1^n 1^n 0^n 0^n 1^n$	i=0
$a^n c^k b^n : n \neq k$	Ogd	$a^{n!+n} c^n b^{n!+n}$	

$L = \{w \in \{0,1,2\}^* w = palindrom\ i\ \lvert w\rvert_0 = \lvert w\rvert_2 mod 13\}$ - hybryda palindromów (da sie zrobic automat) oraz przejście po stanach

L = $\{w \in \{a,b\}^* \lvert w\rvert_a = \lvert w\rvert_b + 5\}$ < bezkontekstowy

L = $\{w \in \{a,b\}^* \lvert w\rvert_a = \lvert w\rvert_b = 2 mod 5\}$ < regularny

Majac $L = \{w \in \{a,b,c\}^* i \lvert w\rvert_a \neq \lvert w\rvert_b lub \lvert w\rvert_a \neq \lvert w\rvert_c lub \lvert w\rvert_b \neq \lvert w\rvert_c\}$
mamy niedeterministyczny automat ze stosem ktory zgaduje poprawny wynik.

$L_1 = \{w : w \in \{a,b,c\}^* \wedge \lvert w\rvert_a \neq \lvert w\rvert_b \neq \lvert w\rvert_c\}$
Niech n stała z lematu Ogdena. Niech m > n. Wybieramy slowo $z = a^{m+m!} b^m c^{m+m!}$ i oznaczamy m liter b jako wyróżnione.

1. nie możemy pompować samego a ani samego c (brak wyróżnionych).
2. nie możemy pompować jednocześnie a oraz c (pomiędzy nimi jest więcej niż n wyróżnionych liter).

Pozostają nam do rozpatrzenia podziały, w których:

1. pompujemy b:
   wyznaczamy i: $\lvert vx\rvert_b = p$
   $m + m! = m + (i-1)p$
   $m! = ip - p$
   $i = \frac{m!}{p} + 1$

   $\lvert z'\rvert_b = \lvert z\rvert_b + (i-1)\lvert vx\rvert_b = m + (\frac{m!}{p} + 1 - 1)p = m + m!$
   Długość b jest taka sama jak długość reszty więc wyszliśmy z języka.
2. pompowanie a i b. Równamy ilość b do ilości c.
   pomowanie b i c. Równamy ilość b do ilości a.

FIRST

1. Szukamy produkcji gdzie na początku stoi terminal i ten terminal dodajemy do zbioru FIRST od nieterminala przed strzałką.
2. Szukamy produkcji z eps i dodajemy ten eps do zbioru FIRST od nieterminala przed strzałką.
3. Szukamy produkcji gdzie na początku stoi nieterminal i FIRST od tego nieterminala dodajemy do FIRST od nieterminala stojącego przed strzałką (bez epsilona). Jeżeli w kopiowanym zbiorze jest epsilon to dodajemy FIRST od następnego symbolu. (Jeśli w każdym symbolu jest epsilon to na końcu dodajemy epsilon).

FOLLOW

1. Do zbioru FOLLOW od symbolu początkowego dodajemy $
2. Szukamy produkcji gdzie za nieterminalem będzie stał jakiś symbol i do FOLLOW od tego nieterminala dodajemy FIRST od następnego symbolu (pomijając eps). Jeśli w dodawanym zbiorze był eps to sprawdzamy kolejny symbol.
3. Szukamy produkcji gdzie na końcu znajduje się nieterminal i do FOLLOW od tego nieterminala kopiujemy zawartość FOLLOW od nieterminala przed strzałką.
4. Szukamy produkcji gdzie za jakimś nieterminalem cała prawa strona będzie się zerowała (czyli w FIRST od całej strony będzie epsilon). Wtedy do FOLLOW od tego nieterminala dodajemy FOLLOW od nieterminala przed strzałką.
   Powtarzaj 3 i 4 dopóki są zmiany.

LL FIRST(alfa) - FIRST od pierwszego znaku eps, jeśli jest epsilon to wchodzimy do kolejnego
bin(n)bin(n+1), n > 0 - nie jest bezkontekstowe
bin(n)bin(n+1)$^R, n > 0 - bezkontekstowe$
$bin(n)hex(n)^R, n > 0 > korzystamyzodpowiednioscibin >> hexiucieczkaodzernapoczatku$