

DFA =(<i>Q</i> , Σ , δ , <i>q</i> ₀ , <i>F</i>)
<i>Q</i> skończony zbiór stanów
Σ skończony alfabet wejściowy
δ funkcja przejścia postaci $Q \times \Sigma \rightarrow Q$
<i>q</i> ₀ stan początkowy
<i>F</i> ⊆ <i>Q</i> zbiór stanów akceptujących
Minimalizacja DFA
<ol style="list-style-type: none"> forall p końcowy, q niekończowy, oznacz (p,q) forall (<i>p</i>, <i>q</i>) ∈ (<i>F</i> × <i>F</i>) ∪ (<i>Q</i> \ <i>F</i> × <i>Q</i> \ <i>F</i>), <i>p</i> ≠ <i>q</i> if ∃_{<i>a</i> ∈ Σ}($\delta(p,a), \delta(q,a)$) jest oznaczona, oznacz (p,q) (rekurencyjnie). nieoznaczone scalamy.
PDA <i>M</i> = (<i>Q</i> , Σ , Γ , δ , <i>q</i> ₀ , <i>Z</i> ₀ , <i>F</i>)
<i>Q</i> skończony zbiór stanów
Σ alfabet wejściowy
Γ alfabet stosowy
<i>q</i> ₀ ∈ <i>Q</i> stan początkowy
<i>Z</i> ₀ ∈ Γ symbol początkowy na stosie
<i>F</i> ⊂ <i>Q</i> zbiór stanów akceptujących (jeśli <i>F</i> = ∅ to akceptujemy przez pusty stos)
δ funkcja przejścia postaci $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$
LOP Zał., że <i>L</i> regularny. Wtedy istnieje stała <i>n</i> , że jeśli <i>z</i> ∈ <i>L</i> oraz <i>z</i> ≥ <i>n</i> , to można podzielić <i>z</i> na <i>z</i> = <i>uvw</i> takie, że:
<ol style="list-style-type: none"> <i>v</i> ≥ 1 <i>uv</i> ≤ <i>n</i> ∀_{<i>i</i> ∈ \mathbb{N}} <i>z</i>' = <i>uv</i>^{<i>i</i>}<i>w</i> ∈ <i>L</i>
Podział $\alpha = uvw$, <i>uv</i> ≤ <i>n</i> oraz <i>v</i> ≥ 1. Wybieramy <i>i</i> dla którego <i>uv</i> ^{<i>i</i>} <i>w</i> ∉ <i>L</i> a powinien.
LOP bezk. Zał., że <i>L</i> bezkontekstowy.Wtedy istnieje stała <i>n</i> , że jeśli <i>z</i> ∈ <i>L</i> oraz <i>z</i> ≥ <i>n</i> , to można podzielić <i>z</i> na <i>z</i> = <i>uvwxy</i> , takie, że:
<ol style="list-style-type: none"> <i>vx</i> ≥ 1 <i>vwx</i> ≤ <i>n</i> ∀_{<i>i</i> ∈ \mathbb{N}} <i>z</i>' = <i>uv</i>^{<i>i</i>}<i>wx</i>^{<i>i</i>}<i>y</i> ∈ <i>L</i>
Lemat Ogdena Niech <i>L</i> język bezkontekstowy. Wtedy istnieje stała <i>n</i> taka, że jeśli <i>z</i> ∈ <i>L</i> oraz <i>z</i> ≥ <i>n</i> i oznaczmy w <i>z</i> <i>n</i> lub więcej pozycji jako wyróżnione, to można podzielić <i>z</i> na <i>z</i> = <i>uvwxy</i> takie, że:
<ol style="list-style-type: none"> <i>v</i> i <i>x</i> zawierają łącznie co najmniej jedną wyróżnioną pozycję <i>vwx</i> zawiera co najwyżej <i>n</i> wyróżnionych pozycji ∀_{<i>i</i> ∈ \mathbb{N}} <i>z</i>' = <i>uv</i>^{<i>i</i>}<i>wx</i>^{<i>i</i>}<i>y</i> ∈ <i>L</i>
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 → α gdzie A ∈ N i α ∈ (N ∪ T)*

S ∈ N - symbol początkowy

Postać normalna Chomsky’ego postaci: *A* → *BC* albo *A* → *a*
Konstrukcje:

- If po prawej terminal *a* to zastępujemy go *C*_{*a*} i dopisujemy *C*_{*a*} → *a*
- If prawa strona dłuższa niz 1 to zastępujemy *A* → *B*₁...*B*_{*n*} przez *A* → *B*₁*D*₁, *D*₁ → *B*₂*D*₂,..., *D*_{*n*-2} → *B*_{*n*-1}*B*_{*n*}

FIRST(X) - dla symboli

- X-terminal, to FIRST(X)=X
- X→ ε to do FIRST(X) dodajemy ε
- X - nieterminal i *X* → *Y*₁*Y*₂...*Y*_{*k*} to dodajemy *a* do *FIRST(X)* jeśli istnieje *i* takie, że *a* ∈ *FIRST(Y*_{*i*}) oraz ε ∈ *FIRST(Y*_{*j*}) dla każdego *j* < *i*. ε ∈ *FIRST(X)* jeśli należy do wszystkich *FIRST(Y*_{*i*}).
- FIRST(Xα)* = *FIRST(X)* gdy ε ∉ *FIRST(X)*
- FIRST(Xα)* = *FIRST(X)* ∪ *FIRST(α)* gdy ε ∈ *FIRST(X)*

FOLLOW(A) - dla nieterminali

- Dla początkowego *S* do *FOLLOW(S)* dodajemy \$
- Jeśli mamy produkcję *A* → α*Bβ* to do *FOLLOW(B)* dodajemy wszystkie symbole z *FIRST(β)* poza ε
- Jeśli *A* → α*B* lub *A* → α*Bβ*, gdzie ε ∈ *FIRST(β)* to do *FOLLOW(B)* dodajemy wszystkie symbole z *FOLLOW(A)*

LL(1) - *A* → α

Tabela: nazwy kolumn terminale i \$!!!⌈

nazwy wierszy nieterminale ⇔

- ∀ produkcji *A* → α z gramatyki wykonaj 2 i 3
- foreach *a* ∈ *T* if *a* ∈ *FIRST(α)* to wpisz *A* → α do *M*[*A*, *a*]
- if ε ∈ *FIRST(α)* to dla każdego *b* ∈ *FOLLOW(A)* wpisz *A* → α do *M*[*A*, *b*] Jeżeli ε ∈ *FIRST(α)* oraz \$ ∈ *FOLLOW(A)*, dodaj *A* → α do *M*[*A*, \$]
- PROTIP: nie ma w tabeli ε!

SLR Tabela:

nazwy kolumn AKCJE (terminale i \$!!!)

i PRZEJŚCIA (nieterminale)

nazwy wierszy stany

- zbiory sytuacji *C* = *I*₀,...,*I*_{*n*}
Zaczynamy od *I*₀ = *domkniecie*([*S*' → .*S*])
- tabelka + redukcje (zaznaczyć ew. konflikty)
konstrukcja tabelki: w części akcji *s*_{*x*} (shift) i *r*_{*x*} (reduce), a w części przejść (nieterminale) *x* (liczba) ACC dla *S*' → *S*.
- redukcja do FOLLOW(A) (if redukcja była z *A* → β.) 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* → α•*Bβ*,*a*] ∈ *I* dla każdej produkcji z *B* → γ dodaj [*B* → •γ, *FIRST(βa)*]

4. tabelka jak SLR ale zamiast redukcja do FOLLOW(A) (if redukcja była z *A* → β.) to redukcja do elementów z podglądu

LALR

- generujemy rodzinę *C* = *I*₀,...,*I*_{*n*} jak w LR(1)
- sklejamy jądra nie patrząc na podgląd, a podglądy łączymy - tabelka analogicznie do LR(1)

LEADING(A)-pierwsze term. z **A**

- a* ∈ *LEADING(A)* jeśli mamy produkcję *A* → *Baβ* lub *A* → *aβ*
- if exists prod. *A* → *Bα* i *a* ∈ *LEADING(B)* to *a* ∈ *LEADING(A)*
- foreach nieterminali liczymy 1 i powtarzamy 2 aż nic się nie zmienia

TRAILING(A)-ostatnie term. z **A**

- a* ∈ *TRAILING(A)* jeśli mamy produkcję *A* → β*aB* lub *A* → β*a*
- if exists prod. *A* → α*B* i *a* ∈ *TRAILING(B)* to *a* ∈ *TRAILING(A)*
- foreach nieterminali liczymy 1 i powtarzamy 2 aż nic się nie zmienia

Tab. priorytetów ≐ <>

TT *T* ≐ *T*

TNT *T* ≐ *T*

TN foreach *a* ∈ *LEADING(N)* do *T* < *a* (wiersze) ⇔

NT foreach *a* ∈ *TRAILING(N)* do *a* > *T* (kolumny) ⌈

\$ zawsze gorszy

Zbiory sytuacji

- Wzbogacenie *S*' → *S*
- Ponumerować produkcje (do redukcji!!!).
- Idziemy od góry z wygenerowanych, jeśli mamy jakąś sytuację *A* → α·*Sβ* (kropka nie na końcu) to generujemy d(*I*_{*teraz*}, *S*)

E → ε ⇒ *E* → .

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

Rekurencja

- A* → *Aα|B*
- A* → β*A*'
- A*' → α*A*'|ε

Faktoryzacja

- A* → αβ₁|...|αβ_{*k*}
- A* → α*A*'
- A*' → β₁|...|β_{*k*}

język	lem	słowo	notes
$\omega = xxy \wedge x \neq \varepsilon$	LOP	ab^na^n	$i = 0$
$\omega = xy yz \wedge y \neq \varepsilon$	reg	$len \geq 4$	dobrac krótsze
$\omega \omega^R \wedge \omega _a \equiv \omega _b \equiv 0(mod13)$	LOP	$a^{13n}b^{13n}b^{13n}a^{13n}$	ozn.
$\omega : \omega _a \equiv \omega _b(mod3)$	reg	mini	
$\omega = xy y^R \wedge y \neq \varepsilon$	reg	2 obok	
$\omega : palindrom \wedge \omega _a = \omega _c$	LOP	$a^nc^nc^na^n$	
$\omega = xcycz \wedge xy \text{ i } yz \in \{a,b\}^* \text{ palindromy}$	Ogd	a^mbca^mcbam	śr. ozn.
$ \omega _a = \omega _b$	bezk.		
$ \omega _a = \omega _b = \omega _c$	LOP	$a^nb^nc^n$	
$\omega : \omega _a \neq \omega _b \neq \omega _c$	Ogd	$a^{m+m!}b^ma^{m+m!}$	ozn b.
$\omega : \omega _a = \omega _b = \omega _c$	LOP	$a^nb^nc^n$	i=0
$\omega : \omega _a = \omega _c > \omega _b$	LOP	$a^{n+1}b^nc^{n+1}$	
$\omega \omega \omega$	LOP	$0^n1^n0^n1^n0^n1^n$	i=0
$\omega \omega^R \omega$	LOP	$0^n1^n1^n0^n0^n1^n$	i=0
$a^nc^kb^n : n \neq k$	Ogd	$a^{n!+n}c^nb^{n!+n}$	

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 epsilon). 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.