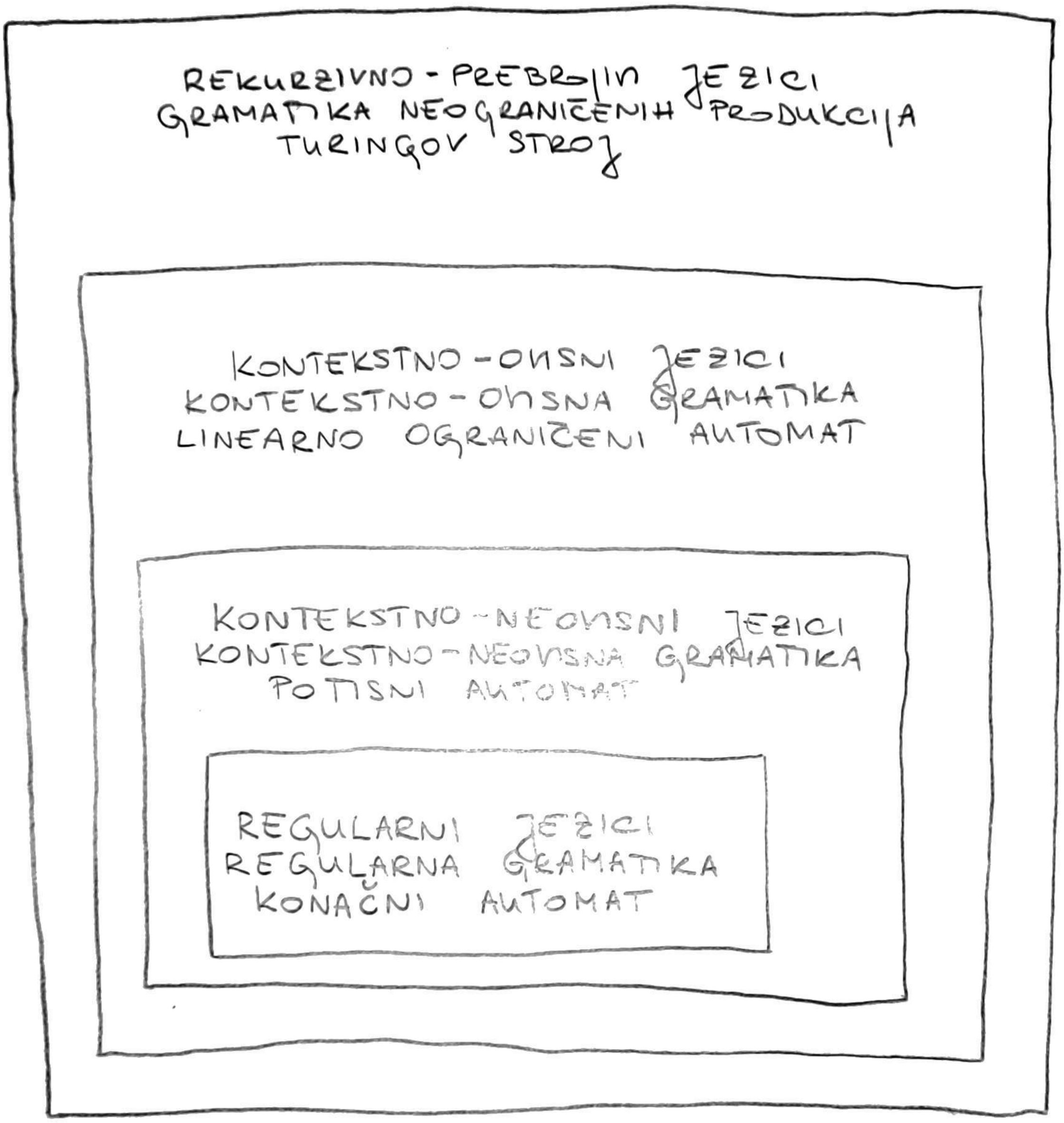
UTR

SKRIPTA

2^{Σ^*}



lipanj, 2015.

Herman Z. Došilović

REGULARNI JEZICI

Def:

Jezik je regularan akko postoji konačni automat koji ga prihvaća. Za bilo koji reg. jez. moguće je izgraditi konačni automat koji ga prihvaća. Bilo koji konačni automat prihvaća jedan od reg. jez.

DETERMINISTIČKI KONAČNI AUTOMAT (DKA)

$$\text{dka} = (Q, \Sigma, \delta, q_0, F)$$

$\vec{\varepsilon}$ - prazan niz

ε - prazan znak

$$\delta(q, \varepsilon) = q$$

Q - konačan skup stanja

Σ - konačan skup ulaznih znakova

δ - fja. prijelaza: $Q \times \Sigma \longrightarrow Q$

$q_0 \in Q$ - početno stanje

$F \subseteq Q$ - prihvatljiva stanja

$$\hat{\delta}(q, \vec{w}a) = \delta(\hat{\delta}(q, \vec{w}), a)$$

$\hat{\delta} : Q \times \Sigma^* \to Q$, Σ^* skup svih mogućih nizova.

dka prihvaća niz $\vec{w}$ akko $\delta(q_0, \vec{w}) \in F$.

dka odbija niz $\vec{w}$ ako u bilo kojem trenutku funkcija $\delta(q, a)$ nije definirana, $a \in \vec{w}$, $q \in Q$.

ili ako $\delta(q_0, \vec{w}) \notin F$.

- MINIMIZACIJA KONAČNOG AUTOMATA

($\equiv$)

stanja p i q su <u>istovjetna</u> akko vrijede oba uvjeta:

1) UVJET PODUDARNOSTI: $(p \in F \wedge q \in F) \vee (p \notin F \wedge q \notin F)$.

2) UVJET NAPREDOVANJA: $\delta(p, a) \underset{\text{istovjetni}}{\equiv} \delta(q, a)$, $\forall a \in \Sigma$.

ODREĐIVANJE ISTOVJETNIH STANJA

- algoritam 1. str. 22.
- algoritam 2. str. 24.
- algoritam 3. str. 25.

NEDOHVATLJIVA STANJA

- dohvatljiva stanja $= \{ q \mid \exists \vec{w} \in \Sigma^* : q = \hat{\delta}(q_0, \vec{w}) \}$

NEDOHVATLJIVA STANJA $= Q \setminus$ dohvatljiva stanja

DKA s minimalnim brojem stanja:

- odbaci nedohvatljiva
- odbaci istovjetna

Dodatno:

- moguće je naći istovjetne <u>znakove</u>.

- DVA AUTOMATA SU ISTOVJETNA AKKO SU NJIHOVA POČETNA STANJA ISTOVJETNA.

NEDETERMINISTIČKI KONAČNI AUTOMAT (NKA)

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

$\delta : Q \times \Sigma \rightarrow 2^Q$, skup svih podskupova od $Q = 2^Q$

$\hat{\delta} : Q \times \Sigma^* \rightarrow 2^Q$:

$$\hat{\delta}(q, \vec{\varepsilon}) = \{q\}$$

$$\hat{\delta}(q, \vec{w}a) = \{p \mid r = \hat{\delta}(q, \vec{w}),\ p \in \delta(r, a)\},\ \vec{w} \in \Sigma^*$$

$$\hat{\delta}(q, \vec{w}a) = \delta(\hat{\delta}(q, \vec{w}), a) = \delta(r, a)$$

$$\hat{\delta}(q, \vec{\varepsilon}a) = \delta(\hat{\delta}(q, \vec{\varepsilon}), a) = \delta(q, a)$$

$$\hat{\delta} \equiv \delta$$

NKA prihvaća $\vec{w}$ akko $\exists q \in \hat{\delta}(q_0, \vec{w}) : q \in F$

NKA odbija $\vec{w}$ akko $\forall q \in \hat{\delta}(q_0, \vec{w}),\ q \notin F$

ili akko fja. prijelaza nije definirana:

$\exists a \in \vec{w} : \delta(q, a)$ nedefinirana $\forall q \in P,\ P \subseteq Q$

PROŠIRIVANJE fje δ :

$$\delta : 2^Q \times \Sigma^*$$

$$\delta(P, w) = \bigcup_{q \in P} \delta(q, \vec{w})$$

$P \in 2^Q,\ P \subseteq Q$ $\rightarrow$ NKA može biti u više stanja

KONSTRUKCIJA: NKA $\to$ DKA

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

$$dka = (Q', \Sigma, \delta', q_0', F')$$

1) $Q' = 2^Q$, $[p_0, p_1, \dots, p_j] \in Q'$, $p_k \in Q$, $0 \le k \le j$

2) F' = skup svih stanja $[p_0, p_1, \dots, p_j]$ gdje je barem jedan $p_k \in F$. $0 \le k \le j$

3) $q_0' = [q_0]$

4) $\delta'([p_0, p_1, \dots, p_l], a) = [r_0, r_1, \dots, r_j]$

akko

$\delta(\{p_0, p_1, \dots, p_l\}, a) = \{r_0, r_1, \dots, r_j\}$

takvi automati nka i dka su istovrijetni.

$nka \equiv dka$ akko $L(nka) = L(dka)$.

NEDETERMINISTIČKI KONAČNI AUTOMAT S ε PRIJELAZIMA (ε-NKA)

$$\varepsilon\text{-nka} = (Q, \Sigma, \delta, q_0, F)$$

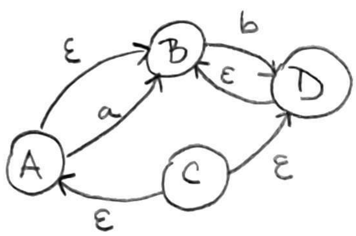
$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$$

ε-okruženje ≡ ε

$$\varepsilon : Q \to 2^Q \cup \{q \in Q\}$$

$$\varepsilon(q) = \{p \mid p = q \text{ ili } p \in \delta(q, \varepsilon^*)\}$$

npr.



$\varepsilon(A) = \{A, B\}$
$\varepsilon(B) = \{B\}$
$\varepsilon(C) = \{C, A, D, B\}$
$\varepsilon(D) = \{B, D\}$

ε na skupu $P \subseteq Q$:

$$\varepsilon(P) = \bigcup_{p \in P} \varepsilon(p)$$

PROŠIRENJE $\hat{\delta} : Q \times \Sigma^* \to 2^Q$

$$\hat{\delta}(q, \vec{\varepsilon}) = \varepsilon(q)$$

$$\hat{\delta}(q, \vec{w}a) = \varepsilon(\delta(\hat{\delta}(q, \vec{w}), a))$$

$$\delta(R, a) = \bigcup_{q \in R} \delta(q, a)$$

$$\hat{\delta}(R, \vec{w}) = \bigcup_{q \in R} \hat{\delta}(q, \vec{w})$$

SKUP $\delta(q_0, \varepsilon) \neq \hat{\delta}(q_0, \vec{\varepsilon})$

$\delta(q_0, \varepsilon)$ - prijelaz za ε znak

$\hat{\delta}(q_0, \vec{\varepsilon})$ - prijelaz za ε^* znakova $\equiv \varepsilon(q_0)$

SKUP $\delta(q_0, a) \neq \hat{\delta}(q_0, \vec{a})$, $|\vec{a}| = 1$

npr.
neka fja δ nije def. za $(q_0, 1)$

$\Rightarrow \delta(q_0, 1) = \emptyset$

ali
$\hat{\delta}(q_0, 1) = \varepsilon(\delta(\hat{\delta}(q_0, \vec{\varepsilon}), 1)) = \varepsilon(\delta(\varepsilon(q_0), 1))$

• BILJEŠKE O IMPLEMENTACIJI ε-NKA:

- skup početnih stanja $= \varepsilon(q_0)$
- nakon pročitanog znaka a NKA prelazi u $\varepsilon(\delta(P, a))$, P je skup stanja u kojem se NKA nalazi prije čitanja a.

ε-NKA prihvaća $\vec{w}$ akko $\exists q \in \hat{\delta}(q_0, \vec{w}) : q \in F$

ε-NKA odbija $\vec{w}$ akko $\forall q \in \hat{\delta}(q_0, \vec{w}),\ q \notin F$

ili akko fja. prijelaza nije definirana:

$\exists a \in \vec{w} : \delta(q, a)$ nije def. $\forall q \in P,\ P \subseteq Q$

KONSTRUKCIJA : ε-NKA → NKA

$$\varepsilon\text{-nka} = (Q, \Sigma, \delta, q_0, F)$$

$$\text{nka} = (Q', \Sigma, \delta', q_0', F')$$

1) $Q' = Q$

2) $q_0' = q_0$

3) <u>ako</u> $\exists q \in \mathcal{E}(q_0) : q \in F$

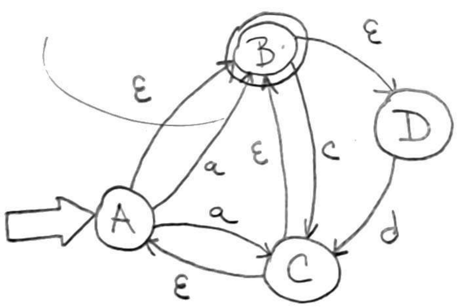
$F' = F \cup \{q_0\}$

<u>inače</u>

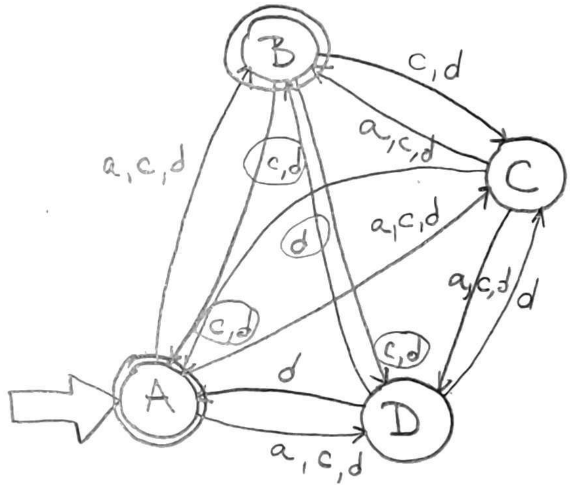
$F' = F$

4) $\delta'(q,a) = \hat{\delta}(q,a) \quad \forall a \in \Sigma \text{ i } q \in Q$

<u>npr.</u>



⇒



$\mathcal{E}(A) = \{A, B, D\}$

$\mathcal{E}(B) = \{B, D\}$

$\mathcal{E}(C) = \{C, A, B, D\}$

$\mathcal{E}(D) = \{D\}$

$$\boxed{\delta'(q,a) = \hat{\delta}(q,a) = \mathcal{E}(\delta(\mathcal{E}(q), a))}$$

SLJEM : MOGUĆE JE DOBITI DKA IZ ε-NKA : ε-NKA → NKA → DKA.

KONAČNI AUTOMATI S IZLAZOM

MOOREOV AUTOMAT

$$moore = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

Δ - skup izlaznih znakova

λ - funkcija izlaza $Q \to \Delta$

npr.

ulazni niz $\vec{w} = a_1 a_2 a_3 \dots a_n \ (n \geq 0,\ a_i \in \Sigma)$

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \dots \xrightarrow{a_n} q_n$$

izlaz automata je niz:

$$\lambda(q_0)\lambda(q_1)\lambda(q_2)\dots\lambda(q_n)$$

za $\vec{w} = \vec{\varepsilon} \Rightarrow$ mooreov automat daje izlaz $\lambda(q_0)$.

MEALYEV AUTOMAT

$$mealy = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$$

λ - f-ja izlaza: $Q \times \Sigma \to \Delta$

npr.

za ulazni niz $\vec{w} = a_1 a_2 \dots a_n \ (n \geq 0,\ a_i \in \Sigma)$

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \dots \xrightarrow{a_n} q_n$$

izlaz automata:

$$\lambda(q_0, a_1)\lambda(q_1, a_2)\dots\lambda(q_{n-1}, a_n)$$

za $\vec{w} = \vec{\varepsilon}$ mealyev automat <u>ne daje izlaz</u>.

KONSTRUKCIJA: Moore → Mealye

$$moore = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$$
$$mealye = (Q, \Sigma, \Delta, \delta, \lambda', q_0)$$

1) $\lambda'(q, a) = \lambda(\delta(q, a))$, $\forall q \in Q$ i $a \in \Sigma$

dodatno:

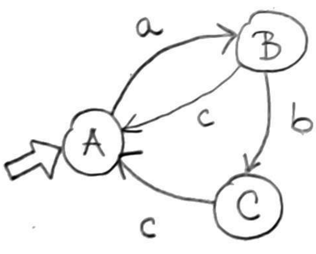
- izlazu mealyevog automata na početak se dodaje $\lambda(q_0)$.

dakle za ulazni niz $\vec{w} = a_1 a_2 a_3 \dots a_n$ $(n \geq 0,\ a_i \in \Sigma)$

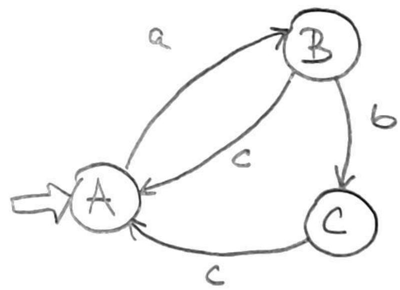
$$\underbrace{\lambda(q_0)\lambda(q_1)\lambda(q_2)\dots\lambda(q_n)}_{\text{MOORE}} = \underbrace{\lambda(q_0)}_{\text{DODATAK}}\underbrace{\lambda'(q_0, a_1)\lambda'(q_1, a_2)\dots\lambda'(q_{n-1}, a_n)}_{\text{MEALYE}}$$

$\Rightarrow$ moore $\equiv$ mealye

npr.



$\Rightarrow$



λ:

q	P
A	x
B	y
C	z

λ'	a	b	c
A	y	/	/
B	/	z	x
C	/	/	x

$$\lambda'(A, a) = \lambda(\delta(A, a))$$
$$= \lambda(B)$$
$$= y$$

+ na početak dodati $\lambda(A) = x$..

KONSTRUKCIJA: Mealye → Moore

$$\text{mealye} = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$$
$$\text{moore} = (Q', \Sigma, \Delta, \delta', \lambda', q_0')$$

1) $Q' = Q \times \Delta$: $[q, b] \in Q'$: $q \in Q$, $b \in \Delta$

2) $q_0' = [q_0, b_0]$, $b_0 \in \Delta$ (proizvoljno)

3) $\delta'([q, b], a) = [\delta(q, a), \Delta(q, a)]$

4) $\lambda'([q, b]) = b$

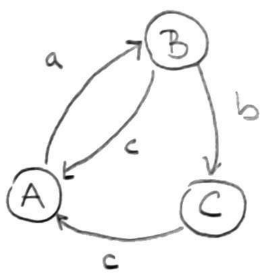
ako za neki niz $\vec{w} = a_1 a_2 \dots a_n$ $(n \geq 1, a_i \in \Sigma)$

mealye ispisuje:

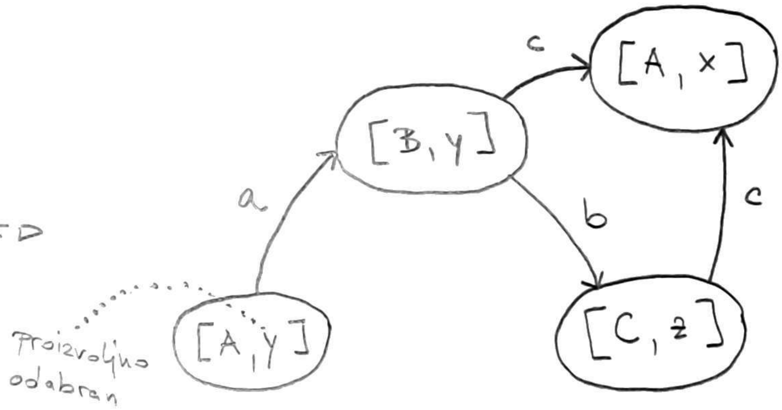
$\lambda(q_0, a_1)\lambda(q_1, a_2) \dots \lambda(q_{n-1}, a_n) = b_1 b_2 \dots b_n$

$\lambda'([q_0, b_0])\lambda'([q_1, b_1]) \dots \lambda'([q_{n-1}, b_n]) = b_0 b_1 b_2 \dots b_n$.

npr.



⟹



λ	a	b	c
A	y		
B		z	x
C			x

q	p
[A, y]	y
[B, y]	y
[C, z]	z
[A, x]	x

+ dodati y na početak izlaza

REGULARNI IZRAZI (REGEX)

- opis reg. jezika

Def.

Ako je jezik moguće opisati reg. izrazom onda je jezik regularan. $\Rightarrow$ $\exists$ dka : $L(dka) = L(regex)$.

Regexi se definiraju rekurzivno nad abecedom Σ i opisuju jezik:

1) $\emptyset$ jest regex : $L(\emptyset) = \{\}$
2) ε jest regex : $L(\varepsilon) = \{\varepsilon\}$
3) $\forall a \in \Sigma$, a jest regex : $L(a) = \{a\}$
4) r i s su regexi : $L(r)$ i $L(s)$:

   a) $(r)+(s) = r+s$: $L((r)+(s)) = L(r+s) = L(r) \cup L(s)$

      $r+s \equiv r|s$

   b) $(r)(s) = rs$: $L((r)(s)) = L(r)L(s)$ - nadovezivanje

   c) $(r)^* = r^*$: $L((r)^*) = L(r)^*$ - Kleeneov operator.

PREDNOST OPERATORA:

1) $*$

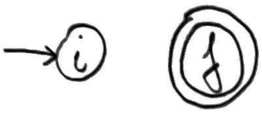
2) nadovezivanje

3) $+$

lijevo asocijativni.

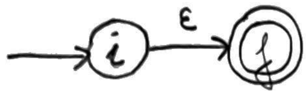
KONSTRUKCIJA: REGEX → ε-NKA

1) $\emptyset$



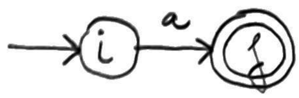
ne prihvaća ništa

2) ε

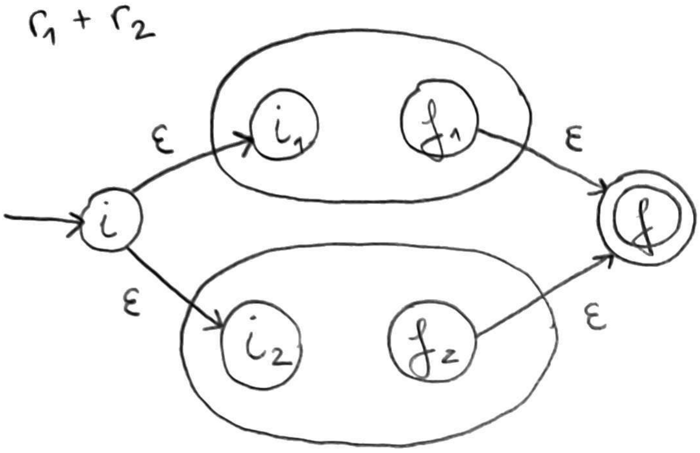


prihvaća samo prazan niz

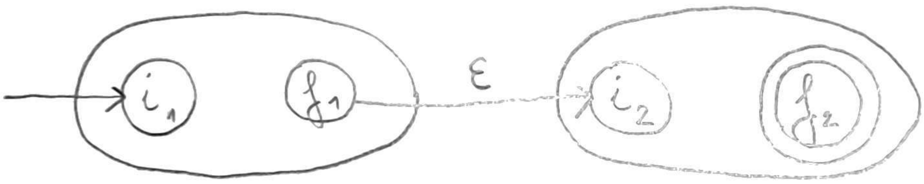
3) a, $a \in \Sigma$



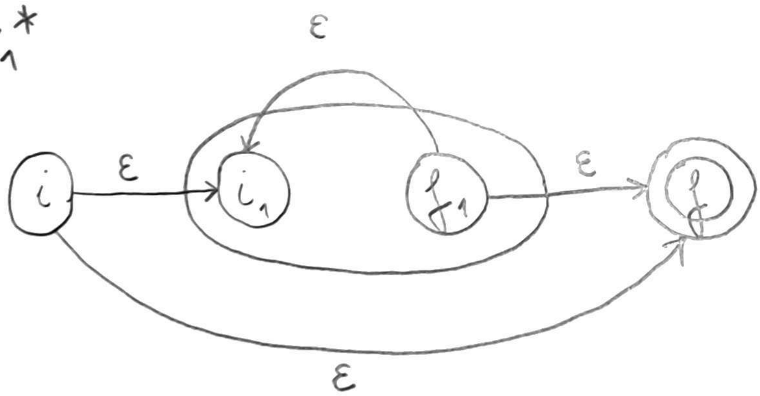
4) $r_1 + r_2$



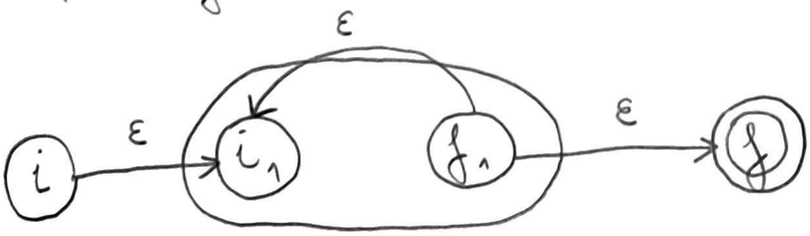
5) $r_1 r_2$



6) r_1^*



7) r_1^+ - jedno ili više ponavljanja r_1



SVOJSTVA ZATVORENOSTI REGULARNIH IZRAZA

Def.
Zatvorenost klase jezika definira se s obzirom na pojedine operacije nad jezicima. Klasa jezika zatvorena je s obzirom na neku operaciju ako primjenom te operacije na bilo koji jezik iz te klase jezik koji je u toj klasi.

REGULARNI JEZICI ZATVORENI su S OBZIROM NA:

- UNIJU $(+, \mid)$
- NADOVEZIVANJE $(\cdot)$
- KLEENEOV OPERATOR
- KOMPLEMENT $(^c)$
- PRESJEK
- SUPSTITUCIJA

KOMPLEMENT

DKA $M=(Q, \Sigma, \delta, q_0, F)$

$L(M)$

$L(M)^c$ DKA $M'=(Q, \Sigma, \delta, q_0, Q \setminus F)$

$$L(M)^c = L(M') = \{\vec{w} \mid \delta(q_0, \vec{w}) \in (Q \setminus F)\} =$$

$$= \Sigma^* \setminus \{\vec{w} \mid \delta(q_0, \vec{w}) \in F\} =$$

$$= \Sigma^* \setminus L(M) = L(M)^c$$

SUPSTITUCIJA

- vidi regularne def.

PRESJEK

$$L \cap N = (\text{De Morgan}) =$$

$$= ((L \cap N)^c)^c = (L^c \cup N^c)^c$$

$\Rightarrow$ operacije $\cup$ i ^{c} su zatvorene $\Rightarrow$ presjek zatvoren.

REGULARNE DEFINICIJE

reg. def. su oblika:

$d_1 \to r_1$
$d_2 \to r_2$
$\vdots$
$d_i \to r_i$
$\vdots$
$d_n \to r_n$

d_i su znakovi označeni različitim imenima,

r_i su regexi nad $\Sigma \cup \{d_1, d_2, \ldots, d_{i-1}\}$

važno! *

* npr.

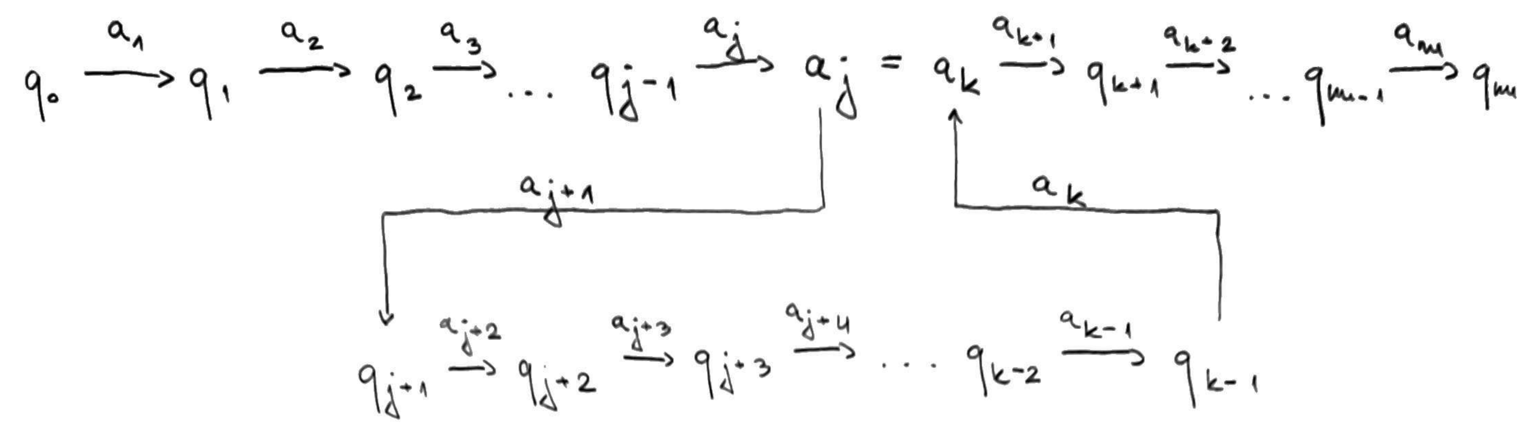
$d_1 \to r_1$
$d_2 \to r_2$
$d_3 \to d_1^* + r_3$, ali nikako npr.
$d_3 \to d_4^+ r_1$

npr.

slovo $\to$ A + B + ... + Z + a + b + ... + z
brojka $\to$ 0 + 1 + ... + 9
varijabla $\to$ slovo (slovo + brojka)*

SVOJSTVO SUPSTITUCIJE OMOGUĆUJE REG. DEF.

SVOJSTVO NAPUHAVANJA - 54. str.



ako je jezik L regularan onda $\exists$ dka $= (Q, \Sigma, \delta, q_0, F) : L = L(dka)$.

neka je:

$$|Q| = n$$

$$\vec{w} = a_1 a_2 \dots a_m, \quad m > n$$

$$\delta(q_0, a_1 a_2 \dots a_i) = q_i$$

u nizu stanja $q_0 q_1 q_2 \dots q_n$ barem se jedno ponavlja i postoje indeksi

$j, k : \; 0 \leq j < k \leq n \;$ i $\; q_j = q_k$.

budući da je $j < k$ i $k \leq n$ za niz $a_{j+1} a_{j+2} \dots a_k$ vrijedi:

$$1 \leq |a_{j+1} a_{j+2} \dots a_k| \leq n$$

neka se niz $a_1 a_2 \dots a_m$ označi sa z.

$z = uvw$, $\quad u = a_1 a_2 \dots a_j$

$v = a_{j+1} a_{j+2} \dots a_k$

$w = a_{k+1} a_{k+2} \dots a_m$

ako je $q_m \in F \Rightarrow z \in L(dka)$ i $uw \in L(dka)$

zato jer $\delta(q_0, uw) = \delta(\delta(q_0, u), w) = \delta(q_j, w) = \delta(q_k, w) = q_m$

Prihvaćaju se svi nizovi oblika $uv^i w$, $i \geq 0$. Podniz v možemo "napuhati".

- Dokaz za neregularnost jezika:

Ako je L regularan onda postoji konstanta $\underline{n}$:

za bilo koji niz z, $|z| > n$ možemo rastaviti na podnizove $z = uvw$ na način:

$$|uv| \leq n \;\wedge\; 1 \leq |v|$$

i $uv^i w$ je također $\in L$

Algoritmi odlučivanja

neka DKA M prihvaća $L(M)$ i $|Q| = n$.

a) reg. jezik $L(M)$ je neprazan akko DKA prihvaća niz z:

$$|z| < n$$

b) reg. jezik je beskonačan akko M prihvaća niz z:

$$n \leq |z| < 2n.$$

FORMALNA GRAMATIKA

- formalna definicija gramatike zasniva se na kontekstno neovisnoj gramatici (novo).

- regularni jezici $\subset$ kontekstno neovisnih jezika

$$G = (V, T, P, S)$$

V - konačan skup nezavršnih znakova

T - konačan skup završnih znakova

P - konačan skup produkcija: $A \rightarrow \alpha$, $A \in V$, $\alpha \in (V+T)^*$

S - početni nezavršni znak.

RELACIJA $\underset{G}{\Rightarrow} : (V+T)^* \rightarrow (V+T)^*$

$$\alpha A \gamma \underset{G}{\Rightarrow} \alpha \beta \gamma \quad , \quad A \in V, \quad \alpha, \beta, \gamma \in (V+T)^*$$

- tj. niz $\alpha\beta\gamma$ generira se neposredno iz niza $\alpha A \gamma$ primjenom produkcije $A \rightarrow B$.
- oznaka G označava kojoj gramatici produkcija pripada (može se izostaviti ako je jasno kojoj gramatici pripada)

$$G \text{ generira jezik } L(G) = \{ \vec{w} \mid \vec{w} \in T^* : S \overset{*}{\Rightarrow} \vec{w} \}$$

G_1 i G_2 su istovjetne akko generiraju isti jezik.

REGULARNA GRAMATIKA

- regularni jezici $\subset$ kontekstno neovisni jezici

KONSTRUKCIJA: DKA $\rightarrow$ REGULARNA GRAMATIKA

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

$$G = (V, T, P, S) \; : \; L(dka) = L(G)$$

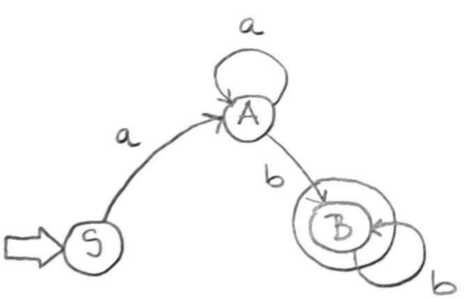
1) $T = \Sigma$
2) $V = Q$
3) $S = q_0$
4) $\delta(A, a) = B$, $A, B \in Q \wedge a \in \Sigma$

$$A \rightarrow aB$$

5) za prihvatljiva stanja grade se produkcije:

$$A \rightarrow \varepsilon, \quad A \in F.$$

npr.



$$G = (\{S, A, B\}, \{a, b\}, P, S)$$

P:

$S \rightarrow aA$

$A \rightarrow aA$

$A \rightarrow bB$

$B \rightarrow bB$

$B \rightarrow \varepsilon$

dka prihvaća jezik $L(a^+b^+)$.

generirati niz aaabb:

$$S \Rightarrow \underline{aA} \Rightarrow a\underline{aA} \Rightarrow aa\underline{aA} \Rightarrow aaa\underline{bB} \Rightarrow aaab\underline{bB} \Rightarrow aaabb.$$

dka prihvaća niz <u>aaabb</u> prolaskom kroz stanja: SAAABB.

KONSTRUKCIJA: REGULARNA GRAMATIKA → NKA

$G = (V, T, P, S)$, produkcije oblika $A \to aB$ ili $A \to \varepsilon$,
$A, B \in V$ i $a \in T$.

$nka = (Q, \Sigma, \delta, q_0, F)$: $L(nka) = L(G)$

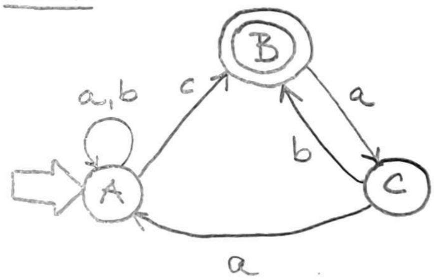
1) $\Sigma = T$
2) $Q = V$
3) $q_0 = S$
4) iz produkcije $A \to aB$ gradi se prijelaz:
$\delta(A, a) = \delta(A, a) \cup \{B\}$, na početku je $\delta(A, a) = \emptyset$.
5) ako postoji prod.: $A \to \varepsilon$ onda je $A \in F$.

npr.

$G = (\{A, B, C\}, \{a, b, c\}, P, A)$

$A \to aA$
$A \to bA$
$A \to cB$
$B \to aC \mid \varepsilon$
$C \to aA \mid bB$

nka:



DESNO-LINEARNA I LIJEVO-LINEARNA GRAMATIKA

- desno-linearna gramatika:
  - bilo koja produkcija gramatike ima najviše jedan nezavršni znak na desnoj strani i to na krajnje desnoj.

$A \to wB$ ili $A \to w$

$A, B \in V$, $w \in T^*$

- lijevo-linearna gramatika:

$A \to Bw$ ili $A \to w$

$A, B \in V$, $w \in T^*$

- lijevo-linearna gramatika i desno-linearna gramatika su regularne gramatike.

…UKCIJA: DESNO-LINEARNA GRAMATIKA $\rightarrow$ NKA

1) za sve produkcije oblika:

$$A \rightarrow w \,, \quad |w| \geq 1$$

dodaje se <u>novi</u> nezavršni znak $[\varepsilon]$ i gradi se produkcija:

$$[\varepsilon] \rightarrow \varepsilon$$

a produkcija $A \rightarrow w$ se zamjenjuje s $A \rightarrow w[\varepsilon]$.

2) sve produkcije oblika:

$$A \rightarrow a_1 \dots a_n B \,, \quad n > 1$$

zamjene se produkcijama:

$$A \rightarrow a_1 [a_2 a_3 \dots a_n B]$$

$$[a_2 a_3 \dots a_n B] \rightarrow a_2 [a_3 a_4 \dots a_n B]$$

$$\vdots$$

$$[a_n B] \rightarrow a_n B$$

3) za sve produkcije:

$$A \rightarrow B$$

grade se nove produkcije:

$A \rightarrow$ desne strane produkcija u kojima je B na lijevoj strani

i izuzmu se produkcije $A \rightarrow B$ i $B \rightarrow A$.

KONSTRUKCIJA: LIJEVO-LINEARNA GRAMATIKA → ε-NKA

$G = (V, T, P, S)$ lijevo-linearna gramatika

ε-nka = M' : $L(M') = L(G)$

1) izgradi se desno-linearna gramatika $G' = (V, T, P', S)$.
Skup produkcija P zapiše se tako da se desne strane napišu obrnutim redoslijedom:

$$P' = \{ A \to \alpha^R \mid A \to \alpha \in P \}$$

$$L(G') = L(G)^R$$

2) iz G' → nka M : $L(M) = L(G') = L(G)^R$

*3) iz M → ε-nka M' : $L(M') = L(M)^R = L(G)$:

a) preuredi nka M:
<u>ako</u> $|F| > 1$:
- napravi novo stanje p i proglasi ga prihvatljivim
- $\forall q \in F$ proglasi neprihvatljivim i povuci ε prijelaze prema p

<u>inače</u>
- sve isto.

b) za početno ε-nka M' uzima se prihvatljivo stanje nka M.
za prihvatljivo stanje ε-nka uzima se početno stanje nka.

c) funkcije prijelaza ε-nka grade se zamjenom smjera grana

KONSTRUKCIJA: NKA → LIJEVO-LINEARNA GRAMATIKA

1) konstrukcija ε-nka M : $L^R = L(\varepsilon\text{-nka})$ ⇒ 3. KORAK *
2) M → desno-linearna gramatika : $L(G_1) = L(M) = L^R$
3) desne strane gramatike G napišu se u obrnutom redoslijedu.

KONTEKSTNO NEOVISNI JEZICI

- jezik jest kontekstno neovisan akko postoji kontekstno neovisna gramatika koja ga generira
- za bilo koji kontekstno neovisan jezik postoji kontekstno neovisna gramatika koja ga generira
- bilo koja kontekstno neovisna gramatika generira jedan od kontekstno neovisnih jezika

NEJEDNOZNAČNOST GRAMATIKE, JEZIKA I NIZA

- interpretacija niza zasniva se na generativnom stablu koje se gradi tijekom gen. niza.
- mogućnost izgradnje više stabala uzrokuje nejednoznačnost u interpretaciji niza.

Def.

Bilo koje generativno stablo moguće je izgraditi primjenom samo jednog postupka generiranja niza zamjenom krajnje lijevog (desnog) nezavršnog znaka.

Nejednoznačnost gramatike G:

- ako je moguće za neki niz $\vec{w} \in L(G)$ izgraditi više različitih gen. stabala onda je KNG G nejednoznačna

ili

- ako je moguće neki niz $\vec{w} \in L(G)$ generirati primjenom više različitih postupaka gen. niza zamjenom krajnje lijevog ili krajnje desnog nezavršnog znaka, onda je G nejednoznačna.

Nejednoznačnost niza $\vec{w} \in L(G)$:

- ako je moguće za niz $\vec{w} \in L(G)$ gen. više različitih generativnih stabala, onda je niz nejednoznačan za G.

Nejednoznačnost jezika L:

- ako nije moguće jezik L generirati niti jednom jednoznačnom gramatikom G, onda je jezik inherentno nejednoznačan.

POJEDNOSTAVLJIVANJE GRAMATIKE

i) bilo koji znak gramatike G koristi se u postupku generiranja barem jednog niza jezika L.

- ako se X koristi u generiranju $S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w$ -> <u>koristan</u>

inače je beskoristan (mrtav ili nedohvatljiv)

refleksivno $\overset{\neq}{i}$ tranzitivno okruženje $\Rightarrow$

- mrtav: ne postoji $X \overset{*}{\Rightarrow} \vec{w}_x$, $\vec{w}_x \in T^*$ (inače živ.)

- nedohvatljiv: ne postoji $S \overset{*}{\Rightarrow} \alpha X \beta$ (inače dohvatljiv.)

- živ i dohvatljiv, ali beskoristan:

neka vrijedi: $S \overset{*}{\Rightarrow} \alpha X \beta$ i $X \overset{*}{\Rightarrow} \vec{w}_x$, $\vec{w}_x \in T^*$

ako α ili β sadrži mrtvi znak, onda X nije koristan

ii) Gramatika G nema produkcija oblika: $A \to B$ (jedinične prod.)

ostalo: nejedinične prod.

iii) ako $\varepsilon \notin L$ onda je moguće izbjeći korištene ε-produkcija: $A \to \varepsilon$.

ODBACIVANJE MRTVIH ZNAKOVA

$$\nexists\, X \overset{*}{\Rightarrow} \vec{w}_x \,,\quad \vec{w}_x \in T^*$$

a) Ako su živi svi znakovi $X_1, X_2, \dots, X_n$ desne strane produkcije

$$A \rightarrow X_1 X_2 \dots X_n$$

onda je živ i A.

$\forall X_i$ vrijedi: $X_i \overset{*}{\Rightarrow} \vec{w}_i \,, \vec{w}_i \in T^* \rightarrow A \Rightarrow X_1 X_2 \dots X_n \overset{*}{\Rightarrow} \vec{w}_1 \vec{w}_2 \dots \vec{w}_n$

b) živi su znakovi koji s desne strane imaju samo završne znakove:

$$A \rightarrow \vec{w} \,,\quad \vec{w} \in T^*$$

ODBACIVANJE NEDOHVATLJIVIH ZNAKOVA

X dohvatljiv akko $S \overset{*}{\Rightarrow} \alpha X \beta$

a) ako je dohvatljiv A onda su dohvatljivi svi završni i nezavršni znakovi s desne strane:

$$A \rightarrow \alpha \beta \gamma$$

- algoritam je sličan kao kod traženja nedohvatljivih stanja DKA.

ODBACIVANJE BESKORISNIH ZNAKOVA

a) odbaci mrtve

b) odbaci nedohvatljive

- nikako obrnuto! Primjer 2.6. str. 80.

ODBACIVANJE ε-PRODUKCIJA

- ako $\varepsilon \notin L(G)$ moguće je izgraditi istovjetnu gramatiku G' koja nema ε-produkcija.

1) Pronađi sve nezavršne znakove koji generiraju prazan niz. Prazni znakovi su oni za koje vrijedi: $A \overset{*}{\Rightarrow} \varepsilon$.

   Traženje takvih -> trivijalno: rekurzivno od svih ε-produkcija.

2) Ako je produkcija od G:

$$A \to X_1 X_2 \dots X_n$$

onda se skupu produkcija G' dodaju produkcije oblika:

$$A \to \xi_1 \xi_2 \xi_3 \dots \xi_n \qquad (\xi \text{ - ksi})$$

$$\xi_i = (X_i \text{ prazan } ?\ X_i : (X_i \vee \varepsilon))$$

- PRODUKCIJE SE GRADE NA TEMELJU SVIH MOGUĆIH VRIJEDNOSTI ξ_i.

- IZBACE SE ε-PRODUKCIJE

- Primjer 2.8. str. 31.

ODBACIVANJE JEDINIČNIH PRODUKCIJA

Iz G moguće je dobiti G' koja nema jediničnih produkcija

a) u PRODUKCIJE G' DODATI SVE IZ G KOJE NISU JEDINIČNE.

b) AKO POSTOJI $A \overset{*}{\Rightarrow} B$ ONDA SE ZA SVE PRODUKCIJE $B \to \alpha$ KOJE NISU JEDINIČNE GRADE PRODUKCIJE $A \to \alpha$.

CHOMSKYJEV NORMALNI OBLIK PRODUKCIJA

Neka $G=(V,T,P,S)$ generira KNJ $L(G)\setminus\{\varepsilon\}$, onda je moguće izgraditi istovjetnu gramatiku G' koja ima sve produkcije oblika $A \to BC$ ili $A \to a$.

0) iz G izbaciti beskorisne znakove, ε-produkcije i jedinične prod.

1) u P' se stave sve iz P koje su već u Chomskyjevom obliku.

2) Za sve produkcije oblika:

$$A \to X_1 X_2 \ldots X_m, \quad X_i \in T \lor X_i \in V$$

završni znakovi X_i zamjenjuju se novim nezavršnim C_i i uvodi se produkcija: $C_i \to X_i$.

3) nakon 2) sve produkcije su u obliku:

$$A \to a \quad \text{ili} \quad A \to B_1 B_2 \ldots B_m, \quad m \geq 2 .$$

- produkcije oblika: $A \to B_1 B_2 \ldots B_m$ za $m \geq 3$ mijenjaju se novim produkcijama i definiraju se novi nezavršni znakovi $D_1, D_2, \ldots, D_{m-2}$ i produkcija $A \to B_1 B_2 \ldots B_m$, $m \geq 3$

zamijeni se skupom produkcija:

$$\begin{aligned}
&A \to B_1 D_1 \\
&D_1 \to B_2 D_2 \\
&D_2 \to B_3 D_3 \\
&\vdots \\
&D_{m-3} \to B_{m-2} D_{m-2} \\
&D_{m-2} \to B_{m-1} B_m
\end{aligned}$$

GREIBACHIN NORMALNI OBLIK PRODUKCIJA

- Neka $G = (V, T, P, S)$ generira jezik $L(G) \setminus \{\varepsilon\}$. Moguće je izgraditi istovjetnu gramatiku G' koja ima sve produkcije oblika $A \to a\alpha$. $(\alpha \in V^*)$

Algoritam zamjene krajnje lijevog nezavršnog znaka:

$$\begin{array}{lcl} D_j \to \alpha_1 & & D_i \to \alpha_1 \gamma \\ D_j \to \alpha_2 & & D_i \to \alpha_2 \gamma \\ \vdots & \Rightarrow & D_i \to \alpha_3 \gamma \\ D_j \to \alpha_r & & \vdots \\ D_i \to D_j \gamma & & D_i \to \alpha_r \gamma \end{array}$$

zamjena $(r+1)$ produkcije s r.

Algoritam razrješavanja lijeve rekurzije:

- isti znak na lijevoj strani produkcije i na krajnje lijevom mjestu desne strane produkcije → produkcija je LIJEVO REKURZIVNA.

$$\begin{array}{lcl} D_i \to D_i \alpha_1 & & D_i \to \beta_l, \quad 1 \le l \le s \\ D_i \to D_i \alpha_2 & & D_i \to \beta_l C_i, \quad 1 \le l \le s \\ \vdots & & D_i \to \alpha_k, \quad 1 \le k \le r \\ D_i \to D_i \alpha_r & \Rightarrow & D_i \to \alpha_k C_i, \quad 1 \le k \le r \\ \hline D_i \to \beta_1 & & \\ D_i \to \beta_2 & & C_i \text{ novi nezavršni znak.} \\ \vdots & & \\ D_i \to \beta_s & & \end{array}$$

1) G u Chomskyjev oblik → $D_i \to a$ su u Greibachinovom obliku!

2), 3), 4) ... str. 85.

PARSIRANJE NIZA

- Prepoznavanje niza: određivanje da li gramatika G generira niz $\vec{w}$...
  - ako niz pripada jeziku $L(G)$ gradi se generativno stablo koje se koristi za interpretaciju niza. (stablo parsiranja)

- Parsiranje niza: prepoznavanje niza i gradnja generativnog stabla.

Parsiranje od vrha prema dnu

- zadana gramatika $G = (V, T, P, S)$
- započinje od S
- završni znakovi $a \in \vec{w}$ određuju produkciju
- produkcije gram. G primjenjuju se sve dok se listovi stabla ne označe isključivo završnim znakovima $\vec{w}$.

LL(1) gramatika i LL(1) parsiranje:

left or right scanning

krajnje lijevi nezavršni znak

odluka se donosi na temelju 1 pročitanog znaka (look ahead)

postoji i LL(k), $k \geq 1$.

- Tehnika rekurzivnog spusta → vidi Labos. Trivijalno.

Parsiranje od dna prema vrhu

- od listova prema korijenu.
- u nizu $\vec{w}$ nastoji se prepoznati jedna od desnih strana produkcije.
  ~ mijenjaju se desne strane lijevima (reducira se)
- produkcije se u ovom postupku nazivaju redukcije.
- vidi Primjer 3.13. str. 97.
- nejednoznačno: ako postoji barem jedno gen. stablo onda $\vec{w} \in L(G)$

LR parser

- left to right, rightmost derivation
- LR(k), k look ahead
- postoje KNG za koje nije moguće izgraditi LR parser: nisu LR gramatike
- složenije od tehnike rekurzivnog spusta
- LR parseri različitih gramatika razlikuju se samo u tablici parsiranja

Model LR parsera

- ulazni spremnik, potisni stog, program LR parsera, tablica parsiranja, izlaz.

Program LR parser: čita znak po znak

Ulazni spremnik: sadrži niz $\vec{w}$ koji se parsira

Program parsiranja koristi se stogom za spremanje niza oblika:

$$s_0 X_1 s_1 X_2 s_2 \ldots X_m s_m \text{ ; } s_m \text{ na vrhu } X_i \in V \vee X_i \in T.$$

s_i su stanja

- Tijekom rada LR parser koristi samo stanja.

Tablica parsiranja

- tablica Akcija i tablica NovoStanje
- četiri vrste akcija:
  - Pomak s
  - Reduciraj $A \to \beta$
  - Prihvati } određuju izlaz parsera, da li zadani
  - Odbaci } niz $\vec{w}$ pripada jeziku $L(G)$.
- odluka na temelju znaka na vrhu stoga (s_m) i znaka a_i niza $\vec{w}$.

KONFIGURACIJA LR parsera:

$$(\underbrace{s_0 X_1 s_1 X_2 s_2 X_3 s_3 \ldots X_m s_m}_{\text{sadržaj stoga}},\ \underbrace{a_i a_{i+1} \ldots a_n}_{\text{nepročitani dio ulaza}})$$

s_m — vrh stoga

a_i — znak koji se čita

neka je LR parser u konfiguraciji:

$$(s_0 X_1 s_1 X_2 s_2 \ldots X_m s_m , \; a_i a_{i+1} \ldots a_n)$$

LR parser čita znak a_i: odredi se sljedeća akcija na temelju $[s_m, a_i]$ iz tablice akcija:

1) Akcija$[s_m, a_i]$ = Pomak s

push(a_i)

push(stanje iz Akcija$[s_m, a_i]$) = push(s_m)

2) Reduciraj $A \to \beta$

pop $2 \cdot |\beta|$ znakova.

novo stanje = NovoStanje$[s_{m-|\beta|}, A]$

push(A)

push(novo stanje)

- ne prelazi na idući znak!

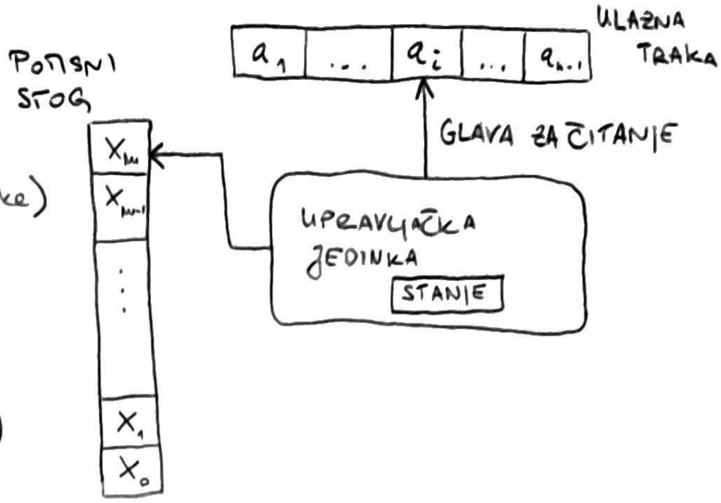
3) Prihvati

$\vec{w}$ se prihvaća, parsiranje je gotovo

4) Odbij

$\vec{w}$ se ne prihvaća, parsiranje je gotovo

POTISNI AUTOMAT (PA)



- prihvaćaju KN jezike
- odluke se donose na temelju:
  - stanja UJ (upravljačke jedinke)
  - znaka na vrhu stoga
  - znaka na ulaznoj traci
- moguće je da (znak na ulaznoj traci) ne sudjeluje u odluci. U tom slučaju glava za čitanje se ne miče jedno mjesto u desno.
- upravljačka jedinka stavlja niz na vrh stoga:

pročitaj znak; skini X_m sa stoga; donesi odluku; stavi neki niz na stog

1) Prazni niz ε ≡ uzimanje jednog znaka s vrha stoga
2) Niz duljine jednog znaka ≡ stavljanjem istog znaka se ne mijena sadržaj
   stavljenjem novog znaka postiže se zamjena znakova s vrha stoga
3) Niz duljine više znakova ≡ simulacija primjene produkcije; lijevi nezavršni lijeve strane produkcije na vrhu stoga.

(q, a, Z)

q – stanje; a – znak sa trake; Z – znak sa vrha stoga

odluke:

1) $(q, a, Z) \to (p, \gamma)$; mijenja se stanje u p, na stog stavi γ (prije toga makni Z sa stoga)

2) $(q, \varepsilon, Z) \to (p, \gamma)$ $\wedge$ glava za čitanje se ne pomiče u desno.

PA prihvaća niz $\vec{w}$ na dva načina:

a) prihvatljivim stanjem: ako PA nakon pročitanih svih znakova niza $\vec{w}$ uđe u prihvatljivo stanje, niz se prihvaća

b) praznim stogom: isprazni li se stog čitanjem svih znakova, niz se prihvaća.

NIZ SE ODBIJA AKO $\exists a \in \vec{w} \wedge \exists Z \in$ stoga : $\delta(p, a, Z)$ nije definirana $\to$ PA staje s radom!

DEFINICIJA POTISNOG AUTOMATA

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

Γ - konačan skup znakova stoga (abeceda znakova stoga)

$\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$

Z_0 - početni znak stoga

oznake:

$a, b, c, d, \ldots \in \Sigma$

$v, w, x, y, z \in \Sigma^*$

$A, B, C, \ldots \in \Gamma$

$\alpha, \beta, \gamma, \ldots \in \Gamma^*$

$$\delta(q, a, Z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \ldots, (p_m, \gamma_m)\},$$

$q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$

- PA prelazi u jedno od m stanja. Niz γ_i na stog se stavlja tako da se prvo stavi krajnje desni znak u nizu γ_i.

- fja δ je nedeterministička (kao i $\delta(q, a)$ kod NKA).

- za fja $\delta(q, \varepsilon, Z)$ glava za čitanje se <u>ne miče</u> u desno jer ulazni znak nema utjecaj na prijelaz.

KONFIGURACIJA PA

- (stanje, nepročitani dio ulaza, znakovi na stogu)
- $(q, \vec{w}, \gamma)$
- $(q, a\vec{w}, Z\alpha) \rightarrow (p, \vec{w}, \beta\alpha) \iff (p, \beta) \in \delta(q, a, Z)$, $a \in \Sigma \cup \{\varepsilon\}$
- $(q, a\vec{w}, Z\alpha) \underset{M}{\succ} (p, \vec{w}, \beta\alpha)$ —||—
- $\overset{*}{\succ}$ refleksivno i tranzitivno okružene $\succ$.

PRIHVAĆANJE JEZIKA PA

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

a) prihvaća prihvatljivim stanjem:

$$L(M) = \{ \vec{w} \mid (q_0, \vec{w}, Z_0) \overset{*}{\succ} (p, \varepsilon, \gamma),\ p \in F \wedge \gamma \in \Gamma^* \}$$

b) prihvaća praznim stogom:

$$N(M) = \{ \vec{w} \mid (q_0, \vec{w}, Z_0) \overset{*}{\succ} (p, \varepsilon, \varepsilon),\ p \in Q \}$$

DETERMINISTIČKI PA (DPA)

PA $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je deterministički akko vrijede oba uvjeta:

1) $\delta(q, \varepsilon, Z) \neq \emptyset \Rightarrow \delta(q, a, Z) = \emptyset, \ \forall a \in \Sigma$

2) $|\delta(q, a, Z)| = 1, \ \forall (q, a, Z) \in Q \times \Sigma \cup \{\varepsilon\} \times \Gamma$

drugim riječima:

- 1) ne dozvoljava mogućnost izbora između ε-prijelaza i prijelaza koji koristi ulazni znak.
- 2) ne dopušta izbor prijelaza!

- nedeterministički potisni automat (PA) prihvaća širu klasu jezika od determinističkog potisnog automata (DPA).

POTISNI AUTOMAT I KONTEKSTNO NEOVISNA GRAMATIKA

$P_1 \equiv P_2 \iff L(P_1) = L(P_2)$.

- pokazati istovjetnost PA koji prihvaća jezik praznim stogom (Z_0-PA) i PA koji prihvaća jezik prihvatljivim stanjem (q_0-PA).

KONSTRUKCIJA: q_0-PA $\longrightarrow$ Z_0-PA

q_0-PA $M_2 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

konstruirati Z_0-PA M_1 : $M_1 \equiv M_2$.

$M_1 = (Q \cup \{q_0', q_e\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, \emptyset)$

X_0 - dodatan znak stoga; stavlja se na početak.

1) $\delta'(q_0', \varepsilon, X_0) = \{(q_0, Z_0X_0)\}$
   - početak rada $M_1 \rightarrow$ automat M_2 stavlja X_0 na stog.

2) $\delta'(q, a, Z) = \delta(q, a, Z)$

3) $\delta'(q, \varepsilon, Z)$ dodaj ε-prijelaz (q_e, ε), $q \in F$.

4) $\delta'(q_e, \varepsilon, Z)$ dodaje se ε-prijelaz (q_e, ε).

~ uđe li M_2 u prihvatljivo stanje, M_1 isprazni stog.

- znak X_0 dodaje se zato što M_2 svojim prijelazima ne može isprazniti stog M_1.

- ako M_2 isprazni stog (tj. ostane samo X_0) i nije u prihvatljivom stanju onda M_1 ne može prihvatiti niz.

KONSTRUKCIJA: Z_0-PA → q_0-PA

Z_0- PA $M_1 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$

q_0- PA $M_2 = (Q \cup \{q_0', q_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, \{q_f\})$

1) $\delta'(q_0', \varepsilon, X_0) = \{(q_0, Z_0X_0)\}$

- na dno stoga X_0. Pročita li M_2 znak X_0 to znači da je M_1 ispraznio svoj stog

2) $\delta'(q, a, Z) = \delta(q, a, Z)$

3) $\delta'(q, \varepsilon, X_0)$ dodaj ε-prijelaz (q_f, ε)

KONSTRUKCIJA: KNG → Z_0-PA

$N(M) = L = L(G)$, $\varepsilon \notin L$.

$G = (V, T, P, S)$, produkcije u Greibachinom obliku
($A \to a\alpha$, $\alpha \in V^*$)

Z_0-PA $M = (\{q\}, \Sigma, \Gamma, \delta, q, S, \emptyset)$

a) jedno stanje q

b) $\Sigma = T$

c) $\Gamma = V$

d) S automata M = S gramatike G

e) $F = \emptyset$

$(q, \gamma) \in \delta(q, a, A) \iff \exists$ produkcija $A \to a\gamma$.

KONSTRUKCIJA: Z_0-PA → KNG

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

$G = (V, T, P, S)$

nezavršni znakovi: $V = \{ [q, A, p] \mid q, p \in Q \text{ i } A \in \Gamma \}$

početni nezavršni: $S \in V$.

1) za početno stanje q_0, nezavršni znak Z_0, i $\forall q \in Q$ gradi se produkcija:

$$S \to [q_0, Z_0, q]$$

2) Ako skup $\delta(q, a, A)$ sadrži

$$(q_1, B_1 B_2 \dots B_m)$$

grade se produkcije:

$$[q, A, q_{m+1}] \to a [q_1, B_1, q_2][q_2, B_2, q_3] \dots [q_m, B_m, q_{m+1}]$$

- za znakove $q_2, q_3, q_4, \dots, q_{m+1}$ uzimaju se sve moguće kombinacije svih stanja iz skupa Q.
- ako $\delta(q, a, A)$ sadrži (q_1, ε) onda se gradi produkcija:

$$[q, A, q_1] \to a$$

SVOJSTVA ZATVORENOSTI KNJ

- unija
- nadovezivanje
- kleeneov operator
- supstitucija

Presjek i komplement

- KNJ nisu zatvoreni s obzirom na operaciju presjeka
- -II- na operaciju komplementa
- vidi na 122. str. - ZANIMLJIVO !

PRESJEK REG. JEZIKA I KNJ = KNJ - str. 123.

SVOJSTVO NAPUHAVANJA

- zasniva se na broju čvorova gen. stabala i broju $|V|$.
- vidi na 124.

- ako je L KNJ onda $\exists\, n : |z| \geq n$ onda je z moguće napisati kao $uvwxy$:

  a) $|vx| \geq 1$

  b) $|vwx| \leq n$

  c) $\forall\, i \geq 0$ vrijedi $uv^i w x^i y \in L$.

TURINGOV STROJ

OSNOVNI MODEL TURINGOVOG STROJA

$$ts = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

Γ - konačan skup znakova trake

$B \in \Gamma$ - znak prazne ćelije

$\Sigma \subseteq (\Gamma - \{B\})$ - konačan skup ulaznih znakova

$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times (L, R)$

- dozvoljava se da je δ nedef. za pojedine argumente.

- glava za čitanje i pisanje
- ulazna traka beskonačno dugačka na desno.
- upravljačka jedinka

KONFIGURACIJA TS

$\alpha_1 q \alpha_2$, α_1 - zapis trake lijevo od glave za čitanje

α_2 - zapis trake desno od glave za čitanje

$q \in Q$ - trenutno stanje TS.

- TS u idućem koraku čita krajnje lijevi znak od α_2.

neka je TS u konfig.:

$$X_1 X_2 \ldots X_{i-1}\, q\, X_i X_{i+1} \ldots X_n$$

i zadana je f-ja:

$$\delta(q, X_i) = (p, Y, L)$$

- $i-1=n$, TS čita B i glava se nalazi skroz desno
- $i=1$, glava se nalazi skroz lijevo $\to$ TS stop!
- inače, $x_1 x_2 \ldots x_{i-1} q x_i x_{i+1} \ldots x_n \succ x_1 x_2 \ldots x_{i-2}\, q\, x_{i-1} Y x_{i+1} \ldots x_n$

TURINGOV STROJ TS $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ prihvaća jezik:

$$L(M) = \{\vec{w} \mid \vec{w} \in \Sigma^* \wedge q_0\vec{w} \overset{*}{\succ} \alpha_1 p \alpha_2,\ p \in F,\ \alpha_1, \alpha_2 \in \Gamma^*\}$$

TS može stati:

a) čim uđe u $p \in F$

b) ako δ nije definirana za (q, a), $q \in Q$, $a \in \Gamma$.

- ako je $q \in F$ niz se prihvaća.

s a) i b) prihvaća se ista klasa jezika $\Rightarrow$ nadalje koristimo a).

METODE IZRADE TS

VIŠEKOMPONENTNA OZNAKA STANJA

$[q_1, q_2, \dots, q_n]$, q_i komponenta složene oznake stanja.

- upravljačka i radna komponenta složene oznake stanja
- upravljačke upravljaju redom
- radne se koristi za spremanje podataka

VIŠEKOMPONENTNI ZNAKOVI TRAKE

$A_j = [a_1, a_2, \dots, a_n]$ složen znak s komponentama a_i, $1 \le i \le n$.

- pojedine komponente složenog znaka pišu se u zasebne tragove trake. Ako je $|A_j| = n \Leftrightarrow$ traka ima n tragova.

- označni trag ($\checkmark$); pomoćna komponenta za izradu TS koji prihvaćaju jezike u kojima se dijelovi niza ponavljaju.

PROŠIRENI MODELI TS

$$\text{TS } T_1 \equiv \text{TS } T_2 \iff L(T_1) = L(T_2).$$

TS S DVOSTRANOM BESKONAČNOM TRAKOM

- osnovne razlike od osnovnog modela:

  1) $\delta(q, X) = (p, Y, L) \to qX\alpha \succ pBY\alpha$

  2) $\delta(q, X) = (p, B, R) \to qX\alpha \succ p\alpha.$

- ideja simulacije dvostranog TS M_2 osnovnim modelom TS M_1

  1) podijeliti ulaznu traku na dva traga:

$\supset$	A_5	A_4	A_{-3}	A_{-2}	A_{-1}	A_0	A_1	A_2	A_3	A_4	A_5	$\subset$

traka M_2

A_0	A_1	A_2	A_3	A_4	A_5	$\subset$
$¢$	A_{-1}	A_{-2}	A_{-3}	A_{-4}	A_{-5}	$\subset$

traka M_1

  2) više komponentna stanja: $[q, G]$ ili $[q, D]$, $q \in Q$.

     – G označava da M_1 koristi gornji trag

     – D označava da M_1 koristi donji trag

  3) prazne ćelije = $[B, B]$

  4) $\Sigma_1 = \{[a, B] \mid a \in \Sigma_2\}$ - nemjestiti tako ulaznu traku M_1.

  5) Ako je $\delta_2(q_2, a) = (q, X, R)$:

$$\delta_1(q_1, [a, B]) = ([q, G], [X, ¢], R),$$

gdje su q_1 i q_2 početna stanja M_1 i M_2. Početno stanje M_1 nije višekomponentno zbog ove def.

  6) slično i za $\delta_2(q_2, a) = (q, X, L)$

  7) ostali prijelazi su logični i lagano ih je sastaviti!

TS s višestrukim trakama

- M_1 ima k traka
- svaka traka ima svoju glavu za čitanje / pisanje

jednim prijelazom TS:

1) promijeni stanje
2) zapiše k znakova na k traka
3) pomakne bilo koju glavu neovisno L / R.

položaj glave 1	X			
sadržaj trake 1	A_1	A_2	...	A_m
položaj glave 2		X		
sadržaj trake 2	B_1	B_2	...	B_m
⋮				
položaj glave k	X			
sadržaj trake k	C_1	...	C_m	

- M_2 koji simulira M_1 ima $2 \cdot k$ tragova.
- Stanje TS M_2 ima $k+2$ komponenti.
  - 1 za stanje M_1
  - k za znakove
  - brojilo za položaje glave

- simulacija M_1 za jedan prijelaz:
  a) glava M_2 miče se od krajnje lijeve oznake do desne položaja glave
  b) ako naiđe na oznaku položaja pročita znak i spremi ga u komponentu stanja
  c) kad pročita svih k znakova donose se odluke:
    - ponovno ide od lijeva na desno i radi prijelaze.

za m pomaka glave M_1, automat M_2 radi $\approx 2m^2$ pomaka glave.

NEDETERMINISTIČKI TS (NTS)

- δ nije jednoznačna

$$|\delta(q, x)| \geq 1.$$

- NTS $\equiv$ TS

- završi li barem jedan slijed prijelaza prihvatljivim stanjem $\Rightarrow$ niz se prihvaća.
- prijelazi δ su numerirani od 1 do $\underset{=r}{\max}(|\delta(q,x)|)$, $\forall (q,x) \in Q \times \Gamma$
- niz brojeva $n_1, n_2, \ldots, n_i, \ldots, n_r$ je sljedećeg značenja:

n_i : i-ta fja prijelaza koristi n_i - trojku.

- NTS će tokom svog rada generirati sve moguće podnizove.
- koristi tri trake: - ulazna
  - traka koja sadrži niz $n_1 n_2 \ldots n_r = \vec{n}$
  - kopija ulazne koja se koristi za simulaciju
- svaki put kad se generira novi niz $\vec{n}$ NTS započinje iznova simulaciju tako da prepiše sadržaj trake 1 na traku 3.

TS S VIŠEDIMENZIONALNIM ULAZNIM POLJEM

- odabere se prostorni (k-dimenzionalni) pravokutnik.
- ta "matrica" se prikaže kao polje (traka) tako da se redci odvoje posebnim znakom *. Dodatno, početak prvog i kraj posljednjeg bloka označeni su s **.

- tijekom rada TS taj pravokutnik se proširi po potrebi ako glava TS izleti iz njegovih dimenzija

- vrlo jednostavno - str. 145.

TS S VIŠE GLAVA ZA R/W

- jedna traka
- više nezavisnih glava
- odluka se donosi na temelju k pročitanih znakova
- istovjetan s osnovnim modelom s 1 glavom
    - gradi se TS s jednom glavom i trakom s k+1 tragom
    - 1 trag za sadržaj, ostalih k za položaj R/W glave.

NEIZRAVNI TS

- za istraživanje prostorne složenosti prihvaćanja jezika i prostorne složenosti računanja cjelobrojnih funkcija
- više radnih traka i jednu ulaznu traku (može se samo čitati)
- niz na traci je ograničen znakovima ¢ i $
- glavu nije moguće pomaknuti izvan granica.
- neiznarni TS istovjetan je TS s više traka.
- dodavanjem jedne trake neizravnom TS i prijepisom niza s ulazne trake na dodatnu traku, može se simulirati bilo koji TS s višestrukim trakama

POJEDNOSTAVLJENI MODELI TURINGOVOG STROJA

STOGOVNI STROJ

- ulazna traka
- dva stoga

ako je konf. TS :

$$X_1 X_2 X_3 \ldots X_{i-1}\, q\, X_i X_{i+1} \ldots X_n$$

onda stog S_1 sadrži niz $X_1 X_2 X_3 \ldots X_{i-1}$ ← vrh stoga , a

stog S_2 sadrži niz $X_i X_{i+1} \ldots X_n$ ← vrh stoga

STROJ S BROJILIMA

- simulacija stogovnog stroja s četiri brojila koji koriste sustav brojeva po bazi k (broj različitih znakova $= |\Gamma|$) kako bi "pohranili" neki znak.

- simulacija s četiri brojila može se smanjiti na samo 2 brojila (potencijama prostih faktora)

⇒ TS s jednom trakom može se simulirati primjenom stroja sa dva brojila. TODO: ZGODNO ZA KODIRATI!

- više magije na str. 147.

TURINGOV STROJ S OGRANIČENIM BROJEM STANJA I ZNAKOVA TRAKE

- ne prihvaća isti skup jezika kao osnovni model
- za prihvaćanje bilo kojeg rekurzivno prebrojivog jezika dovoljno je imati:
  - neograničen skup Γ
  - jedna traka
  - tri stanja (1 prihvatljivo)

  ili
  - neograničen skup Q
  - $\Gamma = \{1, 0, B\}$

    ~ kodiranje znakova u binarni kôd ($|\Gamma_1|$-bitni)

M_1 neki TS koji želimo simulirati.

UNIVERZALNI TS M_u

- simuliranje bilo kojeg TS s 1 trakom
- M_u ima 3 trake:
  - na prvu se zapiše δ i $\vec{w}$ ($<M, w>$)
  - druga traka simulira sadržaj M
  - treća traka sadrži stanje M

- M_u prepisuje $\vec{w}$ s prve na drugu traku
- M_u prihvaća niz $<M, w>$ <=> TS M prihvaća $\vec{w}$.

- moguće je izgraditi M_u sa jednom trakom, pet stanja i 7 znakova trake koji prihvaća niz $<M, w>$ <=> TS M prihvaća $\vec{w}$.

GENERIRANJE JEZIKA TURINGOVIM STROJEM

- Jezik L jest RP (rekurzivno prebrojiv) $\Leftrightarrow \exists$ TS M_1:

$$\underbrace{G(M_1)}_{\text{jezik koji generira TS}} = L.$$

- Za svaki TS M_1 koji generira jezik $G(M_1)$ moguće je izgraditi TS M_2 : $L(M_2) = G(M_1)$

- TS koji generira jezik:
  - višestruke trake (1 izlazna)
  - znak na izlaznoj traci nije moguće promjeniti
  - R/W head se miče samo u desno
  - nizovi nad abecedom Γ_1 ispisuju se na izlaznu traku i odvajaju se simbolom #.
  - ako TS stane $G(M_1)$ je konačan, inače je beskonačan
  - isti niz moguće je gen. više puta.

- TS koji prihvaća jezik $G(M_1)$.
  - M_2 ima traku više od M_1
  - dodatna traka je ulazna
  - M_2 neki niz $\vec{w}$ zapiše na dodatnu (ulaznu) traku
  - M_2 na ostalim trakama simulira M_1 i svaki put kada ispiše # M_2 uspoređuje generirani niz s ulaznim nizom $\vec{w}$.

M_2 prihvaća niz $\vec{w} \Leftrightarrow M_1$ generira $\vec{w}$.

- Za bilo koji TS M_2 koji prihvaća jezik $L(M_2)$ moguće je izgraditi TS M_1 koji generira jezik $L(M_2)$.

$$G(M_1) = L(M_2).$$

  - M_1 generira na radnu traku sve $\vec{w} \in \Sigma^*$. Nakon što je generirao niz $\vec{w_i}$ M_1 simulira rad M_2. Prihvati li M_1 niz $\vec{w_i}$, M_2 ispiše $\vec{w_i}$ na izlaznu traku.
  - M_2 uvijek stane za klasu rekurzivnih jezika.
  - za rekurzivno prebrojive moguće je da M_2 nikad ne stane.

- Jezik $G(M_1)$ koji je moguće gen. kanonskim slijedom jest rekurzivni jezik

GRAMATIKA NEOGRANIČENIH PRODUKCIJA (gramatika tipa 0)

$$\alpha \to \beta \quad , \quad \alpha, \beta \in (V \cup T)^*, \ \alpha \neq \varepsilon$$

$$G = (V, T, P, S)$$

za produkciju $\alpha \to \beta$ definira se relacija:

$$\gamma \alpha \delta \Rightarrow \gamma \beta \delta$$

G generira jezik $L(G) = \{ \vec{w} \mid S \overset{*}{\Rightarrow} \vec{w}, \ \vec{w} \in T^* \}$

- generira klasu rekurzivno prebrojivih jezika

KONSTRUKCIJA: GNP $\to$ TS

- ako G generira $L(G)$ onda postoji TS M koji prihvaća $L(G)$.
- gradi se NTS M:
  - dvije trake:
    - na jednu se zapiše niz završnih znakova $\vec{w}$
    - na drugu početni nezavršni znak S

- M ispisuje na drugu traku međuniz α koje generira G.
- ako je $\alpha = \vec{w}$ TS prihvati $\vec{w}$.
- odavde je jasan NEDERMINIZAM TS-a!

KONSTRUKCIJA: TS → GNP

- G generira međunizove konfiguracija M.
- početna konfiguracija predstavlja se međunizom:

$$q_0 [a_1, a_1][a_2, a_2] \dots [a_n, a_n] \ , \quad a_i \in \Sigma$$

nezavršni znak

- za potrebe generiranja dodaje se neki broj m praznih ćelija:

$$q_0 [a_1, a_1][a_2, a_2] \dots [a_n, a_n][\varepsilon, B]^m .$$

- tijekom simulacije nezavršni znakovi:

$$[b, X] \ , \quad b \in \Sigma \cup \{\varepsilon\} \ , \quad X \in \Gamma$$

- konfiguracija $X_1 X_2 \dots X_{r-1} \, q \, X_r \dots X_s$:

$$[a_1, X_1][a_2, X_2] \dots [a_{r-1}, X_{r-1}] \, q \, [a_r, X_r] \dots [a_s, X_s] \dots [a_{n+m}, X_{n+m}]$$

$a_1 \dots a_n \in \Sigma$

$a_{n+1} = a_{n+2} = \dots = a_{n+m} = \varepsilon$

$X_1 \dots X_s \in \Gamma$

$X_{s+1} = X_{s+2} = \dots = X_{n+m} = B$

- ako je $q \in F$ onda zamijeni sve $\underbrace{[a_i, X]}_{\text{nezavršni}}$ s $\underbrace{a_i}_{\text{završni}}$. i $[\varepsilon, X]$ s ε

- ostatak u knjizi na 154. str. (uh!)

SVOJSTVA REKURZIVNO I REKURZIVNO PREBROJIVIH JEZIKA

- rekurzivno prebrojivi zatvoreni s obzirom na: uniju
- rekurzivni jezici zatvoreni s obzirom na: uniju i komplement

- dokaz korištenjem TS
- dokazi u knjizi str. 157.

IZRAČUNLJIVOST

- problem jest izračunljiv ako postoji automat koji mehaničkim postupkom korak po korak rješava zadani problem.

- proces ne treba stati, ali može

Neizračunljivost digitalnog jezika

- ako je moguće izgraditi TS koji prihvaća jezik L, onda je na temelju def. izračunljivosti i Church-Turingove hipoteze jezik L izračunljiv.

ODLUČIVOST

- rekurzivni jezici su odlučivi, jer ih prihvaćaju TS (uvijek stanu).
- rekurzivno prebrojivi jezici nisu odlučivi

Univerzalni jezik L_u je primjer rekurzivno prebrojivog jezika, tj. izračunljivog jezika, koji nije rekurzivan, odnosno nije odlučiv.

KONTEKSTNO OVISNI JEZICI

- pravi podskup klase rekurzivnih jezika
- L_u i L_d nisu u klasi kontekstno ovisnih jezika

KONTEKSTNO OVISNA GRAMATIKA

oblika:

1) $\alpha \to \beta$, $|\beta| \geq |\alpha|$, $\alpha \neq \varepsilon$

2) $\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$, $\beta \neq \varepsilon$

LINEARNO OGRANIČENI AUTOMAT (LOA)

- ograničeni NTS:

1) $¢, \$ \in \Sigma$, lijevi i desni graničnici

2) zabranjuje se pomak glave izvan graničnika

$$LOA = (Q, \Sigma, \Gamma, \delta, q_0, ¢, \$, F)$$

LOA M prihvaća jezik:

$$L(M) = \{ \vec{w} \mid \vec{w} \in (\Sigma \cup \{¢, \$\})^* \text{ i } q_0 ¢ \vec{w} \$ \overset{*}{\succ} \alpha q \beta, q \in F \}$$

- ako je moguće za prihvaćanje jezika L izgraditi deterministički LOA (DLOA), onda se kaže da jezik L deterministički kontekstno ovisan.

KONSTRUKCIJA: KOG → LOA

- KONTEKSTNO OVISNA GRAMATIKA GENERIRA JEZIK $L(G)$, $\varepsilon \notin L(G)$ ONDA JE ZA $L(G)$ MOGUĆE IZGRADITI LOA M:

$$L(M) = L(G)$$

- postupak sličan postupku gradnje TS za jezik zadan gramatikom neograničenih produkcija
- umjesto dvije trake, LOA koristi jednu s dva traga:
  - gornji: $¢ \vec{w} \$$
  - donji: početni nezavršni znak gramatike G.

KONSTRUKCIJA: LOA → KOG

- Ako LOA M prihvaća L onda postoji KOG G:

$$L(G) = L(M) \setminus \{\varepsilon\}$$

- postupak gradnje je sličan za gradnju GNP.
- početna konfiguracija LOA jest $q_0 ¢ a_1 a_2 \ldots a_n \$$:

1) $A_1 \rightarrow [a, q_0 ¢ a] A_2$

$A_1 \rightarrow [a, q_0 ¢ a \$]$

- ostatak u knjizi na 168. str.

SVOJSTVA KONTEKSTNO OVISNIH JEZIKA

- zatvoreni s obzirom na:
  - uniju
  - nadovezivanje
  - kleeneov operator L^+
  - presjek
  - komplement

ODLUČIVOST KOJ

- KOJ = REKURZIVAN

- Algoritam prihvaćanja KOJ izvodi TS koji stane za bilo koji ulazni niz:
  - gradi se usmjeren graf; čvorovi su $\alpha \in (V \cup T)^*$, $|\alpha| \leq |\vec{w}|$.
  - vrijedi li relacija $\alpha \Rightarrow \beta$, čvorovi α i β se povezuju usmjerenom granom
  - put usmjerenog grafa predstavlja postupak generiranja niza
  - put od S do $\vec{w}$ postoji akko G generira $\vec{w}$.
  - TS se uvijek zaustavi i odluči o prihvaćanju niza
    -> KOJ odlučivi.