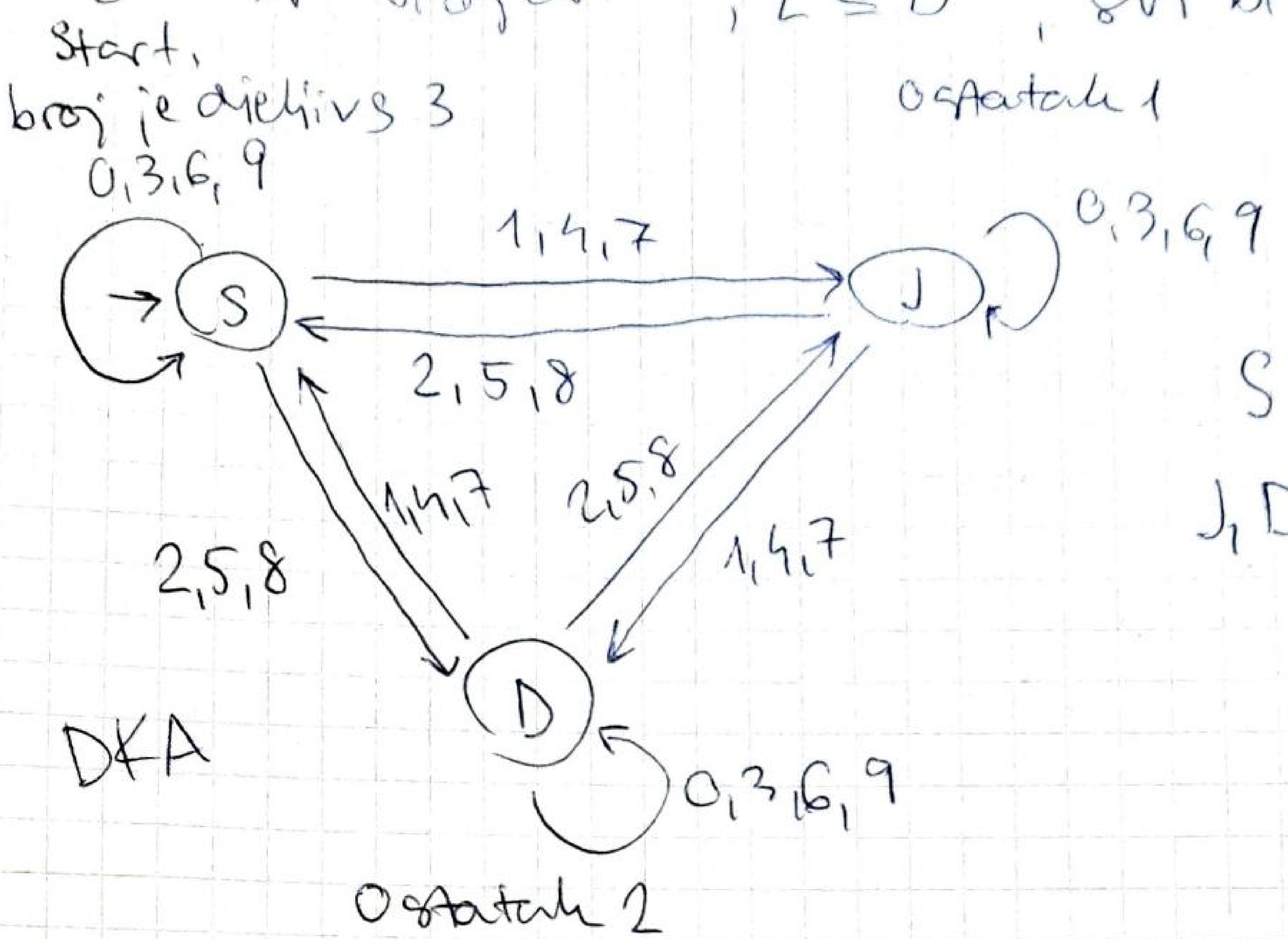
Primjer automata

D^* svi brojevi , $L \subseteq D^*$, svi brojevi djeljivi s 3



S - prihvatljivo stanje

J, D - neprihvatljivo stanje

DKA

	0	1	2	3	4	5	6	7	8	9	
S	S	J	D	S	J	D	S	J	D	S	1
J	J	D	S	J	D	S	J	D	S	J	0
D	D	S	J	D	S	J	D	S	J	D	0

regex definira automat kada traži

gramatika - kreće od 0, mogu se generirati svi nizovi koji zadovoljavaju pravila

$\langle S \rangle \to 0 \langle S \rangle$

$\langle S \rangle \to 3 \langle S \rangle$

$\langle S \rangle \to 6 \langle S \rangle$

$\langle S \rangle \to 9 \langle S \rangle$

$\langle S \rangle \to \varepsilon$

$\langle S \rangle$ - kontrolni znak

0, 3, 6, 9 abeceda

ε - ne nastaje ništa

$\langle S \rangle \to 1 \langle J \rangle | 4 \langle J \rangle | 7 \langle J \rangle$

$\langle J \rangle \to 0 \langle J \rangle | 3 \langle J \rangle | 6 \langle J \rangle | 9 \langle J \rangle$

$\langle J \rangle \to 2 \langle S \rangle | 5 \langle S \rangle | 8 \langle S \rangle$

$\langle J \rangle \to 1 \langle D \rangle | 4 \langle D | 7 \langle D \rangle$

$\langle S \rangle \to 2 \langle D \rangle | 5 \langle D \rangle | 8 \langle D \rangle$

$\langle D \rangle \to 0 \langle D \rangle | 3 \langle D \rangle | 6 \langle D \rangle | 9 \langle D \rangle$

$\langle D \rangle \to 1 \langle S \rangle | 4 \langle S \rangle | 7 \langle S \rangle$

$\langle D \rangle \to 2 \langle J \rangle | 5 \langle J \rangle | 8 \langle J \rangle$

2. REGULARNI JEZICI

- Regularni jezik - može se provjeravati odmornim automatom
  - konačni automat - prihvaća regularne jezike
- Deterministički konačni automat (DKA)
  - ima stanja i prijelaze, prijelazi čitanjem ulaznih znakova
  - konačni - ima konačni broj stanja i prijelaza
  - deterministički - za svako stanje i svaki ulaz ima samo jedan prijelaz
- Upravljačka jedinka - stanje q_0 / S
  - glava za čitanje → ulazna traka (znakovi za čitanje)
- dka = $(Q, \Sigma, \delta, q_0, F)$
  - konačan skup stanja Q
  - konačan skup ulaznih znakova Σ
  - funkcija prijelaza $Q \times \Sigma \to Q$, δ
  - početno stanje $q_0 \in Q$
  - skup prihvatljivih stanja $F \subseteq Q$
- δ(Staro stanje, Ulazni znak) = Novo stanje
- $\hat{\delta} : Q \times \Sigma^* \to$ što bi automat napravio čitanjem niza s > 1 znaka

1) $\hat{\delta}(q, \varepsilon) = q$
2) $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$ - w-niz

DKA prihvaća niz w ako je $\delta(q_0, w) = p$ za neki $p \in F$

DKA prihvaća jezik $L(DKA) = \{ w \mid \delta(q_0, w) \in F \} \subseteq \Sigma^*$

Načini zapisa stanja - izravno (pišemo tablicu stanja)

Tablica [PP, ⊥] = 1 , ⊥ - kraj niza

- posredno (dio koda se izvodi ovisno o stanju)

pretvorba nizova

↳ izvorni niz → pretvorba zapisa → ulazni niz

Minimizacija konačnog automata

Zahtjeva gradnju DKA sa što manjim brojem stanja

Ispitivanje istovjetnosti stanja p i q

↳ (uvjet podudarnosti) moraju oba biti prihvatljiva ili neprihvatljiva

↳ uvjet napredovanja - sljedeća stanja su istovjetna

Algoritam 1 - uspoređujemo sva stanja

Algoritam 2 - u jednoj grupi sva prihvatljiva stanja, u drugu sva neprihvatljiva, njih međusobno uspoređujemo po prihvatljivosti

Algoritam 3 - trokutasta struktura, traži neistovjetna stanja

Nedeterministički konačan automat

↳ jedno stanje i jedan ulaz imaju više mogućih rezultantnih stanja

↳ ako je bilo koji od rezultata prihvatljiv onda prihvaćamo ulaz, ako nijedan nije onda ne prihvaćamo ulaz nka = $(Q, \Sigma, \delta, q_0, F)$

Q - konačan skup stanja

Σ - konačan skup ulaznih znakova

δ - funkcija prijelaza $Q \times \Sigma \to 2^Q$

$q_0 \in Q$ - početno stanje

$F \subseteq Q$ - skup prihvatljivih stanja

δ(stanje, ulazni znak) $= P \subseteq Q$

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

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

2) $\hat{\delta}(q, wa) = P =$

$\{p \mid$ za neko stanje r iz $\hat{\delta}(q, w)$, p jest u $\delta(r, a)\}$

NKA prihvaća niz w ako $\delta(q_0, w)$ sadrži barem jedno stanje iz skupa F

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

DKA $\cong$ NKA (mogu se međusobno prevoditi)

<u>Teorem</u>: Za svaki NKA M postoji DKA M' koji radi istu stvar (prepoznaje isti jezik) $L(M) = L(M')$

<u>Konstrukcija DKA iz zadanog NKA</u>:

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

$Q = \{q_0, q_1, q_2, \dots q_n\}$

$Q' = \{[\emptyset], [q_0], [q_1] \dots [q_j], [q_0, q_1], \dots, [q_{i-1}, q_i], \dots$

$[q_0, q_1, q_2 \dots q_i]\}$

$\delta(q_0, w) = \{p_0, p_1, \dots p_j\}$, $\delta'([q_0], w) = [p_0, p_1 \dots p_j]$

1) $Q' = 2^Q$

2) F' je skup svih stanja $[p_0 \dots p_i]$, gdje je barem jedan $p_k \in F$

3) $q_0' = [q_0]$

4) $\delta'([p_0, p_1 \dots p_i], a) = [r_0, r_1 \dots r_i]$ akko je
$\delta(\{p_0, p_1 \dots p_i\}, a) = \{r_0, r_1 \dots r_i\}$

Istovjetnost DKA i NKA

i) $\delta'([q_0], w) = [r_0, r_1 \dots r_j]$ akko je $\delta(q_0, w) = \{r_0, r_1 \dots r_j\}$

a) $|w| = 0$, $w = \varepsilon$

$\delta'([q_0], \varepsilon) = [q_0]$, $\delta(q_0, \varepsilon) = \{q_0\}$

b) pp da vrijedi za $x \in \Sigma^*$,

dokažemo za $w = xa$, $a \in \Sigma$

$\delta'([p_0, p_1, \dots p_j], a) = [r_0, r_1, \dots r_j]$ akko
$\delta(\{p_0, p_1 \dots p_j\}, a) = \{r_0, r_1 \dots r_j\}$

Nedeterministički konačni automat s ε prijelazima

↳ ε-NKA

↳ ε strelice - može se prošetati bez da se čita ulazna riječ

ε-nka - $(Q, \Sigma, \delta, q_0, F)$

δ - funkcija prijelaza $Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$

ε-okruženje$(P) = \bigcup_{q \in P} \varepsilon$-okruženje$(q)$

1) $\hat{\delta}(q, \varepsilon) = \varepsilon$-okruženje$\{q\}$

2) $\hat{\delta}(q, wa) = \varepsilon$-okruženje$(P)$

$P = \{p \mid$ za neko stanje r iz $\hat{\delta}(q, w)$, p jest u $\delta(r, a)\}$

Konstrukcija NKA iz ε-NKA

1) $Q' = Q$ $\qquad$ ε-NKA $M = (Q, \Sigma, \delta, q_0, F)$

2) $q_0' = q_0$ $\qquad$ NKA $M' = (Q', \Sigma, \delta', q_0', F')$

3) $\delta'(q, a) = \hat{\delta}(q, a), \forall a \in \Sigma, \forall q \in Q$

4) ako ε-okruženje (q_0) sadrži barem jedno stanje skupa F, onda
$F' = F \cup \{q_0\}$, inače, $F' = F$

DKA s izlazom

- radi isto kao i prije, svaki put kada uđe u stanje ispisuje znak, nema prihvatljiva i neprihvatljiva stanja
- ili - svaka strelica ima izlazni znak, koji se ispisuje pri šetanju

- Mooreov automat - izlaz je funkcija stanja
- Mealyjev automat - izlaz je funkcija stanja i ulaznog znaka

Mooreov automat

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

Δ - konačan skup izlaznih stanja

λ - funkcija izlaza $Q \to \Delta$

n ulaza → n+1 izlaza

Mealyjev automat

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

λ funkcija izlaza $Q \times \Sigma \to \Delta$

n ulaza → n izlaza

Konstrukcija Mealyjevog automata iz Mooreovog automata

Za Mooreov automat M vrijedi da je istovjetan Mealyjevom automatu M' ako za svaku riječ $w \in \Sigma^*$ $T_M(w) = bT_{M'}(w)$

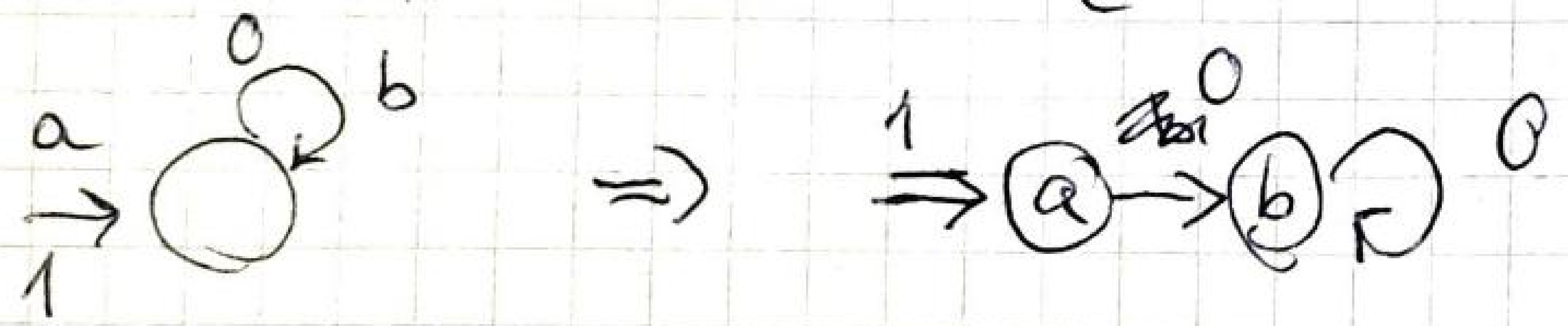
gdje je $b = \lambda(q_0)$

Za svaki Mooreov/Mealyjev automat M postoji istovjetan Mealyjev/Mooreov automat M'

Moore → Mealy
napišemo izlaz stanja u sve strelice koje idu u to stanje

1) $\lambda'(q, a) = \lambda(\delta(q, a))$

Mealy → Moore - radimo nova stanja s različitim izlazima za svako stanje koje ima 2+ strelice sa različitim izlazima



REGULARNI IZRAZI

- algebarski izrazi
  1) $\emptyset$ - jezik $L(\emptyset) = \{\}$
  2) ε - jezik $L(\varepsilon) = \{\varepsilon\}$
  3) a - jezik $L(a) = \{a\}$
  4) $(r)+(s)$ - jezik $L((r)+(s)) = L(r) \cup L(s)$
     $(r)|(s)$
  5) $(r)(s)$ - jezik $L((r)(s)) = L(r)L(s)$
  6) $(r)^*$ - jezik $L((r)^*) = L(r)^*$

Regularni izraz: 1) 01 - $L(01) = \{01\}$
2) $0+1$ - $L(0+1) = \{0, 1\}$
3) $(0+1)(0+1)$ - $L(\dots) = \{00, 01, 10, 11\}$
4) 1^* - $L(1^*) = \{\varepsilon, 1, 11, 111 \dots\}$

Pravila asocijativnosti i prednosti operatora

1) $*$ - unarni operator, lijevo asocijativan, najveće prednosti
2) Operator nadovezivanja - lijevo asocijativan, veće prednosti od $+$
3) $+$ - lijevo asocijativan, najmanje prednosti

$r+s = s+r$ - komutativnost $+$

$r+(s+t) = (r+s)+t$ - asocijativnost $+$

$(rs)t = r(st)$ - asocijativnost nadovezivanja

$r(s+t) = rs+rt$
$(s+t)r = sr+tr$ - distributivnost nadovezivanja nad $+$

$\varepsilon r = r\varepsilon = r$ - ε neutralni element nadovezivanja

$r^* = (r+\varepsilon)^*$ - relacija između $+$ i $*$

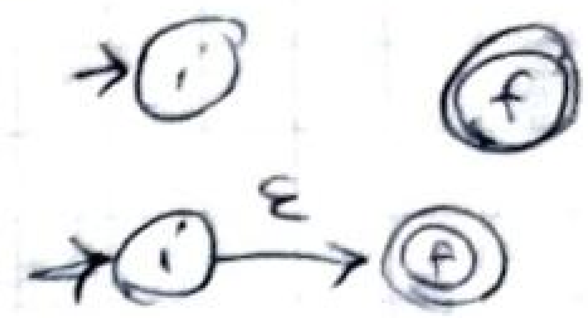
$r^{**} = r^*$ - idempotentnost

regularni izrazi ≅ konačni automati

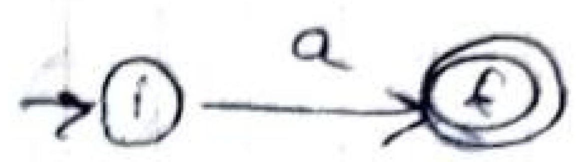
konstrukcija ε-NKA na temelju regex-a

p1) $\emptyset$ - jezik $L(\emptyset) = \{\}$

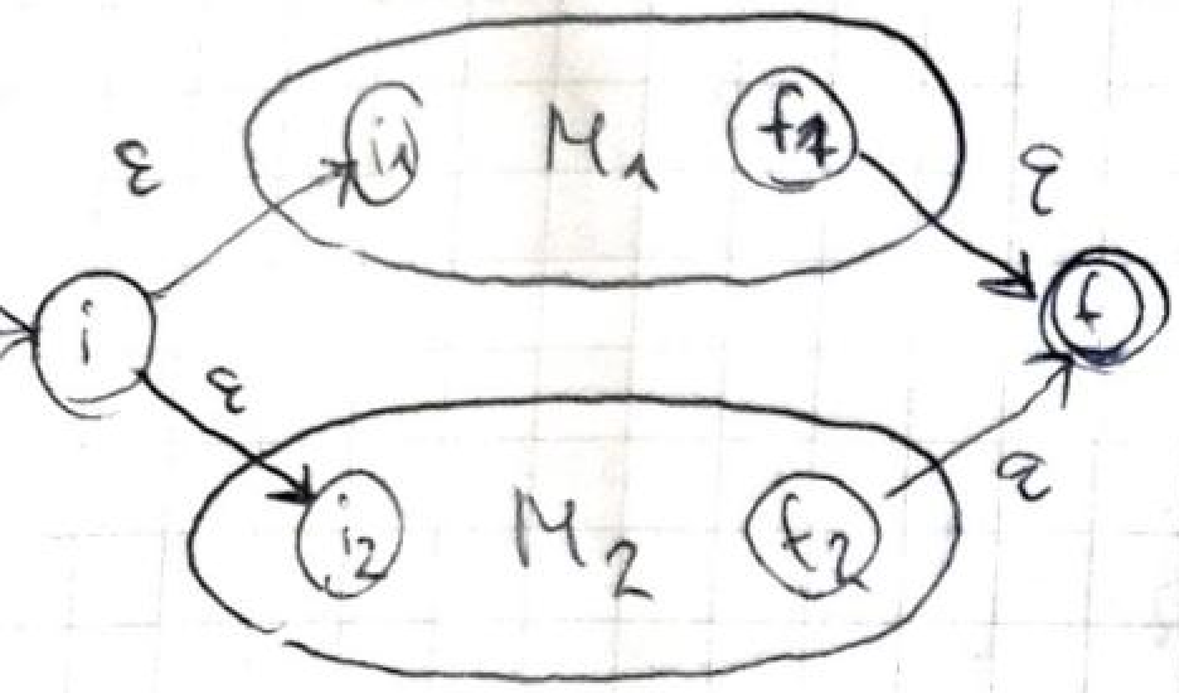
p2) ε - $L(\varepsilon) = \{\varepsilon\}$



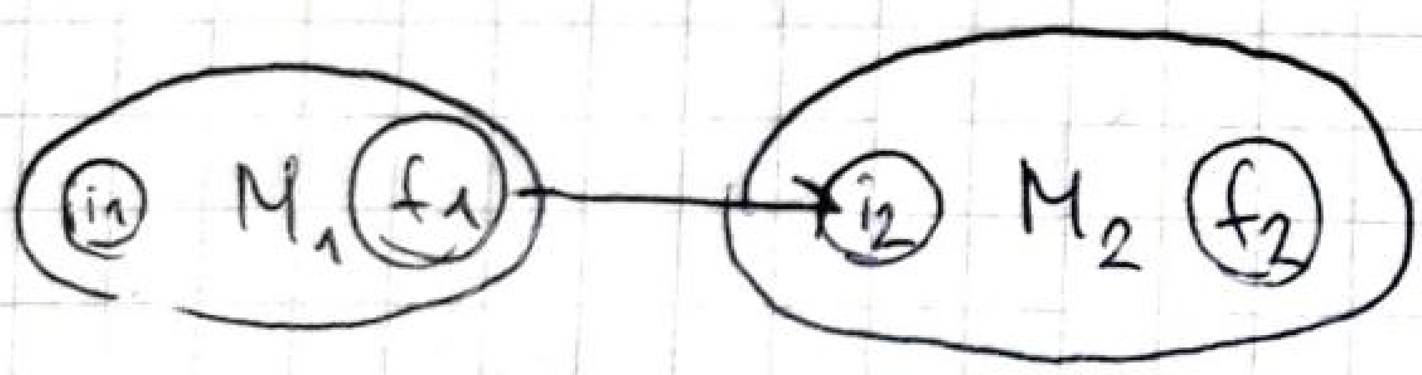
p3) a - $L(a) = \{a\}$



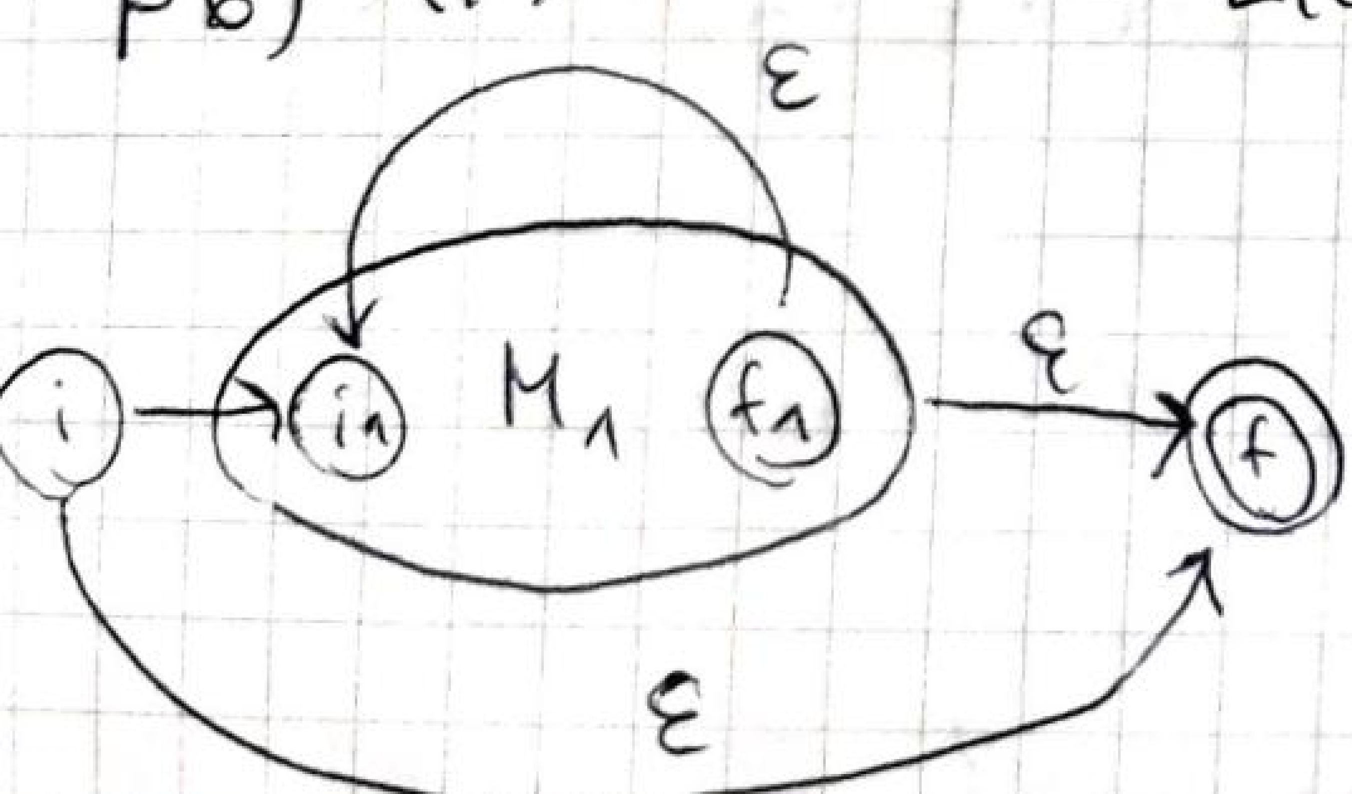
p4) $(r)+(s)$ - $L((r)+(s)) = L(r) \cup L(s)$



p5) $(r)(s)$ - $L((r)(s)) = L(r)L(s)$



p6) $(r)^*$ - $L((r)^*) = L(r)^*$



SVOJSTVA REGULARNIH JEZIKA

Svojstva zatvorenosti regularnih jezika

DKA ≈ NKA ≈ ε-NKA ≈ RI

Jezik L je regularan ako postoji regularni jezik koji ga generira

Neregularni jezici - palindromi, kvadrati $\{a^i b^i\}$...

Klasa jezika je zatvorena s obzirom na operaciju ako primjenom te operacije na bilo koji jezik iz te klase dobijemo jezik koji je u toj istoj klasi

Zatvorena na operaciju: unije, nadovezivanja i Kleeneovog operatora (vrijedi temeljem definicije)

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

↳ presjek (de Morganovo pravilo)
(prihvatljivo stanje presjeka je ono koje je prihvatljivo u oba automata)

$$\left.\begin{array}{l} (\text{DKA } M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1) \\ \text{DKA } M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2) \end{array}\right\} L(M) = L(M_1) \cap L(M_2)$$

1.) $Q = Q_1 \times Q_2$

2) $q_0 = [q_1, p_1]$

3) $F = F_1 \times F_2$

4) $\delta([q, p], a) = [\delta_1(q, a), \delta_2(p, a)]$

↳ supstitucija ($f(R)$ je $\{w_1 \dots w_n$: gdje je $b_1 \dots b_m \in R$ i $w_k \in R_{b_k}\}$)

Regularne definicije i svojstvo napuhavanja

$d_1 \to r_1$ - definiram oznaku d_1 pomoću r_1

$d_2 \to r_2$

... r_i su regularni izrazi nad abecedom $\Sigma \cup \{d_1, d_2, \dots, d_{i-1}\}$

$d_n \to r_n$

Svojstvo napuhavanja - DKA M ima n stanja
↳ prihvaća dovoljni niz $a_1 a_2 \dots a_m$, $m > n$
↳ slijed stanja $q_0 q_1 \dots q_m$ (prihvatljivo)
↳ $0 \leq j < k \leq m$: $q_j = q_k$
↳ podniz $u = a_1 a_2 \dots a_j$,
$v = a_{j+1} a_{j+2} \dots a_k$
$w = a_{k+1} a_{k+2} \dots a_m$
↳ ako pobrišemo v, prihvaća se uw, slijedi da se prihvaća $uv^n w$

Ako je L regularan jezik onda postoji dobrana konstanta n : ako je niz z, $|z| > n$ element jezika L onda je $z = uvw$, $|uv| \leq n$ i $1 \leq |v|$ te za bilo koji $i \geq 0$ niz $uv^i w$ je element jezika L

$N_3 = \{a^n b^n\}$ - dokazujemo da nije regularan
↳ pretpostavljamo da je regularan
↳ neka je $z = a^m b^m$ niz jezika N_3 za koji vrijedi $|z| = 2m$ i $m > n$
↳ niz z rastavlja se na podnizove uvw gdje je $1 \leq |v| \leq |uv| \leq n$
↳ $i = 0$ $|uv| \leq n \Rightarrow$ prefiks uv isključivo znakova a
$\Rightarrow$ niz v isključivo znakova a
$1 \leq |v| \Rightarrow$ niz uw ima manje znakova a od znakova b
↳ niz uw nije element jezika N_3

Algoritmi odlučivanja

↳ nepraznost - Regularni jezik $L(M)$ je neprazan akko DKA M prihvaća niz z duljine manje od n, tj. $|z| < n$ (ako je u skupu dohvatljivih stanja barem jedno prihvatljivo onda je jezik neprazan)

↳ beskonačnost - Regularni jezik $L(M)$ jest beskonačan akko DKA M prihvaća niz duljine l, gdje je $n \leq l < 2n$ (izuzmu se sva beskorisna stanja, ako postoji jedna dohvatljiva zatvorena petlja onda je jezik $L(M)$ beskonačan)

GRAMATIKA

Formalna gramatika

Gramatika su pravila na koje se grade ispravne rečenice.
Elementi gramatike - nezavršni znakovi
riječi iz rječnika - završni znakovi.

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

V - konačan skup nezavršnih znakova
T - konačan skup završnih znakova
$V \cap T = \emptyset$
P - konačan skup produkcija oblika
$A \to \alpha$
A je nezavršni znak
α je niz znakova skupa $(V \cup T)^*$ ili ε
S - početni nezavršni znak $(S \in V)$

$G = (\{E\}, \{a, *, +, (,)\}, \{E \to E+E \mid E*E \mid (E) \mid a\}, E\}$

Generativno stablo za gramatiku $G = (V, T, P, S)$
↳ čvorovi stabla - $V \cup T \cup \{\varepsilon\}$
↳ korijen stabla - S
↳ unutračnji čvorovi - $A \in V$
↳ čvorovi $n_1, n_2, \dots n_k$ - čvorovi djeca čvora n
čvor n - znak A
čvorovi $n_1, n_2, \dots n_k$ - znakovi $X_1, X_2, \dots X_k$
$A \to X_1 X_2 \dots X_k$ - produkcija iz skupa P
↳ znak ε - list stabla, jedino dijete svog roditelja
↳ listovi stabla - znakovima iz skupa $T \cup \{\varepsilon\}$
- čine generirani niz jezika

Regularna gramatika

- $S \to aA \quad S \to bB$

  $G=(V, T, P, S)$, DKA $M=(Q, \Sigma, \delta, q_0, F)$

  - $T = \Sigma$
  - $V = Q$
  - $S = q_0$
  - $A \to aB \qquad \delta(A, a) = B$
  - $A \to \varepsilon \qquad A \in F$

- DKA i gramatika su istovjetni ako DKA prihvaća jezik koji generira gramatika

  $A \overset{*}{\Rightarrow} wB$ akko $\delta(A, w) = B$

  NKA $M=(Q, \Sigma, \delta, q_0, F)$, $G=(V, T, P, S)$

  - $\Sigma = T$
  - $Q = V$
  - $q_0 = S$
  - $\delta(A, a) = \delta(A, a) \cup \{B\} \quad A \to aB$

  $A \to aB, \; A \to aC \Rightarrow \delta(A, a) = \{B, C\}$

- Desno linearna mak

  ↳ produkcije na desnoj strani na krajnjem desnom mjestu imaju najviše jedan nezavršni mak ($A \to wB$, $A \to w$)

  Lijevo linearna gramatika

  ↳ produkcije na desnoj strani na krajnjem lijevom mjestu imaju najviše jedan nezavršni mak ($A \to Bw$, $A \to w$)

Konstrukcija NKA iz desno linearne gramatike

↳ $[\varepsilon] \to \varepsilon$

$A \to w\,[\varepsilon]$ | $A \to w$

↳ $A \to a_1 [a_2 \dots a_n B]$

$[a_2 \dots a_n B] \to a_2 [a_3 \dots a_n B]$ | $a \to a_1 \dots a_n B$

$[a_n B] \to a_n B$

↳ $A \to y$ | $A \to B,\ B \to y$

Konstrukcija ε-NKA iz lijevo linearne gramatike

↳ izgradi se desno linearna gramatika $G' = (V, T, P', S)$

$P' = \{A \to \alpha^R \mid A \to \alpha \text{ je u skupu } P\}$

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

↳ izgradi se NKA M koji prihvaća jezik

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

↳ izgradi se ε-NKA M' koji prihvaća jezik

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

Konstrukcija lijevo linearne gramatike iz NKA

↳ konstruira se ε-NKA M koji prihvaća jezik

$L(M) = L^R$

↳ konstruira se desno linearna gramatika

↳ obrnemo desno linearnu gramatiku

3. KONTEKSTNO NEOVISNI JEZICI

KONTEKSTNO NEOVISNA GRAMATIKA

Nejednoznačnost gramatike, jezika i niza

Kontekstno neovisna gramatika G je nejednoznačna
- ako je za neki niz $w \in L(G)$ moguće izgraditi više različitih generativnih stabala
- ako je neki niz $w \in L(G)$ moguće generirati primjenom više različitih postupaka generiranja niza zamjenom krajnje lijevog nezavršnog znaka
- ako je neki niz .... niza zamjenom krajnje desnog nezavršnog znaka

Niz w je nejednoznačan za gramatiku G ako je moguće za niz $w \in L(G)$ izgraditi više različitih generativnih stabala

Uklanjanje nejednoznačnosti - promjenom gramatike ili promjenom jezika

Pojednostavljenje gramatike

Cilj - bilo koji znak gramatike se koristi u generiranju bar jednog niza

Znak X gramatike $G=(V,T,P,S)$ je koristan

$S \Rightarrow^* \alpha X \beta \Rightarrow^* w$

Znak X gramatike je živ

$X \Rightarrow^* w_X$

Znak X je dohvatljiv

$S \Rightarrow^* \alpha X \beta$

Izbacujemo jedinične produkcije, ε-produkcije

Chomskyjev normalni oblik

$A \to BC$

$A \to a$

Greibachin normalni oblik

$A \to a\alpha$, $a \in T$, $\alpha \in V^*$

odbacivanje mrtvih znakova => odbacivanje nedohvatljivih znakova

Greibachin normalni oblik produkcija

↳ algoritam zamjene krajnje lijevog nezavršnog znaka

↳ $A \to \alpha_1$ $B \to A\gamma \Rightarrow B \to \alpha_i \gamma$

$A \to \alpha_2$

$A \to \alpha_r$

↳ algoritam razrješavanja lijeve rekurzije

↳ $D_i \to D_i \alpha_k$ $\Rightarrow$ $D_i \to \beta_i$ $C_i \to \alpha_k$

$D_i \to \beta_i$ $D_i \to \beta_i C_i$ $C_i \to \alpha_k C_i$

↳ pretvorba : 1. preuredi se u Chomskyev n.o.,

Skup nezavršnih znakova zamijeni se skupom $\{D_1, D_2 \dots D_m\}$

sve produkcije su oblika $D_i \to D_j D_k$

$D_i \to a$

2. $D_i \to D_j D_k$ se preuredi u $D_i \to D_j \beta$, $j > i$

Nezavršni znak D_1, $D_1 \to D_j D_k$

vrijedi $j > 1$ (u zahtijevanom obliku)

$j = 1$ (lijevo rekurzivna, dodajemo C_1)

$D_2 \dots D_m$ u čijem

$D_i \to D_j \beta$, $j < 1$ (zamj. kr. lij. n.z.)

$j = i$ (lijeva rekurzija

$j > 1$ (zahtijevani oblik)

3. nezavršni znakovi $D_{m-1}, \dots D_1$

$D_i \to D_j \beta$ se preurede na oblik

$D_i \to a \alpha \beta$, $a \in T$, α, β nezavršni zn

$D_i \to a \beta$, $C_i \to D_j \varepsilon$

4. preurede se produkcije sa C_i

<u>Parsiranje niza</u>

je li niz $w \in L(G)$

parsiranje niza - prepoznavanje niza i

gradnja generativnog stabla

Parsiranje od vrha prema dnu

↳ gradnja generativnog stabla

- korijen (početni nezavršni znak gramatike)

- prema njegovim listovima (završni znakovi gr.)

Parsiranje od dna prema vrhu

↳ gradnja generativnog stabla

- listovi (završni znakovi gramatike)

- prema njegovom korijenu (početni nez. znak)

Vrh prema dnu

↳ LL(k) gramatika, LL(k) parsiranje

↳ prvi L - ulazni niz se čita s lijeva na desno

↳ drugi L - generiranje niza zamjenom krajnje lijevog nezavršnog znaka

↳ k - odluka o primjeni produkcije donosi se na temelju k pročitanih znakova

↳ LL(1) - sintaksna analiza prog. jezika

↳ tehnika rekurzivnog spusta (programski jezik sa svojstvom rekurzivnog poziva potp. ispituje ako podniz zadanog niza zadovoljava strukturu)

od dna prema vrhu
↳ u dobivenim međunizovima završnih i nezavršnih znakova se nastoji prepoznati jedna od desnih strana produkcije
↳ dio međuniza zamijeni se lijevom stranom produkcije
↳ produkcije - redukcije
↳ LR(K) gramatika, LR(k) parsiranje
↳ L - ulazni niz se čita slijeva na desno
↳ R - (obrnuti postupak generiranja niza zamjenom krajnje desnog nezavršnog znaka
↳ K - broj znakova unaprijed za potrebe donošenja odluke o primjeni redukcije
↳ LR(0), LR(1) - sintaksna analiza prog. jez
LR parser - složeniji od rekurzivnog spusta, gradi se posebnim generatorom, tablica na temelju zadanih produkcija kontekstno neovisne gramatike
- različiti parseri - različite tabl. parsiranja

Program LR Parsera
- tablica parsiranja (Akcija, NovoStanje)
- ulaz
- stog
- izlaz

POTISNI AUTOMAT 1

Model potisnog automata

- koristi LIFO stog
- odluka se donosi ovisno o stanju upravljačke jedinke, pročitanom znaku na ulaznoj traci i pročitanom znaku na vrhu stoga
- izlaz - novo stanje, pomak glave u desno, upis niza znakova na vrh stoga

vrste prijelaza

↳ q - stanje upravljačke jedinke
  a - znak ulazne trake
  Z - znak na vrhu stoga

↳ $(q, a, Z) = (p, \gamma)$ - mijenja stanje u p, pomiče glavu za čitanje, mijenja Z sa γ

↳ $(q, \varepsilon, Z) = (p, \gamma)$ - ne miče se glava za čitanja

Odluka o prihvaćanju niza

↳ PA M prihvaća prihvatljivim stanjem
  - uđe u prihvatljivo stanje kada se pročitaju svi ulazni znakovi
  - prihvaća L(M)

↳ PA M koji prihvaća praznim stogom
  - prihvaća ako se isprazni stog čitanjem svih znakova ulaznog niza
  - prihvaća N(M)

Definicija potisnog automata

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

Q - konačan skup stanja

Σ - konačan skup ulaznih znakova

Γ - konačan skup znakova stoga

δ - funkcija prijelaza, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to Q \times \Gamma^*$

$q_0 \in Q$ - početno stanje

$Z_0 \in \Gamma$ - početni znak stoga

$F \subseteq Q$ - skup prihvatljivih stanja

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

$q \in Q$, $a \in \Sigma$, $Z \in \Gamma$, $p_n \in Q$, $\gamma \in \Gamma^*$

prihvaćanje prihvatljivim stanjem za jezik:

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

prihvaćanje praznim stogom za jezik:

↳ $N(M) = \{w \mid (q_0, w, Z_0) \overset{*}{\underset{M}{\succ}} (p, \varepsilon, \varepsilon)$ za neko stanje $p \in Q\}$

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

↳ ako je $(\delta(q, \varepsilon, Z) \neq \emptyset)$ onda $\delta(q, a, Z) = \emptyset, \forall a \in \Sigma$

↳ u skupu $\delta(q, a, Z)$ je najviše jedan element

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

Konstrukcija PA prazni stog iz PA prihvatljivo stanje

PA $M1 = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', \emptyset)$

PA $M2 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

1. Skup $\delta'(q, a, Z) =$ skup $\delta(q, a, Z)$
2. $q \in F$, $Q' = Q \cup \{q_e\}$

   $\delta'(q, \varepsilon, Z) = \delta'(q, \varepsilon, Z) \cup (q_e, \varepsilon)$

   $(q_e, \varepsilon) \in \delta'(q_e, \varepsilon, Z)$
3. $Z_0' = X_0$, $\Gamma' = \Gamma \cup \{X_0\}$, $Q' = Q' \cup \{q_0'\}$

   $\delta'(q_0', \varepsilon, X_0) = \{(q_0, Z_0 X_0)\}$

Konstrukcija PA prihvatljivim stanjem iz PA prazni stog

PA $M_2 = (Q', \Sigma, \Gamma', \delta', q_0', z_0', F')$

PA $M_1 = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \emptyset)$

1. skup $\delta'(q, a, z)$ = skup $\delta(q, a, z)$
2. $Q' = Q' \cup \{q_f\}$

   $F' = \{q_f\}$

   $(q_f, \varepsilon) \in \delta'(q, \varepsilon, X_0)$
3. $z_0' = X_0$, $\Gamma' = \Gamma \cup \{X_0\}$, $Q' = Q \cup \{q_0'\}$,

   $\delta'(q_0', \varepsilon, X_0) = \{(q_0, z_0 X_0)\}$

Potisni automat i kontekstno neovisna gramatika

PA prazni stog iz kontekstno neovisne gramatike

$G = (V, T, P, S)$, Greibachin normalni oblik

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

a) PA M ima samo jedno stanje q koje je ujedno i početno stanje

b) $\Sigma = T$

c) $\Gamma = V$

d) početni znak stoga $= S$

e) $F = \emptyset$

f) PA M prihvaća praznim stogom

funkcije prijelaza

$(q, \gamma) \in \delta(q, a, A)$ akko $A \to a\gamma$

$(q, x, S) \overset{*}{\underset{M}{\vdash}} (q, \varepsilon, \alpha)$, $S \overset{*}{\underset{G}{\Rightarrow}} x\alpha$

Kontekstno neovisna gramatika iz PA prazni stog

PA $M = (Q, \Sigma, T, \delta, q_0, z_0, F)$

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

$[q, A, p] = V$, $q \in Q$, $p \in Q$, $A \in T$

↳ prelazi iz q u p i briše A

SVOJSTVA KONTEKSNO NEOVISNIH JEZIKA

Jezici koji nisu kontekstno neovisni

↳ ne mogu se generirati kontekstno neovisnom gramatikom

↳ npr $L = \{a^i b^i c^i \mid i \geq 1\}$

Svojstva zatvorenosti konteksno neovisnih jezika:

Unija

↳ $G_1 = (V_1, T_1, P_1, S_1)$, $G_2 = (V_2, T_2, P_2, S_2)$,

$V_1 \cap V_2 = \emptyset$

$L(G_3) = L(G_1) \cup L(G_2)$

↳ konstrukcija gramatike G_3

$V_3 = V_1 \cup V_2 \cup \{S_3\}$, $S_3 \notin V_1$, $S_3 \notin V_2$

$T_3 = T_1 \cup T_2$

$P_3 = P_1 \cup P_2 \cup \{S_3 \to S_1 \mid S_2\}$

Nadovezivanje

↳ $G_1 = (V_1, T_1, P_1, S_1)$, $G_2 = (V_2, T_2, P_2, S_2)$, $V_1 \cap V_2 = \emptyset$

$L(G_4) = L(G_1) L(G_2)$, $G_4 = (V_4, T_4, P_4, S_4)$

↳ konstrukcija gramatike G_4

$V_4 = V_1 \cup V_2 \cup \{S_4\}$, $S_4 \notin V_1$, $S_4 \notin V_2$

$T_4 = T_1 \cup T_2$

$P_4 = P_1 \cup P_2 \cup \{S_4 \to S_1 S_2\}$

Kleeneov operator L^*

↳ $G_1 = (V_1, T_1, P_1, S_1)$

$L(G_5) = L(G_1)^*$, $G_5 = (V_5, T_5, P_5, S_5)$

↳ konstrukcija

$V_5 = V_1 \cup \{S_5\}$, $S_5 \notin V_1$

$T_5 = T_1$

$P_5 = P_1 \cup \{S_5 \to S_1 S_5 \mid \varepsilon\}$

Supstitucija

↳ $G = (V, T, P, S)$, $a_i \in L(G)$ zamijene se nizovima iz $L(G_i)$

$G_i = (V_i, T_i, P_i, S_i)$

$1 \leq i \leq k$, k - kardinalni broj skupa T

↳ $G' = (V', T', P', S')$

$V' = V \cup V_1 \cup V_2 \dots V_k$, $V \cap V_i = \emptyset$, $V_i \cap V_j = \emptyset$ $\quad 1 \leq i \leq k$ $\quad 1 \leq j \leq k$, $i \neq j$

$T' = T_1 \cup T_2 \dots T_k$

$S' = S$

Skup produkcija $P' = P_1 \cup P_2 \dots P_k$ dodaju se preuređene produkcije gramatike G:

a_i zamijenimo s S_i

Nisu zatvoreni na presjek i komplement.

Presjek kontekstno neovisnog jezika i regularnog jezika

↳ PA $M_1 = (Q_1, \Sigma, T, \delta_1, q_0, Z_1, F_1)$

DKA $M_2 = (Q_2, \Sigma, \delta_2, p_0, F_2)$

$L = L_1 \cap L_2 \Rightarrow$ PA $M' = (Q', \Sigma, T, \delta', q_0', Z_0, F')$

$Q' = Q_2 \times Q_1$, $q_0' = [p_0, q_0]$, $F' = F_2 \times F_1$,

$([p', q'], \gamma) \in \delta'([p, q], a, X)$

akko $\;-\; \delta_2(p, a) = p'$ i $-(q', \gamma) \in \delta_1(q, a, X)$

ako $a = \varepsilon$ onda je $p' = p$

Svojstvo napuhavanja

- interpretiramo pomoću generativnog stabla
- broj stanja limitiran
- broj unutrašnjih čvorova stabla > broj nezavršnih znakova
- s lijeve i desne strane stabla podnizovi koji se ponavljaju
- uv^iwx^iy - v i x se ponavljaju (gomilaju)
- za L postoji cjelobrojna konstanta n ako je niz z, $|z| > n$, $z = uvwxy$, $|vwx| \leq n$ i $1 \leq |vx|$ element jezika L, za bilo koji $i \geq 0$ niz uv^iwx^iy je element jezika L

7. REKURZIVNO PREBROJIVI JEZICI

- ## TURINGOV STROJ (TS)

Rekurzivno prebrojiv jezik
↳ Turingov stroj
↳ najopćenitiji matematički model računanja

Osnovni model Turingovog stroja
- proširenje konačnog automata
- tri proširenja - beskonačna ulazna traka na desnu stranu
 - staro stanje + znak na ulaznoj
- traci → novo stanje + upis znaka na traku + odluka
o pomaku glave lijevo ili desno

$TS = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$
Q - konačan skup stanja
Γ - konačan skup znakova trake
$B \in \Gamma$ - znak kojim se označava prazna ćelija
$\Sigma \subseteq (\Gamma - \{B\})$ - konačan skup ulaznih znakova
δ - funkcija prijelaza, $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$
- $q_0 \in Q$ - početno stanje
$F \subseteq Q$ - skup prihvatljivih stanja
$\delta(q, V) = (p, Z, W)$

Rekurzivno prebrojiv jezik
↳ prebrojivi, moguće izgraditi TS koji ispisuje sve nizove jezika
↳ niz $w \in L(M)$, TS M stane i prihvati niz w
↳ niz $w \notin L(M)$, moguće da TS nikada ne stane

- Rekurzivni jezici
↳ moguće izgraditi TS koji uvijek stane za bilo koji prebrojiv niz

Računanje cjelobrojnih funkcija TS
↳ cijeli broj $i \geq 0$ - zapiše se nizom 0^i
↳ cjelobrojna fja k argumenata $i_1, i_2 \dots i_k$
- zapiše se nizom $0^{i_1} 1 0^{i_2} 1 \dots 1 0^{i_k}$
↳ TS se zaustavi i na traci je zapisano 0^m
- vrijednost fje $f(i_1, i_2 \dots i_k) = m$
↳ parcijalno rek. fje - TS ne mora stati
↳ potpuno rekurzivna fj - TS uvijek stane

Metode izrade Turingovog stroja

· Višekomponentni znakovi trake
↳ $[a_1, a_2 \dots a_n]$,
↳ zasebni tragovi ulazne trake

A_1	A_2	A_3
a_1	a_1	a_1
a_2	a_2	a_2
...		

Višekomponentne oznake stanja
↳ $[q_1, q_2 \dots q_n]$
↳ upravljačke komponente (upravljaju radom)
↳ radne komponente (spremaju vrijednosti podataka

Prošireni modeli Turingovog stroja

TS s dvostranom beskonačnom trakom
↳ beskonačna s lijeve i desne strane
↳ TS $M_2 = (Q_2, \Sigma_2, \Gamma_2, \delta_2, q_2, B, F_2)$
TS s višestrukim trakama
↳ veći broj traka, za svaku ima glavu za čitanje i pisanje
↳ staro stanje + k pročitanih znakova sa k traka →
novo stanje + upis k znakova na k traka

Nedeterministički TS
↳ $\delta(q, X) = \{(p_1, Z_1, D_1), (p_2, Z_2, D_2), \dots (p_k, Z_k, D_k)\}$
↳ stablasta struktura pretraženja
TS s višedimenzionalnim ulaznim poljem
↳ traka beskonačna u svim smjerovima
↳ horizontalni i vertikalni pomak unutar pravokutnika

<u>Pojednostavljeni modeli Turingovog stroja</u>

Stogovni stroj
↳ dva stoga simulira Turingov stroj od 1 trake.
↳ 1. stog elementi desno od glave za čitanje
2. stog elementi lijevo od glave za čitanje
↳ bliži elementi na vrhu stoga

Stroj s brojilima
↳ modelira se s 4 brojila
↳ dno stoga X i prazna ćelija B
↳ pomicanjem glave u desno povećavamo vrijednost brojila
↳ stroj s 4 brojila se može pojednostaviti na 2

TS s ograničenim brojem stanja i znakova trake
↳ jedna traka, tri stanja za prihvaćanje svakog rekurzivno prebrojivog jezika (ograničeni broj stanja).
↳ neograničeni broj stanja: znakovi trake $\{0, 1, B\}$,
$2^{k-1} + 1 < $ kardinalni broj skupa znakova trake $< 2^k$
znakovi trake kodiraju se binarnim nizovima duljine k
u k ćelija je k-bitovni binarni kod
1. komp stanje TS M_1
2. komp - pomicanje glave na poč. bin-koda

omaka prazne ćelije B
Univerzalni Turingov stroj Mu
↳ tri trake
1. traka (F-je prijelaza TS M) (niz w)
2. traka (radna traka sadržaj TS M) (poč. w)
3. traka (stanje TS M) (poč. q_0)
↳ pojednostavljen oblik
- jedna traka, pet stanja, sedam znakova trake
<u>Generiranje jezika Turingovim strojem</u>
- 2 radne trake i izlazna traka (generiranje)
- ako TS stane jezik je konačan, ako ne stane je beskonačan
- Prihvaćanje jezika generiranog TS
↳ ulazna traka, radna traka, izlazna traka
↳ uspoređuje ulaznu i izlaznu traku, generira na izlaznu traku dok se ne ispiše ulazni niz
Generiranje jezika koji se prihvaća TS
↳ 2 radne i 1 izlazna traka
↳ na izlaznu traku se ispisuju vrijednosti iz radne trake koje odgovaraju jeziku
↳ jedino za rekurzivne jezike
↳ Rekurzivno prebrojivi jezici
↳ 4 trake, 3. traka prikazuje broj koraka u kojem stroj mora završiti simulaciju
(a, b) → broj koraka za provjeru
↳ element s radne trake
- rekurzivne jezike je moguće generirati kanonskim slijedom

GRAMATIKA NEOGRANIČENIH PRODUKCIJA !

- Produkcije kontekstno neovisne gramatike
  - ↳ na lijevoj strani samo 1 nezavršni znak
  - ↳ $A \to \alpha$

Produkcije regularne gramatike
- ↳ lijevo linearne ili desno linearne
- ↳ $A \to wB$ $\quad A \to Bw$
  $A \to w$ $\quad A \to w$

Produkcije gramatike neograničenih produkcija
- ↳ $\alpha \to \beta$, $\alpha \neq \varepsilon$, $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$

Konstrukcija TS za jezik zadan GNP

- nedeterministički TS s dvije trake
  - ↳ prva traka niz koji provjeravamo
  - ↳ 2. traka početni nezavršni znak, generiraju se svi nizovi gramatike

Konstrukcija gramatike za jezik zadan TS

Početna konfiguracija $q_0 a_1 a_2 \dots a_n$

Međuniz gramatike $\quad q_0 [a_1, a_1][a_2, a_2], \dots [a_n, a_n]$

↳ prva komp znak koji se treba generirati, druga čuva znakove trake koji se koriste za vrijeme simulacija

$A_1 \to q_0 A_2$,

$A_2 \to [a, a] A_2$

↳ također generiramo prazne ćelije $[\varepsilon, B]^m$

$A_2 \to A_3$

$A_3 \to [\varepsilon, B] A_3$

$A_3 \to \varepsilon$

↳ Prijelaz TS $\delta(q, X) = (p, Y, R)$ - pomak glave udesno

$q[a, X] \to [a, Y]p$

↳ Prijelaz TS $\delta(q, X) = (p, Y, L)$ - pomak glave ulijevo

$[b, Z]q[a, X] \to p[b, Z][a, Y]$

↳ za sva stanja q u skupu prihvatljivih stanja F

za sve malepe $a \in \Sigma \cup \{\varepsilon\}$, i $X \in \Gamma$

$[a, X]q \to qaq$

$q[a, X] \to qaq$

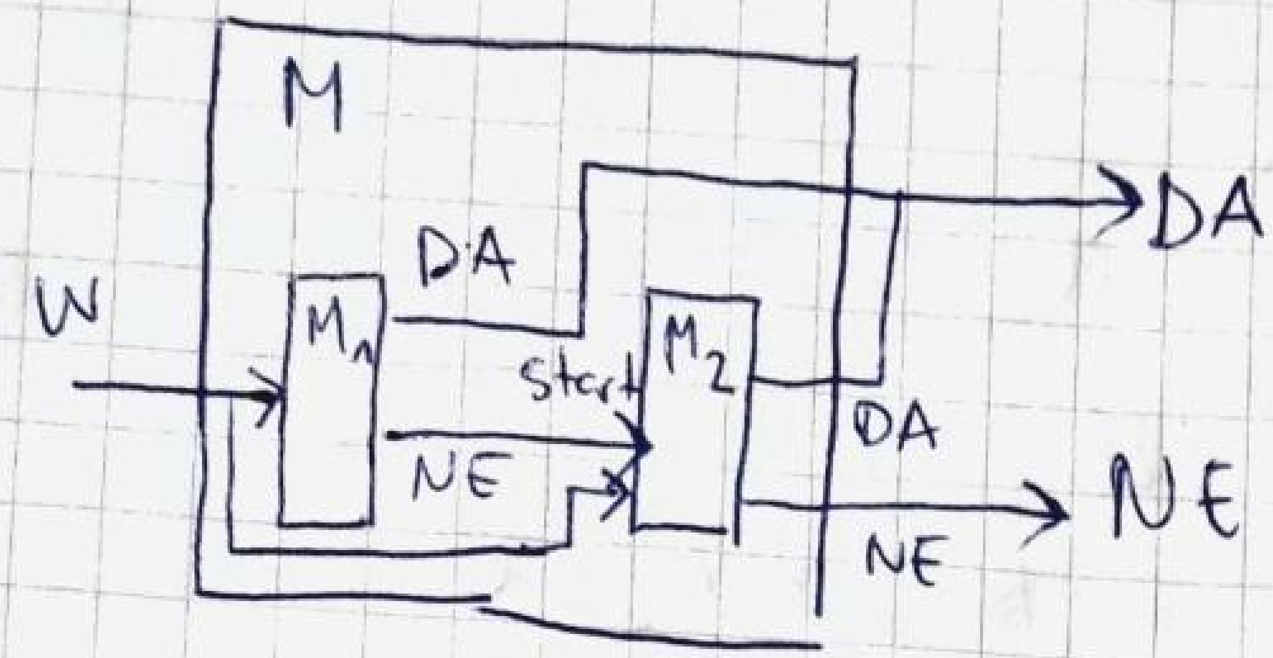
$q \to \varepsilon$

- eliminirano malove koje smo koristili na traci

SVOJSTVA REK. I REK. PREBROJIVIH JEZIKA

Svojstva zatvorenosti rek. i rek. pr. jezika

Unija



Složeni TS od M_1 i M_2
UNIJA RJ je RJ

Unija RP TS daje RPTS

Komplement

Ako su M_1 i $\overline{M_1}$ RP, onda oba moraju biti rekurzivni

Izračunljivost

- postoji automat koji korak po korak rješava problem
- nema ograničenja na broj koraka, veličinu spremnika i ne zahtijeva se da proces računanja ikad stane

Church-Turingova hipoteza $N^k \to N$

↳ izračunljive funkcije - parcijalno rekurzivne funkcije
↳ za njihovo računanje moguće je izgraditi TS
↳ TS ih računa korak po korak primjenom zadanih f-ja prijelaza
↳ TS ne mora nužno stati za sve ulazne nizove

Neizračunljivost dijagonalnog jezika

↳ kodiramo TS M (kod. f-ja prijelaza (makova))

$$\delta(q_i, X_j) = (q_k, X_l, D_m)$$

se kodira:

$$0^i 1 0^j 1 0^k 1 0^l 1 0^m$$

Gradnja dijagonalnog jezika

	1	2	-	x	x+1	x+2	x+3	-
1	0	0	-	0	0	0	0	-
2	0	0	-	0	0	0	0	-
:	-	-	-	-	-	-	-	-
x	0	0	-	0	0	0	1	-
x+1	0	0	-	1	1	1	0	-
x+2	0	0	-	1	0	0	0	-
x+3	0	0	-	0	0	0	1	-
-	-	-	-	-	-	-	-	-

TS M_d koji prihvaća $L_d = L(M_d)$

$(d,d) = 0 \Rightarrow w_d \in L_d$
ali bi trebalo biti $(d,d) = 1$
$(d,d) = 1 \Rightarrow w_d \notin L_d$
ali bi trebalo biti $(d,d) = 0$

Odlučivost

TS umijeh stane i donese odluku o prihvaćanju ili neprihvaćanju niza
rekurzivni jezici su odlučivi i izračunljivi
RP jezici nisu odlučivi ali jesu izračunljivi
Univerzalni TS M_u prihvaća univerzalni jezik L_u
$L_u = \{\langle M, w\rangle \mid \text{TS } M \text{ prihvaća niz } w\}$
univerzalni jezik L_u
↳ rekurzivno prebrojiv - izračunljiv
↳ prihvaća ga univerzalni TS M_u
↳ nije rekurzivan - nije odlučiv

KONTEKSTNO OVISNI JEZICI

- podskup rekurzivnih jezika
  ↳ linearno-ograničeni automat
- ograničena veličina radne trake

KONTEKSTNO OVISNA GRAMATIKA

$\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$
↳ ako je A unutar pravog konteksta, mijenja se u β

$\alpha \to \beta$, $|\alpha| \leq |\beta|$, $\alpha \neq \varepsilon$, $\beta \neq \varepsilon$

- generira nizove rastuće u broju znakova

$L = \{ a^n b^n c^n \mid n \geq 1 \}$ - primjer jezika

LINEARNO OGRANIČEN AUTOMAT

- prihvaća kontekstno ovisne jezike
- TS s restrikcijama, ograničena s lijevim i desnim graničnicima, na njih može stati ali ne može pisati

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

Q - konačan skup stanja

Γ - konačan skup znakova trake

$\Sigma \subseteq \Gamma$ - konačan skup ulaznih znakova

δ - funkcija prijelaza

$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

$q_0 \in Q$ - početno stanje

$F \subseteq Q$ - skup prihvatljivih stanja

$¢, \$$ - lijevi i desni graničnik

$L(M) = \{ w \mid w \in (\Sigma \setminus \{¢, \$\})^*$
$\quad ; q_0 ¢ w \$ \overset{*}{\vdash} \alpha q \beta, \ q \in F \}$

- po definiciji nedeterministički

Konstrukcija LOA za jezik zadan KOG

- slično kao TS
- umjesto 2 trake se stavljaju 2 traga, gore ulazni niz, dolje rezultati
- traka duljine ulaznog niza
- generirani niz kraći od ulazne trake, ako ispadne dulji znači da se neće prihvatiti

Konstrukcija KOG za jezik zadan LOA

početna konfiguracija LOA: $q_0 ¢ a_1 a_2 \dots a_n \$$

međuniz gramatike $[a_1, q_0 ¢ a_1][a_2, a_2] \dots [a_n, a_n \$]$

$A_1 \to [a, q_0 ¢ a \$]$

$A_1 \to [a, q_0 ¢ a] A_2$

$A_2 \to [a, a] A_2$

$A_2 \to [a, a \$]$

prijelaz $\delta(q, X) = (p, Y, R)$

produkcije gramatike -desni pomaci

$[b, qX][a, Z] \to [b, Y][a, pZ]$

$[b, ¢ qX][a, Z] \to [b, ¢ Y][a, pZ]$

$[b, qX][a, Z \$] \to [b, Y][a, pZ \$]$

$[b, qX \$] \to [b, Y p \$]$

$[a, ¢ q X \$] \to [a, ¢ Y p \$]$

za lijeve pomake situacija simetrična

$[b, Z][a, qX] \to [b, pZ][a, Y]$

...

glava pokazuje na graničnike

$[b, q¢X] \rightarrow [b, ¢pX]$

$[b, Xq\$] \rightarrow [b, pX\$]$

$[a, q¢X\$] \rightarrow [a, ¢pX\$]$

$[a, ¢Xq\$] \rightarrow [a, ¢pX\$]$

Za sva stanja $q \in F$, za sve znakove $a \in \Sigma \setminus \{¢, \$\}$

$[a, \alpha q \beta] \rightarrow a$

$[a, \alpha] b \rightarrow ab$

$b [a, \alpha] \rightarrow ba$

SVOJSTVA KONTEKSTNO OVISNIH JEZIKA

Unija

↳ $G_1 = (V_1, T_1, P_1, S_1)$, $G_2 = (V_2, T_2, P_2, S_2)$

↳ $L(G_3) = L(G_1) \cup L(G_2)$, $V_1 \cap V_2 = \emptyset$

↳ $V_3 = V_1 \cup V_2 \cup \{S_3\}$, $S_3 \notin V_1$, $S_3 \notin V_2$

$T_3 = T_1 \cup T_2$

$P_3 = P_1 \cup P_2 \cup \{S_3 \to S_1 \mid S_2\}$

Nadovezivanje

↳ $L(G_4) = L(G_1) L(G_2)$

$V_1 \cap V_2 = \emptyset$

↳ $V_4 = V_1 \cup V_2 \cup \{S_4\}$, $S_4 \notin V_1$, $S_4 \notin V_2$

$T_4 = T_1 \cup T_2$

$P_4 = P_1 \cup P_2 \cup \{S_4 \to S_1 S_2\}$

$a \in T$, $A_a \in V'$, $A_a \to a$

$\alpha' \to \beta'$

Kleeneov operator L^+

↳ $G = (V, T, P, S)$

↳ na lijevim stranama produkcije isključivo nezavršni znakovi gramatike

$L(G_5) = L(G)^+$

↳ gradi se pomoćna gramatika $G' = (V', T, P', S')$ $V \cap V' = \emptyset$

G i G' se koriste za izgradnju G_5

$V_5 = V \cup V' \cup \{S_5, S_5'\}$, $S_5', S_5 \notin V$, $S_5', S_5 \notin V'$

$T_5 = T$

$S_5 \to S S_5' \mid S$

$S_5' \to S' S_5 \mid S'$ (SS'SS...) nema interferencije

Presjek
$L(M_3) = L(M_1) \cap L(M_2)$
↳ dvije trake, prva simulira M1, druga M_2
ako prva traka prihvati niz radimo drugu
traku, tada prihvaća niz
Komplement DKO jezika
↳ s - kardinalni broj skupa stanja Q
n - duljina niza w
t - kardinalni broj skupa znakova trake T
Maksimalan broj različitih konfiguracija DLOA
$s(n+2)t^n$
prijelaz DLOA iz konfiguracije u konfiguraciju
$K_x \rightarrow K_y$
LOA $\overset{?}{=}$ DLOA (ne zna se ako su istovjetni

<u>Odlučivost kontekstno ovisnih jezika</u>
Iterativni algoritam pronalaženja puta u
zadanom grafu
↳ gradi listu K koja sadrži međunizove generirane KOG
$|\alpha| \leq |w|$
↳ K_i - vrijednost liste nakon i-tog koraka iteracije
K_i - međunizovi α, $|\alpha| \leq |w|$, gramatika
generira α primjenom najviše i produkcija
Početni korak iteracije
↳ u listu K se upiše početni nezavršni znak S
i-ti korak iteracije
↳ $K_i = K_{i-1} \cup \{ \beta \mid \alpha \Rightarrow \beta, \alpha \in K_{i-1} \text{ i } |\beta| \leq |w| \}$

Primjer RJ koji nije KOJ

L_R - $w \in L_R$ akko TS M_i ne prihvaća niz w i je cjelobrojna vrijednost binarnog broja w

L_R - ne prihvaća ga niti jedan TS M_i u kojem M_1, M_2, M_3, $L_R \in RJ$, $L_R \notin K \Rightarrow K \subset RJ$

↳ pretpostavi se da TS M_j prihvaća jezik L_R

↳ binarni niz x cjelobrojne vrijednosti j

↳ suprotnost: ako je $x \in L_R \Rightarrow x \notin L(M_j)$

$x \notin L_R \Rightarrow x \in L(M_j)$

- jezik L_R je rekurzivni jezik $L_R \neq L(M_j)$ ...

Kodiranje elemenata gramatike bin. m 1 i 0

RAZREDBA JEZIKA, AUTOMATA I GRAMATIKA

STRUKTURNA SLOŽENOST JEZIKA

Chomsky-jeva hijerarhija jezika

Skup svih jezika nad abecedom: 2^{Σ^*}

(Dijagonalni jezik $L_d \in 2^{\Sigma^*}$ i $L_d \notin RPJ$)

⇓

Rekurzivno prebrojivi jezici: RPJ

(Univerzalni jezik $L_u \in RPJ$ i $L_u \notin RJ$

⇓

Rekurzivni jezici: RJ

(Jezik $L_R \in RJ$ i $L_R \notin KOJ$)

⇓

Kontekstno ovisni jezici: KOJ

(Jezik $L_1 : \{ww \mid w \in (0+1)^*$ i $|w| \geq 1\}$

$L_1 \in KOJ$ i $L_1 \notin NKNJ$

⇓

Nedeterministički kontekstno neovisni jezici: NKNJ

(Jezik $L_2 : \{ww^R \mid w \in (0+1)^*$ i $|w| \geq 1\}$

$L_2 \in NKNJ$ i $L_2 \notin DKNJ$

⇓

Deterministički kontekstno neovisni jezici: DKNJ

(Jezik $L_3 : \{w2w^R \mid w \in (0+1)^*$ i $|w| \geq 1\}$

$L_3 \in DKNJ$ i $L_3 \notin REG$

⇓

Regularni jezici: REG

Hijerarhija gramatika i automata

Regularni jezici: $L_3 = L(G_3) = L(M_3)$
$\Uparrow$
Kontekstno neovisni jezici: $L_2 = L(G_2) = L(M_2)$
$\Uparrow$
Kontekstno-ovisni jezici: $L_1 = L(G_1) = L(M_1)$
$\Uparrow$
Rekurzivno prebrojivi jezici: $L_0 = L(G_0) = L(M_0)$

Regularna gramatika $G3$: $A \to wB$ i $A \to w$
$\Uparrow$ $A \to Bw$ i $A \to w$

Kontekstno neovisna gramatika G_2: $A \to \alpha$
$\Uparrow$
Kontekstno ovisna gramatika G_1 $\alpha \to \beta$, $|\alpha| \leq |\beta|$
$\Uparrow$
Gramatika neograničenih produkcija G_0: $\alpha \to \beta$

Konačni automat: $M_3 = (Q, \Sigma, \delta, q_0, F)$
$\Uparrow$
Potisni automat: $M_2 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
$\Uparrow$
Linearno ograničeni automat: $M_1 = (Q, \Sigma, \Gamma, \delta, q_0, ¢, \$, F)$
$\Uparrow$
Turingov stroj: $M_0 = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

SLOŽENOST PRIHVAĆANJA JEZIKA 1

Definicija prostorne složenosti prihvaćanja jezika

- temelji se na traci na kojoj TS koristi najviše ćelija
- ulazna traka se ne gleda, radne trake se gledaju
- TS M prostorne složenosti $S(n)$ ako koristi najviše $S(n)$ ćelija na nekoj od radnih traka
- prostorna složenost niza $\max(1, \lceil S(n) \rceil)$

Vremenska složenost prihvaćanja jezika

- K dvostrano beskonačnih traka, jedna ulazna (može se na nju pisati)
- broj pomaka glave pri prihvaćanju niza
- TS M je vremenske složenosti $T(n)$ ako se napravi najviše $T(n)$ pomaka glave
- Vremenska složenost jezika $T(n)$
- Vremenska složenost niza $\max(n+1, \lceil T(n) \rceil)$

Svojstva prostorne i vremenske složenosti

Broj traka i prostorna složenost

↳ ne utječe na prostornu složenost

↳ [Stanje TS M_1, Brojilo, Sadržaj tr. 1 ... sad. tr. k]

Broj traka i vremenska složenost

↳ TS M_1 ima k traka, TS M_2 ima 1 traku; vremenske je složenosti $T^2(n)$, $L(M_2) = L(M_1)$

↳ TS sa K glava radi M_m pomaka, onda TS sa 1 glavom radi $\sum_{i=1} 4i \approx 2m^2$ pomaka

Sažimanje prostora za konstantni faktor

↳ ako TS M_1 sa k radnih traka pros. s $S(n)$ prihvaća $L(M_1)$, onda za $c>0$ postoji TS M_2 $cS(n)$ koji prihv. $L(M_2) = L(M_1)$

Klase jezika s obzirom na složenost prihvaćanja jezika

- s obzirom na nedeterministički TS
  ↳ prostorna složenost $S(n)$ (za niti jedan niz duljine n, niti jedan slijed prijelaza, na niti jednoj od radnih traka ne koristi više od $S(n)$ ćelija)
  ↳ Jezik L
    ↳ nedeterminističke prostorne složenosti $S(n)$ (ako postoji Nedeterministički TS M prostorne složenosti $S(n)$ koji prihvaća jezik $L(M)=L$
  ↳ vremenska složenost $T(n)$ (za niti jedan niz duljine n, niti jedan slijed prijelaza, ne pomakne glavu više od $T(n)$ puta)
  → Jezik L
    ↳ nedeterminističke vremenske složenosti $T(n)$ (ako postoji nedeterministički TS M vremenske složenosti $T(n)$ koji prihvaća jezik $L(M)=L$)

Četiri osnovne klase jezika
↳ DSPACE(S(n))
- skup jezika determinističke prostorne složenosti S(n)
↳ NSPACE(S(n))
- skup jezika nedeterminističke prostorne složenosti S(n)
↳ DTIME(T(n))
- skup jezika determinističke vremenske složenosti T(n)
↳ NTIME(T(n))
- skup jezika nedeterminističke vremenske složenosti T(n)

$L \in DTIME(f(n)) \Rightarrow L \in DSPACE(f(n))$
↳ DTS M f(n), primjenom f(n) pomaka glave ne može obići više od f(n)+1 ćelija
↳ sažimanjem sadržaja dviju ćelija u jednu ćeliju gradi se TS koji ne koristi više od $\lceil (f(n)+1)/2 \rceil$
↳ $(\lceil f(n)+1)/2 \rceil < f(n)) \Rightarrow L \in DSPACE(f(n))$

Ako $L \in DSPACE(f(n)) \wedge f(n) \geq \log_2 n) \Rightarrow L \in DTIME(c^{f(n)})$
↳ DTS M_1 f(n) - jedna ulazna i 1 radna traka
Maksimalan broj različitih konfiguracija TS M_1
↳ $s(n+2)f(n)t^{f(n)}$
- s - kardinalni broj skupa stanja Q
- n+2 - broj različitih položaja glave za čitanje na ulaznoj traci (n = dulj. niza w)
- f(+) - broj različitih položaja glave za čitanje na radnoj traci
- $t^{f(n)}$ - broj različitih sadržaja radne trake
  - t - kardinalni broj Γ
  - f(n) max broj ćelija za TS $\hat{M}$

$f(n) \geq \log_2 n \Rightarrow (c^{f(n)} \geq s(n+2)f(n)t^{f(n)}) \Rightarrow L \in DTIME(c^{f(n)})$

Ako $L \in NTIME(f(n)) \wedge f(n) \geq \log_2 n) \Rightarrow L \in DTIME(c^{f(n)})$

↳ NTS M_1 $f(n)$-k radnih traka, primjenom $f(n)$ pomaka glave ne može obići više od $f(n+1)$ ćelija

Maksimalan broj različitih konfiguracija TS M_1

↳ $s(f(n)+1)^k t^{kf(n)}$

- s - kard br. Q
- $(f(n)+1)^k$ - broj položaja glave na k radnih traka
  - $f(n)+1$ - br. položaja glave za čitanje na 1 r.t.
  - k - broj radnih traka
  - n - dulj. niza w
- $t^{kf(n)}$ - broj različitih sadržaja k radnih traka
  - $t^{f(n)}$ - br. razl. s. jedne radne trake
  - t - kard. br. skupa zn. trake Π

Maksimalni broj ćelija potrebnih za spremanje liste konfig.

↳ max br. konfig $V = s(f(n)+1)^k t^{kf(n)}$

↳ broj ćelija za 1 konfig $W = k(f(n)+1)+1$

↳ $B = VW$

- Pomaci glave početak liste <-> kraj liste

↳ broj pomaka glave $= B^2$

↳ $f(n) \geq \log_2 n \Rightarrow (c^{f(n)} \geq B^2) \Rightarrow L \in DTIME(c^{f(n)})$

Prostorna izgradivost $S(n)$

↳ prostorno izgradiva - $\forall n > n_0$ postoji TS M koji računa vrijednost $S(n)$ i koristi točno $S(n)$ ćelija

↳ potpuno prostorno izgradiva - p.i. vrijedi $\forall n$

Vremenska izgradivost $T(n)$

↳ vremenski izgradiva - $\forall n > n_0$ postoji TS M koji računa vrijednost $T(n)$ u točno $T(n)$ koraka

↳ potpuno vremenski izgradiva - v.i. vrijedi $\forall n$

Funkcija $f(n)$

↳ $f(n) \geq \log_2 n$ potpuno prostorno izgradiva

Ako $L \in NSPACE(f(n)) \Rightarrow L \in DSPACE(f^2(n))$

↳ razmatrani nedeter. TS M_1 $f(n)$ 1 ul. t. 1 r. t.

↳ max. br. razl. konf. TS M_1

$s(n+2)f(n)t^{f(n)} \leq c^{f(n)}$

s - k. b. Q

(n+2) - broj razl. pol. glave začit. na ul. tr.

n - duljina niza w

f(n) - br. razl. polj. gl. na rad. tr.

$t^{f(n)}$ - br. razl. sadr. rad. tr

- sadržaj ulazne trake se ne mijenja

Svojstva hijerarhije jezika

- povećavamo vremensku složenost
- $DTIME(n^2) = DTIME(4n^2)$
- postoji li f takva da je svaki rekurzivni jezik u $DTIME(f(n))$?
- hijerarhija jezika je
  ↳ beskonačna
  ↳ neprekinuta za potpuno prostorno izgradive fje
  ↳ nije neprekinuta za opći slučaj fja koje nisu prostorno i vremenski izgradive
  ↳ moguće je zadati jezik za koji ne postoji optimalni TS koji ga prihvaća u min vremenu ili prostoru
  ~~↳ za unutarkl~~

↳ unija klase jezika

Beskonačnost
↳ ako je $f(n)$ potpuno rekurzivna f-ja onda postoji
L takav da $L \notin DTIME(f(n))$, $L \notin NTIME(f(n))$,
$L \notin DSPACE(f(n))$, $L \notin NSPACE(f(n))$
↳ $L \in \{w_i \mid$ TS M_i ne prihvaća w_i u manje od $f(|w_i|)$
pomaka glave$\}$
Neprekinutost h. za potpuno prostorno izgr. funkcije
↳ S_1 i S_2 imaju relativne razlike ~~$DSPACE(S_1(n)) \subset DSPACE(S_2$~~
$DSPACE(S_2(n)) > DSPACE(S_1(n))$
$L \in DSPACE(S_2(n))$, $L \notin DSPACE(S_1(n))$
↳ $S_2(n)$ potpuno prostorno izgradiva f-ja
↳ $\inf_{n \to \infty} S_1(n)/S_2(n) = 0$
↳ $S_1(n)$ i $S_2(n)$ najmanje $\log_2 n$
↳ onda postoji L
- $L \in DSPACE(S_2(n)) \wedge L \notin DSPACE(S_1(n))$
↳ Gradi se TS M za koji vrijedi
- prostorne složenosti $S_2(n)$
- za barem 1 ul. niz donosi suprotnu odluku od
bilo kojeg TS prostorne složenosti $S_1(n)$
→ osigurati prostornu složenost $S_2(n)$
- simulira rad TS M_{S_2} koji za bilo koji niz duljine
n koristi svih $S_2(n)$ ćelija
- $S_2(n)$ je potpuno prostorno izgradiva

Nepremitutost hijerarhije za potp. vr. izgr. fje

↳ $T_2(n)$ potp. vr. izgr. fja

$\inf_{n\to\infty} T_1(n)\log T_1(n)/T_2(n)=0$

Onda postoji L:

$L \in DTIME(T_2(n)) \wedge L \notin DTIME(T_1(n))$

↳ analogan prostornoj složenosti

Praznine u hijerarhiji

↳ ako ne zahtijevamo potp. prostornu izgradivost

onda teorem neće vrijediti

↳ $g(n)$ proizvoljna potp. rek. fja, $g(n)\geq n$

↳ moguće izgraditi potp. rek. fju $S(n)$

tako da $DSPACE(S(n))=DSPACE(g(S(n)))$

$DTIME(f(n))=NTIME(f(n))=DSPACE(f(n))=NSPACE(f(n$

Optimalni TS

↳ ne postoji uvijek optimalno rješenje

↳ $r(n)$ - bilo koja potp. rek. fja, moguće izgraditi

jezik L

- TS M_i složenosti $S_i(n)$, $L(M_i)=L$
- postoji TS M_j složenosti $S_j(n)$, $L(M_j)=L(M_i)=L$
- $S_i(n)\geq r(S_j(n))$ za skoro sve n

Unija klasa jezika

↳ $\{f_i(n) \mid i=1,2,3\}$ - skup rekurzivnih fja

↳ ako je moguće izgraditi TS M koji ispisuje

listu $M_1, M_2 \dots$ koji računaju fje $f_1(n), f_2(n) \dots$

i ako za sve n i i vrijedi $f_i(n)<f_{i+1}(n)$

↳ postoji $S(n)$ za koju vrijedi

$$DSPACE(S(n))=\bigcup_{i\geq 1} DSPACE(f_i(n))$$

JEZICI POLINOMNE SLOŽENOSTI

- jezik je polinomne vremenske složenosti ako postoji TSM koji ga prepoznaje i koji je vremenske složenosti $T(n)$ gdje je $T(n)$ polinom
- Klasa polinomne vremenske složenosti P

$$P = \bigcup_{i \geq 1} DTIME(n^i)$$

- Klasa ned. polinomne vremenske složenosti NP

$$NP = \bigcup_{i \geq 1} NTIME(n^i)$$

- Jezik L_{sat} - modelira odlučivost log. formule
  - ↳ Logički izraz - varijable, zagrade, $\wedge, \vee, \neg$
  - ↳ varijable 1 ili 0
  - ↳ niz znakova w - logički izraz ~~$y_1 \wedge y_2$~~
    - ako je moguće pridružiti vrijednosti 0 ili 1 tako da log. izraz poprimi vrijednost 1
- Ne postoji način da se odredi efikasni deterministički algoritam koji će odrediti ako je $w \in L_{sat}$
- jezik L_{VC} - vertex cover, grafovi
  - ↳ $G=(V,E)$ - neusmjeren graf
  - ↳ obrubje čvorova A
    - podskup skupa čvorova V za koji vrijedi: ako je (x,y) bilo koja grana iz skupa grana E onda je u skupu A barem 1 od čvorova x ili y
- Niz je u jeziku L_{VC} ako za graf G postoji obrubje čvorova u kojemu je k ili manje čvorova

Klasa polinomne vremenske složenosti P

$P = \bigcup_{i \geq 1} DTIME(n^i)$

Klasa ned. polinomne vremenske složenosti NP

$NP = \bigcup_{i \geq 1} NTIME(n^i)$

Klase jezika s obz. na prostornu polinomnu složenost

$PSPACE = \bigcup_{i \geq 1} DSPACE(n^i)$

$NPSPACE = \bigcup_{i \geq 1} NSPACE(n^i)$

$NSPACE(f(n)) \subseteq DSPACE(f(n)^2)$

$P \subseteq NP \subseteq PSPACE = NPSPACE$

Svođenje jezika - TS M_R

↳ jezik L_1 se svodi na L_2 ako

$x \in L_1 \Leftrightarrow y = R(x) \in L_2$

↳ M_R prevodi jednu vrstu problema u drugu vrstu

↳ redukcija je efikasna ako je M_R polinomne vremenske složenosti

L_1 se polinomno svodi na L_2

↳ ako je L_2 u klasi P onda je i L_1 u P

↳ —— || —— NP — || —— NP

L_1 se u log. prostoru svodi na L_2

↳ L_2 u klasi P $\Rightarrow$ L_1 u klasi P

↳ L_2 u $NSPACE(\log^k n) \Rightarrow L_1$ u $NSPACE(\log^k n)$

↳ L_2 u $DSPACE(\log^k n)$ —— || —— $DSPACE(\log^k n)$

NP težak

↳ ako su svi jezici iz klase NP polinomno vremenski svodivi na jezik L

NP potpun

↳ jezik L je u klasi NP i NP težak je

L_{SAT} je NP potpun

$L \in NP$ L_{Sat} se svodi na L, L je NP potpun

P = NP problem