Teorie jazyků a automatů— 1. týden

Marie Demlová
http://math.feld.cvut.cz/demlova

21. 9. 2020

Informace o organizaci semestru

Zápočet

- se získává za aktivní účast na cvičení, upřesní cvičící
- na cvičení se píší dva testy — oba za 20 bodů
- v případě kontaktní výuky pro získání zápočtu je nutné získat z každého testu aspoň 8 ~~20~~ bodů a každý bod nad 12 bodů se připočte k počtu bodů z ústní zkoušky
- v případě online výuky se body ze semestru do zkoušky nezapočítávají

Zkouška

- písemná zkouška — max. 80 bodů
- ústní část pro ty, co získali aspoň **40 bodů** z písemné zkoušky; maximální zisk 20 bodů.

Úvod

Teorie jazyků, automatů a gramatik

Co je obsahem předmětu?

konečný popis nekonečných množin

- přijímání
- popis regulárními výrazy
- generování

Úvod

generování:
dáme pravidla, jak slova z množiny získat.

<číslo> ::= <číslice> | <číslo><číslice>

<číslice> ::= 0 | 1 | 2 | . . . | 9 |

01

Úvod

K čemu je to dobré?

- zpracování přirozeného jazyka
- překladače
- návrh a popis hardware
- vyhledávání v textu

K čemu jsou přednášky?

Všechny informace o předmětu najdete na Moodlu

~~http://~~https://moodle.fel.cvut.cz/course/view.php?id=5400

Úvod

Trocha historie

30. až 40. léta 20. století — první formalizace pojmu algoritmu; Turing, Post, Kleene, Church, Markov

kolem poloviny 20. století — konečné automaty a neuronové sítě; Kleene

60. léta 20. století — gramatiky (Chomsky)
zásobníkové automaty
formální teorie konečných automatů

Jazyky

Abeceda je konečná neprázdná množina symbolů Σ.

Slovo nad abecedou je každá konečná posloupnost prvků (písmen) abecedy.

0100 nad $\Sigma = \{0,1\}$

prázdné slovo ε je posloupnost, která nemá žádné písmeno.

λ

- $\Sigma^\star$ — množina všech slov nad Σ
- Σ^+ — množina všech neprázdných slov nad Σ

$$\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$$

Jazyky

Délka slova u je
rovna délce této konečné posloupnosti, značíme $|u|$;
tj. délka slova $u = a_1 a_2 \ldots a_k$ je $|u| = k$. Délka ε je 0.

Pro písmeno a a slovo u je $|u|_a$ počet výskytů a v u.

$|01011| = 4$ $|\varepsilon| = 0$

$a \in \Sigma$ $|u|_a$ např. $|01011|_0 = 2$

Jazyky

Zřetězení slov.

Pro dvě slova $u = a_1 a_2 \dots a_k$ a $v = b_1 b_2 \dots b_m$ je

$$u \cdot v = a_1 \dots a_k b_1 \dots b_m.$$

Slovo $u \cdot v$ je zřetězením slov u a v.

$$|u \cdot v| = |u| + |v|$$

$u = a_1 a_2 \dots a_k$ $\quad u = a_1 \cdot (a_2 \dots a_k) =$

$a_1 \cdot a_2 \cdot \dots \cdot a_k$

Jazyky

Tvrzení.
Operace zřetězení je asociativní a není komutativní operace na Σ.

je asociativní: $u = a_1 \dots a_k$, $v = b_1 \dots b_n$, $w = c_1 \dots c_m$

$$(u \cdot v) \cdot w = (a_1 \dots a_k b_1 \dots b_n) \cdot (c_1 \dots c_m) =$$
$$= a_1 \dots a_k (b_1 \dots b_n c_1 \dots c_m) = u \cdot (v \cdot w)$$

$\Sigma = \{0, 1\}$ $\quad 0 \cdot 1 = 01 \neq 1 \cdot 0 = 10$. $\quad \varepsilon \cdot v = v = v \cdot \varepsilon$

$(\Sigma^*, \cdot, \varepsilon)$ .. je monoid

Mocniny.
Pro slovo u je

$$\underline{u^0 = \varepsilon}, \qquad \underline{u^{i+1} = u\, u^i \ \text{pro každé } i.}$$

Jazyky

f

Podslovo, prefix, sufix.

Slovo w je podslovo u, jestliže existují x, y, že

$$u = xwy.$$

$$\underbrace{01}_{\text{prefix}}110\overbrace{1}^{\text{suffix}}$$

Je-li $x = \varepsilon$, je w prefix u.

Je-li $y = \varepsilon$, je w suffix u.

w je vlastní podslovo, (prefix, suffix), jestliže $w \neq u$.

Jazyky

Jazyk nad abecedou Σ je jakákoli podmnožina množiny všech slov nad abecedou Σ. Tj. $L \subseteq \Sigma^\star$.

Př.: $L = \{ a u \mid u \in \{a, b\}^* \}$
nad $\Sigma = \{a, b\}$

Σ^* je spočetná množina

$\mathcal{P}(\Sigma^*)$ je nespočetná množina

Konečné automaty

Neformální popis

Všechny typy konečných automatů mají

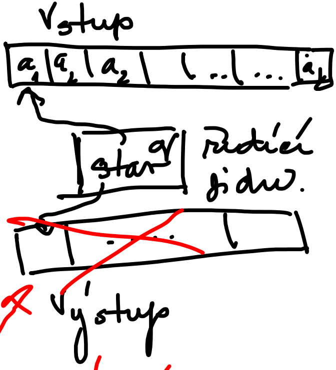
- konečné množiny stavů Q
- konečné množiny vstupů Σ
- přechodové funkce δ
- počáteční stav $q_0 \in Q$

Mealyho a Mooreův automat mají ještě

- konečnou množiny výstupů Y
- „výstupní“ funkci

DFA (též akceptor) má ještě

- množinu koncových stavů



nemá výstup

ano, ne

Konečné automaty

Příklad — posuvný registr.
Stavy jsou uspořádané k-tice naposled přečtených symbolů. Přechodová funkce popisuje, jak přečtením dalšího symbolu přejdeme do nové k-tice (nového stavu).

Příklad - hledání v textu
Úkol: zjistit, zda se v daném slovu textu vyskytuje podslovo zadaná šablona.

text $\underbrace{abab}$ b a a Ano šablona bab

aabaab Ne,

Konečné automaty

Mealyho automat je $(Q, \Sigma, Y, \delta, q_0, \lambda)$, kde

- Q je konečná množina stavů,
- Σ je konečná neprázdná množina vstupních symbolů,
- Y je konečná neprázdná množina výstupních symbolů,
- q_0 je počáteční stav,
- δ je přechodová funkce, tj. zobrazení $\delta\colon Q \times \Sigma \to Q$,
- λ je výstupní funkce, tj. zobrazení $\lambda\colon Q \times \Sigma \to Y$.

Konečné automaty

Mooreův automat je $(Q, \Sigma, Y, \delta, q_0, \beta)$, kde

- Q je konečná množina stavů,
- Σ je konečná neprázdná množina vstupních symbolů,
- Y je konečná neprázdná množina výstupních symbolů,
- q_0 je počáteční stav,
- δ je přechodová funkce, tj. zobrazení $\delta\colon Q \times \Sigma \to Q$,
- β je značkovací funkce, tj. zobrazení $\beta\colon Q \to Y$.

$Y = \{0, 1\}$

$0 = NE$

$1 = ANO$

$\beta : Q \to \{0, 1\}$

$q \in F \subseteq Q \Leftrightarrow \beta(q) = 1$

Konečné automaty

DFA (též akceptor) je $(Q, \Sigma, \cancel{X}, \delta, q_0, F)$, kde

- Q je konečná množina stavů,
- Σ je konečná neprázdná množina vstupních symbolů,
- q_0 je počáteční stav,
- δ je přechodová funkce, tj. zobrazení $\delta\colon Q \times \Sigma \to Q$,
- $F \subseteq Q$ množina koncových stavů.

Zadání automatu

- tabulkou
- stavovým diagramem

Konečné automaty

Nejdříve uvedeme příklad DFA:

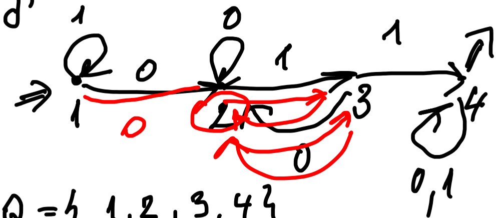
Příklad - hledání v textu

Úkol: zjistit, zda se v binárním slovu *w* vyskytuje podslovo 011.

δ

		0	1
$\rightarrow$	1	2	1
	2	2	3
	3	2	4
$\leftarrow$	4	4	4

δ



$Q = \{1, 2, 3, 4\}$

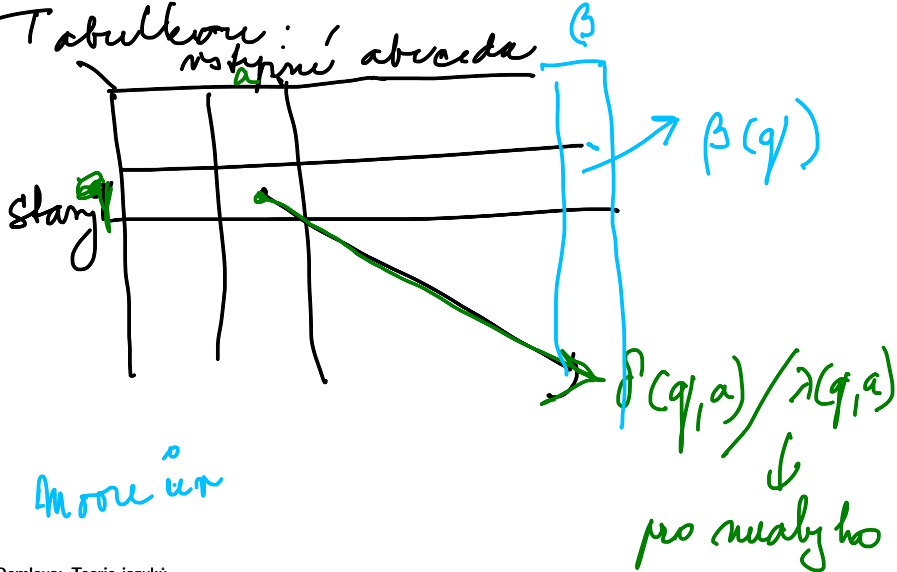
$\Sigma = \{0, 1\}$ $\quad q_0 = 1$

$F = \{4\}$

01010

Konečné automaty

Tabulkou:



Konečné automaty

Stavový diagram — formálně

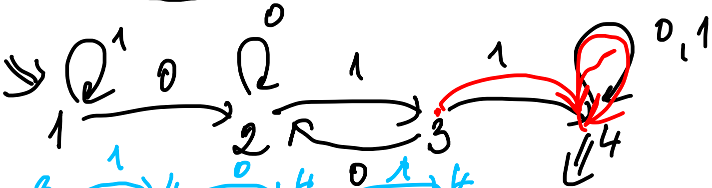
Stavový diagram je orientovaný graf, kde

- vrcholy jsou stavy,
- hrana vede z q do p iff $\delta(q, a) = p$ a je ohodnocena a,
- je vyznačen počáteční stav, $\rightarrow$
- pro Mealyho automat je hrana ohodnocena i $\lambda(q, a)$, $a/\lambda(q,a)$
- pro Mooreův automat je vrchol q ohodnocen $\beta(q)$,
- pro DFA jsou vyznačeny koncové stavy. $\Leftarrow$

Konečné automaty

Máme automat s množinou stavů Q, vstupy Σ, a přechodovou funkci δ. Pak rozšířená přechodová funkce $\delta^\star\colon Q \times \Sigma^\star \to Q$ je definovaná

1. $\delta^\star(q, \varepsilon) = q$, pro všechna $q \in Q$,
2. $\delta^\star(q, ua) = \delta(\delta^\star(q, u), a)$, pro všechna $q \in Q$, $a \in \Sigma$, $u \in \Sigma^\star$.



$3 \xrightarrow{1} 4 \xrightarrow{0} 4, \xrightarrow{1} 4$

101

$\delta^*(3, 101) = \delta(\delta^*(3, 10), 1) = \delta(\delta(\delta^*(3, 1), 0), 1) = \delta(4, 1) = 4$

Konečné automaty

Tvrzení. Pro rozšířenou přechodovou funkci platí

$$\delta^\star(q, uv) = \delta^\star(\delta^\star(q, u), v).$$

nástin zdůvodnění

$\delta^*(q, a_1 \dots, a_k)$

$\delta(q, a_1) = q_1,\ \delta(q_1, a_2) = q_2,\ \dots,\ \delta(q_{k-1}, a_k) = p$

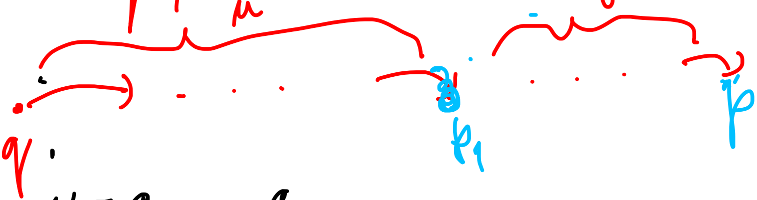
$q \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \longrightarrow \dots \xrightarrow{a_k} p$

$\delta^*(q, a_1 \dots a_k) = p$ iff sled ve stav. diag. označený $a_1 \dots a_k$ z q končí v p.

Konečné automaty

~~**Ještě jiný pohled na DFA:**~~

$\delta^*(q, uv) = p \qquad \delta^*(\delta^*(q,u), v)$



$w = a_1 \dots a_k$

$\delta^*(q, a_1 \dots a_k) = \delta(\dots(\delta(\delta(\delta(q,a_1),a_2),a_3)\dots, a_k)$

Konečné automaty

Jazyk přijímaný DFA.

Pro DFA $M = (Q, \Sigma, \delta, q_0, F)$ je *jazyk přijímaný* L množina slov $L(M)$, kde

$$L(M) = \{w \mid \delta^\star(q_0, w) \in F\}.$$

Regulární jazyky.

Jazyk L, pro který existuje DFA M, že $L = L(M)$, nazýváme regulárním jazyk.

Všechny regulární jazyky značíme Reg a mluvíme o nich jako o třídě regulárních jazyků.

Přednáška 5. října 2020

$L \subseteq \Sigma^*$ je regulární znamená, že k němu existuje DFA M, že $L = L(M)$. Není regulární $L = \{0^n 1^n \mid n \geq 0\}$.

Pumping lemma pro regulární jazyky.

Pro každý regulární jazyk L nad abecedou Σ existuje přirozené číslo n s touto vlastností:

Každé slovo $u \in L$, které je delší než n, lze rozdělit na tři slova $u = xwy$ tak, že

1. $|xw| \leq n$,
2. $w \neq \varepsilon$
3. a pro každé přirozené číslo $i = 0, 1, \ldots$ platí $xw^iy \in L$.

$\big(L$ reg. $\Rightarrow \exists n \geq 1 \ (\forall u \in L \ |u| \geq n \Rightarrow \exists x, w, y$ s vlast. 1) $|xw| \leq n$, 2) $w \neq \varepsilon$, 3) $xw^iy \in L \ \forall i) \big)$

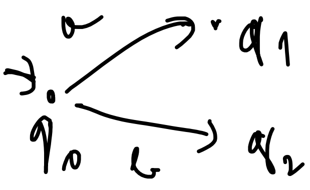
Není pravda, když L je jazyk $\boxed{\forall n \geq 1 \ \exists u \in L \ |u| \geq n \ \forall u = xwy \text{ tak } |xw| \leq n, \ w \neq \varepsilon \Rightarrow \exists i : \ xw^iy \notin L)}$

PL není možné použít k důkazu, že L je regulární.

Důkaz: Víme, že L je regulární. Tj. existuje DFA $M = (Q, \Sigma, \delta, q_0, F)$, že $L = L(M)$.

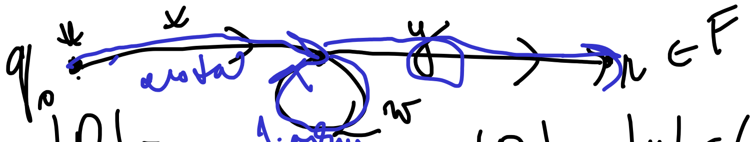
Stavový diagram – orient. graf (Q, E)

$E = \{ (p,q) \mid \delta(p,a) = q \}$ $\quad (p,q)$ ohodnocení a



$\delta^*(q_0, u) = r$ iff sled v (Q,E) ohodnocený u vede z q_0 do r.

$u \in L(M)$ iff $\delta^*(q_0, u) = r \in F$. $\quad |u| \geq n$



máme $|Q| = n$ $\quad n := |Q|$, $|u| \in L$ $|u| \geq n$ obsahuje cyklus.

Fakt. Jazyk $L = \{0^m 1^m \mid m \geq 0\}$ není regulární jazyk.

Ukážeme pomocí pumping lemmatu.

Kdyby byl L regulární, tak existuje $n \geq 1$,

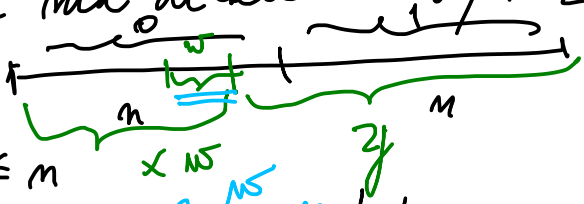
zvolíme $u \in L$ $\quad u = 0^n 1^n$

víme $u \in L$, $|u| = 2n \geq n$

Tudíž by mělo jít rozdělit $u = xwy$ tak, že

$|xw| \leq n$, $w \neq \varepsilon$

když xw je prefix $0^n 1^n$ a má délku $\leq n$, tak



$xw = 0^\ell \quad \ell \leq n$

a $w \neq \varepsilon \quad w = 0^k \quad 1 \leq k \leq n$

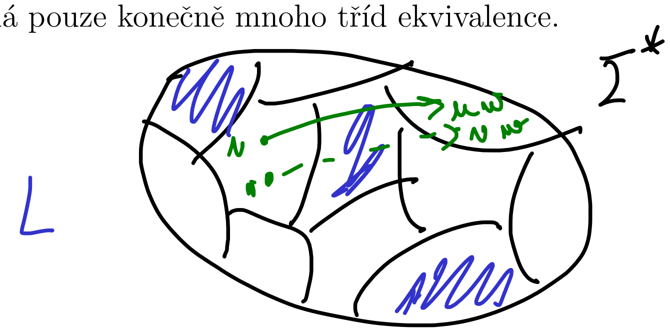
Pak $xw^2y = xwwy = 0^{n+k}1^n \notin L$

xw^2y má více 0 než 1, protože $k \geq 1$

To je spor. Špatně byl předpoklad, že L je regulární.

Nerodova věta.
Je dán jazyk L nad abecedou Σ. Pak L je regulární jazyk právě tehdy, když existuje ekvivalence T na množině všech slov $\Sigma^\star$ taková, že

1. L je sjednocení některých tříd ekvivalence T.
2. Jestliže pro nějaké $u, v \in \Sigma^\star$ platí $u\,T\,v$, pak pro každé slovo $w \in \Sigma^\star$ platí také $uw\,T\,vw$.
3. T má pouze konečně mnoho tříd ekvivalence.



$T \subseteq \Sigma^* \times \Sigma^*$

Důkaz: I. L je regulární. Máme DFA $M=(Q,\Sigma,\delta,q_0,F)$, že $L=L(M)$.

Položme: $\boxed{u\,T\,v \quad u,v\in\Sigma^* \text{ iff } \delta^*(q_0,u)=\delta^*(q_0,v)}$

Je T ekvivalence? Ano

reflexivní $\delta^*(q_0,u)=\delta^*(q_0,u)$ tj. $u\,T\,u$ $\forall u$

sym. Když $\underbrace{\delta^*(q_0,u)=\delta^*(q_0,v)}_{u\,T\,v} \Rightarrow \delta^*(q_0,v)=\delta^*(q_0,u)$ tj. $v\,T\,u$

tranz. Když $u\,T\,v$ $v\,T\,w$

$\delta^*(q_0,u)=\delta^*(q_0,v)=\delta^*(q_0,w)$ tj. $u\,T\,w$

1) $L=\bigcup\{T[u] \mid \delta^*(q_0,u)\in F\}$

2) Když $u\,T\,v$, tj. $\delta^*(q_0,u)=\delta^*(q_0,v)$, w lib.

pak $\delta^*(q_0,uw)=\delta^*(\delta^*(q_0,u),w)$

$\delta^*(q_0,vw)=\delta^*(\delta^*(q_0,v),w)$ $\Rightarrow$ $\delta^*(q_0,uw)=\delta^*(q_0,vw)$ a $uw\,T\,vw$

3) T má nejvýše tolik tříd, kolik je stavů a těch je konečně mnoho.

II. Máme ekvivalenci T s vlastnostmi 1, 2, 3.

Zkonstruujeme DFA $M=(Q,\Sigma,\delta,q_0,F)$, že $L=L(M)$.

Máme T

Pokračování důkazu Nerodovy věty

$Q := \{T[u] \mid u \in \Sigma^*\}$ $\quad$ Q je neprázdná konečná (3)

$q_0 := T[\varepsilon] \in Q$ $\quad$ $F := \{T[u] \mid T[u] \subseteq L\}$ (1)

$\delta(\underbrace{T[u]}_{\in Q}, a) := T[ua]$. $\quad$ $M = (Q, \Sigma, \delta, q_0, F)$

správnost této definice zaručuje (2).

$\delta^*(T[\varepsilon], u) = T[u]$ $\quad$ $u \in L(M)$ iff $T[u] \subseteq L$

iff $u \in L$ $\quad (u \in T[u])$. $\quad$ cbd

Fakt. Jazyk $L = \{0^m 1^m \,|\, m \geq 0\}$ není regulární jazyk.

Zdůvodnění pomocí Nerodovy věty. Sporem.

Kdyby L byl regulární, tak existuje ekvivalence T na Σ^*, že

1) L je sjednocení některých tříd

2) if $u\,T\,v$ tak $uw\,T\,vw$ $\quad \forall w \in \Sigma^*$

3) T má konečně mnoho tříd.

$L = \{0^m 1^m \mid m \geq 0\}$

zvolíme $0, 0^2, 0^3, 0^4, 0^5, \ldots, 0^n, \ldots$

nekonečná posloupnost slov, proto musí existovat

$0^i \, T \, 0^j \qquad i < j$

Zvolíme $w = 1^i$. Pak (2) $\underset{\in L}{0^i 1^i} \; T \; \underset{\notin L}{0^j 1^i} \qquad i \neq j$

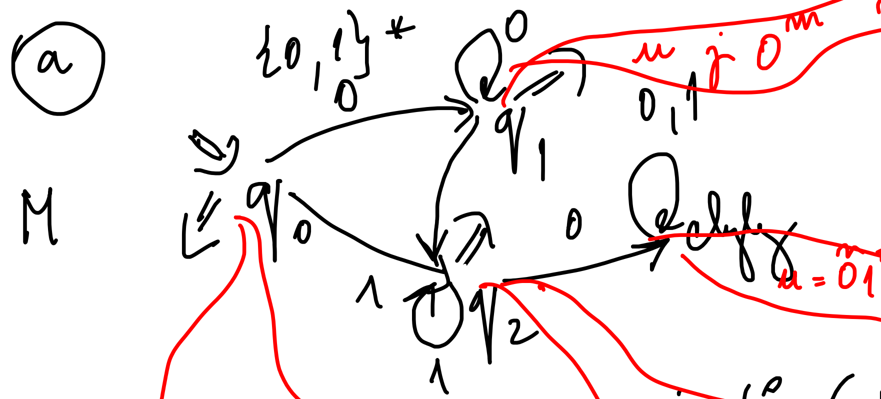
Spor s 1) tj L není sjednocení některých tříd.

Tedíž L není regulární.

cbd

Fakt. Jazyk $L = \{0^m 1^k \mid m, k \geq 0\}$ je regulární jazyk.

(a)



u je 0^m $m \geq 1$ $\delta^*(q_0, u) = q_1$

$Q = \{q_0, q_1, q_2, \text{chyb}\}$

$F = \{q_0, q_1, q_2\}$

q_0 – poč.

$u = 0^i 1 0^j$, $i \geq 1$, $j \geq 1$

Dokážeme pomocí invariantů (Ner. věty), že $L(M) = L$.

u, že $\delta^*(q_0, u) = q_0$

$u = \varepsilon$

$u = 0^m 1^j$

$m \geq 0$ $j \geq 1$

$\delta^*(q_0, u) = q_2$

Jak pomocí Nerodovy věty dokázat, že daný automat přijímá daný jazyk. Užijeme invarianty.

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

charakt.

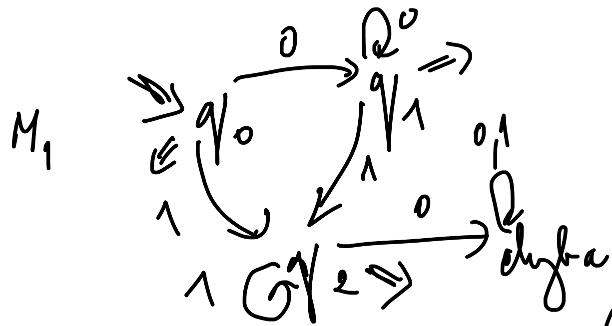
pro každý stav $q \in Q$ popíšeme všechna slova u taková, že $\delta^*(q_0, u) = q$.

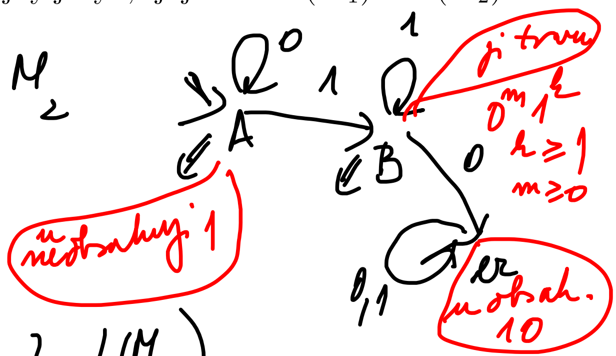
A tyto charakterizace rozdělí Σ^* na navzájem disjunktní množiny.

Ekvivalentní DFA.

Automaty M_1 a M_2 jsou *ekvivalentní*, jestliže přijímají stejný jazyk, tj. jestliže $L(M_1) = L(M_2)$.

Př.:



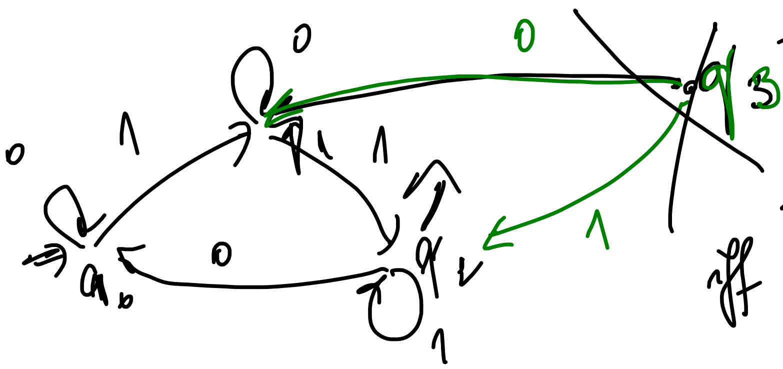


$L(M_1) = \{0^m 1^k \mid m \geq 0, k \geq 0\} = L(M_2)$

Dosažitelné stavy.

Je dán DFA $M = (Q, \Sigma, \delta, q_0, F)$. Řekneme, že stav $q \in Q$ je *dosažitelný*, jestliže existuje slovo $u \in \Sigma^\star$ takové, že $\delta^\star(q_0, u) = q$.

Př.



→ nedosažitelný

$u \in L(M)$ iff $\delta^*(q_0, u) \in F$

Postup nalezení dosažitelných stavů.

```
Q_0 := {q_0}
repeat
        Q_{i+1} := Q_i ∪ {δ(q,a) | q ∈ Q_i, a ∈ Σ}
until Q_{i+1} = Q_i.
return Q' = Q_i
```

prohledávání stavového diagramu z q_0 do šířky.

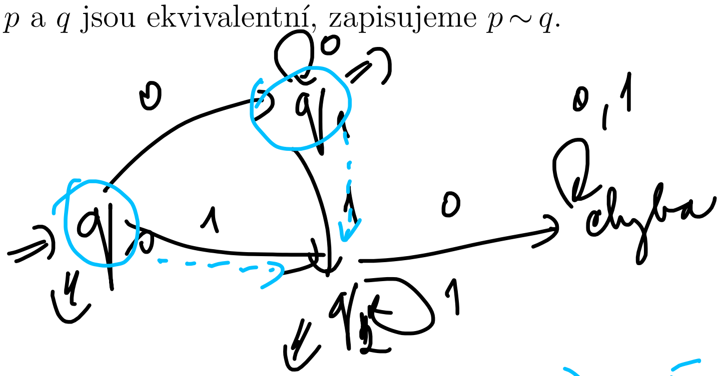
Ekvivalence stavů $\sim$

Máme DFA $M = (Q, \Sigma, \delta, q_0, F)$. Řekneme, že dva stavy $p, q \in Q$ jsou *ekvivalentní*, jestliže pro každé slovo $u \in \Sigma^\star$ platí

$$\delta^\star(p, u) \in F \quad \text{právě tehdy, když} \quad \delta^\star(q, u) \in F.$$

Fakt, že dva stavy p a q jsou ekvivalentní, zapisujeme $p \sim q$.

Př.: M:



$q_0 \sim q_1$

$\boxed{q_0 \not\sim q_2}$

$\delta(q_1, 0) \in F$

$\delta(q_2, 0) = ch \notin F$

$\delta^*(q_0, u) \in F$ iff $\delta^*(q_1, u) \in F$

$u = \varepsilon$ $\quad q_0 \in F \quad q_1 \in F$

$u = 0^n \; n \geq 1 \quad \delta^*(q_0, u) = q_1 \quad \delta^*(q_1, u) = q_1$

Redukovaný automat.

Je dán DFA $M = (Q, \Sigma, \delta, q_0, F)$. Řekneme, že M je *redukovaný*, jestliže nemá nedosažitelné stavy a žádné jeho dva různé stavy nejsou ekvivalentní.

Příště: Ukážeme postup, jak zkonstruovat $\sim$ a M_1, M_2 jsou ekvivalentní právě když redukované automaty k M_1 a M_2 se liší jen pojmenováním stavů.

Jak získat ekvivalence $\sim$

Jak získat ekvivalence $\sim$ – pokračování

Přednáška 12. října 2020

Čím jsme skončili minulou přednášku.

Redukovaný automat splňuje

1) nemá nedosažitelné stavy

2) ekvivalence $\sim$ je identická, tj. nemáme $p \neq q$ a $p \sim q$

Def.: $p \sim q$ iff pro každé slovo $w \in \Sigma^*$ $\delta^*(p,w) \in F$ iff $\delta^*(q,w) \in F$

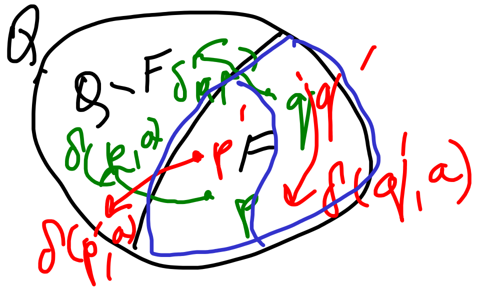
Konstrukce relace $\sim$

Na Q konstruujeme $\sim_i$, kde $i = 0, 1, \ldots$:

a) $p \sim_0 q$ iff buď $p, q \in F$ nebo $p, q \notin F$.

b) Dokud $\sim_{i+1} \neq \sim_i$ konstruujeme $\sim_{i+1}$ takto
$p \sim_{i+1} q$ iff $p \sim_i q$ a pro každé $a \in \Sigma$ je $\delta(p,a) \sim_i \delta(q,a)$.

c) Položíme $\sim := \sim_i$ pro $\sim_{i+1} = \sim_i$.



$\sim_0$ má 2 třídy ekvivalence

$i=0$

$\sim_1$: $p \sim_{i+1} q$ iff $p \sim_0 q$ a $\delta(p,a) \sim_0 \delta(q,a)$

$\ldots p' \not\sim_1 q'$

Končíme, když $\sim_i = \sim_{i+1}$.

$\sim = \sim_i \; (= \sim_{i+1})$

Příklad. Najděte ekvivalenci $\sim$ pro konečný automat M, který je dán stavovým diagramem:

		a	b
$\rightarrow$	1	2	1
	2	2	3
$\leftarrow$	3	4	5
	4	4	3
	5	4	5

$\delta(1,b) = 1 \notin F$

$\delta(2,b) = 3 \in F$

$\delta(4,b) = 3 \in F$

$\delta(5,b) = 5 \notin F$

$\sim_0 \quad \{3\} = F \quad \{1,2,4,5\} = Q \setminus F$

konstruujeme

$\sim_1 \quad \{3\}$

$\delta(1,a) = 2 \in Q \setminus F$

$\delta(2,a) = 2$ —"—

$\delta(4,a) = 4 \in Q \setminus F$

$\delta(5,a) = 4$

$\{1,2,4,5\}$

$\{1,5\} \quad \{2,4\}$

$\delta(2,a) = 2$

$\delta(4,a) = 4$

$\delta(2,b) = 3$

$\delta(4,b) = 3$

$\sim_1 \quad \{3\}, \{1,5\}, \{2,4\}$

Konstruujeme

$\sim_2 \quad \{3\}$

$\delta(1,a) = 2 \quad \delta(1,b) = 1$

$\delta(5,a) = 4 \quad \delta(5,b) = 5$

$\sim_2 = \sim_1$ a $\sim = \sim_1$ $\quad 1 \sim 5 \quad 2 \sim 4$

Př:

δ	a	b	$\sim_0$	a	b	$\sim_1$	a	b	$\sim_2$
$\to$ 1	2	1	0	0	0	0	A	0	0
2	2	3	0	0	K	A	A	K	A
$\leftarrow$ 3	4	5	K	0	0	K	A	0	K
4	4	3	0	0	K	A	A	K	A
5	4	5	0	0	0	0	A	0	0

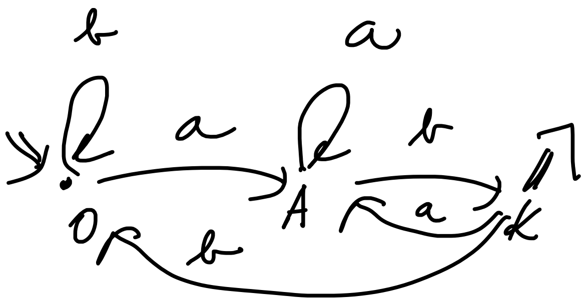
$\sim_1 \quad \overset{0}{\{1,5\}} \quad \overset{A}{\{2,4\}} \quad \overset{K}{\{3\}} \quad = \quad \sim_2 = \sim$

Všechny stavy jsou dosažitelné

Hledaný $\sim$

$K = \{3\} \quad A = \{2,4\} \quad 0 = \{1,5\}$

	a	b
$\Rightarrow$ 0	A	0
A	A	K
$\Leftarrow$ K	A	0



$L(M) = \{uab \mid u \in \{a,b\}^*\}$

Tvrzení. Platí $\sim_0 \supseteq \sim_1 \supseteq \dots \supseteq \sim_i \supseteq \dots$

Navíc, existuje k že $\sim_k = \sim_{k+1}$. Pak pro každé $j \geq 1$ platí $\sim_k = \sim_{k+j}$.

D: Aby $p \sim_{i+1} q$ tak musí $p \sim_i q$ a .....

Proč existuje k, že $\sim_k = \sim_{k+1}$? Protože každá ekvivalence musí $p \sim_i p$. Minimálně (p,p) $\sim_i$ nemůže mít.

Co by znamenalo, že $p \sim_i q \Rightarrow p = q$

je-li DFA jen s dosažitelnými stavy, je DFA redukovaný.

If $p \sim_{i+1} q$ iff $p \sim_i q$, tak $\sim_{i+k} = \sim_i$

If $\sim_i = \sim_{i+1}$, tak $p \sim_{i+2} q$ iff $p \overset{\sim_i}{\sim_{i+1}} q$ a $\delta(p,a) \overset{\sim_i}{\sim_{i+1}} \delta(q,a)$

$\Rightarrow \sim_{i+2} = \sim_{i+1} = \sim_i$

Tvrzení. Pro relace $\sim_i$ platí $p \sim_i q$ iff pro každé slovo u délky menší nebo rovné i je $\delta^\star(p,u) \in F$ iff $\delta^\star(q,u) \in F$.

If $\sim_i = \sim_{i+1} = \sim_{i+k}$ $k \geq 0$, tak

$p \sim_i q$ iff $\forall u \in \Sigma^*$ je $\delta^*(p,u) \in F$ iff $\delta^*(q,u) \in F$

Ano u má délku, $|u| = n$ buď $n \leq i$ nebo $n = i+k$ $k \geq 1$

D.: Indukcí podle $|u| = n$

Z. krok: $u = \varepsilon$ $n = 0$ $\delta^*(p,\varepsilon) = p$ $\delta^*(q,\varepsilon) = q$

$\sim_0$: kdyby $p \in F$, $q \notin F$, tak $p \not\sim_0 q$.

Ind. krok: Předpo., že pro $\sim_i$ platí: $p \sim_i q$

$p \sim_i q$ iff $\forall u, |u| \leq i, \delta^*(p,u) \in F \iff \delta^*(q,u) \in F$

Chceme $\sim_{i+1}$

a) $\underline{p \sim_{i+1} q}$, $|v| \leq i+1$ – if $|v| \leq i$, tak víme, že $\delta^*(p,v) \in F \iff \delta^*(q,v) \in F$ ind. před.

když je $|v| = i+1$ $\quad \delta^*(p, \overset{v}{\widetilde{au}}) = \delta^*(\delta(p,a), \underline{\underline{u}}) \in F$

$v = au \quad |u| = i, a \in \Sigma$ $\qquad \Updownarrow$ ind. p.

$$\delta^*(q, \overset{v}{\widetilde{au}}) = \delta^*(\delta(q,a), \underline{\underline{u}}) \in F$$

Z def. $\sim_{i+1}$ víme, že $\delta(p,a) \sim_i \delta(q,a)$

b) Víme, že každé slovo u délky $\leq i+1$ splňuje

$$\delta^*(p,u) \in F \iff \delta^*(q,u) \in F$$

$u = a w$ $\quad \delta^*(\delta(p,a), w) \iff \delta^*(\delta(q,a), w)$

$|w| \leq i$ $\quad \underline{\delta(p,a) \sim_i \delta(q,a)}$ ind. předp.

$\underline{p \sim_i q}$ protože každé slovo $|x| \leq i$

je také $|x| \leq i+1$

a z definice $\sim_{i+1}$ víme $p \sim_{i+1} q$ $\quad$ cbd

Poznámka. Ekvivalence $\sim_i$ v praxi konstruujeme tak, že konstruujeme odpovídající rozklady R_i množiny stavů Q na třídy ekvivalence $\sim_i$.

Algoritmus redukce.

Je dán DFA $M = (Q, \Sigma, \delta, q_0, F)$.

1. Zkonstruujeme množinu Q' všech dosažitelných stavů automatu M.
2. Zkonstruujeme rozklady R_i ekvivalencí $\sim_i$ proM'. Končíme když $\sim_i \; = \; \sim_{i+1}$. Pak $\sim \; = \; \sim_i$.
3. Označme $[q]_\sim$ třídu ekvivalence $\sim$ obsahující stav q. Pak $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, kde
   - $Q_1 := \{[q]_\sim \mid q \in Q'\}$,
   - $q_1 := [q_0]_\sim$,
   - $\delta_1([q]_\sim, a) \; = \; [\delta(q, a)]_\sim$,
   - $F_1 \; = \; \{[q]_\sim \mid \text{ kde } F \cap [q]_\sim \neq \emptyset\}$.

Věta. Automat M i k němu redukovaný automat M_1 přijímají stejný jazyk, tj. jsou ekvivalentní.

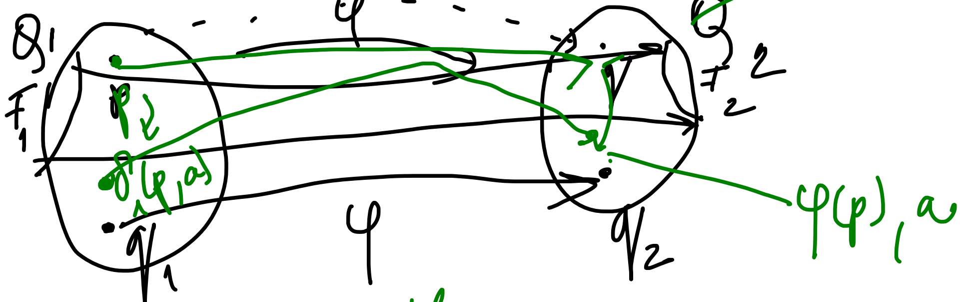
Tj. $L(M) = L(M_1)$

M_1 má jen dosažitelné stavy, nepřísluší jsou o žádné slovo u, pro které $\delta^*(q_0, u) \in F$.

M_1, M_2 jsou ekvivalentní iff jejich redukované automaty se liší pouze v „pojmenování stavů“.

Isomorfní automaty. DFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ a $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ jsou *isomorfní*, jestliže existuje bijekce $\varphi\colon Q_1 \to Q_2$ že

- $\varphi(q_1) = q_2$, ff
- $\varphi(F_1) = F_2$,
- $\varphi(\delta_1(q, a)) = \delta_2(\varphi(q), a)$ pro každé $q \in Q_1$ a $a \in \Sigma$. — $(\varphi(p), a)$



$\varphi(\delta_1(p, a)) = \delta_2(\varphi(p), a)$

$$\begin{array}{ccc} p & \xrightarrow{\varphi} & \varphi(p) \\ \downarrow \delta_1 & & \downarrow \delta_2 \\ \delta_1(p,a) & \xrightarrow{\varphi} & \delta_2(\varphi(p), a) \end{array}$$

Věta. DFA M_1 a M_2 přijímají stejný jazyk (tj. jsou ekvivalentní) právě tehdy, když jejich odpovídající redukované automaty jsou isomorfní.

Zdůvodnění:

① když redukovaný k M_1 ... M_1^R

— " — k M_2 ... M_2^R

jsou isomorfní tak $L(M_1^R) = L(M_2^R)$

ale $L(M_1) = L(M_1^R)$; $L(M_2) = L(M_2^R)$

tj M_1 je ekvivalentní M_2.

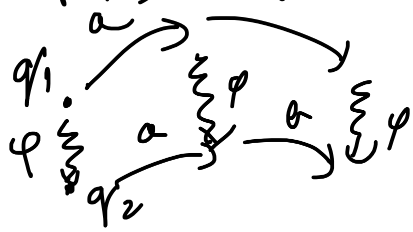
② $L(M_1) = L(M_2)$

$M_1^R = (Q_1, \Sigma, \delta_1, q_1, F_1)$ $M_2^R = (Q_2, \Sigma, \delta_2, q_2, F_2)$

chceme φ isomorfismus:

$\varphi(q_1) = \varphi(q_2)$ když $\delta_1(q_1, a) = p$, $\delta_2(q_2, a) = p$

$$\boxed{\varphi(\delta_1^*(q_1, u)) = \delta_2^*(q_2, u)}$$



φ je na — plyne z faktu, že nemáme nedosažitelné stavy

φ je dobře definované: k tomu potřebujeme, že žádné dva různé stavy ani M_1^R ani M_2^R nejsou ekvivalentní $\sim$.

Nedeterministické automaty – NFA

NFA se od DFA liší tím, že pro $q \in Q$ a $a \in \Sigma$, NFA může přejít do několika (také žádného) stavů.

Může mít více počátečních stavů.

NFA je $(Q, \Sigma, \delta, I, F)$, kde

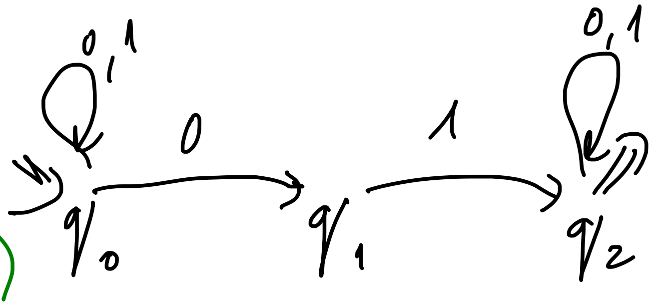
- Q je konečná neprázdná množina stavů,
- Σ je konečná neprázdná množina vstupů,
- δ je přechodová funkce, kde

$$\delta \colon Q \times \Sigma \to \mathcal{P}(Q),$$

  ($\mathcal{P}(Q)$ je množina všech podmnožin množiny stavů Q).
- $I \subseteq Q$ je množina počátečních stavů,
- $F \subseteq Q$ je množina koncových stavů.

Př: $L = \{u01v \mid u, v \in \{0,1\}^*\}$

M NFA



$\delta(\{q_0, q_1\}, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$

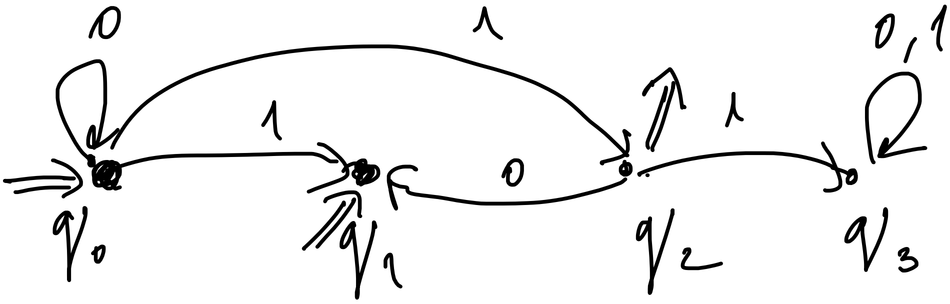
$\delta^*(\{q_0, q_1\}, 01) = \delta(\delta(\{q_0, q_1\}, 0), 1) = \delta(\{q_0, q_1\}, 1) = \{q_0, q_2\}$

Ukážeme, že $L(M) = L$

$$\delta(\{q_0, q_1\}, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$$

Stavový diagram NFA je orientovaný ohodnocený graf, kde

- vrcholy jsou stavy,
- z q vede hrana ohodnocená $a \in \Sigma$ do p iff $p \in \delta(q, a)$,
- jsou vyznačeny počáteční i koncové stavy.



$I = \{q_0, q_1\}$

$$\delta(q_0, 0) = \{q_0\} \quad \delta(q_0, 1) = \{q_1, q_2\} \quad \delta(q_3, a) = \{q_3\}$$
$a = 0, 1$

$$\delta(q_1, 0) = \delta(q_1, 1) = \emptyset$$
$$\delta(q_2, 0) = \{q_1\} \quad \delta(q_2, 1) = \{q_3\}$$

Značení: Pro množinu stavů $X \subseteq Q$ a $a \in \Sigma$ položíme

$$\delta(X, a) = \bigcup\{\delta(q, a) \mid q \in X\}.$$

Rozšířená přechodová funkce

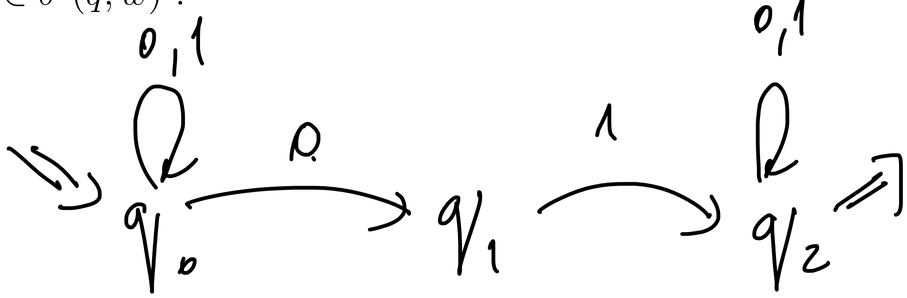
Dán NFA $(Q, \Sigma, \delta, I, F)$. *Rozšířená přechodová funkce* $\delta^\star\colon Q \times \Sigma^\star \to \mathcal{P}(Q)$ je definována:

1. $\delta^\star(q, \varepsilon) = \{q\}$, pro všechna $q \in Q$,
2. $\delta^\star(q, ua) = \delta(\underbrace{\delta^\star(q, u)}_{X}, a)$ pro všechna $q \in Q$, $a \in \Sigma$, $u \in \Sigma^\star$.

$$\bigcup_{a \in \delta^*(q,u)} \delta(q, a) = \delta^*(q, ua)$$

Pozorování. Pro NFA $M = (Q, \Sigma, \delta, I, F)$, stav q a každé slovo $w \in \Sigma^\star$ je ekvivalentní:

1. existuje orientovaný sled ve stavovém diagramu z q do p označený slovem w.
2. $p \in \delta^\star(q, w)$.



$p \in \delta^*(q, w)$ iff existuje sled ve stavovém diagramu z q do p označený w

$\delta^*(q_0, 01) = \{q_0, q_2\}$

$\delta^*(q_0, 010) = \{q_0, q_1, q_2\}$

Slovo přijímané NFA, jazyk přijímaný NFA.

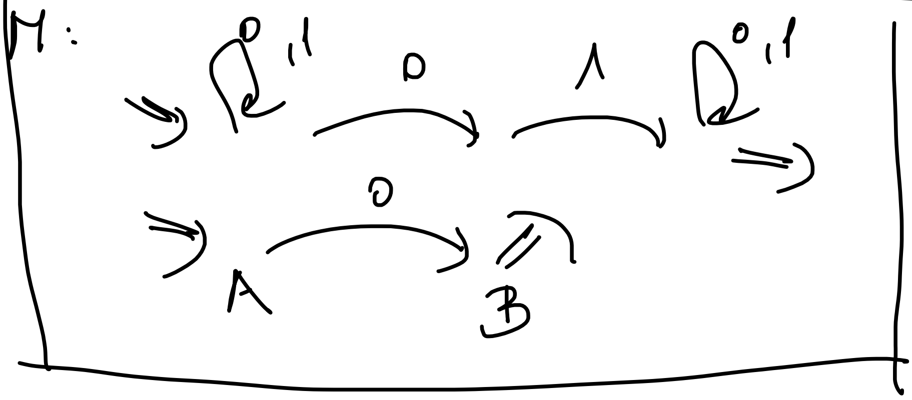
NFA $M = (Q, \Sigma, \delta, I, F)$ *přijímá* slovo $w \in \Sigma^\star$, iff pro nějaký z $q_0 \in I$ platí $\delta^\star(q_0, w) \cap F \neq \emptyset$.

Jazyk $L(M)$ *přijímaný* NFA M se skládá z právě všech slov, které jsou přijímány M.

$\exists q_0 \in I\ \delta^*(q_0, u) \cap F \neq \emptyset$ iff $u \in L(M)$

$u \in L(M)$ iff ex. $q_0 \in I$ a existuje sled z q_0 ohodnocený u, který končí v $p \in F$.

M:



$L(M) =$
$= \{u01v \mid u, v \in \{0,1\}^*\}$
$\cup \{0\}$

Přednáška 19. října 2020

Čím jsme skončili minulou přednášku.

NFA $M = (Q, \Sigma, \delta, I, F)$

$I \subseteq Q \qquad \delta : Q \times \Sigma \to \mathcal{P}(Q)$

stavový diagram

$\delta^*(q, w)$

$w \in L(M)$ iff $\delta^*(p, w) \cap F \neq \emptyset$ pro nějaký $p \in I$.

Tvrzení. Ke každému nedeterministickému automatu M existuje deterministický automat $\widehat{M}$, který přijímá stejný jazyk; tj.

$$L(M) = L(\widehat{M}).$$

Pozn.: Je-li M DFA, pak ho můžeme „považovat" za NFA

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

$I = \{q_0\}$ když $\delta(q, a) = p \iff \delta'(q, a) = \{p\}$

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

Opačnou implikaci; je-li $L = L(M)$, M je NFA, pak existuje DFA $\hat{M}$, že $L = L(\hat{M})$

Podmnožinová konstrukce

NFA

Podmnožinová konstrukce. Máme $M = (Q, \Sigma, \delta, I, F)$. Definujeme DFA $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, q_0, \widehat{F})$ takto:

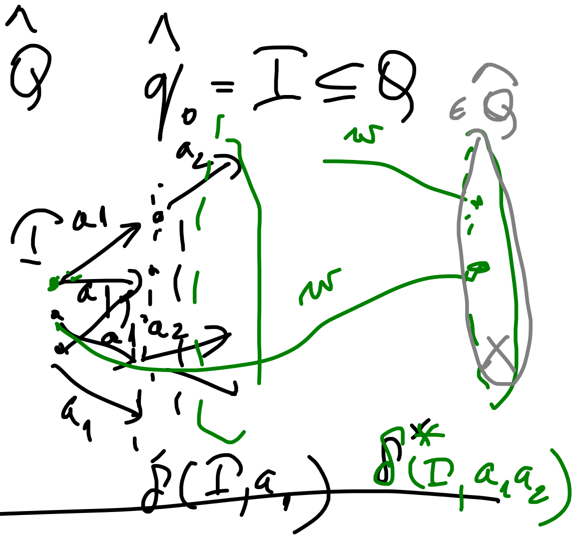
- $\widehat{Q}$ je množina všech podmnožin množiny;
- $q_0 = I$;
- $\widehat{F}$ se skládá z těch podmnožin Q, které obsahují aspoň jeden koncový stav M;
- $\widehat{\delta}(X, a)$ je množina těch stavů, do kterých vede hrana označená a z některého stavu množiny X.

$$\hat{Q} = \{X \mid X \subseteq Q\} \qquad \hat{q}_0 \in \hat{Q} \qquad \hat{q}_0 = I \subseteq Q$$

$$\hat{F} = \{X \mid X \subseteq Q,\ X \cap F \neq \emptyset\}$$

$$\underbrace{\hat{\delta}(X, a) = \bigcup_{q \in X} \delta(q, a)}$$

$$\hat{M} = (\hat{Q}, I, \hat{\delta}, \hat{q}_0, \hat{F})$$



Platí $L(M) = L(\hat{M})$

$w \in L(M)$ iff existuje $p \in I$ takový, že existuje sled ohodnocený w z p do $q \in F$; $w = a_1 \dots a_k$

$$\to p \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \cdots q_{k-1} \xrightarrow{a_k} q = q_k \in F$$

iff $\hat{\delta}^*(I, w) \in \hat{F}$ iff $\hat{\delta}^*(I, w) \cap F \neq \emptyset$

Ano, $w \in L(M)$ iff $w \in L(\hat{M})$

M NFA $\qquad \hat{M}$ DFA vzniklý podmnožinovou konstrukcí

cbd

Když $|Q| = n \quad |\hat{Q}| = 2^n$

Modifikace podmnožinové konstrukce Hledáme jen dosažitelné stavy.

$I = \hat{q}_0 \quad \hat{\delta}(\hat{q}_0, a) \quad a \in \Sigma$

1. $\widetilde{Q} := \{I\}$; $A := \{I\}$;
2. `if` $A \neq \emptyset$ `do` $B := \emptyset$;
   - `for all` $X \in A$ `do`
     - `for all` $a \in \Sigma$ `do` $\widehat{\delta}(X, a) := \delta(X, a)$;
       - `if` $\delta(X, a) \notin \widetilde{Q}$ `then` $B := B \cup \{\delta(X, a)\}$;
3. `if` $B \neq \emptyset$ `do` $\widetilde{Q} := \widetilde{Q} \cup B$; $A := B$ `go to 2`
4. $\widehat{q_0} := I$; $\widehat{Q} := \widetilde{Q}$; $\widehat{F} := \{X \in \widehat{Q} \mid X \cap F \neq \emptyset\}$;
5. `return` $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{q_0}, \widehat{\delta}, \widehat{F})$

Věta. Jazyk L je přijímán nějakým NFA právě tehdy, když existuje DFA M_1 takový, že

$$L = L(M_1).$$

D: 1) Je-li L přijímán DFA, pak je přijímán NFA

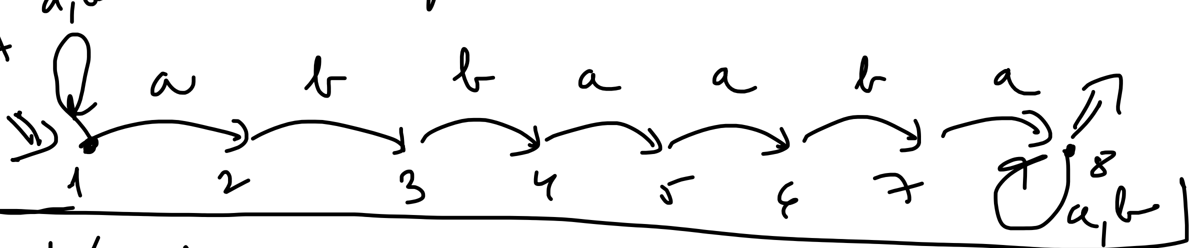
$$\delta'(p, a) = \{\delta(p,a)\}$$

2) Je-li L přijímán NFA M, pak existuje DFA (z podmnožinové konstrukce) $\hat{M}$, že $L(\hat{M}) = L(M) = L$.

Důsledek: NFA přijímají pouze regulární jazyky.

Př: Navrhněte DFA přijímající jazyk všech slov obsahujících podslovo abbaaba

Řeš.: NFA



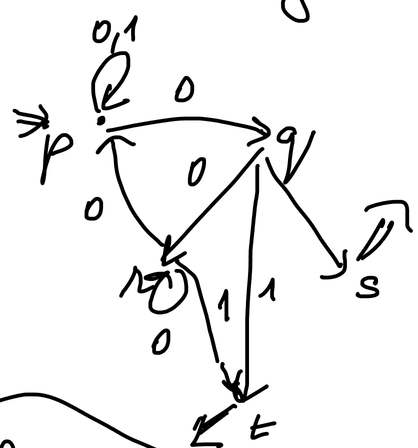
pro každé slovo $w = u\,abbaaba\,v$, $u, v \in \Sigma^*$, existuje sled z 1 do 8 označený w.

a když w neobsahuje abbaaba, tak se z 1 do 8 nedostaneme

Příklad. Je dán NFA tabulkou

		0	1
$\rightarrow$	p	$\{p,q\}$	$\{p\}$
	q	$\{r,s\}$	$\{t\}$
	r	$\{p,r\}$	$\{t\}$
$\leftarrow$	s	$\emptyset$	$\emptyset$
$\leftarrow$	t	$\emptyset$	$\emptyset$

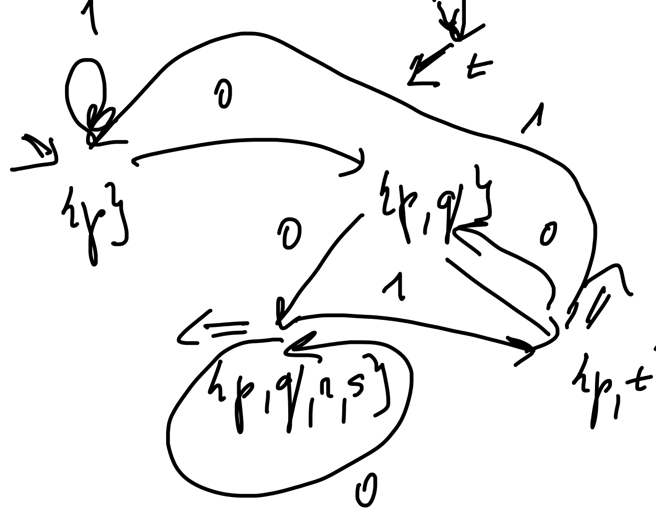
Stav. diagram



Najděte DFA $\widehat{M}$, který přijímá stejný jazyk.

Řeš:

	0	1
$\Rightarrow \{p\}$	$\{p,q\}$	$\{p\}$
$\{p,q\}$	$\{p,q,r,s\}$	$\{p,t\}$
$\Leftarrow \{p,q,r,s\}$	$\{p,q,r,s\}$	$\{p,t\}$
$\Leftarrow \{p,t\}$	$\{p,q\}$	$\{p\}$



Zredukujme

konvence: vynecháme množinové závorky

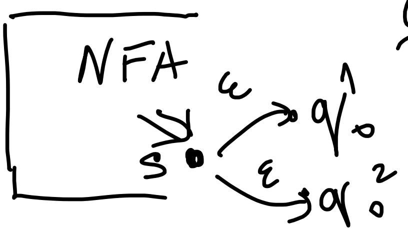
	0	1	$\sim_0$	0	1	$\sim_1$	0	1	$\sim_2 = \sim_1$
$\Rightarrow p$	p,q	p	O	O	O	S	A	S	protože
p,q	p,q,r,s	p,t	O	K	K	A	K	L	$\sim_1$ je
$\Leftarrow p,q,r,s$	p,q,r,s	p,t	K	K	K	K	K	L	identická,
$\Leftarrow p,t$	p,q	p	K	O	O	L	A	S	

automat byl redukovaný

Nedeterministický automat s ε přechody (ve zkratce ε-NFA)
ε-NFA je pětice $M = (Q, \Sigma, \delta, I, F)$, kde Q, Σ, I a F mají stejný význam jako u nedeterministického automatu, a přechodová funkce δ je zobrazení

$$\delta\colon Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q).$$

Rozdíl proti NFA: $\delta(q, \varepsilon)$ může být neprázdné. Jakoby M mohl pracovat, aniž by četl něco na vstupu.

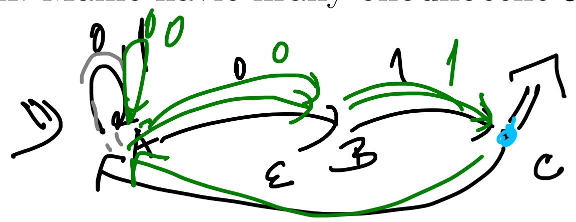


$I = \{q_0^1, q_0^2\}$.

s nový poč. stav

Stavový diagram. Máme navíc hrany ohodnocené ε.

Př.:



ε-NFA

$u = 010001$

1. sled končí v A

2. sled 0100 končí

přijímá $A \xrightarrow{0} B \xrightarrow{1} C$

3.

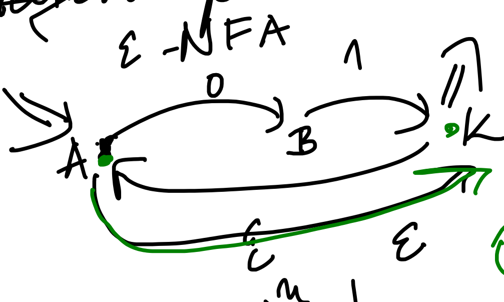
A je poč., C je konc.

$\delta(C, \varepsilon) = A$

~~Pro ε-NFA je složitější definovat δ^* rozšířenou přech.~~

Př.:

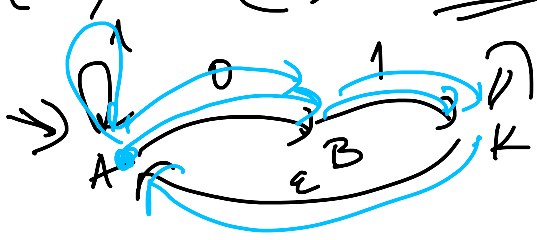
M: ε-NFA



011

$L(M) = \{(01)^n \mid n \geq 1\}$

Př.:



01101 by bylo přijato

Jazyk přijímaný ε-NFA.

$w \in L(M)$ iff existuje sled z nějakého počátečního stavu p označený w do nějakého $q \in F$

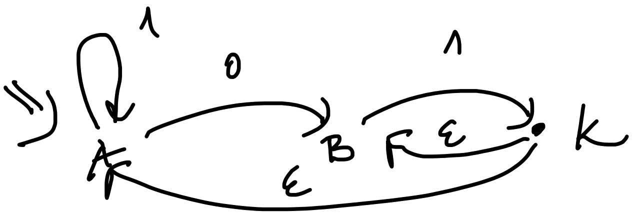
$w = a_1 a_2 \dots a_k$

$p \cdot \xrightarrow{\varepsilon} \cdot \xrightarrow{a_1} \cdot \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} \cdot \xrightarrow{a_2} \cdots \to \cdots \xrightarrow{a_k} \cdot \xrightarrow{\varepsilon} \cdots \xrightarrow{\varepsilon} q$

Dán NFA s ε přechody $M = (Q, \Sigma, \delta, I, F)$. Pro množinu stavů X definujeme ε *uzávěr* ε-UZ(X) indukcí takto:

- $X \subseteq \varepsilon$-UZ(X);
- je-li $p \in \varepsilon$-UZ(X), pak $\delta(p, \varepsilon) \subseteq \varepsilon$-UZ$(X)$.

Má-li množina X jediný stav q, píšeme ε-UZ(q) místo ε-UZ$(\{q\})$.



$X = \{A, K\} \subseteq \{A, B, K\}$

ε-UZ$(\{A, K\}) = \{A, \underline{K}, B\}$ – je množina všech stavů, do kterých vede z $q \in X$ sled označený ε.

$X = \{q\}$ tak píšeme ε-UZ(q) místo ε-UZ$(\{q\})$

Rozšířená přechodová funkce $\delta^\star$ je definována indukcí takto:

- $\delta^\star(q,\varepsilon) = \varepsilon\text{-UZ}(q)$;
- $\delta^\star(q,ua) = \bigcup\{\varepsilon\text{-UZ}(\delta(p,a)) \mid p \in \delta^\star(q,u)\}$, pro $a \in \Sigma$, $u \in \Sigma^\star$.

ε-NFA

$$\delta^*(q,\varepsilon) = \varepsilon\text{-UZ}(q)$$

$$\delta^*(q,ua) = \bigcup_{p\in\delta^*(q,u)} \varepsilon\text{-UZ}(\delta(p,a))$$

NFA

$$\delta^*(q,\varepsilon) = \{q\}$$

$$\delta^*(q,ua) = \delta(\delta^*(q,u),a) = \bigcup_{p\in\delta^*(q,u)} \delta(p,a)$$

Jazyk přijímaný ε-NFA.

Slovo u je *přijímáno* ε*-NFA*, jestliže existuje $q \in I$ a $p \in F$ tak, že $p \in \delta^\star(q, u)$. *Jazyk* $L(M)$ *přijímaný* ε*-NFA* M je množina všech slov přijímaných M.

$u \in L(M)$ iff existuje $q \in I$ a $p \in F$ tak, že $p \in \delta^*(q, u)$

iff existuje $q \in I$, $p \in F$ a sled z q do p označený u.

$$L(M) = \{u \mid \delta^*(I, u) \cap F \neq \emptyset\}$$

Je-li L regulární je přijímán ε-NFA

D.: $L = L(M)$ M DFA

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

stavové diag. M i M' jsou stejné

ε-NFA: $M' = (Q, \Sigma, \delta', \{q_0\}, F)$

$\delta'(q, a) = \{\delta(q, a)\}$ $\delta(q, \varepsilon) = \emptyset$

pro vš. $q \in Q$, $a \in \Sigma$

Důsledek: L je přijímán ε-NFA M iff je přijímán DFA

NFA je zvláštní případ ε-NFA, neboť

$\delta(q, \varepsilon) = \emptyset \quad \forall q \in Q$.

Věta. Jazyk přijímaný libovolným ε-NFA je regulární. ε-NFA $(Q, \Sigma, \delta, I, F)$

D.: Modifikací podmnožinové konstrukce.

$$\hat{M} = (\hat{Q}, \Sigma, \hat{\delta}, \hat{I}, \hat{F})$$

kde $\hat{Q} = \{X \mid X \subseteq Q, \varepsilon\text{-UZ}(X) = X\}$

ε-uzavřené množiny

$$\hat{I} = \varepsilon\text{-UZ}(I) \qquad \hat{F} = \{X \mid X \in \hat{Q}, X \cap F \neq \emptyset\}$$

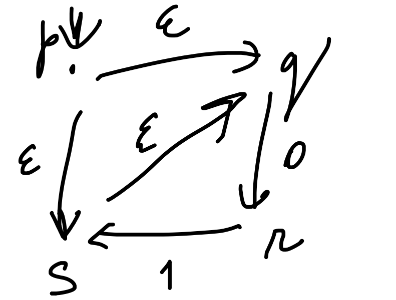
$$\hat{\delta}(X, a) = \varepsilon\text{-UZ}(\delta(X, a))$$

$\underline{X = \varepsilon\text{-UZ}(X)}$

$$= \varepsilon\text{-UZ}\Big(\bigcup_{q \in X} \delta(q, a)\Big)$$

Příklad. Dán ε-NFA tabulkou

	ε	0	1
$\rightarrow p$	$\{q,s\}$	$\emptyset$	$\emptyset$
q	$\emptyset$	$\{r\}$	$\emptyset$
r	$\emptyset$	$\emptyset$	$\{s\}$
$\leftarrow s$	$\{q\}$	$\emptyset$	$\emptyset$



Najděte DFA $\widehat{M}$, který přijímá stejný jazyk.

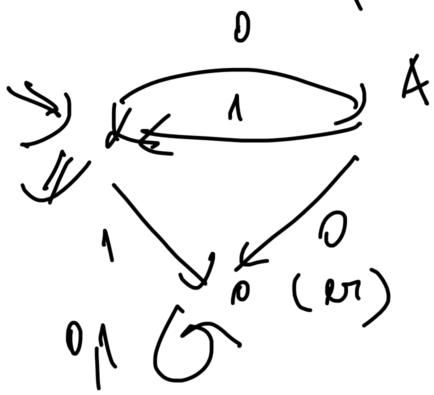
Řeš.:

$\varepsilon\text{-UZ}(p) = \{p, q, s\}$ $\qquad$ $\varepsilon\text{-UZ}(r) = \{r\}$

$\varepsilon\text{-UZ}(q) = \{q\}$ $\qquad$ $\varepsilon\text{-UZ}(s) = \{s, q\}$

	0	1	$\sim_0$	0	1	$\sim_1$	0	1	$\sim_2$
$\Rightarrow$ $\Leftarrow$ p,q,s	r	$\emptyset$	K	0	0	K	A	0	K
r	$\emptyset$	q,s	0	0	K	A	0	K	A
$\Leftarrow$ q,s	r	$\emptyset$	K	0	0	K	A	0	K
$\emptyset$	$\emptyset$	$\emptyset$	0	0	0	0	0	0	0

$\sim_2 = \sim_1$



$L = \{(01)^n \mid n \geq 0\}$

Operace s jazyky. Dány jazyky L_1 a L_2 nad stejnou abecedou. Pak

- $L_1 \cup L_2$ je jejich sjednocení,
- $L_1 \cap L_2$ je jejich průnik,
- $\overline{L_1} = \Sigma^\star - L_1$ je doplněk jazyka L_1.

- *Zřetězení* jazyků L_1 a L_2 je jazyk $L_1\,L_2$ definovaný

$$L_1\,L_2 = \{uv \mid u \in L_1, v \in L_2\}.$$

- *Kleeneho operace* $\star$*.* Je

$$L_1^\star = \{\varepsilon\} \cup L_1 \cup L_1^2 \cup L_1^3 \cup \ldots = \bigcup_{i=0}^{\infty} L_1^i.$$

kde $L_1^0 = \{\varepsilon\}$, $L_1^{i+1} = L_1^i \, L_1$ pro $i \geq 0$.

- *Reverze, též obrácení* $L_1^R = \{w^R \mid w \in L_1\}$.

Pro jazyky L_1 a L_2 obecně **neplatí**:

- $(L_1 \cup L_2)^\star = L_1^\star \cup L_2^\star$.
- $(L_1 \cap L_2)^\star = L_1^\star \cap L_2^\star$.
- $L_1^\star L_2^\star = (L_1 L_2)^\star$.

Věta. Třída regulárních jazyků je uzavřena na následující operace: 1) sjednocení, 2) doplněk, 3) průnik, 4) zřetězení, 5) Kleeneho operaci $\star$ a 6) reverzi.

Přesněji, jestliže L, L_1 a L_2 jsou regulární jazyky, pak také $L_1 \cup L_2$, $\overline{L}$, $L_1 \cap L_2$, $L_1 L_2$, $L^\star$ a L^R jsou regulární jazyky.

Součinová konstrukce Jsou dány DFA $M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$ a $M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$ takové, že $L_1 = L(M_1)$ a $L_2 = L(M_2)$. Definujme DFA $M = (Q, \Sigma, \delta, q_0, F)$ takto:

- $Q = Q_1 \times Q_2$,
- $q_0 = (q_0^1, q_0^2)$,
- $\delta((p,q),a) = (\delta_1(p,a), \delta_2(q,a))$,
- $F = F_1 \times F_2$.

Pak $L(M) = L_1 \cap L_2$.

Přednáška 26. října 2020

Čím jsme skončili minulou přednášku.

Podmnožinová konstrukce:

$$\text{NFA } M \rightsquigarrow \text{DFA } N$$

tak, že $L(M) = L(N)$.

ε-NFA : $M = (Q, \Sigma, \delta, I, F)$

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$$

stavový diagram ε-NFA

Modifikací podmnožinové konstrukce

ε-NFA $M \rightsquigarrow$ DFA N, že $L(M) = L(N)$

Operace s jazyky. Dány jazyky L_1 a L_2 nad stejnou abecedou Σ. Pak

- $L_1 \cup L_2$ je jejich sjednocení,
- $L_1 \cap L_2$ je jejich průnik,
- $\overline{L_1} = \Sigma^\star - L_1$ je doplněk jazyka L_1.

$u \in \overline{L_1}$ iff $u \notin L_1$

Př: $L_1 = \{a^n \mid n \geq 0\}$ $\quad L_2 = \{b^n \mid n \geq 0\}$

$\Sigma = \{a, b\}$

$L_1 \cup L_2 = \{a^n, b^n \mid n \geq 0\}$

$L_1 \cap L_2 = \{\varepsilon\}$ $\quad \overline{L_1} = \{u \mid |u|_b \geq 1, |u|_a \geq 1\}$

Operace vycházející z množinových operací

- *Zřetězení* jazyků L_1 a L_2 je jazyk $L_1 L_2$ definovaný

$$L_1 L_2 = \{uv \mid u \in L_1, v \in L_2\}.$$

Př:

$$L_1 = \{a^n \mid n \geq 0\} \qquad L_2 = \{b^n \mid n \geq 0\}$$

co je $L_1 L_2 = \{a^n b^m \mid n \geq 0, m \geq 0\}$

$$\boxed{\begin{aligned} \emptyset L_2 &= \emptyset \\ L_2 \emptyset &= \emptyset \end{aligned}}$$

protože nemáme slovo uv
$u \in \emptyset$ $v \in L_2$

- *Kleeneho operace* $\star$. Je (Kleeneho uzávěr)

$$L_1^\star = \{\varepsilon\} \cup L_1 \cup L_1^2 \cup L_1^3 \cup \ldots = \bigcup_{i=0}^{\infty} L_1^i.$$

kde $L_1^0 = \{\varepsilon\}$, $L_1^{i+1} = L_1^i\, L_1$ pro $i \geq 0$.

$\{\varepsilon\} L_1 = L_1$

mocnina jazyka $L_1^0 = \{\varepsilon\}$

$L_1^{i+1} = L_1^i L_1 = L_1 L_1^i$

$L_1^0, L_1^1 = L_1, L_1^2, \ldots$

$$L_1^{*} = \bigcup_{i=0}^{\infty} L_1^i$$

a) I kdyby $L_1 = \emptyset$, tak $L_1^* \neq \emptyset$ $L_1^* = \{\varepsilon\}$

b) $\varepsilon \in L^*$ pro každý jazyk L.

c) $\emptyset^* = \{\varepsilon\}$

d) Σ^* jsou všechna slova nad Σ, $\Sigma_1^0 = \{\varepsilon\}$, $\Sigma = \{a, b\}$

Σ_1^i jsou všechna slova délky i

- *Reverze, též obrácení* $L_1^R = \{w^R \,|\, w \in L_1\}$.

$w = a_1 \ldots a_k$ $w^R = a_k a_{k-1} \ldots a_1$

indukce $\varepsilon^R = \varepsilon$

$(au)^R = u^R a$

Př: $L = \{a^n b^m \mid n, m \geq 0\}$, pak

$L^R = \{b^m a^n \mid n, m \geq 0\}$

Pro jazyky L_1 a L_2 obecně **neplatí**:

- $(L_1 \cup L_2)^\star = L_1^\star \cup L_2^\star$.
- $(L_1 \cap L_2)^\star = L_1^\star \cap L_2^\star$.
- $L_1^\star L_2^\star = (L_1 L_2)^\star$.

Platí $L_1^* \cup L_2^* \subseteq (L_1 \cup L_2)^*$

Př: 1 $L_1 = \{a\}$ $L_2 = \{b\}$ $\Sigma = \{a, b\}$

$L_1 \cup L_2 = \{a, b\}$ $(L_1 \cup L_2)^* = \Sigma^*$

$L_1^* \cup L_2^* = \underbrace{\{a^n \mid n \geq 0\}}_{L_1^*} \cup \underbrace{\{b^m \mid m \geq 0\}}_{L_2^*}$

$L_3 = \{ba, a\}$ $L_4 = \{ab, a\}$

$L_3 \cap L_4 = \{a\}$ $(L_3 \cap L_4)^* = \{a^n \mid n \geq 0\}$

$aba \in L_3^* \cap L_4^*$ ano $\quad \underset{\in L_3^*}{aba} \quad \underset{\in L_4^*}{aba}$

Platí $L_1^* \cap L_2^* \supseteq (L_1 \cap L_2)^*$

Př. $L_1 = \{a\}$ $L_2 = \{b\}$

$$L_1^* L_2^* = \{a^n \mid n \geq 0\}\{b^m \mid m \geq 0\}$$
$$= \{a^n b^m \mid n, m \geq 0\}$$

$L_1 L_2 = \{ab\}$

$(L_1 L_2)^* = \{(ab)^n \mid n \geq 0\}$

Věta. Třída regulárních jazyků je uzavřena na následující operace: 1) sjednocení, 2) doplněk, 3) průnik, 4) zřetězení, 5) Kleeneho operaci $\star$ a 6) reverzi.

Přesněji, jestliže L, L_1 a L_2 jsou regulární jazyky, pak také $L_1 \cup L_2$, $\overline{L}$, $L_1 \cap L_2$, $L_1 L_2$, $L^\star$ a L^R jsou regulární jazyky.

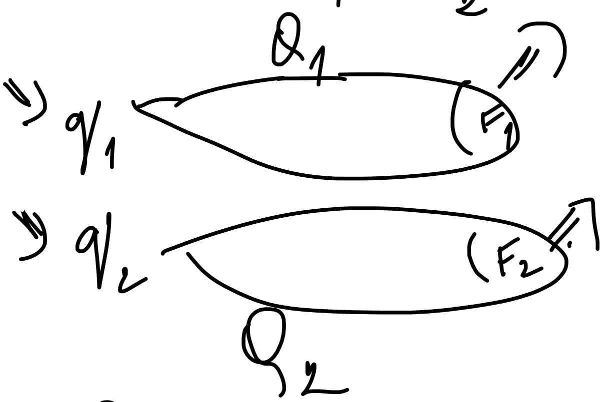
D.: L, L_1, L_2 jsou regulární

tj. DFA M, M_1, M_2 $\qquad L(M) = L$, $L(M_i) = L_i$

$M = (Q, \Sigma, \delta, q_0, F)$ $\qquad M_i = (Q_i, \Sigma, \delta_i, q_i, F_i)$

1) Pro sjednocení $L_1 \cup L_2$

N



NFA N

$\boxed{Q_1 \cap Q_2 = \emptyset}$

$Q_N = Q_1 \cup Q_2$ $\qquad \delta_N$ je sjednocení δ_1, δ_2

$I_N = \{q_1, q_2\}$ $\qquad F_N = F_1 \cup F_2$.

2) doplněk

$L = L(M)$

M DFA $\rightsquigarrow$ $\overline{M}$ DFA pro $\overline{L} = \Sigma^* - L$

v M prohodíme koncové a nekoncové stavy

$\overline{M} = (Q, \Sigma, \delta, q_0, \overline{F})$ $\qquad \overline{F} = Q \setminus F$

3) a) $L_1 \cap L_2 = \overline{\underbrace{\underbrace{\overline{L_1}}_{\text{reg.}} \cup \overline{L_2}}_{\text{reg.}}}$ $\qquad | \qquad \overline{\overline{L_1 \cap L_2}} = \overline{\overline{L_1} \cup \overline{L_2}}$

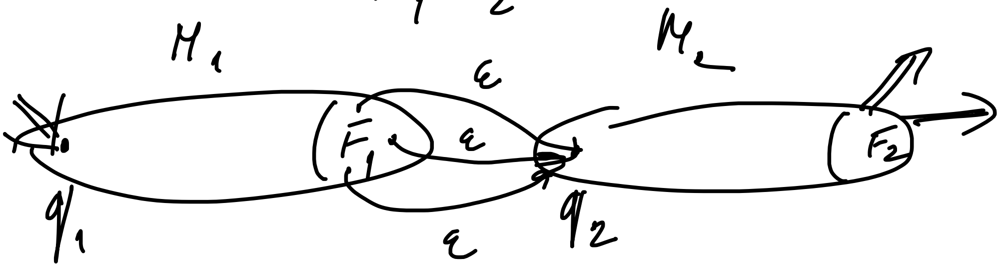
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \| $

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad L_1 \cap L_2$

+) nebo použijeme součinovou konstrukci na M_1, M_2

(4) Zřetězení: $L_1 L_2$

N



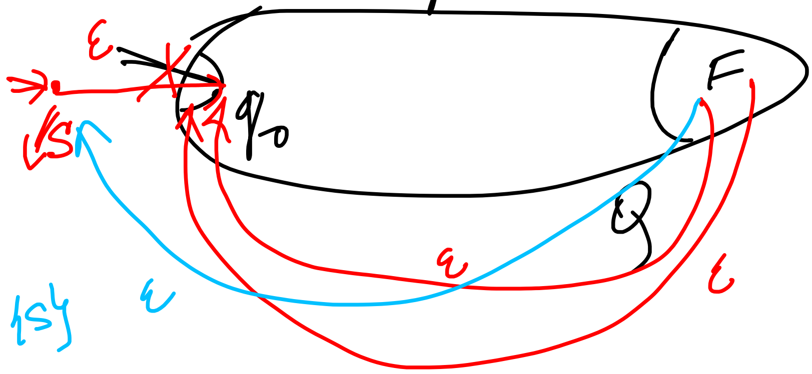
$Q_1 \cap Q_2 = \emptyset$

$Q_N = Q_1 \cup Q_2$ δ_N je sjednocení δ_1, δ_2

$+ \quad \delta(q, \varepsilon) = q_2 \ \forall q \in F_1$

$F_N = F_2$ $q_N^0 = q_1$

(5) Kleeneho operátor L^*



$F_N = \{s\}$

$Q_N = Q \cup \{s\}$

$q_N = s$

δ_N je δ +

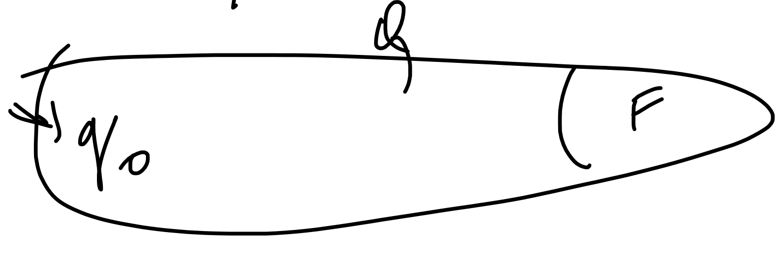
$\delta(s, \varepsilon) = q_0$,

$\delta(q, \varepsilon) = q_0 \ \forall q \in F$

$\boxed{F_N = F \cup \{s\}}$

(6) Reverze L^R

L M DFA



M^R všechny šipky ve stavovém diagramu otočíme M_R je NFA

Regulární výrazy

velice užitečný popis regulárních jazyků

Použití

- V počítačových jazycích lze rozlišit části (podřetězce), tzv. lexikální elementy, též tokeny, které jsou slova z regulárního jazyka. Jsou to např. zápisy konstant, zápisy identifikátorů, klíčová slova, relační znaménka apod. Syntaktický analyzátor (parser) obvykle obsahuje proceduru, lexikální analyzátor, který funguje jako konečný automat. Lexikální analyzátor při každém svém vyvolání přečte ze vstupního souboru nejdelší část, která tvoří jeden lexikální element a volajícímu programu vrátí jeho typ a hodnotu. Zbytek syntaktického analyzátoru může díky tomu být jednodušší, neboť pracuje už jen s lexikálními elementy.

- Vyhledávání výskytů slov daných regulárním výrazem

- Program grep opisuje na výstup ty řádky vstupního souboru, které obsahují podřetězec popsaný regulárním výrazem.

- Mnohé editory dovedou při operacích „vyhledej“, popř. „vyhledej a nahraď“ hledat podle daného regulárního výrazu.

- V Perl, AWK, Ruby je vyhledávání (a popř. náhrada) vzoru daného regulárním výrazem součástí jazyka. V mnoha dalších jazycích jsou k dispozici knihovny.

- Vyhledávání vzorů se také využívá v antivirech a v antispamových filtrech.

Definice. Je dána abeceda Σ. Množina všech **regulárních výrazů** nad Σ je definována induktivně takto:

- $\emptyset$ je regulární výraz,
- ε je regulární výraz,
- $\mathbf{a}$ je regulární výraz pro každé písmeno $a \in \Sigma$,
- jsou-li $\mathbf{r_1}$ a $\mathbf{r_2}$ regulární výrazy, pak $(\mathbf{r_1}\underline{+}\mathbf{r_2})$, $\mathbf{r_1}\mathbf{r_2}$ a $\mathbf{r_1^\star}$ jsou regulární výrazy.

Je to definice indukcí

1) $\underline{\emptyset}$ $\underline{\varepsilon}$ $\underline{a}$ $a \in \Sigma$,

2) $\dfrac{r_1 \quad , \quad r_2}{(r_1 + r_2)}$

$r_1 r_2$

r_1^*

$+ \;, \; {}^* \;, \; (\;) \;, \; \underline{\emptyset} \;, \; \underline{\varepsilon} \;, \; \underline{a}$

Př: a) $r_1 = \underline{a}^* + \underline{b}^*$ je reg. výraz $\quad a \in \Sigma, \; b \in \Sigma$

$L_{r_1} = L_{\underline{a}^*} \cup L_{\underline{b}^*} = (L_a)^* \cup (L_b)^* = \{a\}^* \cup \{b\}^*$

b) $\underline{a}((\underline{a}\underline{b}\underline{b})^* + \underline{a}^*)$

$r = \{a\}(\{abb\}^* \cup \{a\}^*) = \{a((abb)^k \cup a^\ell \mid k, \ell \geq 0\}$

není reg. výraz

$\underline{a} + +$ $\qquad$ $\underline{a}+$

Jazyk reprezentující regulární výraz. Každý regulární výraz **r** nad abecedou Σ reprezentuje jazyk $L_{\mathbf{r}}$ nad abecedou Σ:

nad Σ: $\quad \underline{r} \rightsquigarrow L_{\underline{r}}$ nad Σ!

$$\underline{\emptyset} \rightsquigarrow \emptyset$$

$$\underline{\varepsilon} \rightsquigarrow \{\varepsilon\}$$

$a \in \Sigma \quad \underline{a} \rightsquigarrow \{a\}$

---

když $\underline{r}_1 \rightsquigarrow L_{\underline{r}_1} \quad r_2 \rightsquigarrow L_{\underline{r}_2}$

tak

$$\underline{r}_1 + \underline{r}_2 \rightsquigarrow L_{\underline{r}_1} \cup L_{\underline{r}_2}$$

$$\underline{r}_1 \, \underline{r}_2 \rightsquigarrow L_{\underline{r}_1} L_{\underline{r}_2}$$

$$r_1^* \rightsquigarrow L_{r_1}^*$$

Věta. Každý jazyk reprezentovaný regulárním výrazem je regulární.

D. Stačí ukázat, že $L_{\underline{\emptyset}}$, $L_{\underline{\varepsilon}}$, $L_{\underline{a}}$ jsou regulární jazyky

[protože my víme, že sjednocení, zřetězení regulárních jazyků je regulární jazyk a také Kleeneho operátor reg. jazyka je regulární.

$L_{\underline{\emptyset}} = \emptyset$ NFA $\to q_0$ $\quad \to q_1$ a žádný přechod

$L_{\underline{\varepsilon}} = \{\varepsilon\}$ NFA $\to q_0$ a žádný přechod

$L_{\underline{a}} = \{a\}$ NFA $\to q_0 \xrightarrow{a} q_f$

Kleeneho věta. Každý jazyk přijímaný konečným automatem je možné reprezentovat regulárním výrazem.

Důsledek: L je regulární iff existuje reg. výraz $\underline{r}$, že $L_{\underline{r}} = L$.

Důkaz Kleeneho věty.

L je regulární $\rightsquigarrow$ reg. výraz.

tj. existuje DFA M, že $L = L(M)$.

$M = (Q, \Sigma, \delta, 1, F)$ $\quad Q = \{1, 2, \dots, n\}$

$q_0 = 1$

$X_{ij}^{(k)} = \{ w \mid \delta^*(i,w) = j$ a vnitřní stavy tohoto sledu jsou pouze z množiny $\{1,\dots,k\} \}$

$i \neq j$ $\quad X_{ij}^{(0)} = \{ w \mid \delta^*(i,w) = j$ a nemá vnitřní stavy $\}$

$$= \{ a \mid \delta(i,a) = j,\ a \in \Sigma \}$$

$i = j$ $\quad X_{ii}^{(0)} = \{\varepsilon\} \cup \{ a \mid \delta(i,a) = i \}$

$$X_{ij}^{(n)} = \{ w \mid \delta^*(i,w) = j \}$$

$$\boxed{w \in L(M) \text{ iff } w \in X_{1,l}^{(n)} \quad l \in F}$$

Když zvládneme popsat např. výrazem $r_{ij}^{(k)}$ **každou** množinu $X_{ij}^{(k)}$, $k \le n$, tak

$$L \rightsquigarrow r_{1,l_1}^{(n)} + \dots + r_{1,l_m}^{(n)}, \text{ kde } \{l_1, \dots, l_m\} = F$$

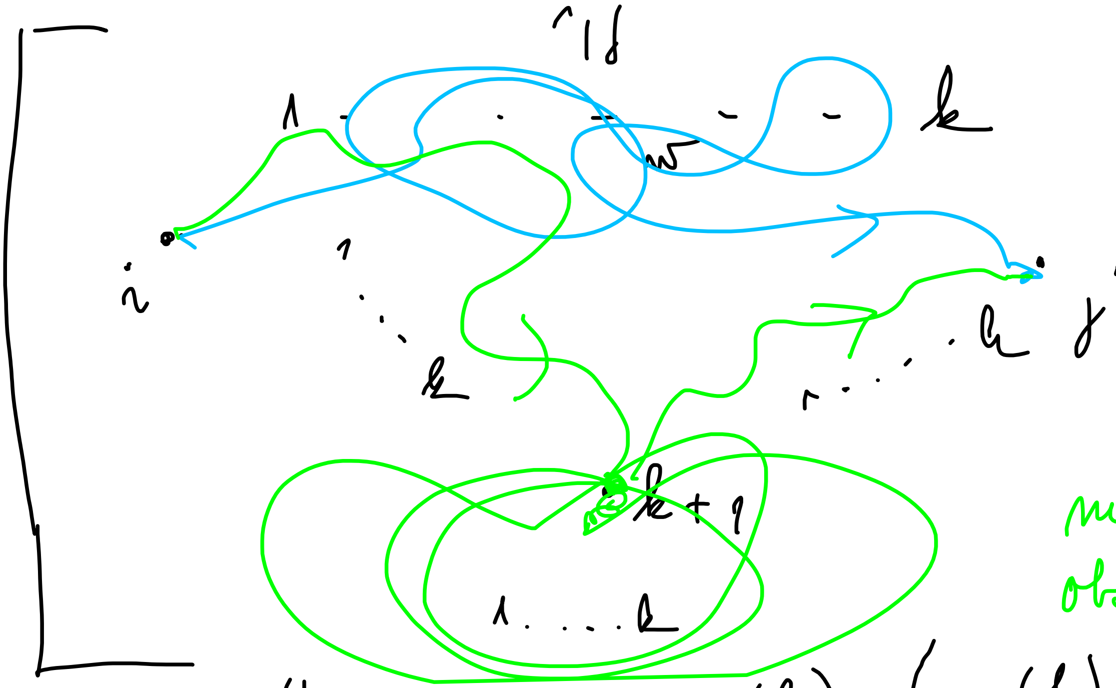
Musíme najít $r_{ij}^{(k)}$

$$X_{ii}^{(0)} \rightsquigarrow r_{ii}^{(0)} = \varepsilon + \sum_{a \in X_{ii}^{(0)}} a$$

$i \neq j$ $\quad X_{ij}^{(0)} \rightsquigarrow \quad r_{ij}^{(0)} = \sum_{a \in X_{ij}^{(0)}} a$

Už máme $r^{(k)}_{ij}$ $\qquad k < n$

a chceme $r^{(k+1)}_{ij}$



buď že vyhýbá k+1

nebo sled obsahuje k+1

$$\underbrace{X^{(k)}_{ij}}_{\text{vyhýbají } k+1} \cup X^{(k)}_{i,k+1}\left(X^{(k)}_{k+1,k+1}\right)^{*} X^{(k)}_{k+1,j}$$

$$\underline{r}^{(k+1)}_{ij} = \underbrace{\underline{r}^{(k)}_{ij}}_{\text{reg. v.}} + \underbrace{\underline{r}^{(k)}_{i,k+1}}_{\text{reg. v.}}\underbrace{\left(\underline{r}^{(k)}_{k+1,k+1}\right)^{*}}_{\text{reg. v.}}\underbrace{\underline{r}^{(k)}_{k+1,j}}_{\text{r.v.}}$$

$$L = \underline{r}^{(n)}_{1 l_1} + \dots + \underline{r}^{(n)}_{1 l_m}$$

$$F = \{l_1, \dots, l_m\}$$

cbd

Definice. Řekneme, že dva regulární výrazy $\mathbf{r}$ a $\mathbf{q}$ jsou *ekvivalentní*, jestliže jimi reprezentované jazyky jsou stejné. Fakt, že regulární výrazy $\mathbf{r}$ a $\mathbf{q}$ jsou ekvivalentní zapisujeme $\mathbf{r} \mathrel{\vDash\!\!\!\Dashv} \mathbf{q}$.

Některé ekvivalentní regulární výrazy. Jsou-li $\mathbf{r}$, $\mathbf{p}$ a $\mathbf{q}$ regulární výrazy, pak

1. $\mathbf{p} + \mathbf{q} \mathrel{\vDash\!\!\!\Dashv} \mathbf{q} + \mathbf{p}$,

2. $(\mathbf{p} + \mathbf{q})\mathbf{r} \mathrel{\vDash\!\!\!\Dashv} \mathbf{p}\,\mathbf{r} + \mathbf{q}\,\mathbf{r}$, $\;\mathbf{r}(\mathbf{p} + \mathbf{q}) \mathrel{\vDash\!\!\!\Dashv} \mathbf{r}\,\mathbf{p} + \mathbf{r}\,\mathbf{q}$,

3. $(\mathbf{r}^\star)^\star = \mathbf{r}^\star$, $\;(\varepsilon + \mathbf{r})\,(\varepsilon + \mathbf{r})^\star \mathrel{\vDash\!\!\!\Dashv} \mathbf{r}^\star$,

4. $(\mathbf{p}+\mathbf{q})^\star \mathrel{\models\!\!\!\dashv} (\mathbf{p}^\star\mathbf{q}^\star)^\star$,

5. $(\mathbf{p}+\mathbf{q})^\star \mathrel{\models\!\!\!\dashv} (\mathbf{p}^\star+\mathbf{q})^\star$,

6. $(\mathbf{p}+\mathbf{q})^\star \mathrel{\models\!\!\!\dashv} (\mathbf{p}^\star\mathbf{q})^\star\mathbf{p}^\star$,

7. $\mathbf{r}^\star \mathrel{\models\!\!\!\dashv} \varepsilon+\mathbf{r}\mathbf{r}^\star$,

8. $\mathbf{r}\mathbf{r}^\star \doteqdot \mathbf{r}^\star\mathbf{r}$,

9. $(\mathbf{p}\,\mathbf{q})^\star \doteqdot \varepsilon + \mathbf{p}\,(\mathbf{q}\,\mathbf{p})^\star\,\mathbf{q}$,

10. $(\mathbf{p}\,\mathbf{q})^\star\,\mathbf{p} \doteqdot \mathbf{p}\,(\mathbf{q}\,\mathbf{p})^\star$,

11. $\mathbf{r}\,\emptyset \doteqdot \emptyset \doteqdot \emptyset\,\mathbf{r}$.

Konstrukce konečného automatu k regulárnímu výrazu.

I. Pomocí důkazu Kleeneho věty.

II. Přímou metodou.

Příklad. Pro regulární výraz

$$\mathbf{r} = (\mathbf{a} + \mathbf{ab})^\star\mathbf{b}.$$

Metoda I.

Metoda II.

Regulární výraz reprezentující jazyk $L(M)$.

Příklad. Vytvořte regulární výraz, který reprezentuje jazyk přijímaný

	a	b
$\rightarrow 1$	1	2
$\leftarrow 2$	3	2
$\leftarrow 3$	1	3

Přednáška 2. listopadu 2020

Čím jsme skončili minulou přednášku.

regulární výrazy

1) $\underline{\emptyset}$, $\underline{\varepsilon}$, $\underline{a}$ $a \in \Sigma$! jsou reg. výrazy

2) Když $\underline{r}_1, \underline{r}_2$ jsou reg. výrazy, tak $(\underline{r}_1 + \underline{r}_2)$, $\underline{r}_1 \underline{r}_2$, $\underline{r}_1^*$ jsou reg. výrazy

reg. výraz reprezentuje jazyk

$\underline{\emptyset} \rightsquigarrow L_{\underline{\emptyset}} = \emptyset$ $\underline{\varepsilon} \rightsquigarrow L_{\underline{\varepsilon}} = \{\varepsilon\}$

$L_{\underline{a}} = \{a\}$, $L_{\underline{r}_1 + \underline{r}_2} = L_{\underline{r}_1} \cup L_{\underline{r}_2}$, $L_{\underline{r}_1 \underline{r}_2} = L_{\underline{r}_1} L_{\underline{r}_2}$, $L_{\underline{r}_1^*} = (L_{\underline{r}_1})^*$

Definice. Řekneme, že dva regulární výrazy $\mathbf{r}$ a $\mathbf{q}$ jsou *ekvivalentní*, jestliže jimi reprezentované jazyky jsou stejné. Fakt, že regulární výrazy $\mathbf{r}$ a $\mathbf{q}$ jsou ekvivalentní zapisujeme $\mathbf{r} \models\!\!\!\dashv \mathbf{q}$.

$$\underline{r} \models\!\!\!\dashv \underline{q} \quad \text{iff} \quad L_{\underline{r}} = L_{\underline{q}}.$$

Některé ekvivalentní regulární výrazy. Jsou-li $\mathbf{r}$, $\mathbf{p}$ a $\mathbf{q}$ regulární výrazy, pak

1. $\mathbf{p} + \mathbf{q} \models\!\!\!\dashv \mathbf{q} + \mathbf{p}$,

ano, $L_{\underline{p}} \cup L_{\underline{q}} = L_{\underline{q}} \cup L_{\underline{p}}$

$$L_{\underline{p}+\underline{q}} = L_{\underline{q}+\underline{p}}$$

$pr \not\dashv\vdash rp \quad L_a = \{a\}, L_b = \{b\}$

$ab \not\dashv\vdash ba$

2. $(\mathbf{p}+\mathbf{q})\mathbf{r} \dashv\vdash \mathbf{p}\,\mathbf{r}+\mathbf{q}\,\mathbf{r}, \quad \mathbf{r}(\mathbf{p}+\mathbf{q}) \dashv\vdash \mathbf{r}\,\mathbf{p}+\mathbf{r}\,\mathbf{q},$

$$(L_p \cup L_q) L_r = L_{pr} \cup L_{qr}$$

$$w \in L_{\varepsilon\varepsilon + r\varepsilon + \varepsilon r + r r^*} \text{ iff } w = \varepsilon, \; w \in L_r, \; w \in \bigcup_{i=1}^{\infty} L_r^i \quad w \in L_{r^*}$$

3. $(\mathbf{r}^\star)^\star = \mathbf{r}^\star, \quad (\varepsilon+\mathbf{r})\,(\varepsilon+\mathbf{r})^\star \dashv\vdash \mathbf{r}^\star,$

$$L_{(r^*)^*} = \{ w_1 w_2 \dots w_l \mid l \geq 0, w_i \in L_{r^*} \}$$

$$= \{ \underbrace{x_1 \dots x_{l_1}}_{w_1} \underbrace{y_1 \dots y_{l_2}}_{w_2} \dots \mid x_i, y_j \dots \in L_r \}$$

$$L_{r^*} = \{ u_1 u_2 \dots u_k \mid k \geq 0, u_i \in L_r \}$$

4. $(\mathbf{p}+\mathbf{q})^\star \dashv\vdash (\mathbf{p}^\star\mathbf{q}^\star)^\star,$

$$L_{p+q} = L_{(p^* q^*)^*}$$

na cvičení jsme ukázali, že

$$(L_1 \cup L_2)^* = (L_1^* L_2^*)^*$$

5. $(\mathbf{p}+\mathbf{q})^\star \dashv\vdash (\mathbf{p}^\star+\mathbf{q})^\star,$

$$L_{(p^*+q)^*} = \Big(\underbrace{(L_p)^*}_{u_1 \dots u_k} \cup L_q \Big)^* \quad x \; u_i \in L_p \quad x \in L_q$$

$$L_{(p+q)^*} = (\underbrace{L_p}_{} \cup L_q)^*$$

$$u_1 u_2 u_3 \dots u_k x$$

6. $(\mathbf{p}+\mathbf{q})^\star \boxminus (\mathbf{p}^\star\mathbf{q})^\star\mathbf{p}^\star$,

4

$$(p+q)^* \dashv\vdash (p^* q^*)^*$$

$$(p^* q^*)^* \dashv\vdash (p^* q)^* p^*$$

$$(\underbrace{u_1 \dots u_k}_{L_{p^*}} \boxed{v_1} \underbrace{v_2 \dots v_l}_{L_{q^*}})$$ opazujmo

7. $\mathbf{r}^\star \boxminus \varepsilon + \mathbf{r}\mathbf{r}^\star$,

$$L_{rr^*} = \bigcup_{i=1}^{\infty} L_r^i$$

$$\{\varepsilon\} \cup \bigcup_{i=1}^{\infty} L_r^i = \sum_{i=1}^{\infty} L_r^i = L_{r^*}$$

8. $\mathbf{r}\mathbf{r}^\star \boxminus \mathbf{r}^\star\mathbf{r}$,

$$L_{rr^*} = L_r\left(\bigcup_{i=0}^{\infty} L_r^i\right) = \bigcup_{i=0} L_r L_r^i = \bigcup_{i=1}^{\infty} L_r^i$$

$$L_{r^*} L_r = \left(\bigcup_{i=0}^{\infty} L_r^i\right) L_r = \bigcup_{i=0}^{\infty} L_r^i L_r = \bigcup_{i=1}^{\infty} L_r^i$$

9. $(\mathbf{p}\,\mathbf{q})^\star \boxminus \varepsilon + \mathbf{p}\,(\mathbf{q}\,\mathbf{p})^\star\,\mathbf{q}$,

$$\varepsilon \in L_{(pq)^*} = (L_{pq})^* = (L_p L_q)^*$$

$$(L_p L_q)(L_p L_q) \dots (L_p L_q)$$

$$L_p\,(\overbrace{L_q L_p\, L_q L_p \dots L_q L_p}^{})\,L_q = L_p\,(L_q L_p)^* L_q$$

10. $(\mathbf{p}\,\mathbf{q})^\star\,\mathbf{p} \dashv\vDash \mathbf{p}\,(\mathbf{q}\,\mathbf{p})^\star$,

obdobně jako 9

$$(L_p L_q) \ldots (L_p L_q) L_p = L_p (L_q L_p) \ldots (L_q L_p)$$

11. $\mathbf{r}\,\emptyset \dashv\vDash \emptyset \dashv\vDash \emptyset\,\mathbf{r}$.

$$L_{\underline{r\emptyset}} = \emptyset = L_{\underline{r}}\, L_{\underline{\emptyset}} = L_{\underline{r}}\,\emptyset = \emptyset$$

Konstrukce konečného automatu k regulárnímu výrazu.

I. Pomocí důkazu Kleeneho věty.

$\underline{r}$ ... r rozdělíme na části, pro které umíme sestrojit ε-NFA reprezentující ji. Vzniklé části „spojíme" jako v důkazu uzavřenosti reg. jazyků na sjednocení, zřetězení a *

II. Přímou metodou.

vhodná spíše k naprogramování

očíslujeme písmena regulárního výrazu

zjistíme
1) čím může slovo začínat
2) která písmena po sobě mohou následovat
3) čím může slovo končit
4) je ε v $L_{\underline{r}}$ nebo ne

Vytvoří NFA.
$s = q_0$ je nový stav, ostatní stavy jsou očíslovaná písmena.
Koncové stavy jsou písmena z 3) s q_0 koncový iff $\varepsilon \in L$

z S vede hrana do všech písmen z 1,

když a_i je vyjmenováno v 1, tak

$$S \xrightarrow{a_i} a_i$$

další hrany: $a_i b_j$ vyjmenovaní v 2,

tak

$$a_i \xrightarrow{b \,=\, b_j} b_j$$

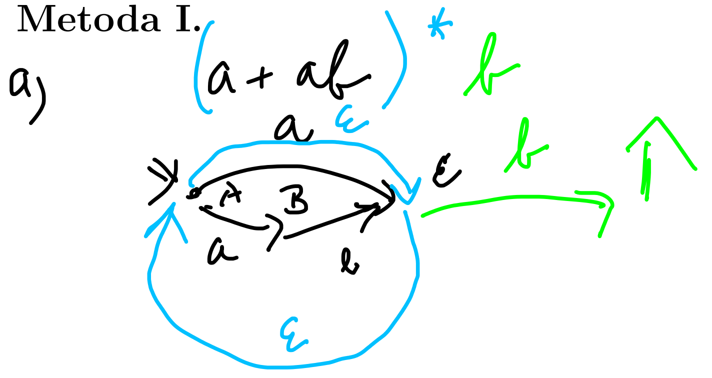
Takto získáme NFA s jedním počátečním stavem a udeláme podmnožinovou konstrukci. Redukujeme.

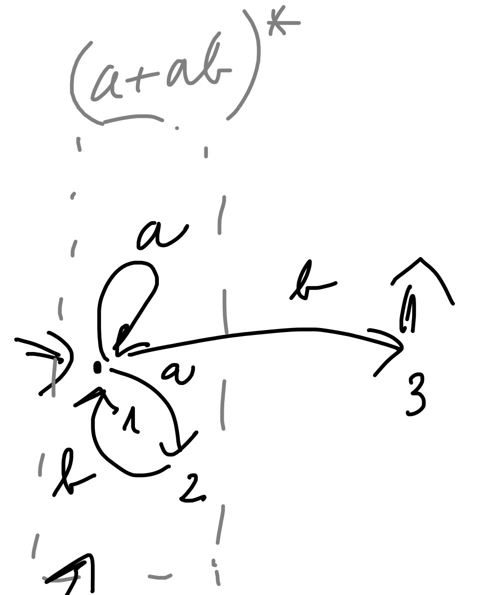
Příklad. Pro regulární výraz

$$\mathbf{r} = (\mathbf{a} + \mathbf{ab})^{\star}\mathbf{b}.$$

Metoda I.

a)

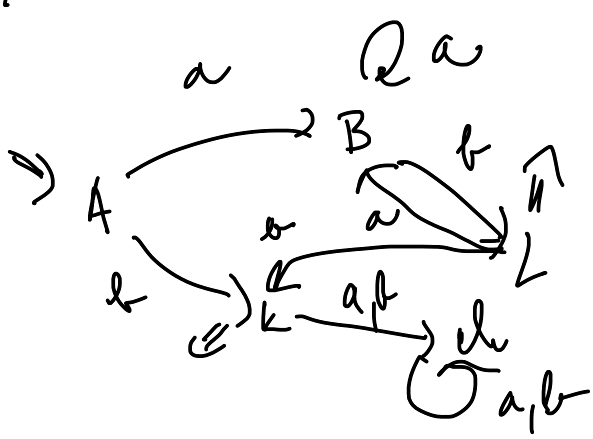




	a	b
$\to$ 1	1,2	3
2	$\emptyset$	1
$\leftarrow$ 3	$\emptyset$	$\emptyset$

NFA

	a	b	$\sim_0$	a	b	$\sim_1$	a	b	$\sim_2$	a	b
$\to$ 1	1,2	3	0	0	K	A	A	K	A	B	K
1,2	1,2	1,3	0	0	K	A	A	L	B	B	L
$\leftarrow$ 3	$\emptyset$	$\emptyset$	K	0	0	K	0	0	K	0	0
$\leftarrow$ 1,3	1,2	3	K	0	K	L	A	K	L	B	K
$\emptyset$	$\emptyset$	$\emptyset$	0	0	0	0	0	0	0	0	0



Metoda II.

$$\underline{r} = (a+ab)^* b$$

$$(a_1 + a_2 b_3)^* b_4$$

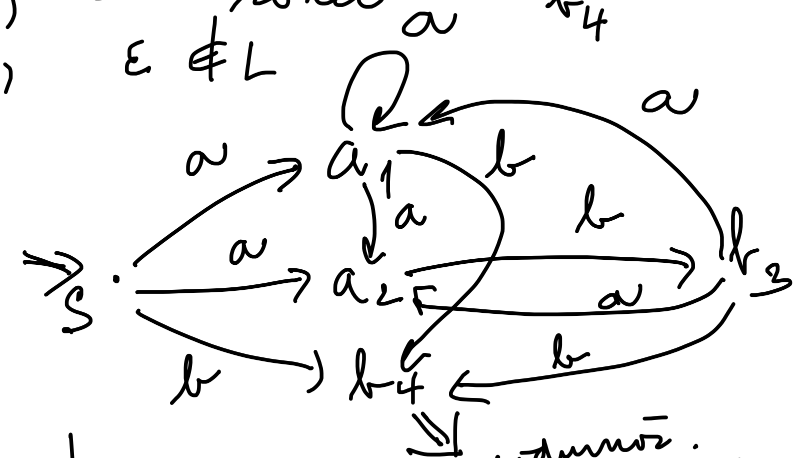
1) začíná a_1 nebo a_2 nebo b_4
2) po sobě mohou následovat

$a_1 a_1,\ a_1 a_2,\ a_1 b_4$

$a_2 b_3,\ b_3 a_1,\ b_3 a_2,\ b_3 b_4$

3) slovo končí b_4
4) $\varepsilon \notin L$

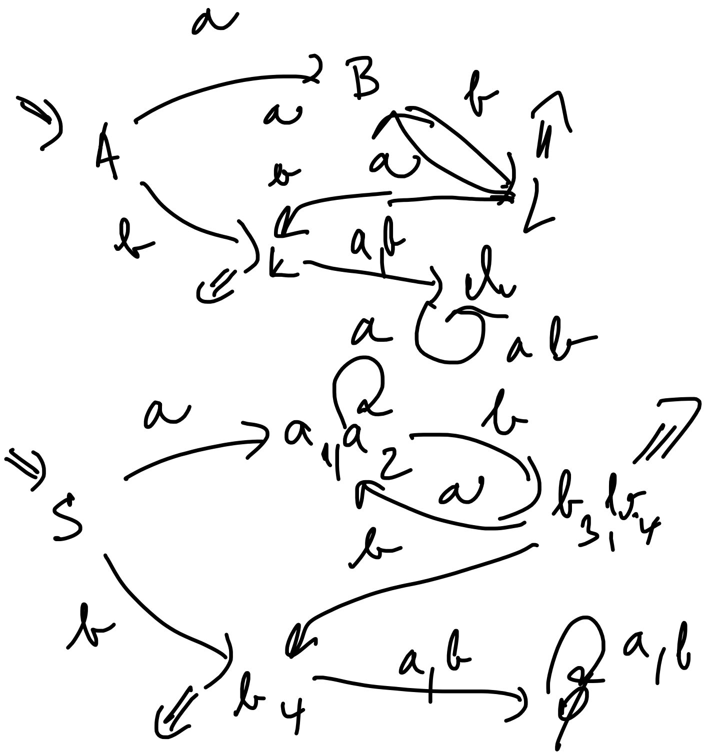
s není koncový, protože $\varepsilon \notin L$



	a	b
$\to s$	a_1, a_2	b_4
a_1	a_1, a_2	b_4
a_2	$\emptyset$	b_3
b_3	a_1, a_2	b_4
$\leftarrow b_4$	$\emptyset$	$\emptyset$

podmnož. konstr.

	a	b
$\Rightarrow s$	a_1, a_2	b_4
a_1, a_2	a_1, a_2	b_3, b_4
$\Leftarrow b_4$	$\emptyset$	$\emptyset$
$\Leftarrow b_3, b_4$	a_1, a_2	b_4
$\emptyset$	$\emptyset$	$\emptyset$



ано

Regulární výraz reprezentující jazyk $L(M)$.

Jak k automatu M vytvořit reg. výraz

Eliminační metoda

úpravy stavového diagramu

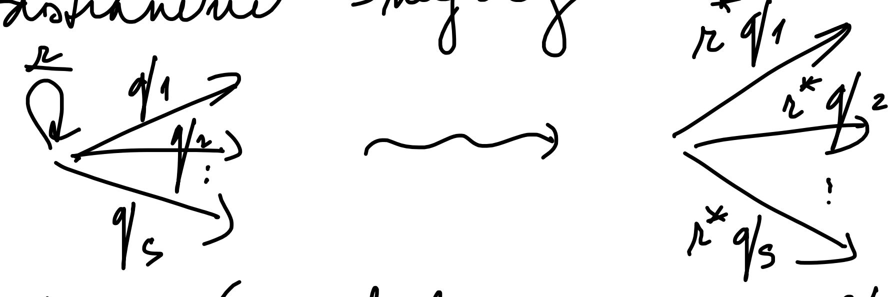
I. K stavovému diagramu přidáme nový „počáteční stav" s a nový stav f

z s vedeme hranu do počátečního stavu M ohodnocenou ε

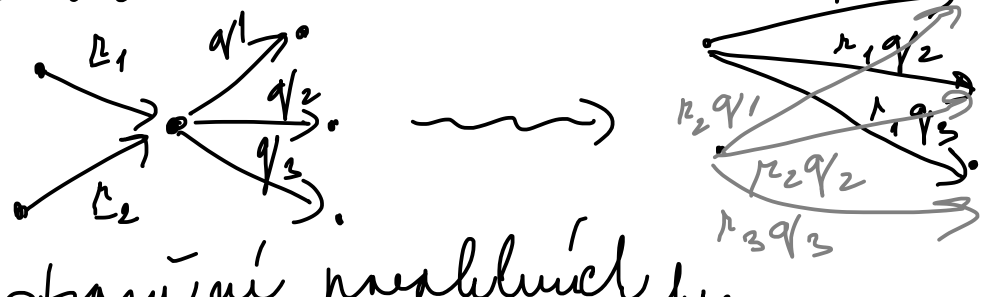
z každého koncového stavu M vedeme hranu do f ohodnocenou ε.

Úpravy:

odstranění smyčky



odstranění vrcholu

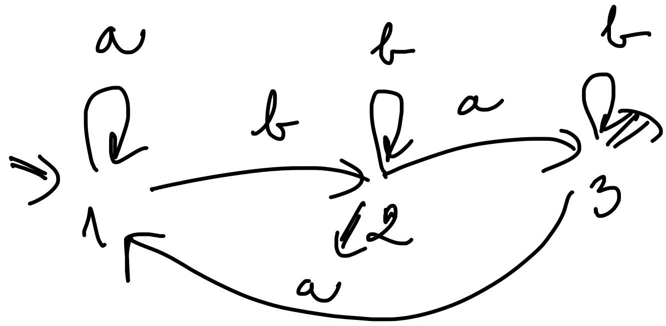


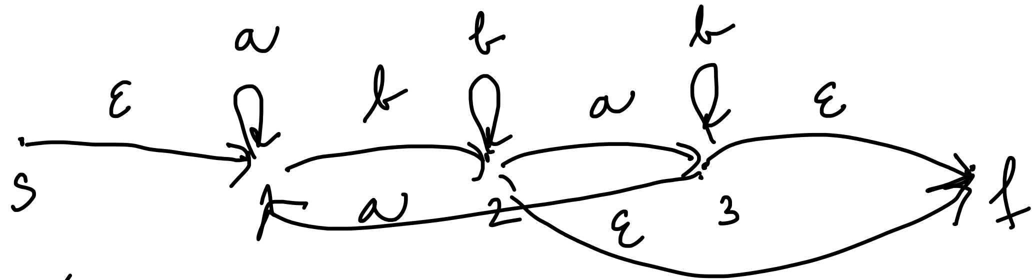
odstranění paralelních hran

úpravy děláme dokud $s \xrightarrow{r} f$ r odpovídá $L(M)$

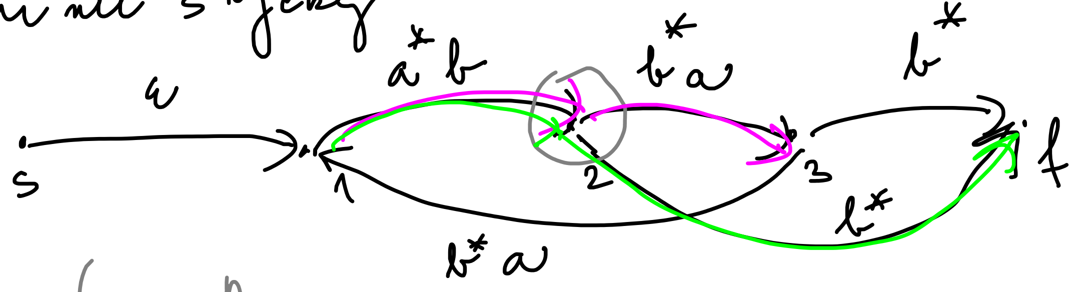
Příklad. Vytvořte regulární výraz, který reprezentuje jazyk přijímaný

	a	b
$\to 1$	1	2
$\leftarrow 2$	3	2
$\leftarrow 3$	1	3

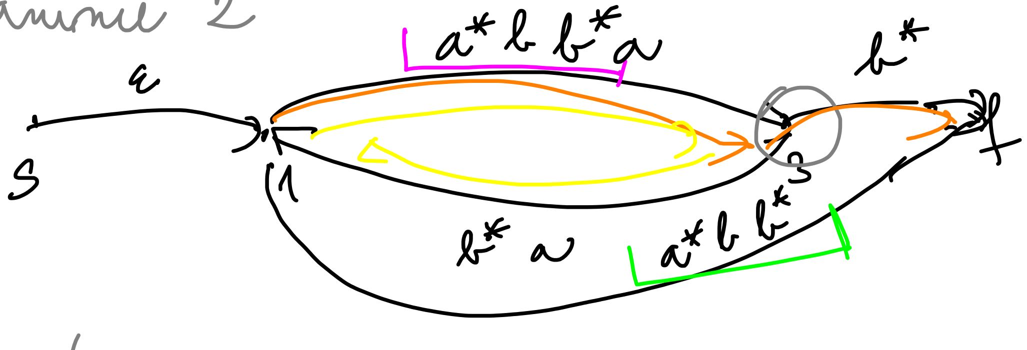




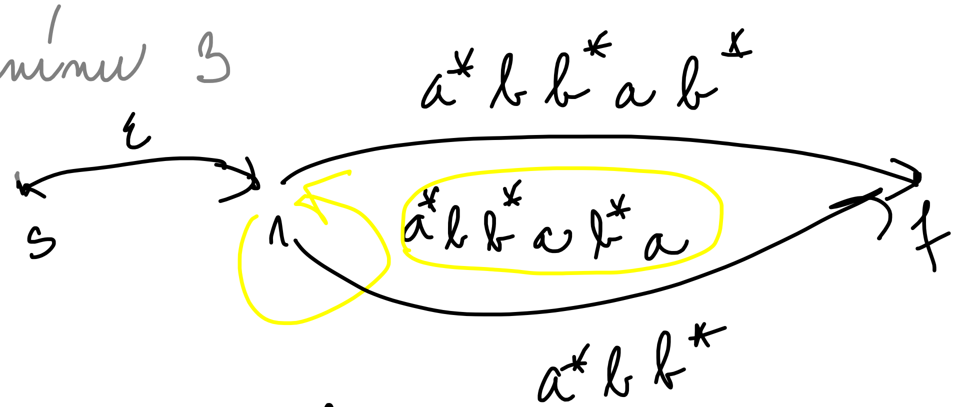
odstraníme smyčky



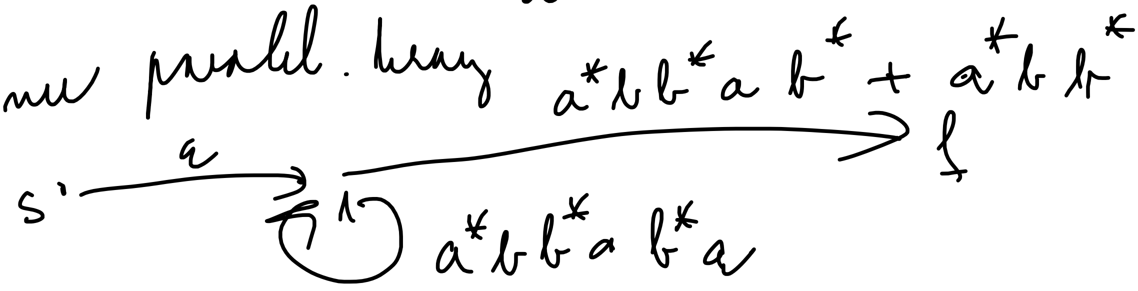
Odstraníme 2



Odstraníme 3



odstraníme parallel. hrany



odstraníme uzel

$$\begin{tikzcd} s' \arrow[r, "\varepsilon"] & 1 \arrow[r, "(a^*bb^*ab^*a)^*(a^*bb^*ab^* + a^*bb^*)"] & f \end{tikzcd}$$

$$(a^*bb^*ab^*a)^*(a^*bb^*ab^* + a^*bb^*)$$

$$\equiv (a^*bb^*ab^*a)^*a^*bb^*(ab^* + \varepsilon)$$

Další uzávěrové vlastnosti třídy regulárních jazyků

Víme, že třída reg. jazyků je uzavřená na sjednocení, doplněk, průnik, zřetězení, *, reverzi

Homomorfismus. Máme dvě abecedy Σ, Γ a zobrazení $h\colon \Sigma \to \Gamma^\star$. h rozšíříme na zobrazení $\Sigma^\star$ do $\Gamma^\star$ takto:

- $h(\varepsilon) = \varepsilon$,
- $h(u\,a) = h(u)\,h(a)$.

$h(a_1 \dots a_k) = h(a_1)\,h(a_2) \dots h(a_k)$

$(\Sigma^\star, \text{zř.}, \varepsilon)$ $\quad h(a) \in \Gamma^\star \quad a \in \Sigma$

Obraz jazyka L **v** h **je**

$$h(L) = \cancel{\bigcup} \{h(w) \mid w \in L\}.$$

$h(L) = \{h(w) \mid w \in L\}$

Př: $\Sigma = \{a, b\}$ $\quad \Gamma = \{0, 1\}$

$h(a) = 010 \quad h(b) = 0$

$h(aba) = \underbrace{010}_{h(a)}\underbrace{0}_{h(b)}\underbrace{010}_{h(a)}$

$L = \{a^n b \mid n \geq 0\}$ $\quad h(L) = \{(010)^n 0 \mid n \geq 0\}$

Substituce. Máme dvě abecedy Σ, Γ a zobrazení $\sigma\colon \Sigma \to \mathcal{P}(\Gamma^\star)$. σ rozšíříme na zobrazení $\Sigma^\star$ do $\mathcal{P}(\Gamma^\star)$ takto:

- $\sigma(\varepsilon) = \{\varepsilon\}$,
- $\sigma(u\,a) = \overline{\sigma(u)}\,\sigma(a)$.

$\sigma(a)$ je jazyk nad Γ

$u = a_1 \ldots a_k \qquad \sigma(u) = \sigma(a_1)\sigma(a_2)\ldots\sigma(a_k)$

Obraz jazyka L **v** σ je

$$\sigma(L) = \underbrace{\bigcup\{\sigma(w) \mid w \in L\}}.$$

Př $\quad \Sigma = \{a, b\} \qquad \Gamma = \{0, 1\}$

$\sigma(a) = \{0^n \mid n \geq 0\} \quad \sigma(b) = \{1^n \mid n \geq 0\}$

$\sigma(ab) = \{0^n 1^m \mid n, m \geq 0\}$

$\cancel{\sigma(ab) \neq \{0^n 1^n \mid n \geq 0\}}$

Věta. Je dána substituce σ z Σ^* do jazyků nad abecedou Γ. Jestliže každý jazyk $\sigma(a)$ je regulární a jestliže L je regulární jazyk nad abecedou $\cancel{\Gamma}$ Σ, pak jazyk $\sigma(L)$ je regulární jazyk nad abecedou Γ.

když L je regulární nad Σ

a $\sigma(a)$ je regulární nad Γ $\forall a \in \Sigma$

tak $\sigma(L)$ je regulární (nad Γ)

D.: $L = L_r$

r reg. výraz nad Σ

$\boxed{s}$ nad Γ

r každý výskyt $a \in \Sigma$ v r

nahradíme reg. výrazem pro $\sigma(a)$ r_a

s je reg. výraz pro $\sigma(L)$

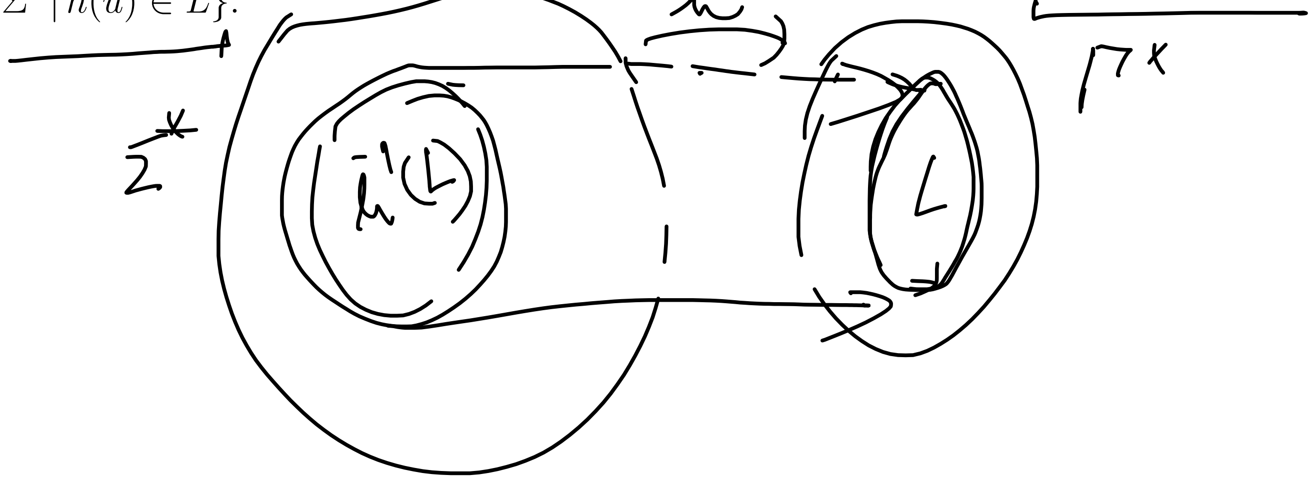
Důsledek. Je-li h homomorfismus a L regulární jazyk, pak jazyk $h(L)$ je také regulární jazyk.

$h(a) \longleftrightarrow \sigma_h(a) = \{h(a)\}$

σ_h je substituce a navíc každé

$\sigma_h(a)$ je reg. jazyk (je konečný)

Inverzní homomorfní obraz jazyka L v homomorfismu h je $h^{-1}(L) = \{u \in \Sigma^\star \,|\, h(u) \in L\}$.



Př. $h(a) = 01 \quad h(b) = 10$

nad $\{0,1\}$ $\quad L \rightsquigarrow (00+1)^*$

Co je $h^{-1}(L)$

$\rightsquigarrow (\underline{b}\,\underline{a})^*$ $\qquad \underbrace{10}_{h(b)}\underbrace{01}_{h(a)}$

Příklad. Uvažujme jazyk L nad abecedou $\Gamma = \{0, 1\}$ popsaný regulárním výrazem $(\mathbf{00} + \mathbf{1})^\star$ a homomorfismus h určený $h(a) = 01$ a $h(b) = 10$.

Určete $h^{-1}(L)$.

Věta. Jestliže h je homomorfismus, $h\colon \Sigma^\star \to \Gamma^\star$ a L je regulární jazyk nad abecedou Γ, pak inversní obraz $h^{-1}(L)$ je také regulární (nad Σ).

D.: M pro L

pro $h^{-1}(L)$

vyrobíme N: $w \in \Sigma^*$

nejprve „přeložíme w na $h(w)$

pustíme M

$w \in h^{-1}(L)$ iff $h(w)$ M přijal

tj. $h(w) \in L(M) = L$

Přednáška 9. listopadu 2020

Čím jsme skončili minulou přednášku.

Ukázali jsme další operace, na které je třída reg. jazyků uzavřená.

Algoritmická řešitelnost úloh pro regulární jazyky

1. Pro daný konečný automat M (ať deterministický nebo nedeterministický) a slovo $w \in \Sigma^\star$ rozhodnout, zda $w \in L(M)$.

Jednoduché:
Vezmeme stavový diagram, ukroužíme sled z q_0 (počátečního stavu) označený w a jestliže končí v $q \in F$, tak $w \in L(M)$
— " — $q \notin F$, tak $w \notin L(M)$

2. Pro daný konečný automat M rozhodnout, zda $L(M) = \Sigma^\star$.

M vytvoříme $\underline{\text{DFA } M'}$ s $L(M) = L(M')$.
M'' vznikne z M' tak, že prohodíme
koncové $\underline{\text{a}}$ nekoncové stavy.

$L(M'') = \overline{L(M)}$.

Zjistíme, zda $L(M'') \neq \emptyset$ (ve stavovém diagramu M'' zjistíme, zda je nějaký koncový stav dosažitelný z počátečního stavu. $L(M) = \Sigma^* \text{ iff } L(M'') = \emptyset$

3. Pro dva konečné automaty M_1 a M_2 rozhodnout, zda $L(M_1) = L(M_2)$.

M_1 zredukujeme M_1'

M_2 zredukujeme M_2'

$\underline{L(M_1) = L(M_2)}$ iff M_1' a M_2' jsou isomorfní

$q_0^1 \longrightarrow q_0^2$ $\quad \delta_1'(q_0^1, a) \rightsquigarrow \delta_2'(q_0^2, a)$

a dále.

Tvrzení. Je dán deterministický konečný automat M s n stavy. Pak

1. Jazyk $L(M)$ je neprázdný právě tehdy, když M přijímá slovo w délky $|w| < n$.

2. Jazyk $L(M)$ je nekonečný právě tehdy, když M přijímá slovo v délky $n \le |v| < 2n$.

D: K M utvoříme stavový diagram

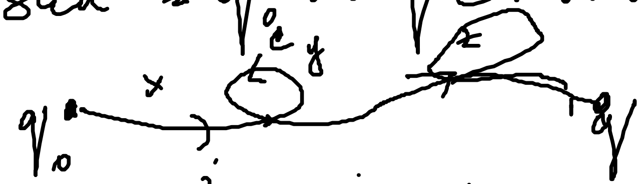
1. a) Když přijímá slovo w, $|w| < n$, tak $L(M) \neq \emptyset$

a) Ať $L(M) \neq \emptyset$. Tak existuje $x \in L(M)$. Tj. ve stavovém diagramu existuje orient. sled $q_0 \rightsquigarrow q \in F$

Každý orient. sled obsahuje orient. cestu z q_0 do q. Každá cesta v grafu s n vrcholy má nejvýše $n-1$ hran. Pak w je slovo, jímž je označena tato cesta, $0 \le |w| \le n-1$.

2) a) Přijímá slovo w délky $n \le w < 2n$. Orient. sled z q_0 do $q \in F$ ohodnocený w má cyklus C

$w = xyz$. zbytek



xy^iz, $i = 0, 1, 2, \dots$, patří do $L(M)$.

Tj. $L(M)$ je nekonečný.

b) Ať $L(M)$ je nekonečný.

Tj. existuje $x \in L(M)$, $|x| \ge n$.

(Slov délky $\le n$ je konečně mnoho.)

Je-li $|x| < 2n$, jsme hotovi.

Předpokládejme, že $|x| \geq 2n$.

Pak sled z q_0 ohodnocený x musí obsahovat cyklus. Vezmeme nějaký cyklus v sledu označeném x a odstraníme ho. Cyklus má délku $\leq n$, tj. dostáváme $|x_1| < |x|$ a $n \leq |x_1|$ (neboť jsme $|x| \geq 2n$)

Je-li $|x_1| < 2n$ jsme hotovi, jinak opět cyklus a odstraníme ho.

Jednou dostaneme $x_i \in L(M)$ s $n \leq |x_i| < 2n$

Gramatiky ... nástroj, který popisuje jak „generovat" slova.

Příklad.

$$\langle \text{číslo} \rangle ::= \langle \text{číslo} \rangle \langle \text{číslice} \rangle \mid \langle \text{číslice} \rangle$$

$$\langle \text{číslice} \rangle ::= 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 0$$

$S \overset{\text{ozn.}}{=} \langle \text{číslo} \rangle \qquad A = \langle \text{číslice} \rangle$

$\longrightarrow \; = \; ::=$

$$S \longrightarrow SA \mid A \qquad \Sigma = \{0, 1, \dots, 9\}$$

$$A \longrightarrow 1 \mid 2 \mid 3 \mid \dots \mid 9 \mid 0$$

Př.: Množina N zadaná induktivně

1) $\varepsilon \in N$

2, a) if $w \in N$, tak $awb \in N$, $bwa \in N$

b) if $u, v \in N$, tak $uv \in N$

$$S \longrightarrow aSb \mid bSa \mid SS \mid \varepsilon$$

To, co tato gramatika „vygeneruje" je množina N.

Gramatika je uspořádaná čtveřice $\mathcal{G} = (N, \Sigma, S, P)$, kde

- N je konečná množina tzv. *neterminálů*;
- Σ je konečná neprázdná množina tzv. *terminálů*, platí $N \cap \Sigma = \emptyset$;
- $S \in N$ je *startovací symbol*;
- P je konečná množina pravidel typu $\alpha \to \beta$, kde α a β jsou slova nad $N \cup \Sigma$ taková, že α obsahuje alespoň jeden neterminál.

$\alpha, \beta \in (N \cup \Sigma)^*$

Př. $N = \{S, A\}$ $\Sigma = \{0, 1, \ldots, 9\}$

S je startovací symbol zkratka za

$\ldots$ $S \to SA$ $S \to A$

P: $S \to SA \mid A$

$A \to 1 \mid 2 \mid \ldots \mid 9 \mid 0$

Přímé odvození. Dána gramatika $\mathcal{G} = (N, \Sigma, S, P)$. Řekneme, že δ se *přímo odvodí* (též *přímo přepíše*) z γ, jestliže existuje v P pravidlo $\alpha \to \beta$ a slova $\varphi, \psi \in (N \cup \Sigma)^\star$ taková, že $\gamma = \varphi\,\alpha\,\psi$ a $\delta = \varphi\,\beta\,\psi$.
Tento fakt zapisujeme $\gamma \Rightarrow_{\mathcal{G}} \delta$.

$\gamma \Rightarrow_{\mathcal{G}} \delta$ iff $\gamma = \varphi\alpha\psi$ $\delta = \varphi\beta\psi$

$\alpha \to \beta \in P$

Př:

$S A A \Rightarrow S 2 A$

$A \to 2 \in P$

Odvození, derivace. Dána gramatika $\mathcal{G} = (N, \Sigma, S, P)$. Řekneme, že δ se *odvodí* z γ, jestliže

- buď $\gamma = \delta$,
- nebo existuje posloupnost přímých odvození

$$\gamma = \gamma_1 \Rightarrow_{\mathcal{G}} \gamma_2 \Rightarrow_{\mathcal{G}} \ldots \Rightarrow_{\mathcal{G}} \gamma_k = \delta.$$

Značíme $\gamma \Rightarrow^\star_{\mathcal{G}} \delta$ a konečná posloupnost je *derivace*.

Př. $S \Rightarrow^* 21A$

protože $S \Rightarrow SA \Rightarrow SAA \xrightarrow{A \to 1} S1A \Rightarrow$

$S \to SA$ $\quad$ $S \to SA$ $\quad$ $S \to A$

$A \to 2$

$A1A \Rightarrow 21A$

Jazyk generovaný gramatikou. Slovo $w \in \Sigma^\star$ je *generováno* gramatikou $\mathcal{G}$, jestliže existuje derivace $S \Rightarrow^\star_{\mathcal{G}} w$.

Jazyk $L(\mathcal{G})$ *generovaný* gramatikou $\mathcal{G}$ je množina všech slov generovaných $\mathcal{G}$, tj

slovo $w \in \Sigma^*$ je generováno $\mathcal{G}$ iff $S \Rightarrow^*_{\mathcal{G}} w$

(uvědomte si, že „dál už nejde generovat")

$$L = \{ w \mid S \Rightarrow^*_{\mathcal{G}} w \}$$

<u>Př</u>: $\mathcal{G}$:
$$S \to SA \mid A$$
$$A \to 0 \mid 1 \mid \dots \mid 9$$

Pak $120 \in L(\mathcal{G})$, ano

$$S \xRightarrow[S \to SA]{} SA \xRightarrow[S \to SA]{} SAA \xRightarrow[S \to A]{} AAA \xRightarrow[A \to 1]{} 1AA \xRightarrow[A \to 2]{} 12A \xRightarrow[A \to 0]{} 120$$

Konvence.

- Neterminály značíme obvykle velkými písmeny $A, B, X, Y, \dots$.
- Terminály značíme obvykle malými písmeny ze začátku abecedy, například $a, b, c, d, \dots$.
- Slova z $(N \cup \Sigma)^\star$ obvykle značíme řeckými písmeny α, β, ...
- Terminální slova, tj. slova z $\Sigma^\star$, značíme malými písmeny z konce abecedy $u, w, x, y, \dots$.

Chomského hierarchie.

- *Gramatiky typu 0* jsou nejobecnější. Jazyky generované gramatikami typu 0 jsou *jazyky typu 0.*

To jsou ta nejobecnější pravidla

$\alpha \to \beta$ $\qquad$ α obsahují neterminál

např. $aBbAC \to bab$

proč dál $\left(\mathcal{L}_3 \subseteq \mathcal{L}_2 \overset{?}{\subseteq} \mathcal{L}_1 \subseteq \mathcal{L}_0 \right)$

- *Gramatiky typu 1*, též *kontextové gramatiky*, jsou gramatiky s pravidly tvaru

$$\alpha\, A\, \beta \to \alpha\, \gamma\, \beta,$$

kde $\alpha, \beta, \gamma \in (N \cup \Sigma)^{\star}$, A je neterminál a $\gamma \neq \varepsilon$. Jedinou výjimku tvoří pravidlo $S \to \varepsilon$, pak se ale S nevyskytuje na pravé straně žádného pravidla. Jazyky generované gramatikami typu 1 jsou *jazyky typu 1*, též *kontextové jazyky.*

$S \to \varepsilon$

Př.:

$aAb \to aab$

kontextové pravidlo

nebo $\quad a\boxed{B}C \to a\boxed{b}C$

- *Gramatiky typu 2*, též *bezkontextové gramatiky*, (CF gramatiky) jsou gramatiky s pravidly tvaru

$$A \to \gamma,$$

kde $\gamma \in (N \cup \Sigma)^\star$ a A je neterminál.

Jazyky generované gramatikami typu 2 jsou *bezkontextové jazyky* nebo *jazyky typu 2*.

CF

Př.

$$S \to SA \mid A$$
$$A \to 0 \mid \ldots \mid 9$$

je CF

- *Gramatiky typu 3* neboli *regulární gramatiky* (též *pravé lineární gramatiky*) jsou gramatiky s pravidly tvaru

$$A \to w\,B,\ A \to w,$$

kde A, B jsou neterminály a w je terminální slovo.

Jazyky generované gramatikami typu 3 jsou *regulární jazyky* nebo *jazyky typu 3*.

$A \to wB$ $\quad A \to w$

$w \in \Sigma^*$ $\quad B \in N$

Nevypouštěcí gramatiky. Gramatiku $\mathcal{G} = (N, \Sigma, S, P)$ nazveme *nevypouštěcí*, jestliže neobsahuje žádné pravidlo typu $A \to \varepsilon$.

Nevypouštěcí gramatika nemůže vygenerovat ε.

Tvrzení. Ke každé CF $\mathcal{G}$ existuje nevypouštěcí gramatika $\mathcal{G}_1$ taková, že

$$L(\mathcal{G}_1) = L(\mathcal{G}) - \{\varepsilon\}.$$

D.: Návod, jak $\mathcal{G}_1$ zkonstruovat.

Označme $U = \{X \mid X \Rightarrow^* \varepsilon\}$

U konstruujeme $U_0 = \{X \mid X \to \varepsilon \in P\}$

když máme U_i, vytvoříme U_{i+1}

$U_{i+1} = U_i \cup \{X \mid X \to A_1 \ldots A_k \in P, A_i \in U_i\}$

$U_0 \subseteq U_1 \subseteq U_2 \subseteq \ldots \subseteq U_i \subseteq \ldots \subseteq N$

N je konečná

existuje n, že $U_{n+1} = U_n$ a pak $U_n = U$.

Máme U

$\mathcal{G}_1$ vytvoříme: $\mathcal{G}_1 = (N, \Sigma, S, P_1)$

P_1: Pro každé pravidlo z $P(\mathcal{G})$

$$A \to \alpha_1 X_1 \alpha_2 X_2 \ldots \alpha_{k-1} X_{k-1} \alpha_k$$

α_i neobsahuje neterminál z U, $X_i \in U$

do P_1 dáme pro každou podmnožinu B $X_1 \ldots X_{k-1}$ pravidlo $A \to$ ___ vynechali vybrané $X_j \in B$

Pravidlo $A \to \varepsilon$ z P do P_1 nedáme

Př: $U = \{S, A\}$ $\quad N = \{S, A, B\}$

v $\mathcal{G}$ $\quad S \to aAbBaA$

$\Downarrow$

v $\mathcal{G}_1$ $\quad S \to aAbBaA \mid abBaA \mid aAbBa \mid abBa$

Příklad. Dána gramatika $\mathcal{G} = (\{A, S\}, \{a, b, c\}, S, P)$ pravidly

$$S \to aSc \,|\, A \quad A \to bAc \,|\, \varepsilon.$$

Ke gramatice $\mathcal{G}$ najdeme nevypouštěcí gramatiku $\mathcal{G}_1$, která generuje jazyk $L(\mathcal{G}) - \{\varepsilon\}$.

Řeš.:

$$\underline{S} \to aSc \,|\, \underline{A} \qquad \underline{A} \to bAc \,|\, \underline{\varepsilon}$$

$$U_0 = \{A\} \qquad U_1 = \{A, S\} = U$$

$$\mathcal{G}_1 : \left(\{S, A\}, \{a, b, c\}, S, P_1\right)$$

$$P_1 \quad \boxed{\begin{array}{l} S \to aSc \,|\, ac \,|\, A \\ A \to bAc \,|\, bc \end{array}}$$

Důsledek. Označme $\mathcal{L}_i$ třídu jazyků typu i. Pak platí:

$$\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0.$$

D.: $\mathcal{G}$ typu 2 (CF)

$\rightsquigarrow$ $\mathcal{G}_1$, $\mathcal{G}_1$ nevypouštěcí (N, Σ, S, P_1)

k $\mathcal{G}_1$ $\rightsquigarrow$ $\underline{\mathcal{G}_2 = (N \cup \{S_0\}, \Sigma, S_0, P_0)}$

$P_0 = P_1 + S_0 \longrightarrow S$

a v případě, že $\varepsilon \in L(\mathcal{G})$, tak $\underline{S_0 \to \varepsilon}$ je v P_0

Tvrzení 1. Je-li L regulární jazyk, pak existuje regulární gramatika $\mathcal{G}$ taková, že $L = L(\mathcal{G})$.

D.: L je regulární, tj. existuje DFA M

tak, že $L = L(M)$

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

$\mathcal{G} = (N, \Sigma, S, P)$ $\quad N := Q$, $\quad S := q_0$

P: $q \to ap \iff \delta(q, a) = p$

$q \to \varepsilon \iff q \in F$

Není těžké ukázat, že $L(M) = L(\mathcal{G})$

$\delta(q_0, a_1) = p_1 \quad q_0 \to a p_1 \quad \ldots$ simulujeme výpočet

Tvrzení 2. Je-li $\mathcal{G}$ gramatika typu 3, pak jazyk $L(\mathcal{G})$ je regulární.

pravidla $A \to wB$

D.: každou pravidla jsou

$A \to aB$ nebo $A \to B$

$A \to \varepsilon$

tak můžeme představit automat

$Q := N$ $q_0 := S$ $A \in F$ iff $A \to \varepsilon$

$B \in \delta(A, a)$ iff $A \to aB$

Když máme $A \to a_1 a_2 \dots a_k B$ ($A \to a_1 \dots a_k$)

přidáme neterminály

$A \to a_1 B_1$, $B_1 \to a_2 B_2$, $B_2 \to a_3 B_3$, ... $B_k \to \varepsilon$

B_i nová $B_{k-1} \to a_k B$ ($B_{k-1} \to a_k B_k$)

Bezkontextové gramatiky příště

Přednáška 16. listopadu 2020

Čím jsme skončili minulou přednášku.

gramatiky $\mathcal{G} = (N, \Sigma, S, P)$

N .. množina neterminálů

Σ —"— terminálů

$S \in N$ startovací symbol

P množina pravidel $\alpha \to \beta$

$\alpha, \beta \in (N \cup \Sigma)^*$

α obsahuje aspoň 1 neterminál

typ 1: $\alpha A \beta \to \alpha \gamma \beta$ typ 2 CF $A \to \alpha$, $A \in N$

typ 3: $A \to wB$ $A \to w$ $w \in \Sigma^*$ $A, B \in N$

Bezkontextové gramatiky

$\mathcal{G} = (N, \Sigma, S, P)$ P: $A \to \alpha$

$A \in N$, $\alpha \in (\Sigma \cup N)^*$

Jazyk L je CF (bezkontextový) if existuje CF $\mathcal{G}$ tak, že $L(\mathcal{G}) = L$.

Př: $\mathcal{G} = (\{S\}, \{0,1\}, S, P)$ kde P: $S \to 0S1 \mid \varepsilon$

Pak $S \underset{S \to 0S1}{\Rightarrow^{(n)}} 0^n S 1^n \underset{S \to \varepsilon}{\Rightarrow} 0^n 1^n$ nic se nedá vygenerovat

$L(\mathcal{G}) = \{0^n 1^n \mid n \geq 0\}$

Tvrzení. Dána CF gramatika $\mathcal{G} = (N, \Sigma, S, P)$ a v ní derivace

$$S \Rightarrow^\star_{\mathcal{G}} \alpha A \beta \Rightarrow^\star_{\mathcal{G}} \gamma,$$

pro $\alpha, \beta, \gamma \in (\Sigma \cup N)^\star$ a $A \in N$.

$$\alpha A \beta \Rightarrow^*_{\mathcal{G}} \gamma = \varphi\psi\mu$$

Pak existují slova $\varphi, \psi, \mu \in (\Sigma \cup N)^\star$ taková, že

$$\gamma = \varphi\psi\mu \quad \text{a} \quad \alpha \Rightarrow^\star_{\mathcal{G}} \varphi, \quad A \Rightarrow^\star_{\mathcal{G}} \psi, \quad \beta \Rightarrow^\star_{\mathcal{G}} \mu.$$

Myšlenka důkazu:

$$\alpha A \beta \Rightarrow^* \gamma$$

v derivaci přehodíme pořadí pravidel, abychom nejdřív používali pravidla která přepisují neterminály vzniklé v α, pak pravidla vzniklá z „prostředního" A, nakonec pravidla vycházející z β.

Levá derivace, levé odvození.

Přímé odvození je *levé*, jestliže se přepisuje ten neterminál, který je nejvíc „vlevo".

Derivace (odvození) je *levá derivace* (*levé odvození*), jestliže se skládá pouze z levých přímých odvození.

Levé přímé odvození:

$\alpha = wA\beta \qquad w \in \Sigma^*$

$wA\beta \xrightarrow{A \to \gamma} w\gamma\beta$

Levá derivace: konečná posloupnost levých přímých odvození.

Tvrzení. Je dána bezkontextová gramatika $\mathcal{G} = (N, \Sigma, S, P)$. Pak pro každou derivaci $S \Rightarrow^\star_\mathcal{G} w$ existuje levá derivace terminálního w z S taková, že používá stejná pravidla jako původní derivace (pouze možná v jiném pořadí).

Ano, postupujeme vždy tak, že přepisujeme vždy nejvíc levý neterminál.

Derivační strom (parse tree).

Dána CF gramatika $\mathcal{G} = (N, \Sigma, S, P)$. *Derivační strom* (anglicky *parse tree*) je kořenový strom, takový, že:

1. Každý vrchol, který není list, je ohodnocen neterminálem.
2. Každý list je ohodnocen terminálem nebo prázdným slovem ε. V případě, že je list ohodnocen prázdným slovem ε, je to jediný následník (svého předchůdce).
3. Jestliže některý vrchol, který není list, je ohodnocen neterminálem A a má následníky (v pořadí zdleva doprava) $X_1, X_2, \ldots, X_k$, $X_i \in N \cup \Sigma$, pak $A \to X_1 X_2 \ldots X_k$ je pravidlo gramatiky $\mathcal{G}$.

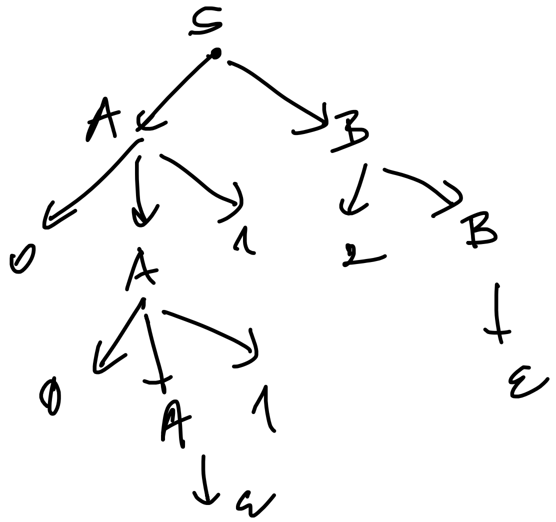
Řekneme, že derivační strom *dává*, nebo *má za výsledek* slovo w, jestliže w je ohodnocení listů derivačního stromu (čteno zleva doprava).

Příklad: $S \to AB \quad A \to 0A1 \mid \varepsilon$

$B \to 2B \mid \varepsilon$

$$S \Rightarrow AB \xRightarrow{A \to 0A1} 0A1B \xRightarrow{B \to 2B} 0A12B \xRightarrow{B \to \varepsilon}$$

$$0A12 \xRightarrow{A \to 0A1} 00A112 \xRightarrow{A \to \varepsilon} 00112$$



Tvrzení.

1. Pro každou derivaci $S \Rightarrow^{\star}_{\mathcal{G}} w$ existuje derivační strom s výsledkem w.
2. Ke každému derivačnímu stromu s výsledkem w existuje aspoň jedna derivace $S \Rightarrow^{\star}_{\mathcal{G}} w$ (takových derivací může být více).
3. Ke každému derivačnímu stromu s výsledkem w existuje právě jedna levá (právě jedna pravá) derivace w z S.

Ke každé derivaci zkonstruovat derivační strom.

a ke každému stromu existuje právě jedna levá derivace

— budeme přepisovat nejlevější neterminál

viz. příklad: $S \Rightarrow AB \xRightarrow{A \to 0A1} 0A1B \xRightarrow{A \to 0A1} 00A11B \xRightarrow{A \to \varepsilon} 0011B \xRightarrow{B \to 2B} 00112B \xRightarrow{B \to \varepsilon} 00112$

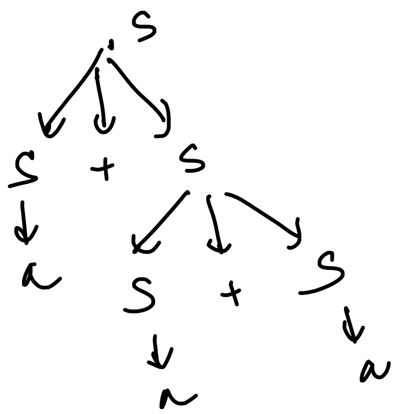
Jednoznačné a víceznačné bezkontextové gramatiky.

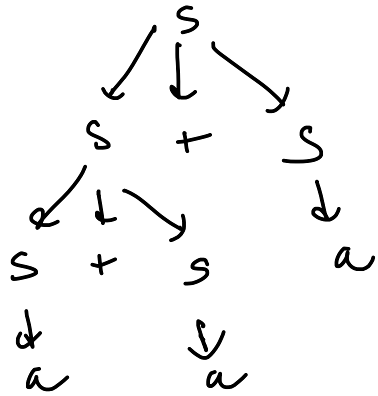
CF gramatika $\mathcal{G} = (N, \Sigma, S, P)$ je *jednoznačná*, jestliže pro každé slovo w generované $\mathcal{G}$ existuje jediný derivační strom s výsledkem w (tj. existuje jediná levá derivace w z S). V opačném případě mluvíme o *víceznačné* gramatice.

$\Sigma = \{+, a\}$ $\quad S \to S+S \mid a$ $\quad N = S$

víceznačná

$w = a+a+a$



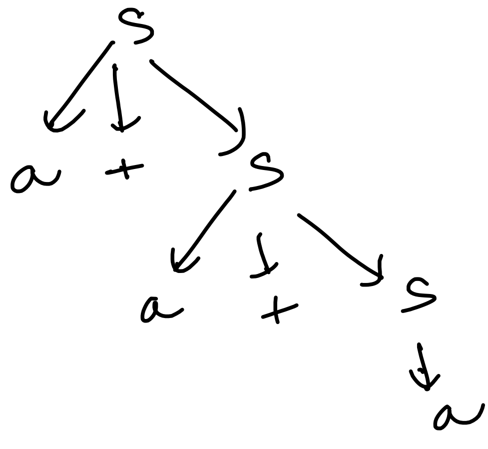


stromy jsou různé

Pro stejný jazyk:

$N = \{S\}$, $\Sigma = \{+, a\}$ P: $S \to a+S \mid a$

$w = a+a+a$



$$L(\mathcal{G}) = \{\underbrace{a+a+\ldots+a}_{n\text{-krát}} \mid n \geq 1\}$$

$$\{a, a+a, a+a+a, \ldots\}$$

Víceznačný jazyk.

Jazyk L je *víceznačný*, jestliže **každá** bezkontextová gramatika, která ho generuje, je víceznačná.

Například jazyk $L = \{a^i b^j c^k d^l \mid i = j \text{ nebo } k = l\}$ je víceznačný.

nedokazujeme.

Redukovaná bezkontextová gramatika.
CF gramatika $\mathcal{G} = (N, \Sigma, S, P)$, která generuje aspoň jedno slovo, je *redukovaná*, jestliže splňuje tyto dvě podmínky:

1. Ke každému neterminálu A existuje aspoň jedno terminální slovo w takové, že $A \Rightarrow^\star_{\mathcal{G}} w$.
2. Ke každému neterminálu A existují slova $\alpha, \beta \in (N \cup \Sigma)^\star$ tak, že
$$S \Rightarrow^\star_{\mathcal{G}} \alpha A \beta.$$

neplatí, že dvě CF gramatiky generují stejný jazyk iff cosi platí o redukovaných

Neexistuje algoritmus který pro dvě CF gramatiky rozhodne, že generují aspoň jedno slovo stejné

Redukovaná CF znamená nemá zbytečné neterminály.

(1) $S \Rightarrow^* \alpha A \beta \Rightarrow^* w \qquad w \in \Sigma^*$

$A \Rightarrow^*$ podslovo $w \qquad A \Rightarrow^* u \quad u \in \Sigma^*$

(2) $S \Rightarrow^* w$ v takové derivaci jsou jen $A \in N$, že $S \Rightarrow^* \alpha A \beta$.

Tvrzení. Ke každé CF gramatice $\mathcal{G} = (N, \Sigma, S, P)$, pro kterou $L(\mathcal{G}) \neq \emptyset$, existuje redukovaná gramatika $\mathcal{G}_1$ taková, že $L(\mathcal{G}_1) = L(\mathcal{G})$.

Důkaz = algoritmus redukce.

Algoritmus redukce.

Dána CF gramatika $\mathcal{G} = (N, \Sigma, S, P)$.

1. Sestrojíme množinu $V = \{A \mid A \in N, A \Rightarrow^\star_{\mathcal{G}} w, w \in \Sigma^\star\}$

Indukcí: 1) $V_1 = \{A \mid A \to w \in P, w \in \Sigma^*\}$

2) Když máme V_i, konstruujeme V_{i+1}:

$$V_{i+1} = V_i \cup \{B \mid B \to \beta \in P, \beta \in (\Sigma \cup V_i)^*\}$$

$X_i \Rightarrow^* w_i$ $\quad B \Rightarrow \beta$, $\beta = u_1 X_1 u_2 \dots X_k u_{k+1} \Rightarrow^* u_1 w_1 \dots w_k u_{k+1}$

$$V_1 \subseteq V_2 \subseteq V_3 \subseteq \dots \subseteq V_i \subseteq \dots \subseteq N$$

tj. ex. n tak, že $V_n = V_{n+1} = V$.

Jestliže $S \notin V$, pak $L(\mathcal{G}) = \emptyset$ a redukovaná gramatika ke gramatice $\mathcal{G}$ neexistuje.

Jinak $S \in V$

Definujeme $\mathcal{G}' = (V, \Sigma, S, P')$: do P' dáme ta pravidla z P, která obsahují pouze neterminály z množiny V.

$$\mathcal{G}' = (V, \Sigma, S, P')$$

z P odstraníme všechna pravidla obsahující nějaký neterminál $X \notin V$.

2. Pro gramatiku $\mathcal{G}' = (V, \Sigma, S, P')$ sestrojíme (opět indukcí) množinu

$$U = \{A \mid A \in V,\ \text{existují } \alpha, \beta \in (V \cup \Sigma)^\star \text{ tak, že } S \Rightarrow^\star_{\mathcal{G}} \alpha A \beta\}.$$

Indukcí: 1) $U_0 = \{S\}$

2) když máme U_i tak U_{i+1} definujeme

$$U_{i+1} = U_i \cup \{A \mid \text{ex. pravidlo } B \to \alpha \in P,\ B \in U_i,\ A \text{ je obsaženo v } \alpha\}$$

$$S \Rightarrow^* \gamma B \beta \xRightarrow{B \to \alpha} \underbrace{\gamma \alpha \beta}_{A \text{ někde}} = \varphi A \psi$$

$$U_0 \subseteq U_1 \subseteq \ldots \subseteq U_i \subseteq \ldots \quad V$$

konečná

ex. i, že $U_i = U_{i+1} = U$

redukovaná gramatika (U, Σ, S, P'')

P'' obsahuje ta pravidla z P', která neobsahují neterminály $X \notin U$.

Hledaná gramatika je $\mathcal{G}_1 = (U, \Sigma, S, P_1)$, kde P_1 je množina všech pravidel z P' (a tedy i z P), které obsahují neterminály pouze z množiny U.

Poznámky.

1) Pro daný CF může existovat řada různých redukovaných CF gramatik.

2) nelze prohodit pořadí kroků v algoritmu redukce.

$N = \{S, A, B, C\}$

$\Sigma = \{a, b\}$

Př.: P:

$S \to aA \mid aSb \mid \varepsilon$

$A \to bCB$

$B \to aA \mid a$

$C \to Ca$

napřed 2. krok, tj. hledáme $U = \{A \mid S \Rightarrow^* \alpha A \beta\}$

$U_0 = \{S\}$, $U_1 = \{S, A\}$, $U_2 = \{S, A, B, C\} = U = N$

gramatika se nezmění.

1. krok, tj. hledáme $V = \{A \mid A \Rightarrow^* w, w \in \Sigma^*\}$

$V_1 = \{S, B\}$ $V_2 = \{S, B\} = V_1 = V$

$(\{S, B\}, \{a, b\}, S, P')$

P': $\boxed{\begin{array}{l} S \to aSb \mid \varepsilon \\ B \to a \end{array}}$

nesplňují podmínku 2)
$S \not\Rightarrow^* \alpha B \gamma$

Chomského normální tvar. Gramatika $\mathcal{G}$ je v *Chomského normálním tvaru*, jestliže má pouze pravidla tvaru

$$A \to BC,\ A \to a \quad \text{pro} \quad A, B, C \in N,\ a \in \Sigma.$$

tj. derivační strom je binární strom a každý neterminál má buď 2 následníky a ty jsou neterminály nebo jednoho následníka a ten je terminální symbol

Věta. Pro každou CF gramatiku $\mathcal{G}$ existuje CF gramatika $\mathcal{G}'$ v Chomského normálním tvaru taková, že

$$L(\mathcal{G}') = L(\mathcal{G}) - \{\varepsilon\}.$$

Důkaz algoritmem na nalezení $\mathcal{G}'$

Postup nalezení ekvivalentní gramatiky v Chomského normálním tvaru.

1. Sestrojíme nevypouštěcí gramatiku $\mathcal{G}_1$, která generuje všechna neprázdná slova z jazyka $L(\mathcal{G})$.

$$L(\mathcal{G}_1) = L(\mathcal{G}) \setminus \{\varepsilon\}$$

už umíme z minulé přednášky

2. V $\mathcal{G}_1$ odstraníme pravidla typu $A \to B$, kde $A, B \in N$.

Pro každé A, C, ť

$$A \Rightarrow X_1 \Rightarrow X_2 \Rightarrow \dots \Rightarrow C$$

a každé pravidlo $C \to \alpha$

přidáme pravidlo $A \to \alpha$

Tím dostaneme gramatiku

s pravidly

$A \to \alpha$ $\qquad |\alpha| \geq 2$

$A \to a$ $\qquad a \in \Sigma$

3. Odstraníme terminály $a \in \Sigma$, které se vyskytují na pravé straně pravidla, které má délky aspoň 2.

a to tak $a \in \Sigma$ zavedeme nový neterminál X_a a pravidlo $X_a \to a$ a každé a v pravé straně delší než 1 nahradíme X_a

Př. $A \to aB$

$\underbrace{A \to X_a B \qquad X_a \to a}$

4. Zkrátíme pravé strany pravidel, které jsou delší než 2.

$A \to X_1 \dots X_k$ $k \geq 3$ přidáme $k-2$ nových neterminálů $Z_1, \dots Z_{k-2}$ a pravidla

$A \to X_1 Z_1 \quad Z_1 \to X_2 Z_2, \dots Z_{k-3} \to X_{k-2} Z_{k-2} \quad Z_{k-2} \to X_{k-1} X_k$

Př: $A \to X_a B C X_b$

zavedeme nové neterminály

$A \to X_a Z_1 \quad Z_1 \to B Z_2 \quad Z_2 \to C X_b$

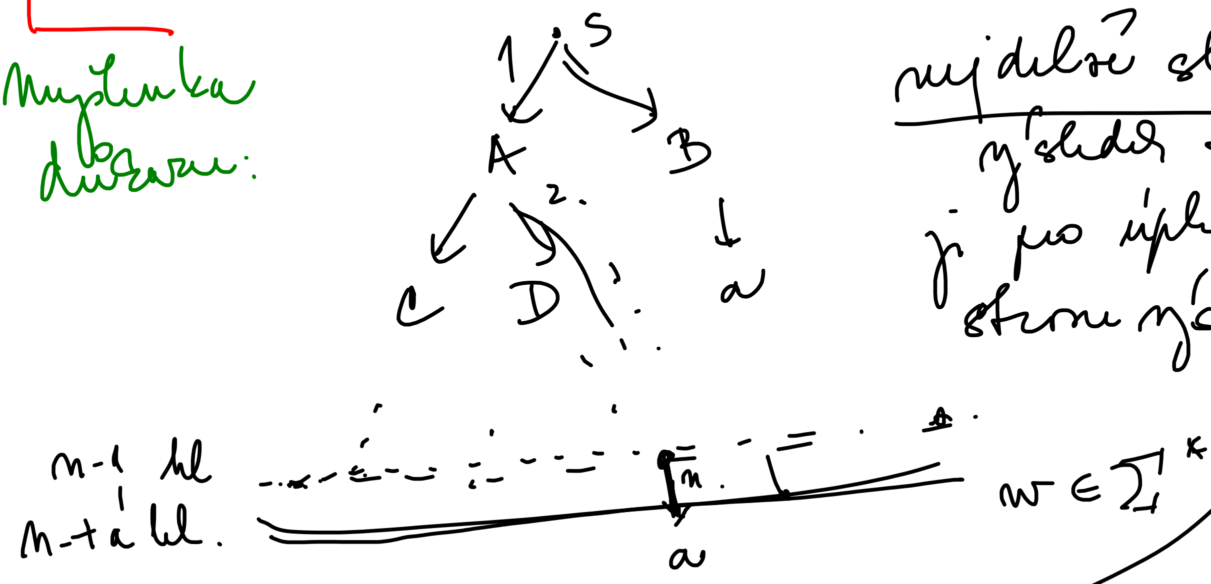
Výsledná gramatika je v Chomského normálním tvaru.

Tvrzení. Dána CF gramatika $\mathcal{G} = (N, \Sigma, S, P)$ v Chomského normálním tvaru a derivační strom s výsledkem $w \in \Sigma^\star$.

Jestliže nejdelší orientovaná cesta v tomto derivačním stromu má délku n, pak pro délku slova w platí

$$|w| \leq 2^{n-1}.$$

Myšlenka důkazu:



nejdelší slovo výsledek stromu je pro úplný binární strom výšky $n-1$

Tento binární strom má v $n-1$ hladině 2^{n-1} uzlů.

To se dokáže jednoduchou indukcí podle n.

Příště: $L = \{0^n 1^n 2^n \mid n \geq 0\}$ není CF

Pumping lemma pro bezkontextové gramatiky.

PL pro reg jazyky:

Je-li L regulární, pak existuje $n \geq 1$ takové, že každé $z \in L$, $|z| \geq n$, lze rozdělit na $z = xyw$ tak, že

1) $|xy| \leq n$, 2) $y \neq \varepsilon$,

3) $\forall i = 0, 1, \ldots$ je $xy^i w \in L$

Věta. Pro každý CF jazyk L existuje přirozené číslo $m \geq 1$ takové, že každé slovo $z \in L$ délky aspoň m lze rozdělit na pět částí $z = uvwxy$ tak, že

- $|vwx| \leq m$, (tj. prostřední část není příliš dlouhá),
- $vx \neq \varepsilon$ (tj. aspoň jedno ze slov v, x není prázdné),
- pro všechna $i \geq 0$ platí $uv^i wx^i y \in L$, (tj. v a x se dají do slova z „napumpovat“ a stále dostaneme slovo z jazyka L).

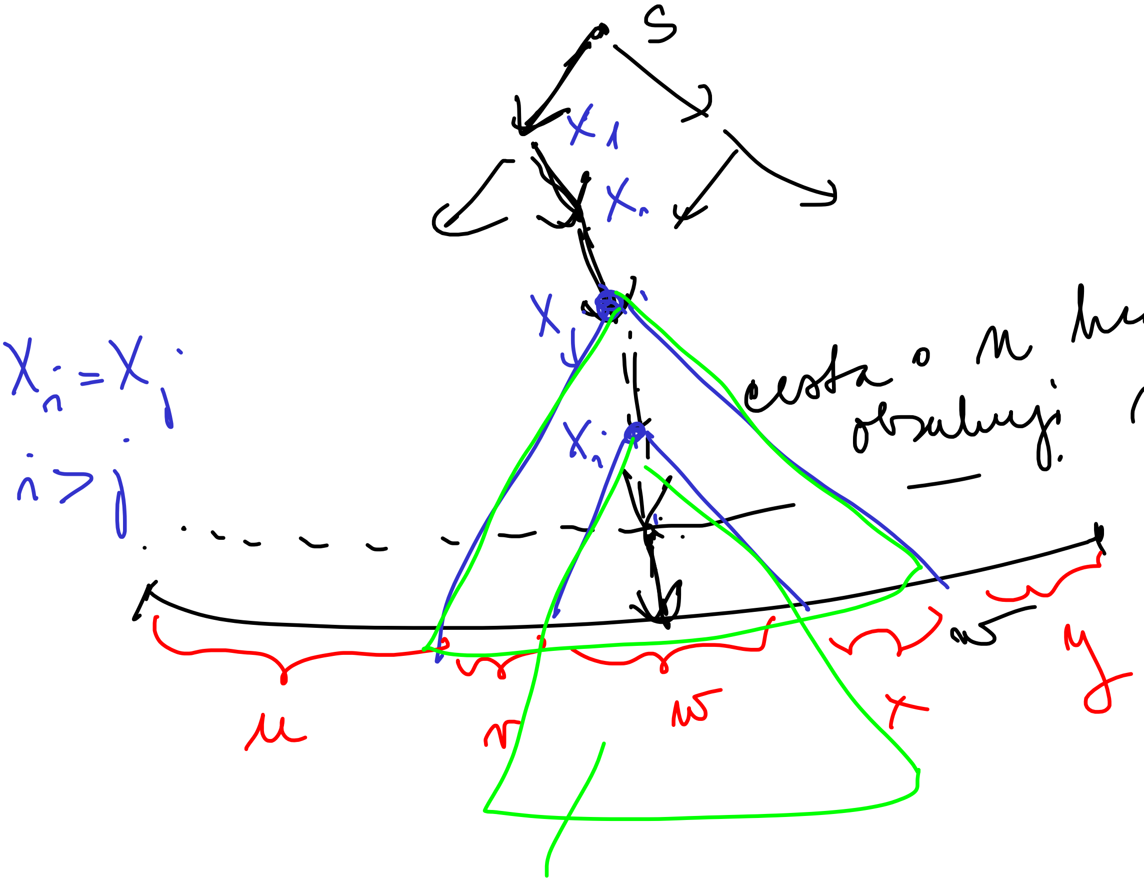
D.: L je CF, tj. existuje k němu CF G v Chom. norm. tvaru, že $L(G)$

$m := 2^n$ $n = |N|$

$G = (N, \Sigma, S, P)$

$w \in L$ $|w| \geq m = 2^n$

Víme, že derivační strom w má orient. cestu o $n+1$ hranách



cesta o n hranách obsahuje $n+1$ neterminálů

$z = u\,v\,w\,x\,y$

Využití Pumping lemmatu pro bezkontextové gramatiky.

Příklad. Ukážeme, že jazyk $L = \{0^n 1^n 2^n \mid n \geq 0\}$ není bezkontextový.

Přednáška 23. listopadu 2020

Čím jsme skončili minulou přednášku.

Pumping lemma

Pumping lemma pro bezkontextové gramatiky.

Pro každý CF jazyk L existuje přirozené číslo $m \geq 1$ takové, že každé slovo $z \in L$ délky aspoň m lze rozdělit na pět částí $z = uvwxy$ tak, že

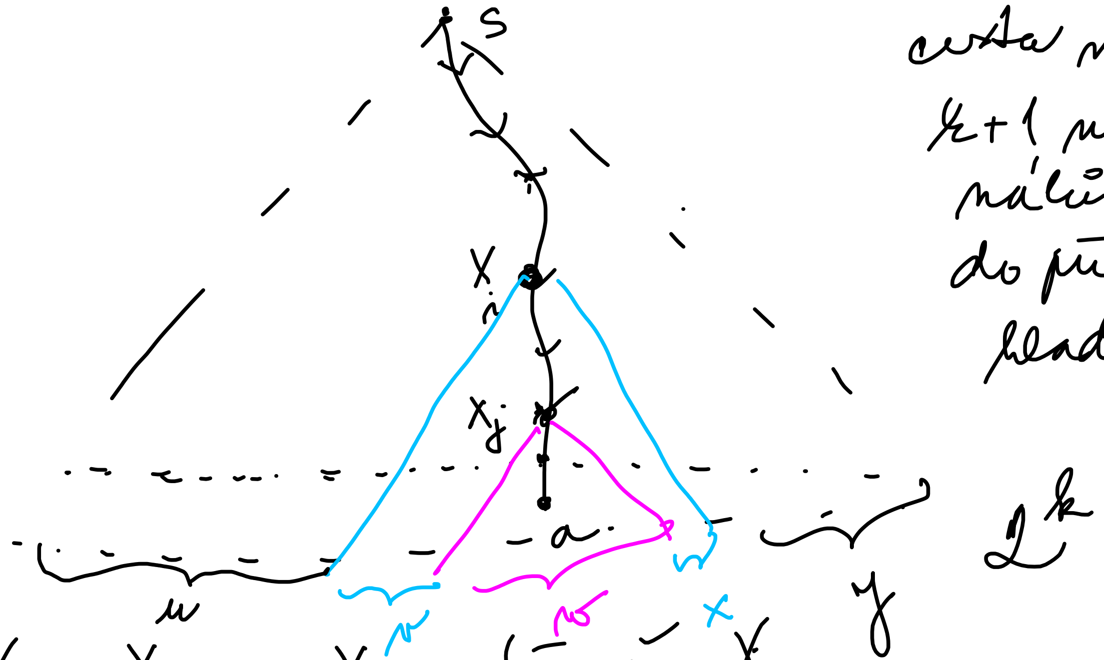
- $|vwx| \leq m$, (tj. prostřední část není příliš dlouhá),
- $vx \neq \varepsilon$ (tj. aspoň jedno ze slov v, x není prázdné),
- pro všechna $i \geq 0$ platí $uv^iwx^iy \in L$, (tj. v a x se dají do slova z „napumpovat" a stále dostaneme slovo z jazyka L).

D.: $m = 2^k$, kde $k = |N|$ $\mathcal{G} = (N, \Sigma, S, P)$ v Chom. norm. tvaru

když $z \in L$, $|z| \geq m$, tak derivační strom z má výšku aspoň k, tj. z kořene S vede do některého listu orient. cesta o k hranách, tedy o $k+1$ uzlech (neterminálech).

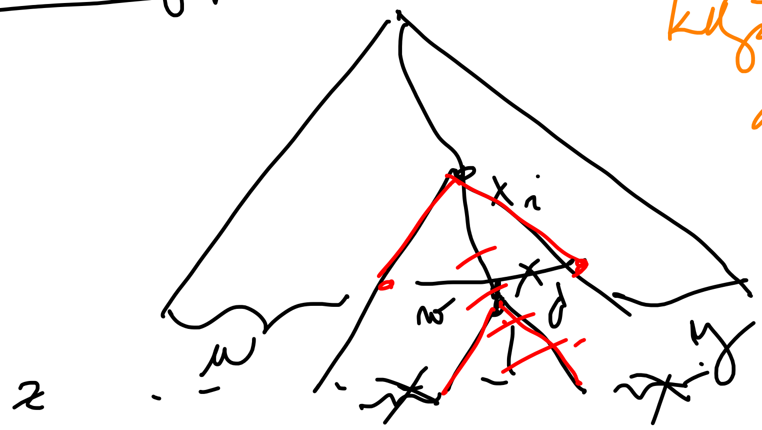
z $S \Rightarrow^* z$

cesta má $k+1$ neterminálů do předposledního leading



$X_i = X_j$ $\quad$ X_i je výš než X_j

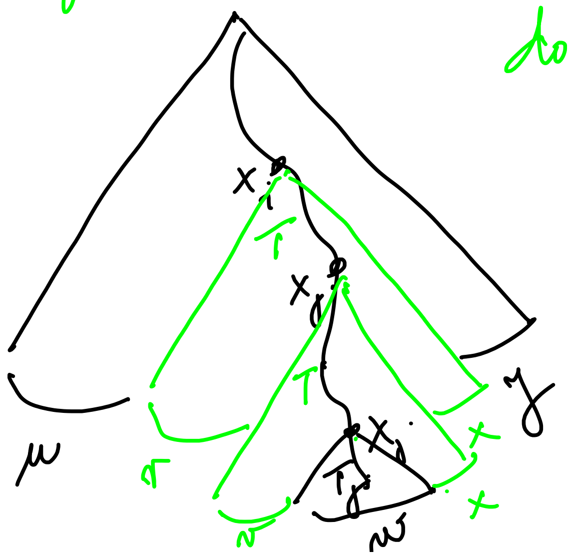
$\boxed{z = u v w x y}$



Když místo stromu X_i dáme strom X_j. Dostaneme derivační strom pro

$u v^0 w x^0 y = u w y$

Když místo stromu T_j dáme strom včetně T_i dostaneme deriv. strom pro $u v^2 w x^2 y$



atd.

Využití Pumping lemmatu pro bezkontextové gramatiky.

k důkazu faktu, že jazyk L není CF.

Nemůže se použít pro to, abyste ukázali, že L je CF

Příklad. Ukážeme, že jazyk $L = \{0^n 1^n 2^n \mid n \geq 0\}$ není bezkontextový.

$L = \{0^n 1^n 2^n \mid n \geq 0\}$

Kdyby L byl CF, tak existuje $m \geq 1$ tak, že každé $z \in L$, $|z| \geq m$, lze rozdělit na $z = u\,v\,w\,x\,y$ splňující 1) $|vwx| \leq m$ 2) $vx \neq \varepsilon$ $(|vx| \geq 1)$ 3) $uv^iwx^iy \in L$ $\forall i \geq 0$.

Předpokládejme, že máme $m \geq 1$.

Zvolíme $z = 0^m 1^m 2^m \in L$, $|z| = 3m \geq m$

Kdyby $0^m 1^m 2^m = uvwxy$ tak, že

(1) $|vwx| \leq m$ tak vwx buď neobsahuje 2 nebo neobsahuje 0 (možná neobsahuje 0 i 2).

Víme (2) $vx \neq \varepsilon$.

ani v ani x nemůže obsahovat dva různé symboly (např. $v \neq 0^i 1^l$ $i, l > 0$)

$$x^2 = 0^i 1^l 0^i 1^l$$

Takže jestliže $v\,w\,x$ neobsahují 2

tak uv^2wx^2y (y obsahuje 2^m)

nebude mít stejný počet 0 a 1 jako 2

$|uv^2wx^2|_0 = r$ a $|uv^2wx^2|_1 = s$

aspoň jedno z r a s je větší než m

Analogicky, jestliže $v\,w\,x$ neobsahují 0,

tak u obsahuje 0^m (jako prefix)

a uv^2wx^2y obsahuje víc 1 nebo 2 než m

Proto $uv^2wx^2y \notin L$ spor s (3).

Tudíž L není CF.

cbd

$L = \{0^n 1^n 2^n \mid n \geq 0\}$ je kontextový

→ na jedné ze závěrečných přednášek

Algoritmus CYK

Dostanu: $\mathcal{G}$ v Chomského norm. tvaru

$w \in \Sigma^*$, $|w| = k$,

tak v čase $O(k^3)$ rozhodnu, zda $u \in L$ nebo

$u \notin L$.

Označme $\mathcal{G} = (N, \Sigma, S, P)$ v Chomského normálním tvaru a $w = a_1 a_2 \dots a_k$. Vytváříme množiny $X_{i,j}$ pro $1 \le i \le j \le k$, kde

$$X_{i,j} = \{A \in N \mid A \Rightarrow^*_{\mathcal{G}} a_i a_{i+1} \dots a_j\}.$$

.... $X_{11}, X_{22}, X_{33}, \dots, X_{kk}$ ... chci vyg. podslovo délky 1

$X_{12}, X_{23} \dots X_{k-1,k}$ ... podslovo délky 2

$X_{13}, X_{24}, X_{35}, \dots, X_{k-2,k}$ ... — 11 — 3

$\vdots$

X_{1k} podslovo délky k w

půjde se z gramatiky

$$X_{ii} = \{A \mid A \Rightarrow^* a_i\} = \{A \mid A \to a_i \in P\}$$

$$X_{12} = \{A \mid A \Rightarrow^* a_1 a_2\}$$

$A \to BC \quad B \to a_1 \quad C \to a_2$

$$A \Rightarrow^* a_1 a_2 \qquad A \Rightarrow BC \Rightarrow a_1 C \Rightarrow a_1 a_2$$

$$A \to BC \qquad B \in X_{11} \qquad C \in X_{22}$$

$$X_{i,i+1} = \{A \mid A \to BC \in P,\ B \in X_{ii},\ C \in X_{i+1,i+1}\}$$

$$X_{13} = \{A \mid A \Rightarrow^* \underbrace{a_1 a_2 a_3}\}$$

$$A \xRightarrow{A \to BC} BC \mid B \Rightarrow^* a_1 \quad C \Rightarrow^* a_2 a_3$$

$$\mid B \Rightarrow^* a_1 a_2 \quad C \Rightarrow^* a_3$$

tj. $\boxed{A \to BC \in P \text{ buď } \mid B \in X_{11} \quad C \in X_{23} \text{ nebo } \mid B \in X_{12} \quad C \in X_{33}}$

$$X_{i,i+2} = \{A \mid A \to BC \in P \text{ a } B \in X_{ii},\ C \in X_{i+1,i+2} \text{ nebo } B \in X_{i,i+1},\ C \in X_{i+2,i+2}\}$$

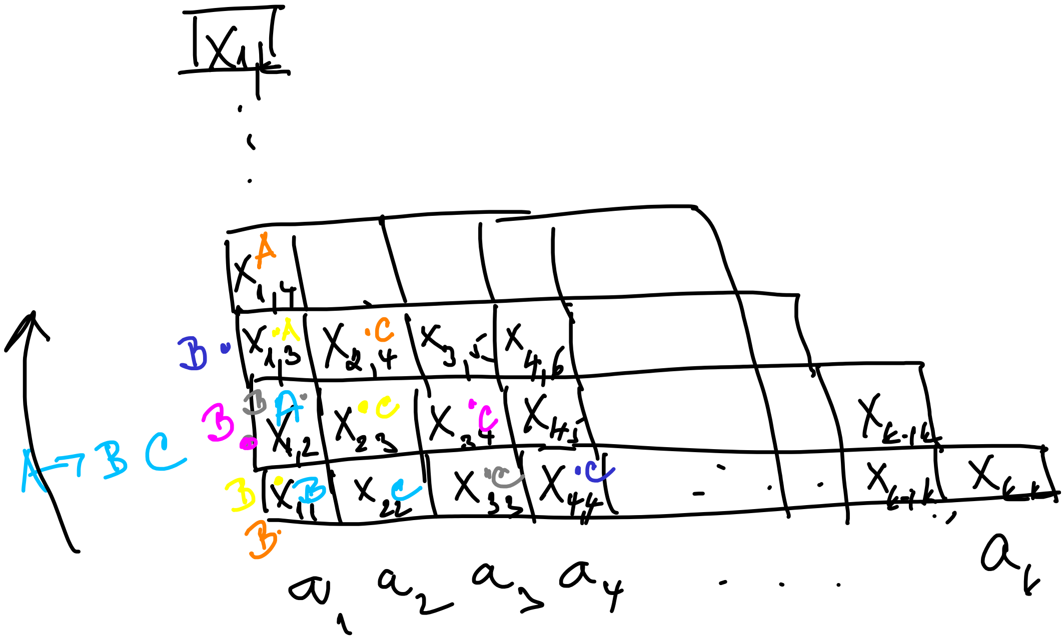
$X_{ij} = \{A \mid A \to BC \in P$ a

$a_i a_{i+1} \dots a_j$

$B \in X_{ii}$, $C \in X_{i+1,j}$ — $a_i, a_{i+1} \dots a_j$

nebo $B \in X_{i,i+1}$, $C \in X_{i+2,j}$ — $a_i a_{i+1}, a_{i+2} \dots a_j$

nebo $\vdots$ $B \in X_{i,j-1}$, $C \in X_{jj}$ — $a_i \dots a_{j-1}, a_j$



X1k
X1,4
X1,3
X2,4
X3,5
X4,6
X1,2
X2,3
X3,4
X4,5
X11
X22
X33
X44
Xk-1k
Xk-1,k
Xkk
A→B C
a1 a2 a3 a4 . . . ak

Platí $w \in L(\mathcal{G})$ právě tehdy, když $S \in X_{1,k}$.

$$X_{1,k} = \{A \mid A \Rightarrow \underbrace{a_1 \dots a_k}_{w}\}$$

$w \in L(\mathcal{G})$ $S \Rightarrow^* w$ iff $w \in X_{1,k}$.

Příklad. Je dána gramatika $\mathcal{G}$ pravidly

$$\begin{aligned} S &\to AB \mid BC \\ A &\to BA \mid a \\ B &\to CC \mid b \\ C &\to AB \mid a \end{aligned}$$

Pomocí algoritmu CYK rozhodněte, zda slovo *aabab* je generováno bezkontextovou gramatikou $\mathcal{G}$.

Řeš.:

$AB \leftarrow S, C$

$BA \leftarrow A$

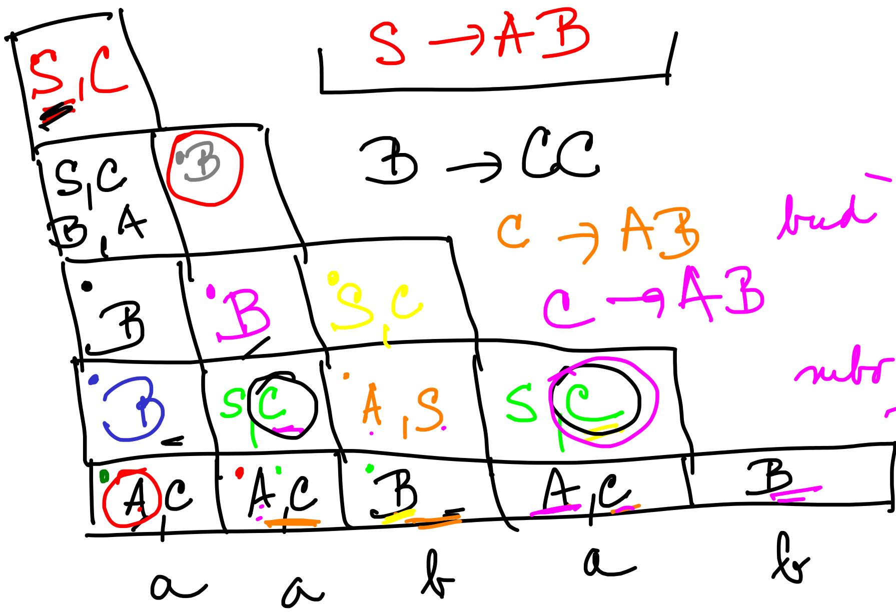
$BC \leftarrow S$

$CC \leftarrow B$

$w = aabab$

$\{D \mid D \to a \in P\} = \{A, C\}$

$\{D \mid D \to b \in P\} = \{B\}$



$\underline{S \to AB}$

$B \to CC$

$C \to AB$

$C \to AB$

buď $\longrightarrow$ ~~AA~~, ~~AS~~, ~~CA~~, ~~CS~~

nebo

$B \longleftrightarrow$ ~~SA~~, ~~SC~~, ~~CA~~, $\underline{CC}$

$S \to AB$
$C \to AB$
$B \to CC$

$B \to CC$

$D \Rightarrow^* aa$

$D \to UV$

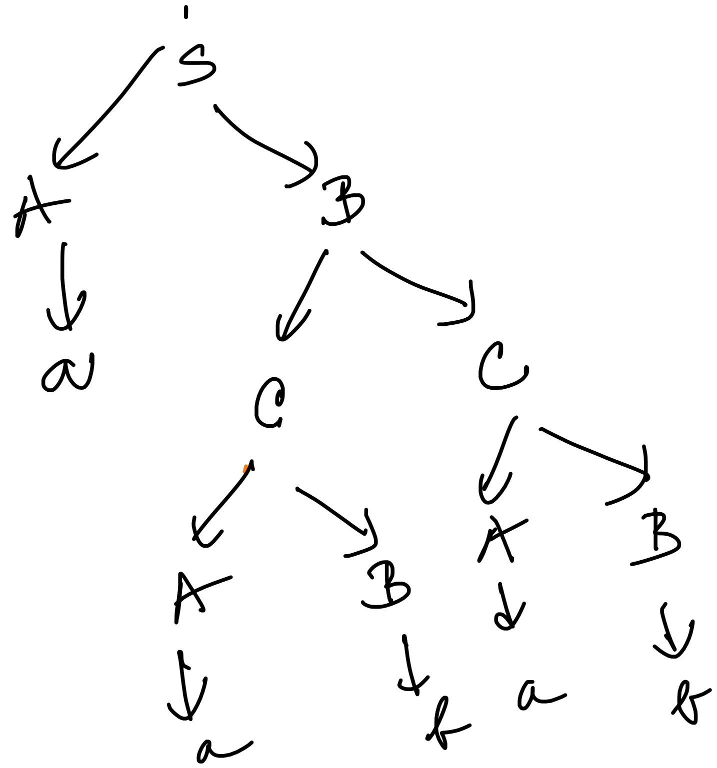
$A \to BA \in P$

$S \to BC \in P$

X_{13} $B \longrightarrow$ ~~AS~~, ~~AC~~, ~~CS~~, CC nebo $\longrightarrow$ BB ano, $S \in X_{15}$ tudíž

$aabab \in L(\mathcal{G})$

Derivační strom je např.



je derivační strom slova <u>aabab</u>

Je vlastně alg. založený na dynamickém programování.

Zásobníkové automaty

L je CF iff existuje CF gramatika $\mathcal{G}$ taková, že $L = L(\mathcal{G})$

zásobníkové automaty jsou automaty, které přijímají právě CF jazyky.

Př: zásobník na talíře v jídelně

dva úkony: přidat talíř a odebrat talíř

chci: pomocí tohoto nástroje zjistit, zda $w \in \Sigma^*$ je tvaru $0^n 1^n$, $n \geq 0$.

N

q_0 když čtu 0 – dám na zásobník talíř

když čtu 1, přechodíme do jiného stavu p a odebereme talíř

v p při čtení 1 odebereme talíř, o havarujeme

Zásobníkový automat je sedmice $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

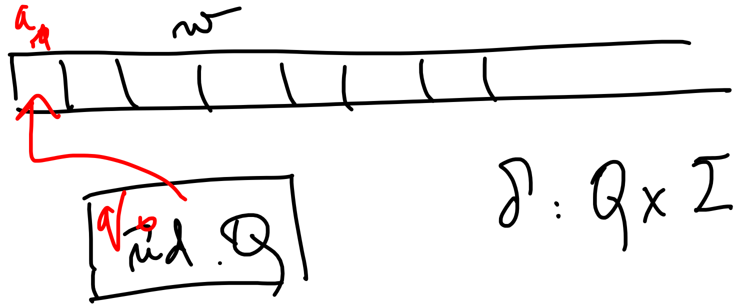
- Q je konečná množina stavů,
- Σ je konečná množina vstupních symbolů, neprázdná
- Γ je konečná množina zásobníkových symbolů,
- δ přiřazuje každé trojici (q, a, X), $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $X \in \Gamma$, konečnou množinu dvojic (p, α), kde $p \in Q$ a $\alpha \in \Gamma^\star$. Formálně:

$$\delta\colon Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_f(Q \times \Gamma^\star).$$

  ($\mathcal{P}_f(A)$ značí množinu všech konečných podmnožin množiny A.)
- $q_0 \in Q$ je počáteční stav,
- $Z_0 \in \Gamma$ je počáteční zásobníkový symbol a
- $F \subseteq Q$ je množina koncových stavů.

DFA

NFA



DFA

$\delta : Q \times \Sigma \to Q$

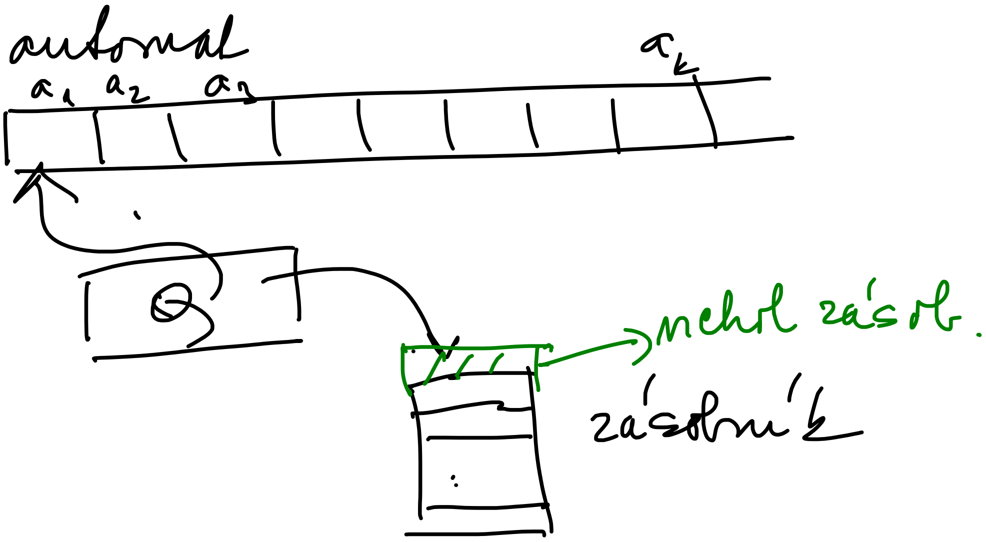
NFA

$Q \times \Sigma \to \mathcal{P}(Q)$

ε-NFA

$Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

Zásobníkový automat



Z_0 ... podložka, T talíř

$\Gamma = \{Z_0, T\}$ $\quad z_0 = Z_0$

$Q = \{q_0, p\}$ $\quad q_0 = q_0$

$\Sigma = \{0, 1\}$

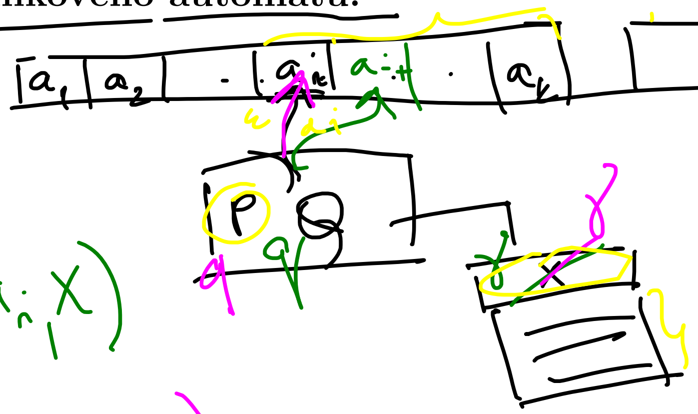
$\delta(q_0, 0, X) = (q_0, TX)$ $\quad X = T$, $X = Z_0$

$\delta(q_0, 1, T) = (p, \varepsilon)$

$\delta(p, 1, T) = (p, \varepsilon)$ $\quad \delta(p, \varepsilon, Z_0) = (p, \varepsilon)$

$p \in F$

Situace zásobníkového automatu.



$a_i \in \Sigma$

$\delta(p, a, X)$

$a = \begin{cases} \varepsilon \\ a_i \end{cases}$

$a = \varepsilon$

$(q, \gamma) \in \delta(p, a_i, X)$

$(q, \gamma) \in \delta(p, \varepsilon, X)$

$(p, aw, X\alpha)$ situace

$p \in Q$, $a = \begin{cases} \varepsilon \\ a_i \in \Sigma \end{cases}$ $w \in \Sigma^*$, $X \in \Gamma$ $\alpha \in \Gamma^*$

Relace $\vdash_A^\star$.

krok

$(p, aw, X\alpha)$

if $(q, \gamma) \in \delta(p, a, X)$, tak

$(p, aw, X\alpha) \vdash_M (q, w, \gamma\alpha)$

X je nahrazeno γ

chceme-li, aby po provedení kroku se nezměnil zásobník, tak

$(q, X) \in \delta(p, a, X)$

$\vdash_M^*$ to je reflexivní a tranzitivní uzávěr $\vdash_M$

tj. $S \vdash^* S'$ iff

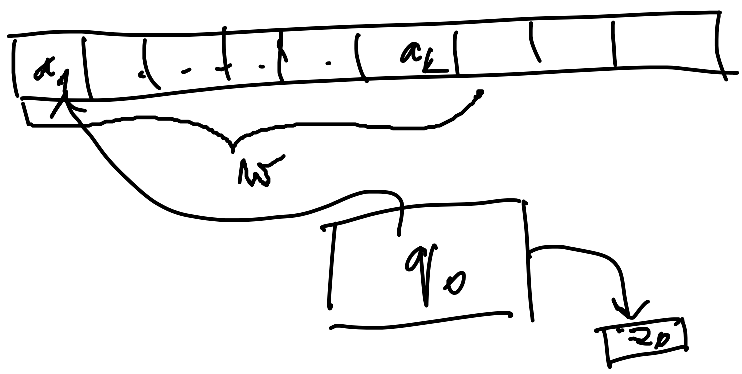
buď $S = S'$

nebo existují situace $S_1, S_2, \ldots, S_n$

tak, že $S = S_1 \quad S_1 \vdash S_2 \vdash \ldots \vdash S_n$

$S_n = S'$

Počáteční situace ZA M



Přednáška 30. listopadu 2020

Čím jsme skončili minulou přednášku.

Zásobníkové automaty

počáteční zás. symbol

ZA $\quad A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

zásobníkových symbolů

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \longrightarrow \mathcal{P}_f(Q \times \Gamma^*)$$

situace ZA $\quad (q, w, \gamma)$

w je dosud nepřečtené slovo vstupní, γ je stav zásobníku

$$(q, au, X\alpha) \vdash (p, u, \beta\alpha) \text{ iff}$$
$$(p, \beta) \in \delta(q, a, X)$$

počáteční situace práce nad slovem w

je (q_0, w, Z_0) $\quad$ na zásobníku je Z_0

~~Situace a počáteční situalce zásobníkového automatu.~~

$\vdash^*$ na množině všech situací

je reflexivní a transitivní uzávěr $\vdash$

bý S, S' jsou situace

$S \vdash^* S'$ iff buď $S = S'$

nebo $S \vdash S_2 \vdash S_3 \vdash \ldots \vdash S_k \vdash S'$

pro $k \geq 0$

Stavový diagram zásobníkového automatu.

ZA může být zadán (δ ZA)

buď tabulkou

sloupce jsou označeny (a, X) $a \in \Sigma \cup \{\varepsilon\}$, $X \in \Gamma$

nebo stavovým diagramem, který se liší od stavového diagramu FA jen označením hran

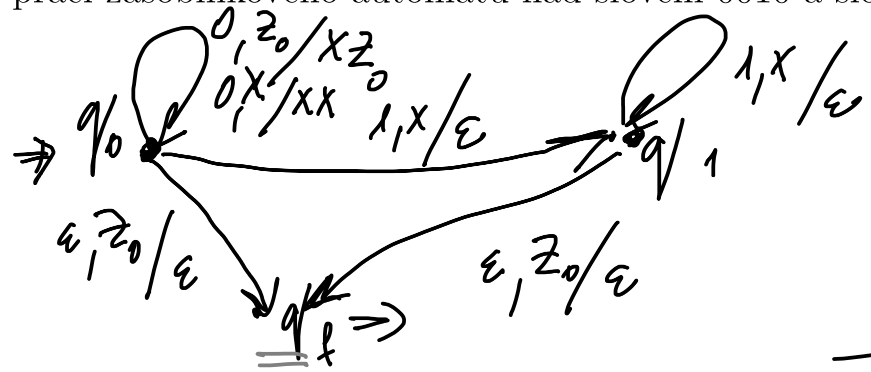
když $(p, \gamma) \in \delta(q, a, X)$ $\quad q \xrightarrow{a, X/\gamma} p$

Příklad. Je dán zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde $Q = \{q_0, q_1, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{Z_0, X\}$ a přechodová funkce je daná tabulkou

	$(0, Z_0)$	$(0, X)$	$(1, Z_0)$	$(1, X)$	(ε, Z_0)	(ε, X)
$\rightarrow q_0$	(q_0, XZ_0)	(q_0, XX)	$-$	(q_1, ε)	(q_f, ε)	$-$
q_1	$-$	$-$	$-$	(q_1, ε)	(q_f, ε)	$-$
$\leftarrow q_f$	$-$	$-$	$-$	$-$	$-$	$-$

1. Nakreslete stavový diagram zásobníkového automatu A.
2. Ukažte práci zásobníkového automatu nad slovem 0010 a slovem 0011.

a)



$\boxed{a, X/\gamma}$

b) $w = 0010$ $\qquad \delta(q_0, \varepsilon, Z_0) = (q_f, \varepsilon)$

práce A nad w $\overset{1.}{\vdash}$ $(q_0, 0010, \varepsilon)$ musíme skončit a to neúspěšně

$\underline{(q_0, 0010, Z_0)}$ $\overset{2.}{\vdash}$ $(q_0, 010, XZ_0) \vdash (q_0, 10, XXZ_0)$

$\vdash (q_1, 0, XZ_0)$ nex. další přechod

c) $u = 0011$ nedíláme slepou větev

$(q_0, 0011, Z_0) \vdash (q_0, 011, XZ_0) \vdash (q_0, 11, XXZ_0)$

$\vdash (q_1, 1, XZ_0) \vdash (q_1, \varepsilon, Z_0) \vdash \underline{\underline{(q_f, \varepsilon, \varepsilon)}}$

Zde je $L(A) = \{0^n 1^n \mid n \geq 0\}$

$N(A) = \{0^n 1^n \mid n \geq 0\}$

(pro slovo $0^3 1^4$ sice vyprázd.

$(q_0, 0^3 1^4, Z_0) \vdash^* (q_1, 1, Z_0) \vdash (q_f, 1, \underline{\underline{\varepsilon}})$

$0^3 1^4 \notin N(A)$

Jazyk přijímaný koncovým stavem $L(A)$, kde $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je zásobníkový automat, je definován takto:

$$L(A) = \{u \in \Sigma^\star \mid (q_0, u, Z_0) \vdash_A^\star (p, \varepsilon, \gamma), p \in F\}.$$

$\gamma \in \Gamma^*$

je to obdoba ε-NFA (ovšem ZA má „paměť" v zásobníku)

(ZA je definován jako nedeterministický s ε-přechody)

Jazyk přijímaný prázdným zásobníkem $N(A)$, kde $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je zásobníkový automat, je definován takto:

$$N(A) = \{u \in \Sigma^\star \mid (q_0, u, Z_0) \vdash_A^\star (p, \varepsilon, \varepsilon), p \in Q\}.$$

přijímací situace

jsme schopni přečíst celé slovo a současně vyprázdnit zásobník.

Má dva typy jazyků pro A

$$A \longrightarrow \begin{array}{l} L(A) \\ N(A) \end{array} \quad \begin{array}{l} (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F) \\ (Q, \Sigma, \Gamma, \delta, q_0, Z_0) \end{array}$$

Tvrzení.
Ke každému zásobníkovému automatu A existuje zásobníkový automat B takový, že

$$N(A) = L(B).$$

D.: Máme $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$.
Zkonstruujme $B = (Q', \Sigma, \Gamma', \delta', q_0', Z_0', F')$

$Q' = Q \cup \{q_0', q_f\}$ $\quad q_0', q_f \notin Q$

$\Gamma' = \Gamma \cup \{Z_0'\}$ $\quad Z_0' \notin \Gamma$ $\quad F' = \{q_f\}$

δ' obsahuje δ + přidáme

$$\delta'(q_0', \varepsilon, Z_0') = (q_0, Z_0 Z_0')$$

tímhle přechod způsobí, že „na dně zásobníku je stále Z_0'"

dál pracujeme v A

Tj. vyprázdnění zásobníku A

$$(q_0', w, Z_0') \vdash_B (q_0, w, Z_0 Z_0') \vdash_B^* (q, \varepsilon, \varepsilon Z_0')$$

kde $(q_0, w, Z_0) \vdash_A^* (q, \varepsilon, \varepsilon)$

$$\delta'(q, \varepsilon, Z_0') = (q_f, Z_0) \quad \forall q \in Q$$

δ' je δ + 2 modrá pravidla

Platí $N(A) = L(B)$

Tvrzení.
Ke každému zásobníkovému automatu A existuje zásobníkový automat B takový, že

$$L(A) = N(B).$$

D.: $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Zkonstruujeme B

$B = (Q', \Sigma, \Gamma', \delta', q_0', Z_0')$

$Q' := Q \cup \{q_0', q_M\}$ q_M „mazací" stav

$\Gamma' := \Gamma \cup \{Z_0'\}$ $q_0', q_M \notin Q$

δ':

k δ přidáme přechody

$\delta'(q_0', \varepsilon, Z_0') = (q_0, Z_0 Z_0')$

pro každé $q \in F$

$\delta(q, \varepsilon, X) = (q_M, \varepsilon)$ $X \in \Gamma'$

$\delta(q_M, \varepsilon, X) = (q_M, \varepsilon)$ $X \in \Gamma'$

Tvrdíme: $N(B) = L(A)$

a) $L(A) \subseteq N(B)$ je zřejmé

$(q_0, w, Z_0) \vdash_A^* (q, \varepsilon, \gamma)$ $q \in F$

$(q_0', w, Z_0') \vdash_B (q_0, w, Z_0 Z_0') \vdash_{A,B}^* (q, \varepsilon, \gamma Z_0') \vdash$

$\vdash_B (q_M, \varepsilon, \gamma Z_0') \vdash_B^* (q_M, \varepsilon, \varepsilon)$

tj. $w \in N(B)$

b) $N(B) \subseteq L(A)$

$(q_0', w, Z_0') \vdash$ $\boxed{(q_0, w, Z_0 Z_0') \vdash_B^* (q, \varepsilon, \gamma Z_0')} \vdash \cdots \vdash (\underline{\underline{q_M}}, \varepsilon, \varepsilon)$

1\. přechod

poslední přechod

$(q_0, w, Z_0 Z_0')$

(q, ε, Z_0')

nebo (q_M, ε, Z_0')

* tato část je práce A s jedinou změnou: na dně zásobníku je Z_0' a $q \in F$

tj. $(q_0, w, Z_0) \vdash_A^* (q, \varepsilon, \gamma)$

$\underline{\underline{q \in F}}$

tj. $w \in L(A)$.

Věta. Ke každé CF gramatice $\mathcal{G} = (N, \Sigma, S, P)$ existuje zásobníkový automat A takový, že

$$L(\mathcal{G}) = N(A).$$

D: $\mathcal{G} = (N, \Sigma, S, P)$ $\quad P: X \to \alpha$

$X \in N, \ \alpha \in (N \cup \Sigma)^*$

Vytvoříme ZA $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$

$Q := \{q\} \qquad q_0 = q \qquad Z_0 = S$

$\Gamma := N \cup \Sigma$

δ:

$\delta(q, \varepsilon, X) = \{(q, \alpha) \mid X \to \alpha \in P\} \qquad X \in N$

$\delta(q, a, a) = \{(q, \varepsilon)\} \qquad a \in \Sigma$

→ je kontrola, že vrchol zás. je stejný jako první čtené písmeno.

Př. $\mathcal{G}$: $N = \{S, A\} \quad \Sigma = \{0, 1, 2\}$

$S \to 0S1 \mid A$

$A \to 2A \mid \varepsilon$

$L(\mathcal{G}) = \{0^n 2^k 1^n \mid n, k \geq 0\}$

$A = (\{q\}, \Sigma, \Gamma, \delta, q, S)$

$\Gamma = \{S, A, 0, 1, 2\}$

$$\delta(q, \varepsilon, S) = \{(q, 0S1), (q, A)\}$$
$$\delta(q, \varepsilon, A) = \{(q, 2A), (q, \varepsilon)\}$$
$$\delta(q, a, a) = (q, \varepsilon) \qquad a \in \{0, 1, 2\}$$

$$S \Rightarrow 0S1 \underset{S \to A}{\Rightarrow} 0A1 \underset{A \to 2A}{\Rightarrow} 02A1 \underset{A \to \varepsilon}{\Rightarrow} 021$$

sl. v.

$$(q, 021, S) \vdash (q, \underline{0}21, \underline{0}S1) \vdash (q, 21, S1)$$
$$\vdash (q, 21, A1) \vdash (q, 21, 2A1) \vdash$$
$$(q, 1, A1) \vdash (q, 1, 1) \vdash (q, \varepsilon, \varepsilon).$$

Indukcí podle délky buď odvození (pro $L(\mathcal{G}) \subseteq N(A)$) nebo délky práce A (pro $N(A) \subseteq L(\mathcal{G})$) se dokáže rovnost $L(\mathcal{G}) = N(A)$.

Příklad. Je dána bezkontextová gramatika $\mathcal{G}$ pravidly:

$$\begin{aligned} S &\to aSa \mid A \\ A &\to bA \mid \varepsilon \end{aligned}$$

Zkonstruujte zásobníkový automat přijímající jazyk $L(\mathcal{G})$.

$$L(\mathcal{G}) = \{ a^n b^k a^n \mid n, k \geq 0 \}$$

Každý může udělat.

Věta. Ke každému zásobníkovému automatu A existuje bezkontextová gramatika $\mathcal{G}$ taková, že

CF

$$N(A) = L(\mathcal{G}).$$

Idea důkazu.

Ta <u>těžší část</u> je ukázat, že ke každému ZA A existují ZA <u>B</u> s <u>jediným stavem</u>, který splňuje $N(A) = N(B)$.

Pak už se dá udělat "opačná" konstrukce k konstrukci z minulé věty.

Kdo chce důkaz i s detaily odkazuji na Chytil Gramatiky . . .

Deterministický zásobníkový automat

DZA ... ZA automat, kde vždy máte nejvýše jedinou možnost, jak pokračovat.

Zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je **deterministický zásobníkový automat**, jestliže

- Pro každé $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ a $X \in \Gamma$ je $|\delta(q, a, X)| \leq 1$.
- Jestliže pro nějaké $q \in Q$ a $X \in \Gamma$ je $\delta(q, \varepsilon, X) \neq \emptyset$, pak pro každé $a \in \Sigma$ je $\delta(q, a, X) = \emptyset$.

nemůžeme se rozhodovat, že nepřečteme vstupní symbol nebo ho přečteme.

Př. $L = \{0^n 1^n \mid n \geq 1\}$

pro L je možné udělat DZA

(v 1. příkladu definujeme $\delta(q_0, \varepsilon, Z_0) = \emptyset$.)

Tady neplatí, že přijímání prázdným zásobníkem je to samé jako koncovým stavem.

Neplatí, že každý jazyk, který je $L(B)$ pro nějaký DZA B je přijímán DZA A, $L = N(A)$.

Jazyky přijímané deterministickým zásobníkovým automatem.

Třída deterministických jazyků je třída $\{L \mid L = L(A)$ pro DZA $A\}$

Jazyk L je **deterministický**, jestliže je přijímán deterministickým zásobníkovým automatem koncovým stavem.

příkladem je např.

$$L = \{wcw^R \mid w \in \{a,b\}^+\} \subseteq \{a,b,c\}^*$$

Jazyk L je **bezprefixový**, jestliže je přijímán deterministickým zásobníkovým automatem prázdným zásobníkem.

Třída bezprefixových jazyků (CF jazyků)

$\{L \mid$ existuje DZA A, že $L = N(A)\}$

Platí: $w \in N(A)$ pro A DZA, tak $wu \notin N(A)$ pro žádné u, $|u| \geq 1$.

Přednáška 7. prosince 2020

Čím jsme skončili minulou přednášku.

Jazyk L je **deterministický**, jestliže je přijímán deterministickým zásobníkovým automatem koncovým stavem.

DZA $\quad (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$\forall a \in \Sigma \cup \{\varepsilon\}$ $\forall X \in \Gamma$ platí

1) $|\delta(q, a, X)| \le 1$ 2) když $\delta(q, \varepsilon, X) \neq \emptyset$ tak $\delta(q, a, X) = \emptyset$ $\forall a \in \Sigma$

Jazyk L je **bezprefixový**, jestliže je přijímán deterministickým zásobníkovým automatem prázdným zásobníkem.

L je determ. iff ex. DZA A, že $L = L(A)$

je přijímáno konc. stavem

$w \in L$ iff $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \alpha)$ pro $q \in F$

L je bezprefix. iff ex. DZA B, že $L = N(B)$

$w \in L$ iff $(q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$

---

bezprefixový: pro bezprefixový jazyk L neexistují $u, v \in L$, $u \neq v$, že u je prefix v.

$v = ux \quad |x| \neq \varepsilon \qquad u \in L = N(B)$

$(q_0, v, Z_0) = (q_0, ux, Z_0) \vdash^* (q, x, \varepsilon)$

další krok není definován. $ux \notin N(B)$

Tvrzení. Pro každý deterministický zásobníkový automat A existuje deterministický zásobníkový automat B takový, že

$$N(A) = L(B).$$

D.: A DZA $(Q_A, \Sigma, \Gamma_A, \delta_A, q_A, Z_A)$

$L = N(A)$

B $\quad Q_B = Q_A \cup \{q_B, q_f\}$ $\quad \Gamma_B = \Gamma_A \cup \{Z_B\}$

$q_B, q_f \notin Q_A$ $\quad F = \{q_f\}$

$B = (Q_B, \Sigma, \Gamma_B, \delta_B, q_B, Z_B, F)$

δ_B $\quad \delta_A$ s dalšími přechody

$$\delta_B(q_B, \varepsilon, Z_B) = (q_A, Z_A Z_B)$$

$$\delta_B(q, \varepsilon, Z_B) = (q_f, Z_B)$$

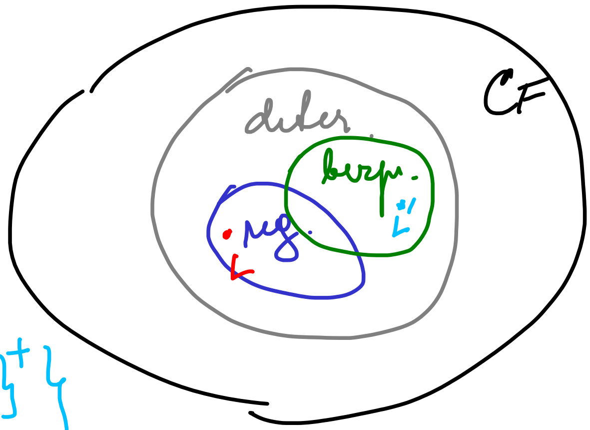
$\forall q \in Q_A$ $\quad Z_B \notin \Gamma_A$

Když A byl DZA, tak B je také DZA.

Vztah mezi třídami regulárních jazyků, bezprefixových jazyků a deterministických jazyků.

Reg. jazyky, CF, determ. jazyky a bezprefix.

• $L = \{a, aa\}$



na DFA se můžeme dívat jako na DZA $\Gamma = \{Z\}$

• $L' = \{wcw^R \mid w \in \{a,b\}^+\}$
...

Uzávěrové vlastnosti bezkontextových jazyků

Reg. jazyky jsou uzavřené na sjednocení, zřetězení, doplněk, průnik, Kleeneho *, reverzi, substituci, homomorf., inverzní homom.

Tvrzení. Bezkontextové jazyky jsou uzavřeny na sjednocení, zřetězení, Kleeneho operaci $\star$.

D.: Když L_1, L_2 jsou CF, tak $L_1 \cup L_2$, $L_1 \cdot L_2$, L_1^* jsou CF

$L_1 \ldots \mathcal{G}_1 = (N_1, \Sigma, S_1, P_1)$ $L_2 \ldots \mathcal{G}_2 = (N_2, \Sigma, S_2, P_2)$

$$N_1 \cap N_2 = \emptyset$$

$L_1 \cup L_2 \quad \ldots \quad (N_1 \cup N_2 \cup \{S\}, \Sigma, S, P_{L_1 \cup L_2})$

$S \notin N_1 \cup N_2$

$P_{L_1 \cup L_2}$: P_1 a P_2 plus

$$S \to S_1 \mid S_2$$

nic nového: první pravidlo musí být buď $S \to S_1$ anebo $S \to S_2$ a generujeme v $\mathcal{G}_1$ / gen. v $\mathcal{G}_2$

$L_1 \cdot L_2 \quad (N_1 \cup N_2 \cup \{S\}, \Sigma, S, P_{L_1 L_2})$ $S \notin N_1 \cup N_2$

$P_{L_1 L_2}$ $\quad P_1, P_2$ a $S \to S_1 S_2$

L_1^* $\quad (N_1 \cup \{S\}, \Sigma, S, P_{L_1^*})$

$S \notin N_1$

$P_{L_1^*}$ $\quad P_1$ přidáme $S \Rightarrow SS_1 \mid \varepsilon$

$S \Rightarrow^* S S_1^n \overset{S \to \varepsilon}{\Rightarrow} \underbrace{S_1 \dots S_1}_{n\text{-krát}}$

Tvrzení. Bezkontextové jazyky **nejsou** uzavřeny na průnik.

D

$$L_1 = \{a^i b^j c^k \mid i = j, \ i,j,k \geq 0\} = \{a^i b^i \mid i \geq 0\} . \{c\}^*$$

CF reg

$N = \{S, A, B\}$

$S \to AB \quad A \to aAb \mid \varepsilon \quad B \to cB \mid \varepsilon$

$$L_2 = \{a^i b^j c^k \mid j = k, \ i,j,k \geq 0\}$$

$\{a\}^* \{b^j c^j\}$

$N = \{S', C, D\}$

$S' \to CD \quad C \to aC \mid \varepsilon \quad D \to bDc \mid \varepsilon$

$$L_1 \cap L_2 = \{a^i b^j c^k \mid i = j, \ j = k\} = \{a^i b^i c^i \mid i \geq 0\} \text{ není CF}$$

CF jazyky nejsou uzavřeny na doplňky

Když by byly, tak $\overline{L_1}, \overline{L_2} \in CF$ pro $L_1, L_2 \in CF$

Spor a $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ $\quad$ $\overline{L_1} \cup \overline{L_2}$ by byl CF a jeho doplněk také

Tvrzení. Třída bezkontextových jazyků je uzavřena na průniky s regulárními jazyky.

D.: L je CF nad Σ, R je regulární

Chceme dokázat, že $L \cap R$ je CF.

L je CF iff existuje ZA A tak, že $L = L(A)$

$$A = (Q_A, \Sigma, \Gamma, \delta_A, q_A, Z_A, F_A)$$

R je regulární jazyk iff existuje DFA M, že $R = L(M)$; $M = (Q_M, \Sigma, \delta_M, q_0, F_M)$

Sestrojíme ZA B tak, že $L(B) = L \cap R$

B ... součinovou konstrukcí

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

$$Q := Q_A \times Q_M \qquad F := F_A \times F_M$$

$$\delta((q,r), a, X) \qquad a \in \Sigma \cup \{\varepsilon\}$$

$$= \{ ((p,s), \gamma) \mid \begin{cases} s = \delta_M(r,a) & \text{pro } a \in \Sigma \\ s = r & \text{pro } a = \varepsilon \end{cases} \quad s = \delta_M^*(r,a)$$

$$(p,\gamma) \in \delta_A(q,a,X) \}$$

máme dokázat, že $L(B) = L \cap R$.

a) $L \cap R \subseteq L(B)$ (tj. každé $w \in L \cap R$ je přijato B.)

$w \in L \cap R$ $\quad ((q_A, q_0), w, Z_A)$

$w \in L$ $\quad (q_A, w, Z_A) \vdash_A^* (q, \varepsilon, \alpha) \qquad q \in F_A$

$w \in R$ $\quad \delta_M^*(q_0, w) = r \in F_M$

tak jednoduchou indukcí se dokáže

$$((q_A, q_0), w, Z_A) \vdash_B^* ((q, r), \varepsilon, \alpha)$$

$$(q, r) \in F_A \times F_M$$

b) $L(B) \subseteq L \cap R$

Protože $F_B = F_A \times F_M$

$$((q_A, q_0), w, Z_A) \vdash_B^* ((p, s), \varepsilon, \beta)$$

$(p, s) \in F_B = F_A \times F_M$ tj. $p \in F_A$, $s \in F_M$

$w \in L(A) = L \qquad w \in L(M) = R$

Andiž $\quad w \in L \cap R$.

cbd

Tvrzení. Bezkontextové jazyky jsou uzavřeny na reverzi.

D.: Když L je CF, tak L^R je CF.

$$L^R = \{ w^R \mid w \in L \}.$$

Když $\mathcal{G} = (N, \Sigma, S, P)$, že $L(\mathcal{G}) = L$

tak $\mathcal{G}^R = (N, \Sigma, S, P^R)$

$$X \to \alpha \in P^R \text{ iff } X \to \alpha^R \in \mathcal{G}.$$

Levé derivaci $S \Rightarrow^*_{\mathcal{G}} w$

odpovídá pravá derivace $S \Rightarrow^*_{\mathcal{G}^R} w^R$.

Tvrzení. Třída bezkontextových jazyků je uzavřena na substituce a tedy i homomorfismy.

Připomeňme pojem substituce:

Σ, abeceda Π abeceda

substituce z Σ do Π je

$\sigma : a \longmapsto \sigma_{(a)} \subseteq \Pi^*$

$a \in \Sigma_1^*$ → je jazyk

$\varepsilon \longmapsto \phi$

$w = a_1 \dots a_k \longmapsto \sigma_{(a_1)} \sigma_{(a_2)} \dots \sigma_{(a_k)}$

$$\sigma(L) = \bigcup \{ \sigma(w) \mid w \in L \}$$

$L \subseteq \Sigma^*$

homomorfismus $f : a \longmapsto f(a) \in \Pi^*$

$a_1 a_2 \dots a_k \longmapsto f(a_1) f(a_2) \dots f(a_k)$

$L \subseteq \Sigma_1^* \quad f(L) = \{ f(w) \mid w \in L \}$

Přesnující změnění tvrzení:

Jestliže $\sigma(a)$ jsou CF jazyky nad Π pro každé $a \in \Sigma$ a jestliže L je CF nad Σ, pak $\sigma(L)$ je CF nad Π.

D.: Máme $\mathcal{G}_a = (N_a, \Pi, S_a, P_a)$ $\forall a \in \Sigma$ a platí $L(\mathcal{G}_a) = \sigma(a)$

máme $\mathcal{G} = (N, \Sigma, S, P) \quad L(\mathcal{G}) = L$

Chceme $\mathcal{G}_1 = (N_1, \Pi, S_1, P_1)$, že $L(\mathcal{G}_1) = \sigma(L)$

$w \in L$

$S \overset{*}{\underset{\mathcal{G}}{\Longrightarrow}} w$ pomocí pravidel P

$w_1 = a_1 \dots a_k \quad \sigma(w) = \sigma(a_1) \dots \sigma(a_k)$

generujeme v S, ale vždy, když na pravé straně máme $a \in \Sigma$!, tak tam dáme místo a, S_a

$$S \Rightarrow^* S_{a_1} S_{a_2} \dots S_{a_k} \Rightarrow^*_{\mathcal{G}_{a_1}} u_1 S_{a_2} \dots S_{a_k} \Rightarrow^*_{\mathcal{G}_{a_2}}$$

$$\mathcal{G}_{a_1} \quad \mathcal{G}_{a_2} \quad \mathcal{G}_{a_k} \qquad u_1 \in \sigma(a_1)$$

$\Rightarrow^* \dots$ dostaneme slovo $u_1 u_2 \dots u_k$

$u_i \in \sigma(a_i)$

$S_1 = S \qquad N_1 = N \cup \{N_a \mid a \in \Sigma\}$

N i N_a jsou po dvou disjunktní

P_1: pravidla P_a, $a \in \Sigma$ + pravidla vzniklá z P tak, že každý terminál a na pravé straně je nahrazen S_a.

---

homomorfismus je zvláštní případ substituce, kde $\sigma(a)$ je jedno slovo (tedy dokonce regulární).

Tvrzení. Třída bezkontextových jazyků je uzavřena na inversní homomorfní obrazy.

nedokazujeme

Bezkontextový jazyk L je **Dyckův jazyk**, jestliže je generován gramatikou $\mathcal{G} = (\{S\}, \Sigma, S, P)$, kde $\Sigma = \{a_1, a_1', a_2, a_2', \ldots, a_n, a_n'\}$ a P:

$$S \to \varepsilon \,|\, SS \,|\, a_1 S a_1' \,|\, \ldots \,|\, a_n S a_n'$$

$$S \to \varepsilon \,|\, a_1 S a_1' \,|\, a_2 S a_2' \,|\, \ldots \,|\, a_n S a_n' \,|\, SS$$

$a_1 = ($ $a_1' =)$ $a_2 = [$ $a_2' =]$ $a_3 = \{$ $a_3' = \}$

$n = 3$

$[\,(\,)\,]\,\{\,[\,\{\,\}\,\{\,\}\,]\,\}$

Dyckův jazyk: je jazyk „správně uzávorkovaných výrazů"

Tvrzení. Pro každý bezkontextový jazyk L existuje regulární jazyk R, Dyckův jazyk D a homomorfismus h takový, že $L = h(R \cap D)$.

$h(R \cap D)$ je CF je zřejmé.

Pro každý L CF najdeme R, $\underline{D}$, h

Dyckův jazyk: každý vložený symbol na zásobník, tak ho ze zásobníku musíme odstranit

$\underbrace{\text{vložení } X}_{a_1}$ odstranění X $\bar{a}_1$

Přednáška 14. prosince 2020

Čím jsme skončili minulou přednášku.

Uzávěrové vlastnosti třídy CF jazyků

1) Třída CF jazyků je uzavřena na sjednocení, zřetězení, Kleeneho operace, reverzi, substituci, homomorfní obrazy, průnik s regulárním jazykem

2) Třída CF jazyků není uzavřena na průniky, doplňky

Říkla jsem věta:

Třída CF jazyků je uzavřena na inverzní homomorfní obrazy.

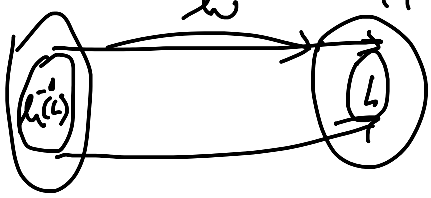
Σ^* $\quad h: \Sigma \to \Pi^*$ $\quad \Pi^*$

rozšíříme

$h: \Sigma^* \to \Pi^*$

$h(a_1 \ldots a_k) = h(a_1) \ldots h(a_k)$

$h(\varepsilon) = \varepsilon$



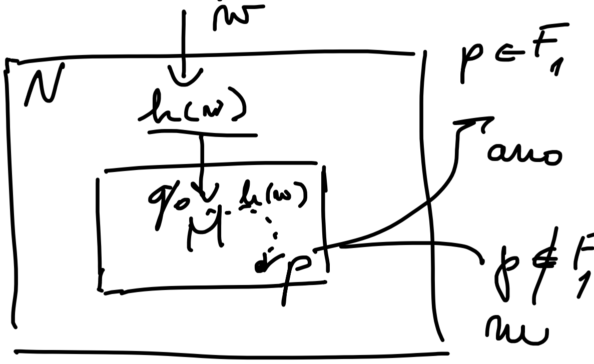
Je-li L CF, je $h^{-1}(L)$ CF.

Tvrzení. Třída regulárních jazyků je uzavřena na inversní homomorfní obrazy.

D.: $\Sigma^* \xrightarrow{h} \Pi^*$ $L \in Reg$ nad Π

$L \in Reg$ tj. existuje DFA $M = (Q_1, \Pi, \delta_1, q_1, F_1)$
že $L = L(M)$.

Chceme DFA $N = (Q, \Sigma, \delta, q_0, F)$
tak, že $L(N) = h^{-1}(L)$



Formálně:

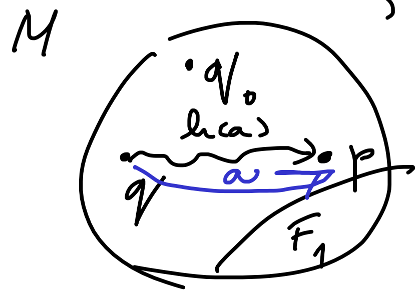
N: $Q := Q_1$ $q_0 := q_1$

$F := F_1$

$$q \in Q \quad \delta(q, a) \stackrel{def.}{=} \delta_1^*(q, h(a))$$
$a \in \Sigma$

$Q = Q_1$

$q_0 = q_1$



N má přechod $q \xrightarrow{a} p$

$F = F_1$

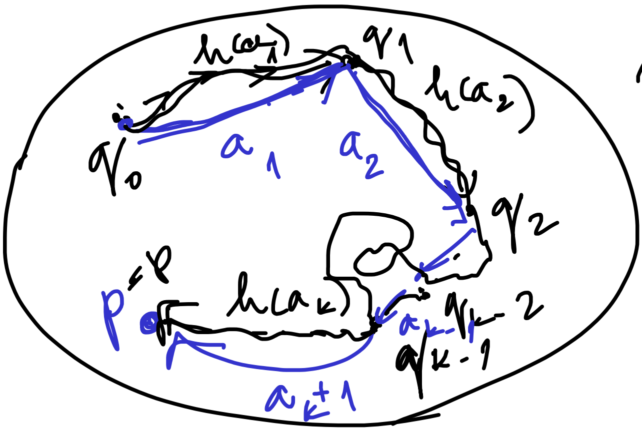
Chceme, ať platilo:

$\delta^*(q, w) = p$ iff $\delta_1^*(q, h(w)) = p$

$q \in Q = Q_1$ $w \in \Sigma^*$

$\underline{N}$

$w = a_1 \dots a_k$



na M s modrými šipkami (hranami)

$$h(w) = h(a_1 \dots a_k) = h(a_1)\, h(a_2) \dots h(a_k)$$

Dokážeme indukcí:

dokazujeme $\delta^*(q, w) = p$ iff $\delta_1^*(q, h(w)) = p$

indukcí podle $|w| = k$

1) $|w| = 0$, $k = 0$, $w = \varepsilon$

$\delta^*(q, \varepsilon) = q$ $\qquad \delta_1^*(q, h(\varepsilon)) = q$

ano pro $k = 0$ platí $\qquad h(\varepsilon) = \varepsilon$

2) Indukční předpoklad: Platí

$\delta^*(q, w) = p$ iff $\delta_1^*(q, h(w)) = p$

pro $|w| = k$.

Vezmeme $u \in \Sigma_1^*$ $\qquad |u| = k + 1$

$u = va$ $\qquad |v| = k$

$a \in \Sigma_1$

$$\delta^*(q, va) = \delta(\underbrace{\delta^*(q, v)}_{r \,=\, \delta_1^*(q, h(v))}, a) = \delta(r, a)$$

$$\stackrel{def}{=} \delta_1^*(r, h(a)) = \delta_1^*(\delta_1^*(q, h(v)), h(a)) = \delta_1^*(q, h(v)h(a))$$

IP $\qquad$ homomorf.: $h(v)h(a) = h(va) = h(u)$ atd.

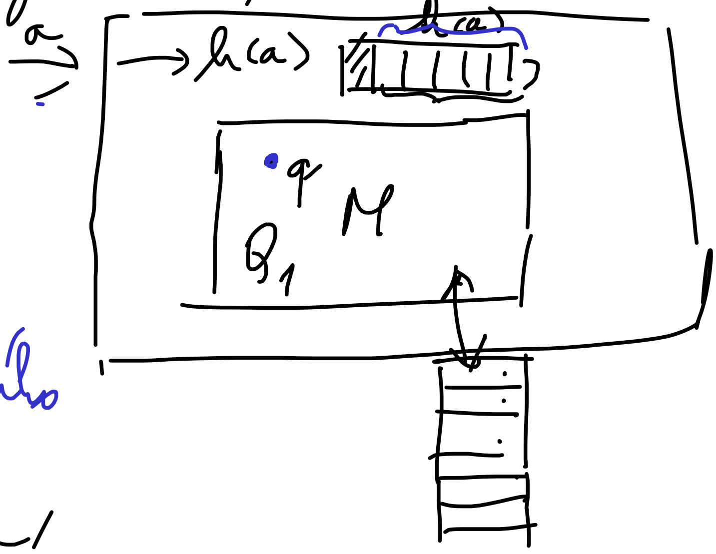
Tvrzení. Třída CF jazyků je uzavřena na inversní homomorfní obrazy.

D.: $h^{-1}(L) \subseteq \Sigma_1^*$ $\quad L \subseteq \Pi^*$ $\quad h : \Sigma_1^* \to \Pi^*$

pro L máme ZA $M = (Q_1, \Pi, \Gamma, \delta_1, q_0, Z_0, F_1)$

$L = L(M)$ $\quad M$ přijímá L koncovým stavem

a $\quad |h(a)| = 7$



stav N

(q, zbytek nepřečteného $h(a)$)

Problém se vyřeší

$Q := \{(q, u) \mid q \in Q_1,\ u$ je suffix nějakého $h(a),\ a \in \Sigma_1\}$

Q je konečná množina, protože Q_1 je konečná, $h(a)$ je konečné, tak má $|h(a)|+1$ suffixů, Σ je konečná.

δ: 1) $\delta((q, \varepsilon), a, X) = \{((q, h(a)), X)\}$ $\quad a \in \Sigma_1 \quad X \in \Gamma$

2) if $(p, \gamma) \in \delta_1(q, b, X)$ $\quad X \in \Gamma,\ b \in \Pi \cup \{\varepsilon\}$

M

$N: \quad ((p,u),\gamma) \in \delta((q, bu), \varepsilon, X)$

vyprázdňujeme buffer, na začátku bylo $\underbrace{h(a)}$

Dá se dokázat: (ne težká indukce)

$$\underline{((q_0,\varepsilon), w, Z_0)} \vdash_N^* \underline{((p,\varepsilon), \varepsilon, \gamma)} \quad \text{iff}$$

$$(q_0, h(w), Z_0) \vdash_M^* (p, \varepsilon, \gamma)$$

To stačí k tomu, aby $w \in L(N)$ iff $h(w) \in L(M)$.

$$N = (Q, \Sigma, \Gamma, (q_0,\varepsilon), Z_0, F \times \{\varepsilon\})$$

to už se nevztahuji k homomorfismu

Použití CF gramatik v programovacích jazycích

Příklad. Gramatika odpovídající správnému použití konstrukce **if ... then**, **else** v C.

$\mathcal{G} = (N, \Sigma, S, P)$, kde $N = \{S\}$, $\Sigma = \{i, e\}$ a P je

$$S \to SS \mid iS \mid iSe \mid \varepsilon$$

i ... if then

e ... else

$$S \to iS \mid iSe \mid SS \mid \varepsilon$$

v $L(\mathcal{G})$ leží slova w, která splňují:

pro každý prefix u slova w platí

$|u|_i \geq |u|_e$.

Co je potřeba pro analýzu shora

$\mathcal{G}$ je CF gramatika

CYK .. pro dané slovo w rozhodne, zda $w \in L(\mathcal{G})$.

Analýza zdola

$\mathcal{G}$ CF

Spojení pravidel. Máme pravidlo $A \to \alpha B \beta$. Jestliže

$$B \to \gamma_1 \,|\, \ldots \,|\, \gamma_k$$

jsou všechna pravidla s levou stranou B, pak nahradíme $A \to \alpha B \beta$ pravidly

$$A \to \alpha\gamma_1\beta \,|\, \ldots, \,|\, \alpha\gamma_k\beta$$

Tím nezměníme co je vygenerováno.

$$\underbrace{(1) \quad A \to \alpha B \beta \qquad B \to \gamma_1 | \ldots | \gamma_k}$$

$$\Downarrow$$

$$(2) \quad A \to \alpha\gamma_1\beta \,|\, \alpha\gamma_2\beta \,|\, \ldots \,|\, \alpha\gamma_k\beta$$

Z pravidel $(1)^{+P}$ vygenerujeme přesně stejná terminální slova jako, když v P zaměníme (1) za pravidla (2).

ig

Odstranění levé rekurze. Máme pravidla

$$A \to A\alpha_1,\ A \to A\alpha_2,\ \ldots,\ A \to A\alpha_k$$

a pravidla

$$A \to \beta_1,\ A \to \beta_2,\ \ldots,\ A \to \beta_m$$

jsou všechna ostatní pravidla.

Pak tato pravidla nahradíme pravidly

$$A \to \beta_i,\ A \to \beta_i Z,\ Z \to \alpha_j,\ Z \to \alpha_j Z, i = 1, \ldots m, j = 1, \ldots, k;$$

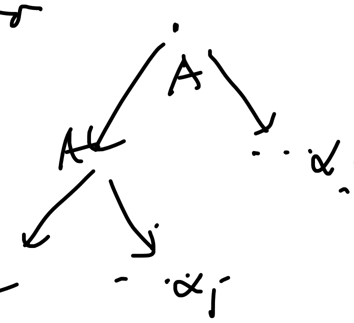
kde Z je nový neterminál.

Tím nezměníme co je vygenerováno.

$$A \to A\alpha_i \xRightarrow{A \to A\alpha_j} A\alpha_j\alpha_i \xRightarrow{A \to A\alpha_n} A\alpha_n\alpha_j\alpha_n \ldots$$

když chceme vyg. terminální slovo

$$\xRightarrow{A \to \beta_l} \beta_l \underbrace{\alpha_i\alpha_j\alpha_n \ldots \alpha_s}$$



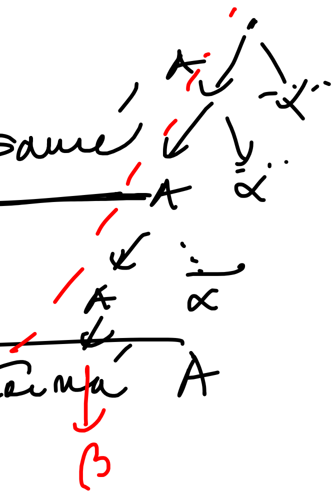
nahradíme pravou rekurzí

$$\boxed{\begin{aligned} A &\to \beta_1 \mid \ldots \mid \beta_m \mid \beta_1 Z \mid \ldots \mid \beta_m Z \\ Z &\to \alpha_1 \mid \alpha_1 Z \mid \alpha_2 \mid \alpha_2 Z \mid \ldots \mid \alpha_k \mid \alpha_k Z \end{aligned}}$$

bez $\smile$ se nezbavíme Z

na googlu najdete

$$\left.\begin{aligned} A &\to \beta_1 Z \mid \ldots \mid \beta_m Z \\ Z &\to \alpha_1 Z \mid \ldots \mid \alpha_k Z \mid \varepsilon \end{aligned}\right\}$$ to je to samé



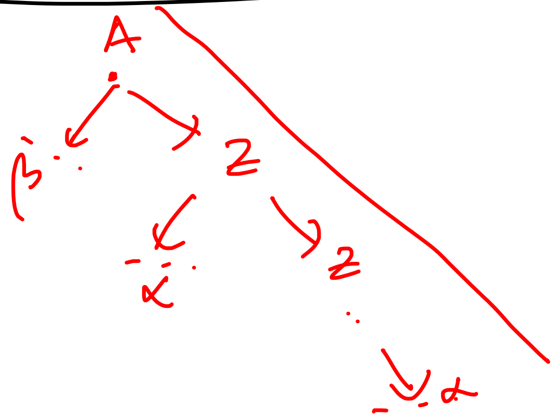
Př: $A \to A\alpha$ $\quad$ $A \to \beta \mid \beta$ nahradí A

$$A \xrightarrow{(3)} A\alpha^3 \Rightarrow \beta\alpha^3$$

$A \to A\alpha$

$A \Rightarrow \beta \underbrace{Z}$ $\qquad Z \overset{*}{\Rightarrow} \alpha^3$

$$A \Rightarrow \beta Z \xRightarrow{Z \to \alpha Z} \beta \alpha Z \xRightarrow{Z \to \alpha Z} \beta \alpha^2 Z \xRightarrow{Z \to \alpha} \beta \alpha^3$$



Greibachové normální forma (tvar).

GNF

CF gramatika $\mathcal{G}$ je v Greibachové normální formě, jestliže všechna pravidla mají tvar:

$$A \to a\alpha, \quad \text{kde } a \in \Sigma \text{ a } \alpha \in N^\star.$$

$A \to b\,CD \qquad A \to b$

CF v GNF negeneruje ε

Věta. Ke každému CF jazyku L existuje CF gramatika v Greibachové normálním tvaru $\mathcal{G}$ taková, že

$$L(\mathcal{G}) = L - \{\varepsilon\}.$$

D: L je CF, tak k němu existuje CF gramatika $\mathcal{G}_1$, že $L = L(\mathcal{G}_1)$.

ke $\mathcal{G}_1$ zkonstruujme nevypouštějící $\mathcal{G}_2$

$L(\mathcal{G}_2) = L - \{\varepsilon\}$

Postup nalezení gramatiky v Greibachové normálním tvaru.

CF $\mathcal{G}$ je nevypouštěcí gramatika.

- Očíslujeme její neterminály, tj. $N = \{A_1, A_2, \ldots, A_n\}$.

$A_1 = S$ startovací symbol.

- Postupným spojováním pravidel a odstraňováním levé rekurze získáme pouze pravidla, kde pravá strana začíná buď terminálem nebo jsou tvaru $A_i \to A_j\alpha$ pro $i < j$.

$A_1 = S \quad A_2 = A \quad B = A_3$

mají
$$S \to A\alpha \mid \beta$$
$$A \to B\beta$$
$$B \to a\gamma$$

$a \in \Sigma$

- Zajistíme, aby pravidla obsahující nové neterminály (vzniklé při odstraňování levé rekurze) také měla požadovaný tvar.

ty se zařadí na konec a zajistíme, aby i tyto neterminály splňovaly podmínky

- Postupným spojováním pravidel (od n do 1) dosáhneme toho, aby pravidla s levou stranou A_i měla tvar $A_i \to a\alpha$, kde $a \in \Sigma$.

$$S \to A\alpha$$
$$A \to B\beta$$
$$B \to a\gamma \mid \ldots$$

- Jestliže nyní máme terminální symbol uvnitř pravé strany některého pravidla, zavedeme za něj nový neterminál.

$$S \to a_1 \alpha_1 \mid a_2 \alpha_2 \mid \ldots$$

$$A \to b_1 \beta_1 \mid \ldots$$

aby byla v GNF α_i, β_j nesmí obsahovat terminál!
pro ní zavedeme nový neterminál $X_b \to b$

Příklad. Dána CF gramatika $\mathcal{G}$ pravidly

$$
\begin{aligned}
E &\to E + T \mid T \\
T &\to T * F \mid F \\
F &\to (E) \mid a
\end{aligned}
$$

Převeďte ji do Greibachové normální formy.

$N := \{E, T, F\}$
$S := E$
$\Sigma := \{+, *, (,), a\}$

Řeš.: 1) $A_1 = E$ $A_2 = T$ A_3 F

2) $E \to T$ musíme odstranit $E \to E + T$

tj $E \to E \underbrace{+T}_{\alpha}$ a $E \to \underbrace{T}_{\beta}$

nahradíme

$$E \to T \mid TX \qquad X \to +T \mid +TX$$

$$T \to T \underbrace{*F}_{\alpha} \mid \underbrace{F}_{\beta}$$

nahradíme $T \to F \mid FY \qquad Y \to *F \mid *FY$

Tj dostali jsme pravidla

$$\boxed{\begin{aligned} E &\to T \mid TX \\ T &\to F \mid FY \\ F &\to (E) \mid a \\ X &\to +T \mid +TX \\ Y &\to *F \mid *FY \end{aligned}}$$

3) $E \to (E\overset{Z}{)} \mid a \mid (E\overset{Z}{)}Y \mid aY \mid (E\overset{Z}{)}X \mid aX \mid (E\overset{Z}{)}YX \mid aYX$

$T \to (E) \mid a \mid (E)Y \mid aY$

$F \to (E) \mid a$

$X \to +T \mid +TX$

$Y \to *F \mid *FY$

Z dosadíme za)

4) $+ \quad Z \to)$

Přednáška 4. ledna 2021

Čím jsme skončili minulou přednášku.

Greibachové normální tvar CF gramatiky

$X \to a\alpha \quad a \in \Sigma, \quad \alpha \in N^*$

Zásobníkové automaty

s nimi můžeme zkoušet pro $\mathcal{G}$, zda jistá $w \in \Sigma^*$, $w \in L(\mathcal{G})$

zásobník

$a_1 \dots a_k \; q \; S$

$a_1 \dots a_k \; q \; \alpha$

$a_1 \dots a_k \; q \; \gamma\beta$

$\delta(q, \varepsilon, S) = \{(q, \alpha) \mid S \to \alpha \in P\}$

$\alpha = X\beta$

$\delta(q, \varepsilon, X) = \{(q, \gamma) \mid X \to \gamma \in P\}$

Pro Greibach. normální formu

$a_1 \dots a_k \quad S$

$a_1 \dots a_k \quad a_1\alpha$

Analýza shora

zkusíme pravidla s levou stranou S

Dána CF gramatika $\mathcal{G}$ a slovo $w \in L(\mathcal{G})$. **Levý rozbor** slova w je posloupnost pravidel levé derivace slova w.

Ukážeme na příkladu

Příklad. Je dána CF gramatika $\mathcal{G}$ pravidly:

$$S \stackrel{1.}{\to} aSa, S \stackrel{2.}{\to} bSb, S \stackrel{3.}{\to} c.$$

Očíslujme jednotlivá pravidla

1. $S \to aSa$
2. $S \to bSb$
3. $S \to c$

Levý rozbor slova *bbacabb* je 2, 2, 1, 3.

$$S \overset{2.}{\Rightarrow} bSb \overset{2.}{\Rightarrow} bbSbb \overset{1.}{\Rightarrow} bbaSabb \overset{3.}{\Rightarrow} bbacabb$$

levá derivace

$(bbacabb; S; -) \overset{2.}{\dashrightarrow} (\cancel{b}bacabb; \cancel{b}Sb; 2)$

krácení

$(bacabb, Sb, 2) \overset{2.}{\dashrightarrow} (\cancel{b}acabb; \cancel{b}Sbb; 2,2)$

$(acabb; Sbb; 2,2) \overset{1.}{\dashrightarrow} (\cancel{a}cabb, \cancel{a}Sabb; 2,2,1)$

$(cabb; Sabb; 2,2,1) \overset{3.}{\dashrightarrow} (\cancel{cabb}; \cancel{cabb}; 2,2,1,3)$

$(\varepsilon;\ \varepsilon;\ 4\,2\,1\,3)$ → levý rozbor

Jednoduchá $LL(1)$ **gramatika** je CF gramatika $\mathcal{G}$, jestliže pro každé její pravidlo platí:

1. Pravá strana začíná terminálním symbolem.
2. Jestliže dvě pravidla mají stejnou levou stranu, pak se liší terminálním symbolem, kterým začíná pravá strana pravidla.

$S \to aSa$

$S \to bSb$

$S \to c$

$LL(1)$ **analyzátor** je zobrazení

$$\mathcal{M}\colon (N \cup \Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\}) \to \{\textbf{krátit}, \textbf{chyba}, \textbf{přijmout}\} \cup \\ \cup \{(\alpha; i) \mid \alpha \text{ je pravá strana } i\text{-tého pravidla}\}$$

takové, že $\mathcal{M}(a, a) =$ **krátit**, $\mathcal{M}(\varepsilon, \varepsilon) =$ **přijmout** a jestliže $\mathcal{M}(X, a) = (\beta; i)$, pak $X \to \beta$ je i-té pravidlo.

β začíná a

$a \neq b$

$\mathcal{M}(a, b) =$ chyba

Pracuje se s $\mathcal{M}$ jako se zásobníkovým automatem (s jedním stavem)

$(w;\ \underbrace{\alpha}^{(N \cup \Sigma)^*};\ \underbrace{\quad}_{\text{posloupnost pravidel}})$

Příklad. Dána CF gramatika s $N = \{S, A, B\}$, $\Sigma = \{a, b\}$, S a pravidly

1. $S \to \underline{a}SA$
2. $S \to \underline{b}B$
3. $A \to \underline{a}Ab$
4. $A \to \underline{b}$
5. $B \to ab$

$LL(1)$ analyzátor z příkladu můžeme prezentovat v tabulce, kde řádky odpovídají terminálům, neterminálům a prázdnému slovu, sloupce odpovídají terminálům a prázdnému slovu.

Její $LL(1)$ analyzátor je následující tabulka:

	a	b	ε
S	$aSA; 1$	$bB; 2$	**chyba**
A	$aAb; 3$	$b; 4$	**chyba**
B	$aA; 5$	**chyba**	**chyba**
a	**krátit**	**chyba**	**chyba**
b	**chyba**	**krátit**	**chyba**
ε	**chyba**	**chyba**	**přijmout**

Použijeme na slovo *baabb*.

$(baabb; S; -) \vdash (baabb; bB; 2) \vdash$

$(aabb; B; 2) \vdash (\cancel{a}abb; \cancel{a}A; 2,5) \vdash (abb; A; 2,5)$

$(abb; \cancel{a}Ab; 2,5,3) \vdash (bb; Ab; 2,5,3) \vdash$

$\vdash (\cancel{b}b; \cancel{b}b; 2,5,3,4) \vdash (\cancel{b}; \cancel{b}; 2,5,3,4) \vdash$

$(\varepsilon; \varepsilon; 2,5,3,4)$ přijmout $\quad 2,5,3,4$

levý rozbor

Podobný analyzátor je možné vytvořit i pro obecnější CF gramatiky.

Dána CF gramatika $\mathcal{G}$. Pro slovo $\underline{\omega} \in (N \cup \Sigma)^\star$ je $\mathbf{Fi}(\omega)$ množinu všech $a \in \Sigma$ na něž začíná některé $w \in \Sigma^\star$ generované z $\underset{\omega}{\cancel{\alpha}}$. Jestliže $\underset{\omega}{\cancel{\alpha}} \Rightarrow^\star \varepsilon$,tak $\varepsilon \in \mathbf{Fi}(\underset{\omega}{\cancel{\alpha}})$.

<u>$LL(1)$ gramatika je CF gramatika splňující:</u>

Jestliže v $\mathcal{G}$ existují dvě pravidla $\underline{A} \to \alpha$ a $\underline{A} \to \beta$ se stejnou levou stranou, pak pro každé dvě levé derivace $S \Rightarrow^\star uA\mu$ a $S \Rightarrow^\star vA\chi$, $u, v \in \Sigma^\star$, $\mu, \chi \in (N \cup \Sigma)^\star$ platí $\mathbf{Fi}(\alpha\mu) \cap \mathbf{Fi}(\beta\chi) = \emptyset$. Jestliže navíc $\mathbf{Fi}(A)$ obsahuje ε, pak $\mathbf{Fi}(\mu)$ je také disjunktní s $\mathbf{Fi}(\alpha\mu)$ (a tedy i $\mathbf{Fi}(\beta\chi)$).

V tomto případě lze vytvořit $LL(1)$ analyzátor

$LL(k)$ $\quad k > 1$

analyzátor na základě prvních k písmen

Turingovy stroje

Reg. jazyky . . . DFA, NFA

CF jazyky . . . . ZA

kontextové jazyky . . . lineárně omezené TM (TS)

$\mathcal{L}_0$ jazyky . . . . . . Turingovy stroje, které je přijímají

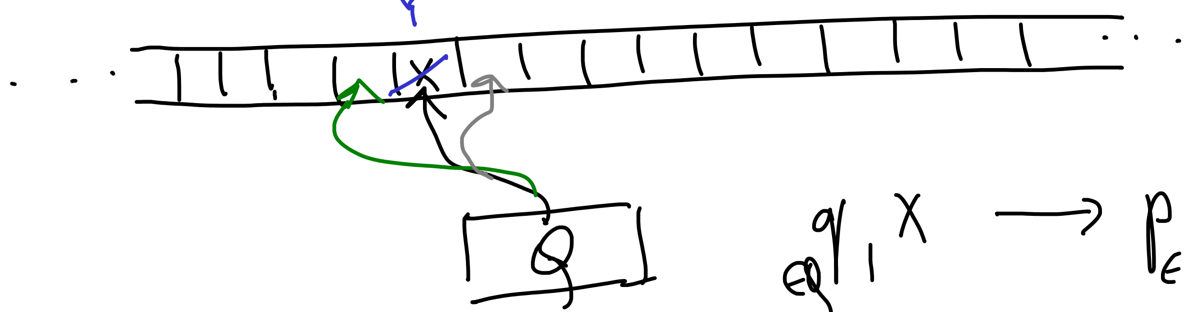
$L \in \mathcal{L}_0$ iff $\exists$ gramatika typu 0

$\alpha \to \beta$ α obsahuje neterminál

Turingův stroj je sedmice $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, kde

- Q je konečná množina stavů,
- Σ je konečná množina vstupních symbolů,
- Γ je konečná množina páskových symbolů, přitom $\Sigma \subset \Gamma$,
- B je prázdný symbol (též nazývaný *blank*), jedná se o páskový symbol, který není vstupním symbolem, (tj. $B \in \Gamma \setminus \Sigma$),
- δ je přechodová funkce, tj. parciální zobrazení z množiny $\cancel{Q \times \Gamma}$ $(Q \setminus F) \times \Gamma$ do množiny $Q \times \Gamma \times \{L, R\}$, (zde L znamená pohyb hlavy o jedno pole doleva, R znamená pohyb hlavy o jedno pole doprava),
- $q_0 \in Q$ je počáteční stav a
- $F \subseteq Q$ je množina koncových stavů. – též akceptující stavy

Obrázek TM

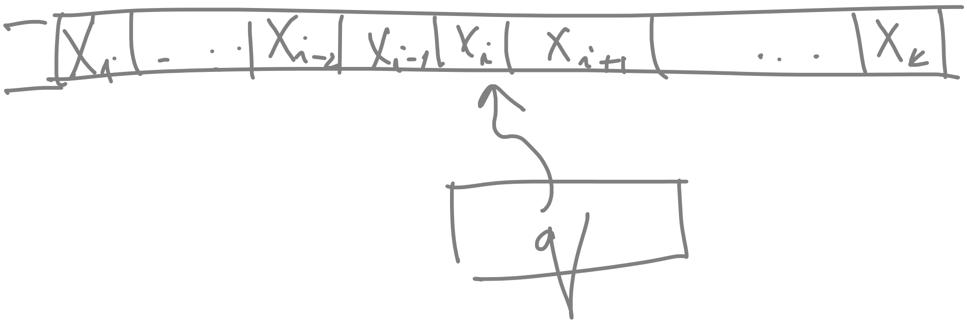


$\underset{\in Q}{q}, X \longrightarrow \underset{\in Q}{p}, Y, \overset{L}{\cancel{R}}$

Konfigurace je

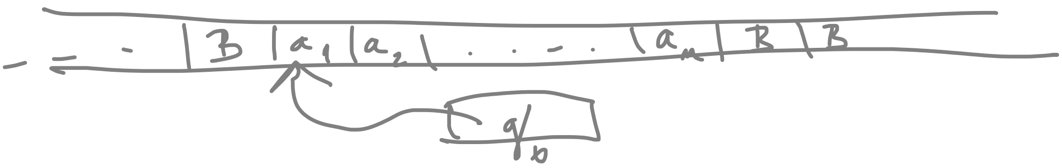
$$X_1\, X_2\, \dots\, X_{i-1}\, q\, X_i\, X_{i+1}\, \dots\, X_k.$$

$(X_1 \dots X_{i-1}, q, X_i X_{i+1} \dots X_k)$



Počáteční konfigurace je

$$q_0\, a_1\, \dots\, a_n.$$



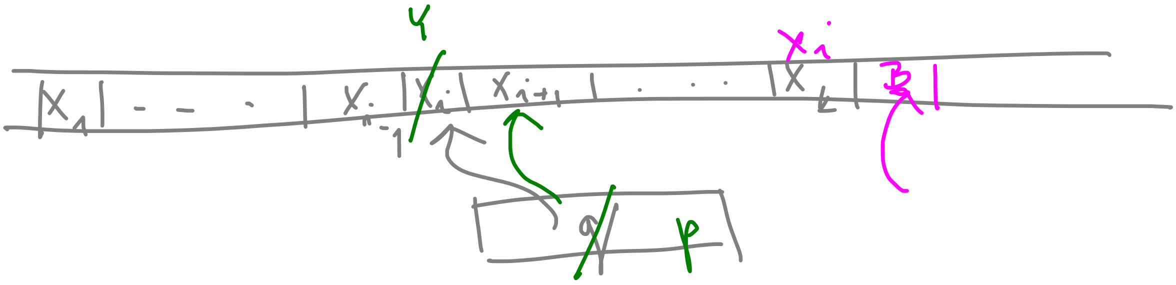
Krok Turingova stroje.

Turingův stroj nachází v konfiguraci $X_1\, X_2\, \dots\, X_{i-1}\, q\, X_i\, \dots\, X_k$, pak:
Jestliže $\delta(q, X_i) = (p, Y, R)$, pak

$$X_1\, X_2\, \dots\, X_{i-1}\, q\, X_i\, \dots\, X_k \vdash X_1\, X_2\, \dots\, X_{i-1}\, Y\, p\, X_{i+1}\, \dots\, X_k.$$

Je-li $i = k$, pak

$$X_1\, X_2\, \dots\, q\, X_k \vdash X_1\, X_2\, \dots\, X_{k-1}\, Y\, p\, B.$$

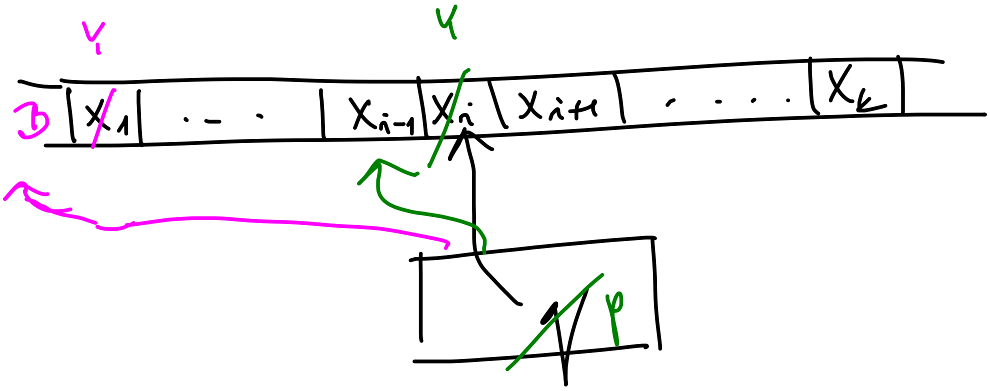


Jestliže $\delta(q, X_i) = (p, Y, L)$, pak

$$X_1\, X_2\, \ldots\, X_{i-1}\, q\, X_i\, \ldots\, X_k \vdash X_1\, X_2\, \ldots\, X_{i-2}\, p\, X_{i-1}\, Y\, X_{i+1}\, \ldots\, X_k.$$

Je-li $i = 1$, pak

$$B\; q\, X_1\, X_2\, \ldots\, X_k \vdash p\, B\, Y\, X_2\, \ldots\, X_k.$$



Výpočet Turingova stroje je posloupnost jeho kroků, která začíná v počáteční konfiguraci.

$$S_0 \vdash S_1 \vdash S_2 \vdash \cdots \vdash S_\ell$$

$q_0 a_1 \ldots a_k$

Slovo $w \in \Sigma^\star$ je **přijato Turingovým strojem** jestliže

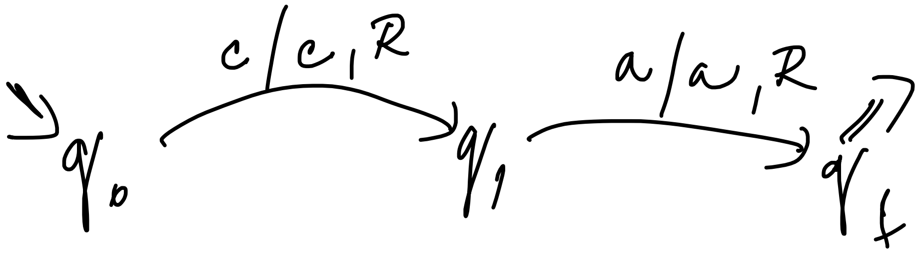
$$q_0\, w \vdash^\star \alpha \boxed{q} \beta \ \text{ pro nějaké } \ q \in F \text{ a } \alpha, \beta \in \Gamma^\star.$$

Jazyk přijímaný Turingovým strojem je množina slov $w \in \Sigma^\star$, které Turingův stroj přijímá a značíme ho $L(M)$.

Př.: kdybychom chtěli TM M, že $L(M)$ přijme všechna slova s druhým symbolem a

$\delta(q_0, c) = (q_1, c, R)$ jinak

$\delta(q_1, a) = (q_f, a, R)$ není definováno



Platí: $L \in \mathcal{L}_0$ iff existuje TM M, že $L(M) = L$.

$$L(M) = \{ w \mid q_0 w \vdash^* \alpha q_f \beta,\ q_f \in F \}$$

Nerozhodnutelnost

Jazyk L je *rekursivně spočetný*, jestliže existuje Turingův stroj M, který tento jazyk přijímá.

L je rekursivně spočetný iff $L \in \mathcal{L}_0$

Jazyk L je *rekursivní*, jestliže existuje Turingův stroj M, který tento jazyk přijímá a na každém slově se zastaví.

M rozhoduje nebo řeší úlohu:
Patří w do L nebo nepatří do L?

Jazykům, které nejsou rekursivní, říkáme, že jsou **nerozhodnutelné**. Rozhodovací úlohy, pro které instance s odpověďmi ANO tvoří nerozhodnutelný jazyk, říkáme **algoritmicky neřešitelné**.

Věta. Problém zastavení Turingova stroje (tzv. halting problem) je algoritmicky ~~nerozhodnutelný~~.

neřešitelný

Je dán TM M a slovo $w \in \Sigma^*$.

Zastaví se TM při práci nad w?

Vstup: TM M, $w \in \Sigma^*$

Je pravda, že $w \in L(M)$?

Postův korespondenční problém (PCP).

Dány dva seznamy slov A, B nad abecedou Σ.

$$A = (w_1, w_2, \ldots, w_k), \quad B = (x_1, x_2, \ldots, x_k),$$

kde $w_i, x_i \in \Sigma^+$, $i = 1, 2, \ldots, k$. Dvojice A, B **má řešení**, jestliže existuje posloupnost $i_1, i_2, \ldots, i_r$ indexů, tj. $i_j \in \{1, 2, \ldots, k\}$, taková, že

$$w_{i_1} w_{i_2} \ldots w_{i_r} = x_{i_1} x_{i_2} \ldots x_{i_r}.$$

Otázka: Existuje řešení dané instance?

Příklady.

1. Dány seznamy

	1	2	3	4	5
A	011	0	101	1010	010
B	1101	00	01	00	0

$A = (011, 0, 101, 1010, 010)$

$B = (1101, 00, 01, 00, 0)$

$A: \quad \overbrace{0}^{2}\ \overbrace{0\,1\,1}^{1}\ \overbrace{0\,1\,0}^{5}$

$B: \quad \underbrace{0\,0}_{2}\ \underbrace{1\,1\,0\,1\,0}_{1}\ \underbrace{0}_{5}$

ano má řešení

2, 1, 5

2. Dány seznamy

	1	2	3	4	5
A	11	0	101	1010	010
B	101	00	01	00	0

2 5

0 0 1 0

0 0 0

2 2 2

0 0 0

0 0 0 0 0 0

nikdy se nesejdou

5 1 1

0 1 0 1 1 1 1

0 1 0 1 1 0 1

nelze

nemá řešení

Další nerozhodnutelné problémy.

- Pro dané bezkontextové gramatiky $\mathcal{G}_1$ a $\mathcal{G}_2$ rozhodnout, zda obě generují aspoň jedno stejné slovo, tj. zda $L(\mathcal{G}_1) \cap L(\mathcal{G}_2) \neq \emptyset$.

Vrážíme tím, že pro A, B z PKP sestrojíme $\mathcal{G}_A, \mathcal{G}_B$ CF gramatiky a dostaň, že $L(\mathcal{G}_A) \cap L(\mathcal{G}_B) \neq \emptyset$

iff A, B má řešení

PKP chceme vyřešit pomocí postupu pro zjištění, že $\mathcal{G}_A, \mathcal{G}_B$ generují spol. slovo

$$A = (w_1, w_2, \dots, w_k) \qquad w_i \in \Sigma^+$$

$$\mathcal{G}_A = (N_A, \Sigma', S_A, P_A) \qquad N_A = \{S_A\}$$

$$\Sigma' = \Sigma \cup \{a_1, \dots, a_k\} \qquad a_i \notin \Sigma$$

$$P_A: \quad S_A \longrightarrow w_1 S_A a_1 \mid w_2 S_A a_2 \mid \dots \mid w_k S_A a_k$$

$$S_A \longrightarrow w_1 a_1 \mid w_2 a_2 \mid \dots \mid w_k a_k$$

$$S_A \overset{1.}{\underset{\mathcal{G}_A}{\Longrightarrow}} w_1 S_A a_1 \overset{3.}{\Longrightarrow} w_1 w_3 S_A a_3 a_1$$

$$\overset{S_A \to w_5 a_5}{\Longrightarrow} w_1 w_3 w_5 a_5 a_3 a_1$$

$L(\mathcal{G}_A)$ se skládá z

$$\boxed{S_A \Rightarrow^* w_{i_1} w_{i_2} \dots w_{i_k} a_{i_k} \dots a_{i_3} a_{i_2} a_{i_1}}$$

$$B = (x_1, x_2, \dots, x_k) \qquad \Sigma_1'$$

$$S_B \to x_1 S_B a_1 \mid \dots \mid x_k S_B a_k$$

$$S_B \to x_1 a_1 \mid \dots \mid x_k a_k$$

$$S_B \Rightarrow^* x_{j_1} \dots x_{j_\ell} a_{j_\ell} \dots a_{j_1} \in L(\mathcal{G}_B)$$

$$L(\mathcal{G}_A) \cap L(\mathcal{G}_B) \neq \emptyset \quad \text{iff}$$

$$\underbrace{w_{i_1} \dots w_{i_k}}_{\in \Sigma_1^+} \overbrace{a_{i_k} \dots a_{i_2} a_{i_1}}^{\in L(\mathcal{G}_A)} = \underbrace{x_{j_1} \dots x_{j_\ell}}_{\in \Sigma^+} \overbrace{a_{j_\ell} \dots a_{j_2} a_{j_1}}^{\in L(\mathcal{G}_B)}$$

$$a_{i_1} = a_{j_1} \qquad i_1 = j_1 \qquad i_2 = j_2$$

$$a_{i_2} = a_{j_2} \qquad \text{a dokonce } k = \ell \quad i_3 = j_3$$

a i po zkrácení jsou to stejná slova

opravdu $w_{i_1} \dots w_{i_k} = x_{i_1} \dots x_{i_k}$

Takže A, B má řešení

- Pro dané bezkontextové gramatiky $\mathcal{G}_1$ a $\mathcal{G}_2$ rozhodnout, zda přijímají stejný jazyk, tj. zda $L(\mathcal{G}_1) = L(\mathcal{G}_2)$.

- Pro danou bezkontextovou gramatiku $\mathcal{G}_1$ rozhodnout, zda je víceznačná.

- Pro danou bezkontextovou gramatiku $\mathcal{G}_1$ rozhodnout, zda přijímá všechna slova, tj. zda $L(\mathcal{G}_1) = \Sigma^\star$.

CF $\mathcal{G}$ Otázka: je $L(\mathcal{G}) = \emptyset$?

Ano je řešitelné – algoritmus redukce, 1. krok.

- Pro danou bezkontextovou gramatiku $\mathcal{G}_1$ a regulární jazyk R rozhodnout, zda $R \subseteq L(\mathcal{G}_1)$.