# Automaty a gramatiky

4

Roman Barták, KTIML

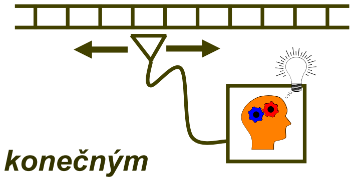
bartak@ktiml.mff.cuni.cz
http://ktiml.mff.cuni.cz/~bartak

## Pro připomenutí

Automat může také ovládat čtecí hlavu – dvousměrný (dvoucestný) automat

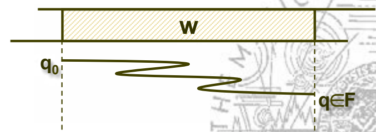
*přechodová funkce:*

$Q \times X \rightarrow Q \times \{-1,0,+1\}$



Slovo w je *přijato dvousměrným konečným automatem*, pokud:

- výpočet začal na prvním písmenu slova w vlevo v počátečním stavu
- čtecí hlava poprvé opustila slovo w vpravo v některém přijímacím stavu
- mimo čtené slovo není výpočet definován (výpočet zde končí a slovo není přijato)






## Věta o dvousměrných automatech

**Jazyky přijímané dvousměrnými konečnými automaty jsou právě jazyky přijímané konečnými automaty.**

Možnost pohybovat čtecí hlavou po pásce nezvětšila sílu konečného automatu!

Pozor, na pásku nic nepíšeme!

Pokud můžeme na pásku psát, dostaneme Turingův stroj.

*Zřejmé:* konečný automat → dvousměrný konečný automat

dvousměrný automat vždy posouvá hlavu doprava

KA $A=(Q,X,\delta,q_0,F) \rightarrow$ 2KA $B=(Q,X,\delta',q_0,F)$, $\delta'(q,x)=(\delta(q,x),+1)$

*Zbývá:* dvousměrný konečný automat → konečný automat



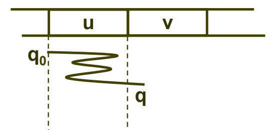
## Důkaz věty o dvousměrných automatech (1)


1) Formální popis vlivu slova *u* na výpočet nad slovem *v*

(i) kdy poprvé opustíme slovo *u* vpravo (v jakém stavu poprvé vstoupíme nad *v*)

$f(q'_0) = q$ poprvé přejdeme na *v* ve stavu q

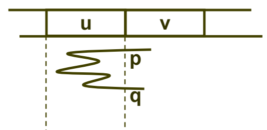
$f(q'_0) = 0$ nikdy neopustíme *u* vpravo




(ii) pokud opustíme slovo *v* vlevo, kdy se nad *v* opět vrátíme

$f(p) = q$ vrátíme se nad *v* ve stavu q

$f(p) = 0$ nikdy už se nevrátíme




2) Výpočet nad *u* máme popsaný funkcí $f_u$

$f_u: Q \cup \{q'_0\} \rightarrow Q \cup \{0\}$

$f_u(q'_0)$ popisuje situaci (i): v jakém stavu poprvé odejdeme vpravo, pokud začneme výpočet vlevo v počátečním stavu $q_0$

$f_u(p)$ ($p \in Q$) popisuje situaci (ii): v jakém stavu opět odejdeme vpravo, pokud začneme výpočet vpravo v p

symbol 0 značí, že daná situace nenastane (odejdeme vlevo nebo cyklus)

## Důkaz věty o dvousměrných automatech (2)

Pro každé slovo *u* máme funkci $f_u$ popisující výpočet dvousměrného automatu A nad *u*

Definujme ekvivalenci slov takto: $u \sim w \Leftrightarrow_{def} f_u = f_w$

tj. slova jsou ekvivalentní, pokud mají stejné „výpočtové" funkce

Vlastnosti ~:

- je to ekvivalence (zřejmé, definováno pomocí =)
- má konečný index (maximální počet různých funkcí je $(n+1)^{n+1}$ pro n-stavový dvousměrný automat)
- je to pravá kongruence (zřejmě $u \sim w \Rightarrow uv \sim wv$, protože rozhraní u|v a w|v je stejné a nad v se automat chová stejně)
- L(A) je sjednocením jistých tříd rozkladu $X^*/\sim$

  stačí si uvědomit, že $w \in L(A) \Leftrightarrow f_w(q'_0) \in F$

  $u \sim w \Rightarrow f_u(q'_0) = f_w(q'_0) \Rightarrow (u \in L(A) \Leftrightarrow w \in L(A))$

Podle Nerodovy věty je L(A) regulární jazyk.

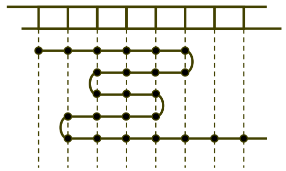


## Převod 2KA na NKA

Konstruktivní důkaz věty o dvousměrných automatech.

Jak výpočet s návraty převést na lineární výpočet?

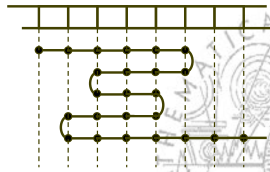
- zajímají nás jen přijímací výpočty
- díváme se na přechody mezi symboly (v jakém stavu se přechází na další políčko)



*Pozorování:*

- stavy se v přechodu (řezu) střídají (doprava/doleva)
- první stav jde doprava, poslední také doprava
- v deterministických přijímajících výpočtech nejsou cykly
- první a poslední řez obsahují jediný stav

1. Najdeme všechny možné řezy - posloupnosti stavů (je jich konečně mnoho).
2. Mezi řezy definujeme (nedeterministické) přechody podle čteného symbolu.
3. Rekonstruujeme výpočet skládáním řezů (jako puzzle).





## Formální převod 2KA na NKA

Nechť $A=(Q,X,\delta,q_0,F)$ je dvousměrný konečný automat.

Definujme ekvivalentní nedeterministický konečný automat $B=(Q',X,\delta',(q_0),F')$, kde:

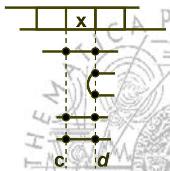
$Q'$ = všechny korektní přechodové posloupnosti

posloupnosti stavů $(q^1,\ldots,q^k)$ z Q takové, že

- délka posloupnosti je lichá ($k=2m+1$)
- žádný stav se neopakuje na liché ani na sudé pozici $(\forall i \neq j\ q^{2i} \neq q^{2j}) \wedge (\forall i \neq j\ q^{2i+1} \neq q^{2j+1})$

$F' = \{(q) \mid q \in F\}$ přechodové posloupnosti délky 1 obsahující koncový stav

$\delta'(c,x) = \{ d \mid d \in Q' \wedge c \rightarrow d$ je lokálně konzistentní přechod pro x$\}$




L(A)=L(B)?

„trajektorie 2KA A odpovídá řezům KA B"

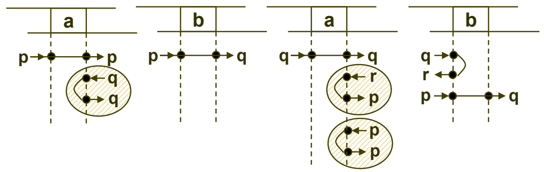


## Příklad převodu 2KA na NKA

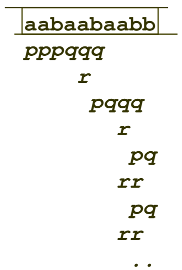
Mějme následující dvousměrný konečný automat:

| | a | b |
|---|---|---|
| → p | p,+1 | q,+1 |
| ← q | q,+1 | r,-1 |
| r | p,+1 | r,-1 |

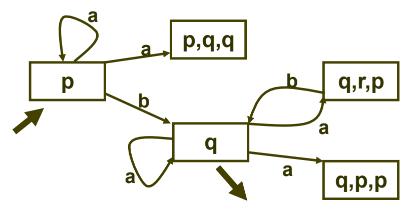
Možné řezy a jejich konzistentní přechody:




Ukázka výpočtu:




Výsledný nedeterministický KA:



## Množinové operace nad jazyky

*Sjednocení jazyků* $L_1 \cup L_2 = \{ w \mid w \in L_1 \vee w \in L_2 \}$

*Příklad*: jazyk obsahuje slova začínající baba nebo končící baa

*Průnik jazyků* $L_1 \cap L_2 = \{ w \mid w \in L_1 \wedge w \in L_2 \}$

*Příklad*: jazyk obsahuje slova se sudým počtem nul a každý symbol 1 je bezprostředně následován 0

*Rozdíl jazyků* $L_1 - L_2 = \{ w \mid w \in L_1 \wedge w \notin L_2 \}$

*Příklad*: jazyk obsahuje slova začínající baba a neobsahující abb

*Doplněk jazyka* $-L = \{ w \mid w \notin L \} = X^* - L$

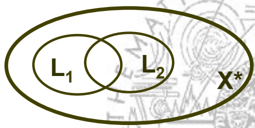
*Příklad*: slova jazyka neobsahují posloupnost tří symbolů 1

Platí tradiční de Morganova pravidla

$L_1 \cap L_2 = -(-L_1 \cup -L_2)$

$L_1 \cup L_2 = -(-L_1 \cap -L_2)$

$L_1 - L_2 = L_1 \cap -L_2$






## Uzavřenost na množinové operace

Nechť $L_1$ a $L_2$ jsou jazyky rozpoznávané konečnými automaty. Potom $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 - L_2$ a $-L_1$ jsou také jazyky rozpoznávané konečnými automaty (třída $\mathcal{F}$ je uzavřena na uvedené operace).

*Konstruktivní důkaz:*

doplněk

stačí prohodit koncové a nekoncové stavy přijímajícího det. automatu

sjednocení, průnik a rozdíl

idea: paralelní běh přijímajících automatů

$A_1 = (Q_1,X,\delta_1,q_1,F_1)$, $A_2 = (Q_2,X,\delta_2,q_2,F_2)$

uděláme spojený automat $A = (Q,X,\delta,q,F)$

$Q = Q_1 \times Q_2$, $q = (q_1,q_2)$

$\delta((p_1,p_2),x) = (\delta_1(p_1,x), \delta_2(p_2,x))$

sjednocení $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$

průnik $F = F_1 \times F_2$

rozdíl $F = F_1 \times (Q_2 - F_2)$



## Množinové operace v příkladě

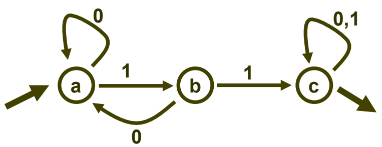
Navrhněte konečný automat přijímající slova, která obsahují 3k+2 symbolů 1 a neobsahují posloupnost 11.

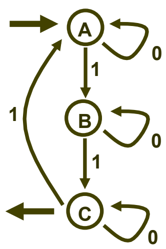
Přímá konstrukce komplikovaná!

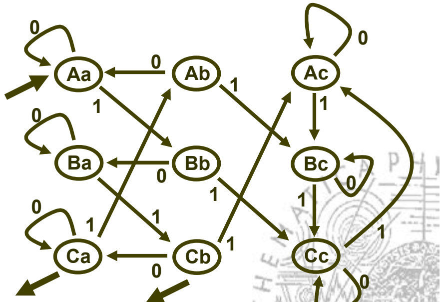
$L_1 = \{w \mid w \in \{0,1\}^* \wedge |w|_1 = 3k+2\}$

$L_2 = \{w \mid u,v \in \{0,1\}^* \wedge w = u11v\}$

$L = L_1 - L_2$



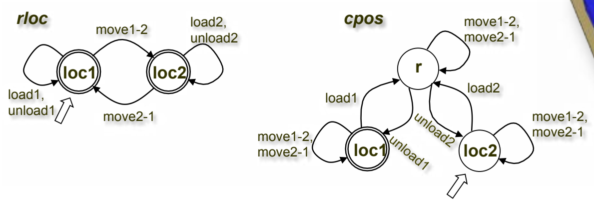



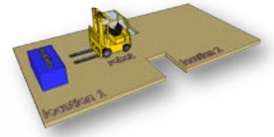



## K čemu to je?

Můžeme operace s automaty někde přímo využít?

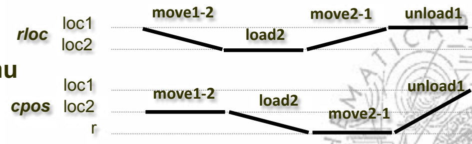
Například v plánování, kde automat popisuje, jak se mění hodnota nějaké stavové proměnné.



Plán se potom může hledat jako průnik automatů.

V každém stavovém diagramu se provede stejná posloupnost akcí.



## Řetězcové operace nad jazyky

| | |
|---|---|
| *Zřetězení jazyků* | $L_1 . L_2 = \{ uv \mid u \in L_1 \wedge v \in L_2 \}$ |
| *Mocniny jazyka* | $L^0 = \{\lambda\}$ |
| | $L^{i+1} = L^i . L$ |
| *Pozitivní iterace* | $L^+ = L^1 \cup L^2 \ldots = \cup_{i\geq 1} L^i$ |
| *Obecná iterace* | $L^* = L^0 \cup L^1 \ldots = \cup_{i\geq 0} L^i$ |
| zřejmě $L^* = L^+ \cup \{\lambda\}$ | |
| *Otočení jazyka* | $L^R = \{ u^R \mid u \in L \}$ |
| reverse, zrcadlový obraz | $(x_1x_2\ldots x_n)^R = x_n\ldots x_2x_1$ |
| *Levý kvocient* $L_1$ podle $L_2$ | $L_2 \setminus L_1 = \{ v \mid uv \in L_1 \wedge u \in L_2 \}$ |
| *Levá derivace* L podle w | $\partial_w L = \{w\} \setminus L$ |
| *Pravý kvocient* $L_1$ podle $L_2$ | $L_1 / L_2 = \{ u \mid uv \in L_1 \wedge v \in L_2 \}$ |
| *Pravá derivace* L podle w | $\partial^R_w L = L / \{w\}$ |



## Uzavřenost zřetězení

$L_1,L_2 \in \mathcal{F} \Rightarrow L_1.L_2 \in \mathcal{F}$

| $L_1$ | $L_2$ |
|---|---|

idea:

nejprve počítá automat $A_1=(Q_1,X,\delta_1,q_1,F_1)$ potom $A_2 =(Q_2,X,\delta_2,q_2,F_2)$

realizace:

pomocí nedeterministického konečného automatu $B =(Q,X,\delta,S,F)$

nedeterminismus slouží při rozhodování kdy přepnout do $A_2$

$Q = Q_1 \cup Q_2$ (předpokládáme různá jména stavů, jinak přejmenuj)

$S = \{q_1\}$ pokud $\lambda \notin L_1$ $(q_1 \notin F_1)$

$= \{q_1, q_2\}$ pokud $\lambda \in L_1$ $(q_1 \in F_1)$, tj. rovnou přejdeme také do $A_2$

$F = F_2$ končíme až po přečtení slova z $L_2$

| | | | |
|---|---|---|---|
| $\delta(q,x)$ | $= \{\delta_1(q,x)\}$ | pokud $q \in Q_1 \wedge \delta_1(q,x) \notin F_1$ | (počítáme v $A_1$) |
| | $= \{\delta_1(q,x), q_2\}$ | pokud $q \in Q_1 \wedge \delta_1(q,x) \in F_1$ | (přechod $A_1$ do $A_2$) |
| | $= \{\delta_2(q,x)\}$ | pokud $q \in Q_2$ | (počítáme v $A_2$) |

DCV: ověřit $L(B) = L(A_1) . L(A_2)$



## Uzavřenost iterace

$L \in \mathcal{F} \Rightarrow L^* \in \mathcal{F}$

| $u_1 \in L$ | $u_2 \in L$ | $u_3 \in L$ |
|---|---|---|

idea: opakovaný výpočet automatu $A=(Q,X,\delta,q_0,F)$

realizace: nedeterministické rozhodnutí, zda pokračovat nebo restart

pozor! $\lambda \in L^*$ i když $\lambda \notin L$, řešíme pomocí speciálního stavu

hledáme nedeterministický automat $B =(Q',X,\delta',S,F')$

$Q' = Q \cup \{s\}$ přidáme nový stav pro příjem $\lambda$

$S = \{q_0, s\}$ nový stav

$F' = F \cup \{s\}$ končíme po přečtení slova z L nebo v s (pro $\lambda$)

| | | | |
|---|---|---|---|
| $\delta'(q,x)$ | $= \{\delta(q,x)\}$ | pokud $q \in Q \wedge \delta(q,x) \notin F$ | (počítáme uvnitř A) |
| | $= \{\delta(q,x), q_0\}$ | pokud $q \in Q \wedge \delta(q,x) \in F$ | (možný restart) |
| $\delta'(s,x)$ | $= \{\}$ | žádné přechody z nového stavu | |

$L \in \mathcal{F} \Rightarrow L^+ \in \mathcal{F}$

stejná konstrukce, pouze bez použití stavu s



## Uzavřenost reverse

$L \in \mathcal{F} \Leftrightarrow L^R \in \mathcal{F}$

$\xleftarrow{\quad}$

| L |
|---|

zřejmě $(L^R)^R = L$, a tedy stačí ukázat $L \in \mathcal{F} \Rightarrow L^R \in \mathcal{F}$

idea: obrátíme „šipky“ ve stavovém diagramu

realizace: nedeterministický konečný automat

$A=(Q,X,\delta,q_0,F) \quad \rightarrow \quad B=(Q,X,\delta',F,\{q_0\})$

$\delta'(q,x) = \{p \mid \delta(p,x)=q \}$ $\qquad (\delta(p,x)=q \Leftrightarrow p \in \delta'(q,x) )$

$w = x_1x_2\ldots x_n$

$q_0, q_1, \ldots q_n$ je přijímající výpočet pro w automatu A

tj. $\delta(q_i,x_{i+1})=q_{i+1}$ a $q_n \in F$

$\Leftrightarrow$

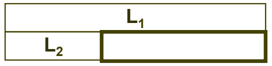
$q_n, q_{n-1}, \ldots q_0$ je přijímající výpočet pro $w^R$ automatu B

$q_i \in \delta'(q_{i+1},x_{i+1})$

*Poznámka*:

někdy L nebo $L^R$ má výrazně jednodušší přijímající automat

## Uzavřenost kvocientu




$L_1, L_2 \in \mathcal{F} \Rightarrow L_2 \setminus L_1 \in \mathcal{F}$

idea: automat $A_1$ budeme startovat ve stavech, do kterých se lze dostat slovem z $L_2$

realizace: nedeterministický automat B „téměř“ totožný s $A_1$ (rozdíl ve startovních stavech)

$S = \{q \mid q \in Q_1\ \exists u \in L_2\ q = \delta_1(q_1, u)\}$ nové startovní stavy

lze nalézt algoritmicky ($A_q = (Q_1, X, \delta_1, q_1, \{q\})$, pak $q \in S \Leftrightarrow L(A_q) \cap L_2 \neq \emptyset$)

$v \in L_2 \setminus L_1$

$\Leftrightarrow \exists u \in L_2\ uv \in L_1$

$\Leftrightarrow \exists u \in L_2\ \exists q \in Q_1\ \ \delta_1(q_1, u) = q \wedge \delta_1(q, v) \in F_1$

$\Leftrightarrow \exists q \in S\ \ \delta_1(q, v) \in F_1$

$\Leftrightarrow v \in L(B)$

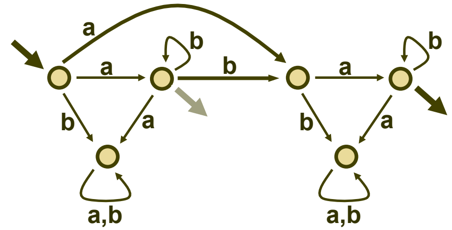
$L_1, L_2 \in \mathcal{F} \Rightarrow L_1 / L_2 \in \mathcal{F}$

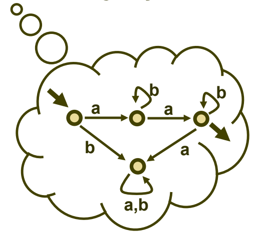
obdobně nebo pomocí $L_1 / L_2 = (L_2^R \setminus L_1^R)^R$



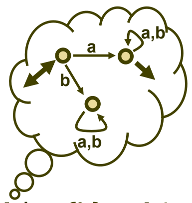
## Příklady řetězcových operací

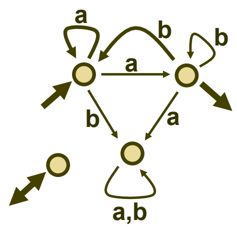
$L = \{ab^i,\ i \geq 0\}$ $L.L = \{ab^iab^j,\ i \geq 0,\ j \geq 0\}$

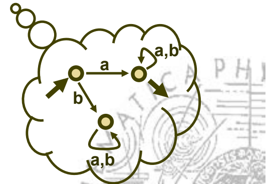



$L^+ = \{ab^{i_1}\ ab^{i_2} \ldots ab^{i_n},\ n > 0,\ i_j \geq 0\}$



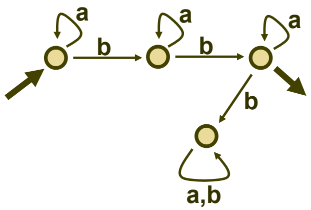




$L^* = \{\lambda\} \cup L^+$

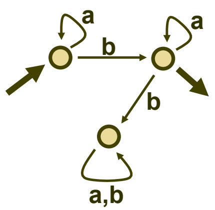


## Příklad kvocientu

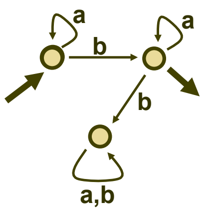
$L_1 = \{a^{i_1}ba^{i_2}ba^{i_3},\ i_j \geq 0\}$




$L_2 = \{a^{i_1}ba^{i_2},\ i_j \geq 0\}$




$L_2 \setminus L_1 = \{a^{i_1}ba^{i_2},\ i_j \geq 0\} = L_2$






## Substituce jazyků

nechť X je konečná abeceda

pro každé $x \in X$ budiž $\sigma(x)$ jazyk v nějaké abecedě $Y_x$

dále položme:

$\sigma(\lambda) = \{\lambda\}$

$\sigma(u.v) = \sigma(u)\ .\ \sigma(v)$

zobrazení $\sigma$: $X^* \to P(Y^*)$, kde $Y = \cup_{x \in X} Y_x$ se nazývá *substituce*

$\sigma(L) = \cup_{w \in L}\ \sigma(w)$

*nevypouštějící substituce*, žádné $\sigma(x)$ neobsahuje $\lambda$

Příklad:

$\sigma(0) = \{a^ib^j,\ i,j \geq 0\}$, $\sigma(1) = \{cd\}$

$\sigma(010) = \{a^ib^jcda^kb^l,\ i,j,k,l \geq 0\}$

*homomorfismus h*: $h(x) = w_x$ (speciální případ substituce)

*nevypouštějící homomorfismus*: $w_x \neq \lambda$

***Věta:*** $L \in \mathcal{F},\ \forall x \in X\ \sigma(x) \in \mathcal{F} \Rightarrow \sigma(L),\ h(L),\ h^{-1}(L) \in \mathcal{F}$

$h^{-1}(L) = \{w \mid h(w) \in L\}$

## Poznámky k uzávěrovým vlastnostem

**Zjednodušení návrhu automatů**

$L.\emptyset = \emptyset.L = \emptyset$

$\{\lambda\}.L = L.\{\lambda\} = L$

$(L^*)^* = L^*$

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

$(L_1.L_2)^R = L_2^R . L_1^R$

$\partial_w(L_1 \cup L_2) = \partial_w L_1 \cup \partial_w L_2$

$\partial_w(X^* - L) = X^* - \partial_w L$

$h(L_1 \cup L_2) = h(L_1) \cup h(L_2)$

**Důkaz regulárnosti**

$L = \{w \mid w \in \{0,1\}^*, |w|_1 = |w|_0\}$ není regulární

$L \cap \{0^i 1^j, i,j \geq 0\} = \{0^i 1^i, i \geq 0\}$