AUTOMATY A GRAMATIKY

5

Pavel Surynek

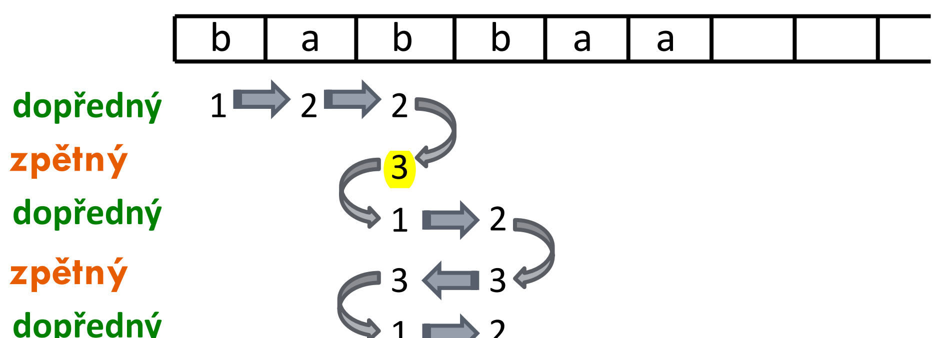
Univerzita Karlova v Praze

Matematicko-fyzikální fakulta
Katedra teoretické informatiky a matematické logiky

Výpočet dvousměrného automatu

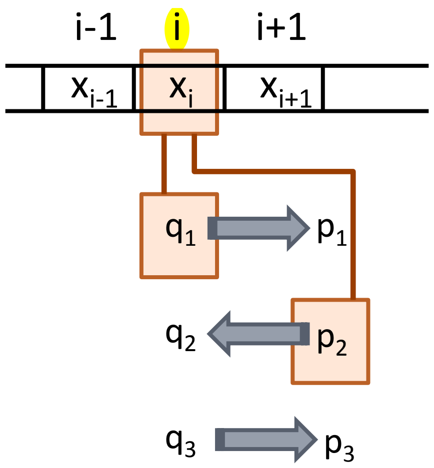
- 2KA $A = (Q, X, \delta, q_0, F)$
  - lze se omezit na $\delta': Q \times X \rightarrow Q \times \{-1, 1\}$
    - výpočet na místě nahradíme jeho výsledkem v okamžiku přesunu na další buňku
      - $\delta'(q,x) = \delta(p_i,x)$, kde $i \in \mathbb{N}_0$ je takové, že $\delta(p_i,x)_2 \neq 0$, $p_1=q$ a $[p_{j+1}, 0] = \delta(p_j,x)$ pro $j = 1,2, \ldots, i-2$
  - při výpočtu se pak střídá **dopředný** a **zpětný** chod
    - speciálně přijímající výpočet začíná dopředným chodem a končí dopředným
      - počet chodů v přijímajícím výpočtu je lichý

	a	b
→1	1, +1	2, +1
←2	2, +1	3, -1
3	1, +1	3, -1



Ekvivalence KA a 2KA (1)

- KA $A = (Q_A, X, \delta_A, q_0, F_A) \Rightarrow$ 2KA B
  - ihned: $B = (Q_A, X, \delta_B, q_0, F_A)$, kde $\delta_B(q,x)=[\delta_A(q,x), +1]$ pro $q \in Q$ a $x \in X$
- 2KA $\Rightarrow$ KA
  - omezme se na deterministický 2KA
    - bez újmy na obecnosti $\delta: Q \times X \rightarrow Q \times \{-1, 1\}$
  - i-tá přechodová (*crossing*) posloupnost
    - posloupnost stavů v <u>přijímajícím</u> výpočtu při práci nad symboly x_{i-1} a x_i (tj. nad (i-1)-ní a i-tou buňkou) ve slově $w=x_1x_2\ldots x_n$
      - směry výpočtu se střídají
        - dopředný chod pracuje se symbolem x_i
        - zpětný chod pracuje se symbolem x_{i-1}
      - posloupnosti jsou liché délky
    - stavy se v přechodové posloupnosti na lichých resp. sudých pozicích neopakují
      - jinak zacyklení, tj. nepřijímající výpočet
  - umíme ověřit, zda dvě dané přechodové posloupnosti po sobě mohou v přijímajícím výpočtu následovat



Př.: [q1, q2,q3] je i-tá přechodová posloupnost

Ekvivalence KA a 2KA (2)

- (deterministický) 2KA $A = (Q_A, X, \delta_A, q_{A0}, F_A)$
  - definujeme ekvivalentní KA $B = (Q_B, X, \delta_B, q_{B0}, F_B)$:
    - Q_B nechť je množina všech přechodových posloupností vzhledem k A
      - Q_B konečná ← pod-posloupnosti na sudých resp. lichých pozicích jsou bez opakování stavů
    - $q_{B0} = [q_{A0}]$ $\qquad q_{A0} \Rightarrow$
    - $\delta_B([q_1,q_2,\ldots,q_k], x)=\{[p_1,p_2,\ldots,p_l] \mid$
      $[p_1,p_2,\ldots,p_l]$ může následovat po $[q_1,q_2,\ldots,q_k]$ při čtení $x\}$
      - definujeme pro každou přechodovou posloupnost $[q_1,q_2,\ldots,q_k]\in Q_B$ a $x\in X$
    - $F_B = \{[f] \mid f\in F_A\}$ $\qquad f \Rightarrow$

Ekvivalence KA a 2KA (3)

- nedeterministický 2KA A
  - stavy se v přechodových posloupnostech mohou opakovat
    - omezíme se na prosté přechodové posloupnosti
      - jestliže $[q_0,i_0]$, $[q_1,i_1]$, …, $[q_\alpha,i_\alpha]$, …, $[q_\beta,i_\beta]$,…., $[q_k,i_k]$, kde $[q_\alpha,i_\alpha]=[q_\beta,i_\beta]$, je přijímající výpočet, pak $[q_0,i_0]$, $[q_1,i_1]$, …, $[q_\alpha,i_\alpha]$, $[q_{\beta+1},i_{\beta+1}]$,…., $[q_k,i_k]$ je rovněž přijímající výpočet
      - jestliže $w \in L(A)$, pak existuje prostý výpočet, který přijímá w
      - v přechodových posloupnostech vzhledem k prostým přijímajícím se neopakují stavy na lichých ani sudých pozicích
  - přechodová funkce je nedeterministická
    - návaznost přechodových posloupností se bude ověřovat nedeterministicky
  - množina počátečních stavů
    - $S_{B0} = \{[q] \mid q \in S_0\}$

Regulární **substituce**

- (obecná) **substituce** abecedy
  - nechť X a Y jsou abecedy
    - zobrazení $f: X \rightarrow 2^{Y^*}$ je substituce z X do Y
      - symbolům přiřazujeme množiny slov
  - nechť L je jazyk nad X
    - definujeme $f(L)=\{w_1w_2\ldots w_n \mid (\exists x_1x_2\ldots x_n \in L) w_1 \in f(x_1) \wedge w_2 \in f(x_2) \wedge \ldots \wedge w_n \in f(x_n)\}$
  - nechť K je jazyk nad Y
    - definujeme $f^{-1}(K)=\{x_1x_2\ldots x_n \mid (\exists w_1w_2\ldots w_n \in K)$ $w_1 \in f(x_1) \wedge w_2 \in f(x_2) \wedge \ldots \wedge w_n \in f(x_n)\}$
- **regulární** substituce
  - substituce z X do Y je regulární, jestliže $f(x)$ je regulární jazyk pro každé $x \in X$
  - speciálně, když $|f(x)|=1$, tj. symbolům jsou přiřazena slova, se f nazývá **homomorfismus**

Př.: X={0, 1}; Y={a,b}
L={000, 101, 11}
f(0) = {aba, abba, aa}
f(1) = {λ, bb, bab}
f(L) = {aba.abba.aa, aa.aa.aa, aba.aba.aa, ..., aa.bb, bb.abba.bab, ...}

Uzavřenost regulární **substituce**

- regulární jazyky jsou uzavřené na regulární substituci
  - nechť L je regulární jazyk nad X, K je regulární jazyk nad Y a f regulární substituce, pak $f(L)$ i $f^{-1}(K)$ je regulární jazyk
    - speciálně je-li h homomorfismus, pak $h(L)$ i $h^{-1}(K)$ jsou regulární jazyky
  - **Důkaz**
    - regularita $f(L)$ resp. $h(L)$ snadno pomocí regulárních výrazů
    - regularita $f^{-1}(K)$ resp. $h^{-1}(K)$
      - KA $A = (Q, Y, \delta, q_0, F)$, že $L(A) = K$
      - pro každé $x \in X$ máme KA $A_x = (Q_x, Y, \delta_x, q_{x0}, F_x)$, že $L(A_x) = f(x)$, kde množiny stavů jsou disjunktní, tj. $Q_x \cap Q_{x'} = \emptyset$ pro $x, x' \in X$, že $x \neq x'$
      - zkonstruujme dvousměrný konečný automat $B = (Q_B, X, \delta_B, S_{B0}, F_B)$, kde
        - $Q_B = \{f_B\} \cup Q \times (\bigcup_{x \in X} Q_x)$, kde f_B je nový stav
        - $S_{B0} = \{(q_0, q_{x0}) \mid x \in X\}$
        - $F_B = \{f_B\}$
        - $$\delta_B([q,r],x) = \begin{cases} \{[(\delta(q,y), \delta_x(r,y)),0] \mid y \in Y\}, \text{ když } r \in Q_x - F_x \\ \{[(\delta(q,y), \delta_x(r,y)),0] \mid y \in Y\} \cup \{[(q,q_{x0}),+1] \mid x \in X\}, \text{ když } r \in F_x \text{ a } q \notin F \\ \{[(\delta(q,y), \delta_x(r,y)),0] \mid y \in Y\} \cup \{[(q,q_{x0}),+1] \mid x \in X\} \cup \{[f_B,+1]\}, \\ \quad \text{když } r \in F_x \text{ a } q \in F \\ \emptyset, \text{ když } r \notin Q_x \end{cases}$$
        - $\delta_B(f_B,x) = \emptyset$ pro $x \in X$

Regulární jazyky algebraicky

- nad danou abecedou X zavedeme třídu jazyků $\mathcal{R}$ a ukážeme, že $\mathcal{R}$ jsou právě regulární jazyky nad X
  - $\mathcal{R}$ bude definována algebraicky pomocí uzávěrových vlastností jako <u>nejmenší</u> třída jazyků splňující následující podmínky:
    - (i) $\emptyset \in \mathcal{R}$ (*prázdný* jazyk)
    - (ii) $\{x\} \in \mathcal{R}$ pro $x \in X$ (*jednopísmenné* jazyky)
      - společně s $\emptyset$ a $\{\lambda\}$ tvoří tzv. elementární jazyky
    - (iii) když $K,L \in \mathcal{R}$, pak $K \cup L \in \mathcal{R}$
      - uzavřenost na *sjednocení*
    - (iv) když $K,L \in \mathcal{R}$, pak $K.L \in \mathcal{R}$
      - uzavřenost na *konkatenaci*
    - (v) když $L \in \mathcal{R}$, pak $L^* \in \mathcal{R}$
      - uzavřenost na *iteraci*
  - speciálně platí:
    - $\{\lambda\} \in \mathcal{R}$ $\{\lambda\} = \emptyset^*$
    - $X^* \in \mathcal{R}$ protože $X \in \mathcal{R}$ a libovolné slovo $x_1x_2...x_n \in X.X.....X$, kde $x_i \in X$ pro $i=1,2,...,n$
      - pokud X chápeme jako jazyk jednopísmenných slov, je $X = \bigcup_{x \in X} \{x\}$

Pozn.: bez požadavku na nejmenší třídu může být $\mathcal{R} = 2^{X^*}$ (všechny jazyky nad X)

Kleeneho věta (1)

- předpokládejme fixní abecedu X
  - třídu $\mathcal{R}$ tvoří právě jazyky přijímané konečnými automaty tj. regulární jazyky
    - tedy $L \in \mathcal{R}$, právě když existuje KA A, že $L(A)=L$
  - **Důkaz**
    - $\Rightarrow$ (již víme)
      - elementární jazyky jsou regulární
      - při budování $\mathcal{R}$ díky uzavřenosti regularity na sjednocení, konkatenaci a iteraci přidáme vždy regulární jazyk
    - $\Leftarrow$ (idea z Floyd-Warshallova algoritmu)
      - mějme KA $A=(Q, X, \delta, q_1, F)$, že $L(A)=L$, kde
        - $Q=\{q_1,q_2,\ldots,q_n\}$
      - definujeme $L_{i,j}=\{w \mid w \in X^* \wedge \delta^*(q_i,w)=q_j\}$
        - platí, že $L(A)=\bigcup_{q_i \in F} L_{1,i}$

Kleeneho věta (2)

- Pokračování **důkazu**
  - $\Leftarrow$
    - definujeme $L_{i,j}^k=\{w \mid w \in X^* \wedge \delta^*(q_i,w)=q_j$ tak, že výpočet je tvaru $q_i p^1 p^2 \ldots p^m q_j$, kde $p^l \in \{q_1,q_2,\ldots q_k\}$ pro $l=1,2,\ldots,m\}$
      - vnitřní stavy výpočtu se **omezují** na prvních k stavů
    - zřejmě platí $L_{i,j}=L_{i,j}^n$, protože omezení je prázdné
    - $L_{i,j}^{k+1}=L_{i,j}^k \cup L_{i,k+1}^k.(L_{k+1,k+1}^k)^*.L_{k+1,j}^k$
      - výpočet buď stav q_{k+1} nevyužije, nebo ano, a sice libovolněkrát (proto *)
    - indukcí podle k
      - $L_{i,j}^0$ jsou elementární jazyky, tedy $L_{i,j}^0 \in \mathcal{R}$
      - předpokládejme, že $L_{i,j}^k$, $L_{i,k+1}^k$, $L_{k+1,k+1}^k$, $L_{k+1,j}^k \in \mathcal{R}$, pak z uzavřenosti $\mathcal{R}$ na sjednocení, konkatenaci a iteraci je $L_{i,j}^{k+1} \in \mathcal{R}$
      - celkem dostáváme $L_{i,j} \in \mathcal{R}$
    - z uzavřenosti $\mathcal{R}$ na sjednocení a z $L(A)=\bigcup_{q_i \in F} L_{1,i}$ máme $L(A) \in \mathcal{R}$

Regulární výrazy

- Kleeneho věta poskytuje alternativní popis regulárních jazyků nad danou abecedou X
  - umožňuje zavedení **regulárních výrazů**, což jsou slova nad abecedou X∪{∅, λ ,+, ., *, (,)} vytvořené podle následujících pravidel:
    - předpokládáme, že {∅, λ ,+, ., *, (,)}∩X=∅
      - (i) ∅ a λ jsou regulární výrazy
      - (ii) x je regulární výraz pro každé x∈X
      - (iii) (α+β) je regulární výraz, když α, β jsou regulární výrazy
      - (iv) (α.β) je regulární výraz, když α, β jsou regulární výrazy
      - (v) α* je regulární výraz, když α je regulární výraz
      - (vi) každý regulární výraz vznikne konečným použitím pravidel (i)-(v)

Př: X = {a,b,c,d}
a(bc)*a+cd
(ab)*(cd)*+ca

  - regulární výraz α **reprezentuje** jazyk [α], kde
    - [∅]=∅, [λ]={λ}, [x]={x} pro x∈X
    - [(α+β)]=[α]∪[β]
    - [(α.β)]=[α].[β]
    - [α*]=[α]*

Př: [a(bc)*a+cd] =
[a(bc)*a]∪[cd] =
{a}.[bc]*.{a} ∪{cd} =
{abca,abcbca, ...
abcbc...bca, cd}

  - z Kleeneho věty vidíme, že reprezentovaný jazyk je regulární a naopak libovolný regulární jazyk lze reprezentovat nějakým regulárním výrazem

Regulární **výraz** ⇒konečný **automat**

- regulárním jazykem **není** reprezentující regulární výraz určen **jednoznačně**
  - chceme rozpoznávat, zda dvojice regulárních výrazů reprezentuje stejný jazyk, tj. zda jsou *ekvivalentní*
    - převodem na konečný automat $A=(Q\cup\{q_0\}, X, \delta, \{q_0\}, F)$
      - regulární výrazy umožňují úspornou reprezentaci jazyka ⇒ rozhodnutí o ekvivalenci regulárních výrazů je PSPACE-úplný problém
    - 1. očíslujeme symboly v regulárním výrazu
      - množina stavů
    - 2. zjistíme, které očíslované symboly mohou stát **na začátku** reprezentovaného slova
      - počáteční stavy
    - 3. zjistíme, které dvojice očíslovaných symbolů mohou v reprezentovaném slově stát **vedle sebe**
      - přechodová funkce
    - 4. zjistíme, které očíslované symboly mohou stát **na konci** reprezentovaného slova
      - přijímající stavy
    - 5. speciálně ošetřit případ, kdy λ je reprezentováno regulárním výrazem

Př: $X = \{a,b,c,d\}$
$a(bc)^*a+cd$
$a_1(b_2c_3)^*a_4+c_5d_6$
$Q=\{a_1, b_2, c_3, a_4, c_5,d_6\}$

Př: začátek
$\{a_1, c_5\}$

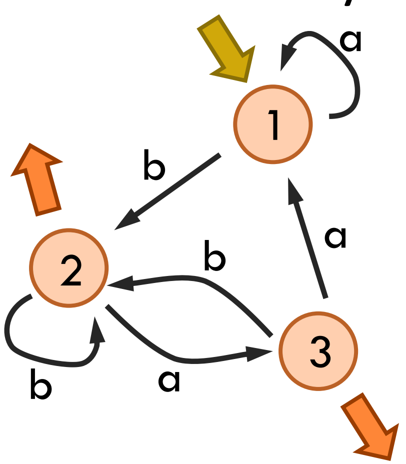
Př: sousedství
$\{[a_1,b_2], [b_2,c_3], [c_3,b_2], [c_3,a_4], [c_5,d_6]\}$

Př: konec
$\{a_4, d_6\}$

Př: λ reprezentováno není
q_0 nebude v F

Konečný **automat** ⇒ regulární **výraz**

- mějme KA $A=(Q, X, \delta, q_1, F)$, kde $Q = \{q_1, q_2, \ldots, q_n\}$
  - využijeme důkaz Kleeneho věty, tj. zkonstruujeme regulární výrazy pro $L_{i,j}$
    - induktivně podle: $L_{i,j}^{k+1} = L_{i,j}^{k} \cup L_{i,k+1}^{k}.(L_{k+1,k+1}^{k})^*.\ L_{k+1,j}^{k}$
    - výsledný regulární výraz sestavíme (aditivně) z regulárních výrazů pro $L_{1,i}$, kde $q_i \in F$



(a+b)*b + (a+b)*ba

$L_{i,j}^{0}$	1	2	3
1	a+λ	b	∅
2	∅	b+λ	a
3	a	b	λ

$L_{i,j}^{1}$	1	2	3
1	a*	a*b	∅
2	∅	b+λ	a
3	aa*	a*b	λ

$L_{i,j}^{2}$	1	2	3
1	a*	a*bb*	a*bb*a
2	∅	b*	b*a
3	aa*	a*bb*	a*bb*a+λ

$L_{i,j}^{3}$	1	2	3
1	_	(a+b)*b	(a+b)*ba
2	_	_	_
3	_	_	_