Automaty a gramatiky

5

Roman Barták, KTIML

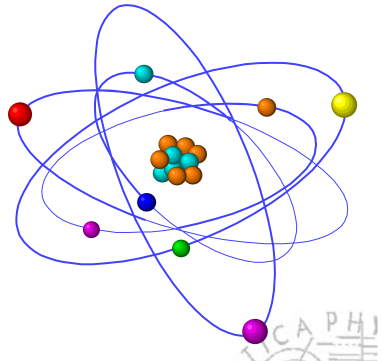
bartak@ktiml.mff.cuni.cz
http://ktiml.mff.cuni.cz/~bartak

Trochu motivace

L = { w | w=babau ∨ w=uabbv ∨ w=ubaa, u,v∈{a,b}* }

L = L_1 ∪ L_2 ∪ L_3, kde

- L_1 = { w | w=babau, u∈{a,b}* },
- L_2 = { w | w=uabbv, u,v∈{a,b}* }
- L_3 = { w | w=ubaa, u∈{a,b}* }



Můžeme jít ještě dál!

- L_1 = {baba} . {a,b}*
- L_2 = {a,b}* . {abb} . {a,b}*
- L_3 = {a,b}* . {baa}

Pojďme ještě dál

- L_3 = ({a} ∪ {b})* . {b}.{a}.{a}

Nešlo by všechny regulární jazyky „poskládat“ z nějakých triviálních jazyků?

Regulární jazyky

Třída regulárních jazyků RJ(X) nad konečnou neprázdnou abecedou X je nejmenší třída jazyků, která:

- obsahuje prázdný jazyk ∅
- pro každé písmeno x∈X obsahuje jazyk {x}
- A,B∈RJ(X) ⇒ A∪B ∈RJ(X) uzavřená na sjednocení
- A,B∈RJ(X) ⇒ A.B ∈RJ(X) uzavřená na zřetězení
- A∈RJ(X) ⇒ A* ∈RJ(X) uzavřená na iteraci

Vlastně algebraický popis jazyků!

Speciálně:

- {λ}∈RJ(X) protože {λ} = ∅*
- X∈RJ(X) protože X = $\cup_{x\in X}$ {x} (pozor! je to konečné sjednocení)
- $\{x_{i_1},\dots,x_{i_k}\}$∈RJ(X)
- X*∈RJ(X)

Kleeneova věta

Libovolný jazyk je regulární právě když je rozpoznatelný konečným automatem.

Konečnými automaty lze rozpoznávat jen triviální jazyky (prázdný a jednopísmenné) a jazyky, které z nich lze složit operacemi sjednocení, zřetězení a iterace.

Důkaz RJ ⇒ $\mathcal{F}$

regulární jazyky jsou rozpoznatelné konečnými automaty

- triviální jazyky jsou rozpoznatelné konečným automatem
- operace sjednocení, zřetězení a iteraci dávají opět jazyk rozpoznatelný konečným automatem

Důkaz Kleeneovy věty

jazyky rozpoznatelné konečnými automaty jsou regulární

máme automat $A=(Q,X,\delta,q_1,F)$, který přijímá jazyk L(A)
chceme ukázat, že L(A) dostaneme z elementárních jazyků a operací

definujme $R_{ij} = \{w \in X^* | \delta^*(q_i,w)=q_j\}$ — slova převádějící stav q_i na q_j

potom $L(A) = \cup_{q_i \in F} R_{1i}$ — slova převádějící počáteční stav q_1 na nějaký koncový stav q_i

jsou jazyky R_{ij} regulární?

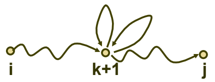
pokud ano, potom L(A) je také regulární, protože ∪ zachovává regulárnost

definujme R^k_{ij} =slova převádějící stav q_i na q_j bez meziprůchodu stavy q_m m>k

zřejmě $R_{ij} = R^n_{ij}$ (n je počet stavů automatu)

jsou jazyky R^k_{ij} regulární?

- R^0_{ij} je regulární (žádné mezistavy, tj. maximálně jednopísmenná slova)
- $R^{k+1}_{ij} = R^k_{ij} \cup R^k_{i,k+1}.(R^k_{k+1,k+1})^*.\ R^k_{k+1,j}$ je regulární (sjednocení a iterace regulárních jazyků)

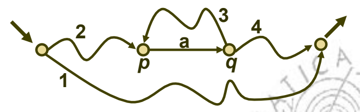


Alternativní důkaz Kleeneovy věty

jazyky rozpoznatelné konečnými automaty jsou regulární

Indukcí podle počtu hran v nedeterministickém automatu $A = (Q,X,\delta,S,F)$ pro daný jazyk L(A)

- *žádná hrana*
  - pouze jazyky ∅ nebo {λ}
- *(n+1) hran*
  - vybereme si jednu hranu: $p \to^a q$ tj. $q \in \delta(p,a)$
  - sestrojíme čtyři automaty bez této hrany (δ‘)
    - $A_1 = (Q,X,\delta',S,F)$
    - $A_2 = (Q,X,\delta',S,\{p\})$
    - $A_3 = (Q,X,\delta',\{q\},\{p\})$
    - $A_4 = (Q,X,\delta',\{q\},F)$



Potom $L(A) = L(A_1) \cup (L(A_2).a).(L(A_3).a)^*L(A_4)$

Jazyky $L(A_1)$, $L(A_2)$, $L(A_3)$, $L(A_4)$ jsou regulární (n hran)

Regulární výrazy

Množina regulárních výrazů RV(X) nad konečnou neprázdnou abecedou $X=\{x_1,\dots,x_n\}$ je nejmenší množina slov v abecedě $\{x_1,\dots,x_n,\emptyset, \lambda, +, . ,^*, (,)\}$, která:

- obsahuje výraz ∅ a výraz λ — $\emptyset \in RV(X)$, $\lambda \in RV(X)$
- pro každé písmeno $x \in X$ obsahuje výraz x — $x \in RV(X)$
- $\alpha,\beta \in RV(X) \Rightarrow (\alpha+\beta) \in RV(X)$
- $\alpha,\beta \in RV(X) \Rightarrow (\alpha.\beta) \in RV(X)$
- $\alpha \in RV(X) \Rightarrow \alpha^* \in RV(X)$

Příklad: ((a+((b.c)+d)*)+e)

Konvence:

- vnější závorky lze vynechat — (a+((b.c)+d)*)+e
- závorky lze vynechat u . a + díky asociativitě — a+((b.c)+d)*+e
- tečku lze vynechat — a+((bc)+d)*+e
- priorita operací (nejvyšší) *, . , + (nejnižší) — a+(bc+d)*+e

Hodnota regulárního výrazu

Hodnotou regulárního výrazu $\alpha \in RV(X)$ je množina slov $[\alpha]$ (jazyk) definovaná následovně:

- $[\emptyset] = \emptyset$, $[\lambda] = \{\lambda\}$, $[x] = \{x\}$
- $[(\alpha+\beta)] = [\alpha] \cup [\beta]$
- $[(\alpha.\beta)] = [\alpha]\ .\ [\beta]$
- $[\alpha^*] = [\alpha]^*$

Regulární výrazy odpovídají regulárním jazykům

- hodnotou regulárního výrazu je regulární jazyk
- každý regulární jazyk lze reprezentovat pomocí regulárního výrazu (jazyk je hodnotou tohoto výrazu)

Příklady:

[baba(a+b)* + (a+b)*abb(a+b)* + (a+b)*baa] =
= { w | w=babau ∨ w=uabbv ∨ w=ubaa, u,v∈{a,b}* }

[(0*10*10*1)*0*] =
= {w | w∈{0,1}* , $|w|_1$ =3k }

Použití regulárních výrazů

Praktický

přehledný zápis jazyka

Teoretický

zjednodušení některých důkazů

Věta: $L \in \mathcal{F}$, $\forall x \in X\ \sigma(x) \in \mathcal{F} \Rightarrow \sigma(L) \in \mathcal{F}$

L a $\sigma(x)$ jsou regulární jazyky, lze je tedy reprezentovat regulárními výrazy

každý výskyt x ve výrazu pro L stačí nahradit výrazem pro $\sigma(x)$

Rozšířené regulární výrazy

máme i další „regulární“ operace, např. průnik (α & β)

Ekvivalence regulárních výrazů

$\alpha \equiv \beta$ jestliže $[\alpha] = [\beta]$ (tj. výrazy reprezentují stejné jazyky)

Příklad: $(0^*1)^* \equiv \lambda + (0+1)^*1$

Jak to zjistíme?

Převod regulárního výrazu na konečný automat

Metoda 1 (inkrementální):

- převeď elementární jazyky (prázdný, jednopísmenné)
- spoj použitím regulárních operací podle výrazu

Metoda 2 (přímá)

$a+(bc+d)^*+a$

- očísluj symboly ve výrazu (zleva do doprava) — $a_1+(b_2c_3+d_4)^*+a_5$
- zjisti všechny možné páry symbolů, které se mohou vyskytovat za sebou — b_2c_3, c_3d_4, c_3b_2, d_4d_4, d_4b_2
- zjisti symboly, které mohou být první ve slově — a_1, b_2, d_4, a_5
- zjisti symboly, které mohou být poslední ve slově — a_1, c_3, d_4, a_5
- zjisti, zda jazyk obsahuje prázdné slovo — ANO
- vytvoř nedeterministický automat

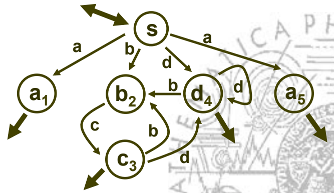
stavy: s + očíslované symboly

počátek = s

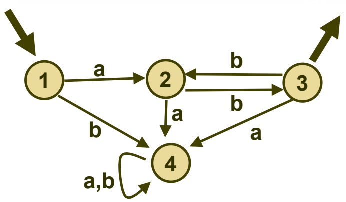
konec = poslední symboly (+s pro λ)

přechody: s→první symbol

$x_i \to x_j$, pokud je pár x_ix_j



Od automatu k regulárnímu výrazu



Pomocí Kleeneovy věty:

- R^0_{ij}
- $R^{k+1}_{ij} = R^k_{ij} \cup R^k_{i,k+1}.(R^k_{k+1,k+1})^*.\ R^k_{k+1,j}$

Pozn.: uzel 4 můžeme ignorovat (nevedou přes něj žádné cesty do ostatních uzlů)

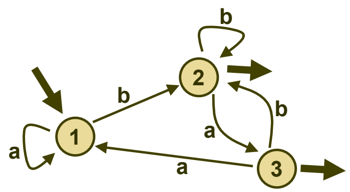
R^0	1	2	3
1	λ	a	$\emptyset$
2	$\emptyset$	λ	b
3	$\emptyset$	b	λ

R^1	1	2	3
1	λ	a	$\emptyset$
2	$\emptyset$	λ	b
3	$\emptyset$	b	λ

R^2	1	2	3
1	λ	a	ab
2	$\emptyset$	λ	b
3	$\emptyset$	b	$\lambda+b^2$

R^3	1	2	3
1	λ	$a(b^2)^*$	$ab(b^2)^*$
2	$\emptyset$	$(b^2)^*$	$b(b^2)^*$
3	$\emptyset$	$b(b^2)^*$	$(b^2)^*$

Od automatu k regulárnímu výrazu (příklad 2)



Pomocí Kleeneovy věty:

- R^0_{ij}
- $R^{k+1}_{ij} = R^k_{ij} \cup R^k_{i,k+1}.(R^k_{k+1,k+1})^*.\ R^k_{k+1,j}$

R^0	1	2	3
1	$a+\lambda$	b	$\emptyset$
2	$\emptyset$	$b+\lambda$	a
3	a	b	λ

R^1	1	2	3
1	a^*	a^*b	$\emptyset$
2	$\emptyset$	$b+\lambda$	a
3	a^+	a^*b	λ

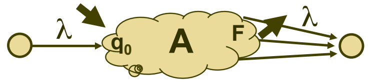
R^2	1	2	3
1	a^*	a^*b^+	a^*b^+a
2	$\emptyset$	b^*	b^*a
3	a^+	a^*b^+	a^*b^+a

R^3	1	2	3
1	?	$(a+b)^*b$	$(a+b)^*ba$
2	?	?	?
3	?	?	?

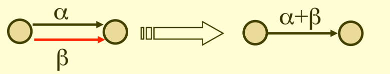
Od automatu k regulárnímu výrazu jinak

Ohodnocení hran regulárním výrazem

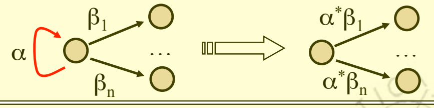
Nejprve vytvoříme automat s jedním vstupem a jedním výstupem



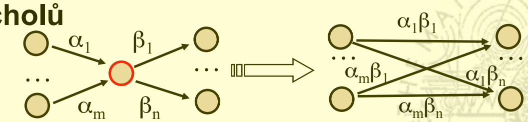
– spojení hran



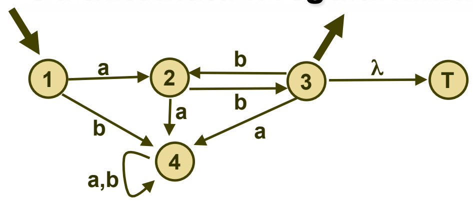
– eliminace smyček



– eliminace vrcholů



Od automatu k regulárnímu výrazu v příkladě

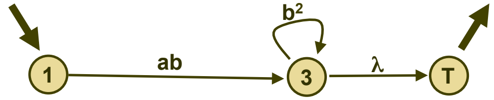


Stačí přidat pouze nový koncový stav.

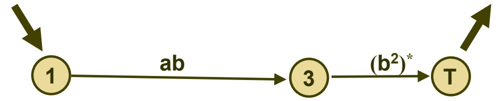
Eliminujeme smyčku 4.

Eliminujeme uzel 4.

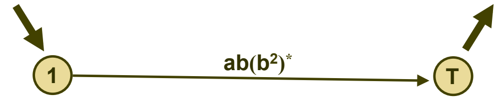
Eliminujeme uzel 2.



Eliminujeme smyčku 3.



Eliminujeme uzel 3.



Automaty s výstupem (motivace)

… aneb jak zaznamenat výpočet automatu?

Dosud jediná zpráva z automatu - jsme v přijímajícím stavu.
Můžeme z konečného automatu získat více informací?
Můžeme zaznamenat trasu výpočtu?

1) indikace stavů (všech, nejen koncových)
   v každé chvíli víme, kde se automat nachází
   Příklad: různé (regulární) čítače

2) indikace přechodů
   po přečtení každého symbolu víme, co automat udělal
   Příklad: (regulární) překlad slov

Automat už není tak docela černá skříňka.

Mooreův stroj

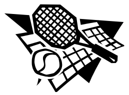
Mooreovým (sekvenčním) strojem nazýváme šestici
$A = (Q,X,Y,\delta,\mu,q_0)$ resp. pětici $A = (Q,X,Y,\delta,\mu)$, kde:

- Q - konečná neprázdná množina stavů (*stavový prostor*)
- X - konečná neprázdná množina symbolů (*vstupní abeceda*)
- Y - konečná neprázdná množina symbolů (*výstupní abeceda*)
- δ - zobrazení $Q \times X \to Q$ (*přechodová funkce*)
- μ - zobrazení $Q \to Y$ (*značkovací funkce*)
- $q_0 \in Q$ (*počáteční stav*)

Poznámky:

- někdy nás nezajímá počáteční stav, ale jen práce automatu
- značkovací funkce umožňuje suplovat roli koncových stavů
  $F \subseteq Q$ nahradíme značkovací funkcí $\mu : Q \to \{0,1\}$ takto:
  $\mu(q) = 0$, pokud $q \notin F$
  $= 1$, pokud $q \in F$

Příklad Mooreova stroje



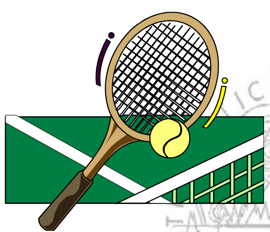
Navrhněte automat počítající tenisové skóre.

Vstupní abeceda: ID hráče, který uhrál bod

Výstupní abeceda/stavy: skóre (tj. Q=Y a $\mu(q)=q$)

Stav/výstup	*A*	*B*
00:00	15:00	00:15
15:00	30:00	15:15
15:15	30:15	15:30
00:15	15:15	00:30
30:00	40:00	30:15
30:15	40:15	30:30
30:30	40:30	30:40
15:30	30:30	15:40
00:30	15:30	00:40
40:00	A	40:15
40:15	A	40:30
40:30	A	shoda
30:40	shoda	B
15:40	30:40	B
00:40	15:40	B

Stav/výstup	*A*	*B*
shoda	A:40	40:A
A:40	A	shoda
40:A	shoda	B
A	15:00	00:15
B	15:00	00:15



Mealyho stroj

Mealyho (sekvenčním) strojem **nazýváme šestici**

A = (Q,X,Y,δ,λ,q_0) resp. pětici A = (Q,X,Y,δ,λ), kde:

- Q - konečná neprázdná množina stavů (*stavový prostor*)
- X - konečná neprázdná množina symbolů (*vstupní abeceda*)
- Y - konečná neprázdná množina symbolů (výstupní abeceda)
- δ - zobrazení $Q \times X \rightarrow Q$ (*přechodová funkce*)
- λ - zobrazení $Q \times X \rightarrow Y$ (*výstupní funkce*)
- $q_0 \in Q$ (*počáteční stav*)

Poznámka:

výstup je určen stavem a vstupním symbolem

tj. Mealyho stroj je obecnějším prostředkem než stroj Mooreův

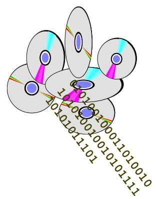
značkovací funkci $\mu : Q \rightarrow Y$ lze nahradit výstupní funkcí $\lambda: Q \times X \rightarrow Y$

například takto:

$\forall x \in X \;\; \lambda(q,x) = \mu(q)$

nebo takto $\forall x \in X \; \lambda(q,x) = \mu(\delta(q,x))$

Příklad Mealyho stroje

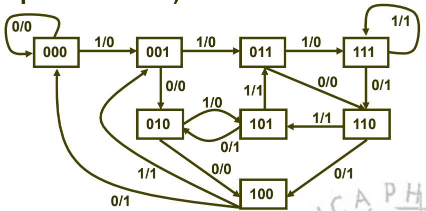


Navrhněte automat, který dělí vstupní slovo v binárním tvaru číslem 8 (celočíselně).

Realizace:

- posun o tři bity doprava (1101010 → 0001101)
- potřebujeme si pamatovat poslední trojici bitů (vlastně dynamická tříbitová paměť-buffer)

Stav\symbol	0	1
000	000/0	001/0
001	010/0	011/0
010	100/0	101/0
011	110/0	111/0
100	000/1	001/1
101	010/1	011/1
110	100/1	101/1
111	110/1	111/1



Vadí nám, když nevíme, kde automat startuje?

NE - po třech symbolech začne počítat správně

Výstup sekvenčních strojů

slovo ve vstupní abecedě → slovo ve výstupní abecedě

Mooreův stroj

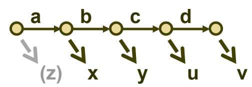
značkovací funkce $\mu: Q \rightarrow Y$

$\mu^* : Q \times X^* \rightarrow Y^*$

$\mu^*(q,\lambda) = \lambda$ (někdy $\mu^*(q,\lambda) = \mu(q)$)

$\mu^*(q,wx) = \mu^*(q,w) \,.\, \mu(\delta^*(q,wx))$

Příklad: μ^*(00:00,AABA) = (00:00 .) 15:00 . 30:00 . 30:15 . 40:15



Mealyho stroj

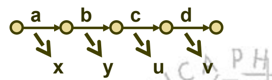
výstupní funkce $\lambda: Q \times X \rightarrow Y$

$\lambda^* : Q \times X^* \rightarrow Y^*$

$\lambda^*(q,\lambda) = \lambda$

$\lambda^*(q,wx) = \lambda^*(q,w) \,.\, \lambda(\delta^*(q,w),x)$

Příklad: μ^*(‚000‘,1101010) = 0001101



Převod Mooreova stroje na Mealyho

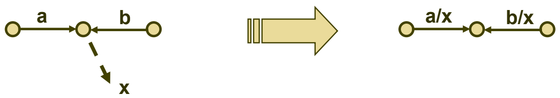
Nechť $A = (Q,X,Y,\delta,\mu,q_0)$ je Mooreův stroj.

Umíme najít Mealyho stroj B tak, že $\forall q,w\ \mu^*(q,w) = \lambda^*(q,w)$?

ANO!

položme $B = (Q,X,Y,\delta,\lambda,q_0)$, kde $\lambda(q,x) = \mu(\delta(q,x))$

tj. λ vrací značku stavu, do kterého přejdeme



Příklad:

stav	0	1	výstup
a	a	b	0
b	b	c	1
c	c	a	2

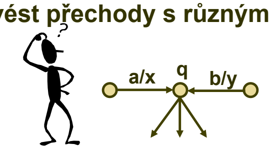
stav	0	1
a	a/0	b/1
b	b/1	c/2
c	c/2	a/0

Převod Mealyho stroje na Mooreův

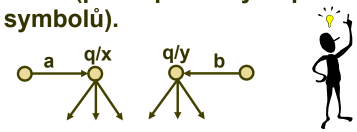
Nechť $A = (Q,X,Y,\delta,\lambda,q_0)$, je Mealyho stroj.

Sestrojme Mooreův stroj B tak, že $\forall q,w\ \lambda^*(q,w) = \mu^*(q,w)$.

Problém: do jednoho stavu mohou vést přechody s různým výstupem!



Řešení: stav rozdělíme na více stavů (podle počtu výstupních symbolů).



Teď už je to jednoduché! $B = (Q\times Y,X,Y,\delta',\mu,(q_0,\_))$, kde

$\delta'((q,y),x) = (\delta(q,x), \lambda(q,x))$ a $\mu((q,y)) = y$

Příklad:

stav	0	1
a	a/0	b/0
b	a/1	b/1

stav	0	1	výstup
(a,0)	(a,0)	(b,0)	0
(a,1)	(a,0)	(b,0)	1
(b,0)	(a,1)	(b,1)	0
(b,1)	(a,1)	(b,1)	1

Konečné automaty - shrnutí

Konečný automat

- jednoznačný redukovaný automat
- nedeterminismus (2^n), dvousměrný KA (n^n)

Automaty a jazyky

- regulární jazyky
- uzavřenost na množinové operace
- uzavřenost na řetězcové operace
- uzavřenost substituce

Charakteristika regulárních jazyků

- Nerodova věta (kongruence)
- Kleeneova věta (elementární jazyky a operace)
- Iterační lemma (iterace podslov, jen nutná podmínka)