Automaty a gramatiky

9

Roman Barták, KTIML

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

Co bylo minule

Determinismus vs.nedeterminismus

- deterministický ZA - každý krok je jednoznačně určen
  - $\forall p \in Q, \forall a \in X \cup \{\lambda\}, \forall Z \in Y \quad |\delta(p,a,Z)| \le 1$
  - $\forall p \in Q, \forall Z \in Y \ (\delta(p,\lambda,Z) \neq \emptyset \Rightarrow \forall a \in X \ \delta(p,a,Z) = \emptyset)$
- determinismus oslabuje ZA (ww^R)
- determinismus rozlišuje přijímání zásobníkem a stavem
  - deterministické BK jazyky (přijímání stavem)
  - bezprefixové BK jazyky (přijímání zásobníkem)

Zásobníkové automaty a bezkontextové gramatiky

- převod BKG na ZA
  - pravidla gramatiky → pravidla automatu
  - + pravidla pro krácení terminálů
- převod ZA na BKG
  - pro jednostavové ZA jednoduché
  - pro vícestavové ZA - stavy simulovány v neterminálech

Jak na bezprefixové jazyky?

Jak se od deterministického jazyka dostaneme k bezprefixovému?

Co nám vadí?

po přečtení prefixu, který patří do jazyka, máme prázdný zásobník.

Řešení:

přidáme speciální znak na konec slova (až po přečtení tohoto znaku vyprázdníme zásobník)

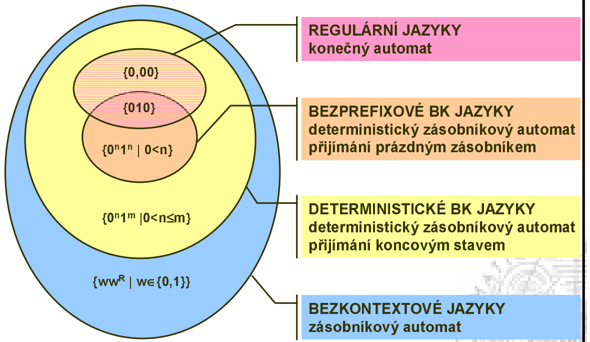
Nechť $L \subseteq X^*$ je deterministický jazyk a $\# \notin X$, potom $L\#$ je bezprefixový jazyk.

\# opraví nedeterministické λ -kroky při převodu automatů

Příklad: $\{0^n1^m\# \mid 0<n \le m\}$

$\delta(p,0,Z) = \{(p,AZ)\}$	… čte první symbol 0
$\delta(p,0,A) = \{(p,AA)\}$	… čte další symboly 0
$\delta(p,1,A) = \{(q,\lambda)\}$	… čte první symbol 1
$\delta(q,1,A) = \{(q,\lambda)\}$	… čte další symboly 1
$\delta(q,\lambda,Z) = \{(q_{OK},Z)\}$	… nyní se počet 0 a 1 vyrovnal
$\delta(q_{OK},1,Z) = \{(q_{OK},Z)\}$	… dále čteme jen 1
$\delta(q_{OK},\#,Z) = \{(q_{OK},\lambda)\}$	… na konci vyprázdníme zásobník

Jazyky a automaty - přehled



Od gramatiky k automatu

$G=(V_N,V_T,S,P) \quad \rightarrow \quad M=(\{p\},V_T,V_N \cup V_T,\delta,p,S,\{\})$

$\delta(p,\lambda,X)=\{(p,w) \mid (X \rightarrow w) \in P\} \quad \forall X \in V_N$ … opracování zásobníku

$\delta(p,a,a)=\{(p,\lambda)\} \quad \forall a \in V_T$ … krácení terminálů

Příklad: $L= \{a^ib^jc^k \mid i+j=k\}$

Gramatika	*Zásobníkový automat*
$S \rightarrow aSc \mid A$ $A \rightarrow bAc \mid \lambda$	$\delta(p,\lambda,S) = \{ (p,aSc), (p,A) \}$ $\delta(p,\lambda,A) = \{ (p,bAc), (p,\lambda) \}$ $\delta(p,x,x) = \{ (p,\lambda) \} \quad \forall x \in \{a,b,c\}$

$S \Rightarrow aSc \Rightarrow aAc \Rightarrow abAcc \Rightarrow abbAccc \Rightarrow abbccc$

$(p,abbccc,S) \vdash (p,abbccc,aSc) \vdash (p,bbccc,Sc) \vdash (p,bbccc,Ac)$
$\vdash (p,bbccc,bAcc) \vdash (p,bccc,Acc) \vdash (p,bccc,bAccc)$
$\vdash (p,ccc,Accc) \vdash (p,ccc,ccc) \vdash (p,cc,cc) \vdash (p,c,c) \vdash (p, \lambda, \lambda)$

Od automatu ke gramatice

Pro jednostavový ZA, stačí reverzní proces k BKG→ZA.

Pro vícestavový ZA:

- převést na jednostavový ZA
- přímo na gramatiku

Převod vícestavového ZA na gramatiku:

pravidla gramatiky zachycují všechny možné výpočty

neterminální symboly: $[q,Z,p]$, kde $p,q \in Q$, $Z \in Y$

q je stav automatu těsně před tím, než se Z přepíše

p je stav automatu, když začínáme počítat pod Z

pravidla gramatiky:

$S \rightarrow [q_0,Z_0,q]$ nastartování výpočtu (q - nevíme, kde skončí)

$[q,A,p] \rightarrow a\,[q_1,B_1,q_2]\,[q_2,B_2,q_3] \dots [q_m,B_m,p]$

$\delta(q,a,A) \ni (q_1, B_1 B_2 \dots B_m)$

$q_2,\dots, q_m,p$ libovolné stavy - nevíme, jak výpočet vypadá

speciálně: $[q,A,p] \rightarrow a$, pro $\delta(q,a,A) \ni (p,\lambda)$

Výpočet automatu → derivace

$(q,w,A) \vdash^* (p,\lambda,\lambda) \quad \Rightarrow \quad [q,A,p] \Rightarrow^* w$

indukcí dle délky výpočtu

k=1

$w \in X \cup \{\lambda\}$, $\delta(q,w,A) \ni (p,\lambda)$, tj. máme pravidlo $[q,A,p] \to w$

k>1 (pro výpočty kratší než k platí)

$(q,au_1 \dots u_l,A) \vdash (q_1, u_1 \dots u_l, B_1 \dots B_l)$, $l \geq 1$ první krok výpočtu dle přechodu $\delta(q,a,A) \ni (q_1, B_1 B_2 \dots B_l)$

u_i jsou slova nutná ke zpracování zásobníkového symbolu B_i

tj. $(q_i,u_i,B_i) \vdash^* (q_{i+1},\lambda,\lambda)$, pro vhodná q_i ($q_{l+1}=p$)

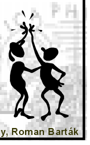
tyto výpočty jsou nutně kratší než *k*

tj. dle indukčního předpokladu $[q_i,B_i,q_{i+1}] \Rightarrow^* u_i$

dohromady:

$[q,A,p] \to a\, [q_1,B_1,q_2]\, [q_2,B_2,q_3] \dots [q_l,B_l,p]$

$[q,A,p] \Rightarrow^* a \quad u_1 \quad u_2 \quad \dots \quad u_l$



Derivace → výpočet automatu

$[q,A,p] \Rightarrow^* w \quad \Rightarrow \quad (q,w,A) \vdash^* (p,\lambda,\lambda)$

indukcí dle délky (levé) derivace

k=1

jediné pravidlo $[q,A,p] \to w$ muselo vzniknout z $\delta(q,w,A) \ni (p,\lambda)$

k>1 (pro derivace kratší než k platí)

$[q,A,p] \to a\, [q_1,B_1,q_2]\, [q_2,B_2,q_3] \dots [q_l,B_l,p]$ první použité pravidlo vzniklo z přechodu $\delta(q,a,A) \ni (q_1, B_1 B_2 \dots B_l)$

potom $w = a\, u_1 \dots u_l$, kde $[q_i,B_i,q_{i+1}] \Rightarrow^* u_i$ ($q_{l+1}=p$)

tyto derivace jsou nutně kratší než *k*

tj. dle indukčního předpokladu $(q_i,u_i,B_i) \vdash^* (q_{i+1},\lambda,\lambda)$

dohromady: „slepíme" výpočty a dostaneme

$(q,w,A) \vdash (q_1, u_1 \dots u_l, B_1 B_2 \dots B_l) \vdash^* (p,\lambda,\lambda)$

Derivace vždy začíná nějakým pravidlem $S \to [q_0,Z_0,q]$, tj. $L(G)=N(M)$.

Příklad převodu automatu na gramatiku

$\delta(q,x,A) \ni \{(q_1,B^1..B^m)\}$ …….. ${}_qA_p \to x\, [{}_{q_1}B^1{}_{q_2}] \dots [{}_{q_m}B^m{}_p]$

Příklad: $L = \{0^n1^n \mid n \geq 0\}$

Automat	*Gramatika*
	$S \to {}_pZ_p \mid {}_pZ_q$
$\delta(p,\lambda,Z) = \{(p,\lambda)\}$	${}_pZ_p \to \lambda$
$\delta(p,0,Z) = \{(p,A)\}$	$\cancel{{}_pZ_p \to 0\, {}_pA_p}$ ${}_pZ_q \to 0\, {}_pA_q$
$\delta(p,0,A) = \{(p,AA)\}$	$\cancel{{}_pA_p \to 0\, {}_pA_p\, {}_pA_p \mid 0\, {}_pA_q\, {}_qA_p}$ ${}_pA_q \to \cancel{0\, {}_pA_p\, {}_pA_q} \mid 0\, {}_pA_q\, {}_qA_q$
$\delta(p,1,A) = \{(q,\lambda)\}$	${}_pA_q \to 1$
$\delta(q,1,A) = \{(q,\lambda)\}$	${}_qA_q \to 1$

$S \Rightarrow {}_pZ_q \Rightarrow 0\, {}_pA_q \Rightarrow 00\, {}_pA_q\, {}_qA_q \Rightarrow 001\, {}_qA_q \Rightarrow 0011$

Greibachové normální forma

Co nám vadí při analýza slova?

Nevíme, jaké pravidlo vybrat!

Speciálně vadí pravidla tvaru $A \to Au$ (levá rekurze).

Definice: Říkáme, že gramatika je v ***Greibachové normální formě*** (tvaru), jestliže všechna pravidla mají tvar: $A \to au$, kde $a \in V_T$, $u \in V^*_N$.

K čemu je tento tvar dobrý?

Srovnáním terminálu na pravé straně pravidel a čteného symbolu můžeme zjistit, jaké pravidlo použít, pokud je ovšem takové pravidlo jediné.

Věta: Ke každému bezkontextovému jazyku L existuje bezkontextová gramatika G v Greibachové normální formě taková, že $L(G) = L - \{\lambda\}$.

Spojení pravidel a odstranění levé rekurze

Lemma (spojení pravidel):

Nechť $A \to uBv$ je pravidlo gramatiky G a $B \to w_1, \dots, B \to w_k$ jsou všechna pravidla pro B. Potom nahrazením pravidla $A \to uBv$ pravidly $A \to uw_1v, \dots, A \to uw_kv$ dostaneme ekvivalentní gramatiku.

Důkaz: $A \Rightarrow uBv \Rightarrow^* u'Bv \Rightarrow u'w_iv$ v původní gramatice

$A \Rightarrow uw_iv \Rightarrow^* u'w_iv$ v nové gramatice

Lemma (odstranění levé rekurze):

Nechť $A \to Au_1, \dots, A \to Au_k$ jsou všechna levě rekurzivní pravidla gramatiky G pro A a $A \to v_1, \dots, A \to v_m$ jsou všechna ostatní pravidla pro A. Potom nahrazením všech těchto pravidel pravidly:

1) $A \to v_i$, $A \to v_iZ$, $Z \to u_j$, $Z \to u_jZ$, nebo
2) $A \to v_iZ$, $Z \to u_jZ$, $Z \to \lambda$

(Z je nový neterminál) dostaneme ekvivalentní gramatiku.

Důkaz:

$A \Rightarrow Au_{i_n} \Rightarrow \dots \Rightarrow Au_{i_1} \dots u_{i_n} \Rightarrow v_ju_{i_1} \dots u_{i_n}$ (G)

$A \Rightarrow v_jZ \Rightarrow v_ju_{i_1}Z \Rightarrow \dots \Rightarrow v_ju_{i_1} \dots u_{i_{n-1}}Z \Rightarrow v_ju_{i_1} \dots u_{i_n}$ (1)

$A \Rightarrow v_jZ \Rightarrow v_ju_{i_1}Z \Rightarrow \dots \Rightarrow v_ju_{i_1} \dots u_{i_n}Z \Rightarrow v_ju_{i_1} \dots u_{i_n}$ (2)

Převod na Greibachové NF

Věta: Libovolnou bezkontextovou gramatiku lze převést na gramatiku v Greibachové normální formě.

Důkaz:

spojování pravidel a odstraňování levé rekurze

1) neterminály očíslujeme $\{A_1, \dots, A_n\}$
2) rekurzivní pravidla pouze tvaru $A_i \to A_ju$, kde $i<j$
   postupnou iterací od 1 do n
   $A_i \to A_ju$ pro $j<i$ odstraníme spojováním pravidel
   pro $j=i$ odstraníme levou rekurzi
   získáme pravidla tvaru $A_i \to A_ju$ ($i<j$), $A_i \to au$ ($a \in V_T$), $Z_i \to u$
3) pravidla s A_i (původní neterminály) pouze tvaru $A_i \to au$
   postupným spojování pravidel od n do 1 (pro n již platí)
4) pravidla s Z_i (nové neterminály) pouze tvaru $Z_i \to au$
   žádné pravidlo Z_i pro nezačíná vpravo Z_j
   buď je v požadovaném tvaru nebo se spojí z pravidlem $A_j \to au$
5) odstranění terminálů uvnitř pravidel

Příklad převodu na Greibachové NF

Původní gramatika

E → E+T | T
T → T*F | F
F → (E) | a

Odstranění levé rekurze

E → T | TE'
E' → +T | +TE'
T → F | FT'
T' → *F | *FT'
F → (E) | a

(téměř) Greibachové normální forma

E → (E) | a | (E)T' | aT' | (E)E' | aE' | (E)T'E' | aT'E'
E' → +T | +TE'
T → (E) | a | (E)T' | aT'
T' → *F | *FT'
F → (E) | a

Chomského normální forma

Podívejme se nyní na derivační stromy.

Jak odhadnout výšku stromu podle délky slova?

Definice: **Říkáme, že gramatika je v *Chomského normální formě* (tvaru), jestliže všechna pravidla mají tvar:**
$X \to YZ$ nebo $X \to a$, kde $a \in V_T$, $X,Y,Z \in V_N$.

K čemu je tento tvar dobrý?

Derivační strom je (skoro) binární.

Je-li maximální cesta délky k, potom terminální slovo $\leq 2^{k-1}$.

Pumping lemma pro bezkontextové jazyky

Věta: **Ka každému bezkontextovému jazyku L existuje bezkontextová gramatika G v Chomského normální formě taková, že $L(G) = L - \{\lambda\}$.**

Odstranění pravidel A→B

Lemma (odstranění A→B):
Pro každou bezkontextovou gramatiku G lze sestrojit ekvivalentní gramatiku G' takovou, že neobsahuje pravidla tvaru A→B.

Důkaz:

hledáme $U(X) = \{Y \mid X \Rightarrow^* Y\}$

$U_1(X) = \{Y \mid X \to Y\}$

$U_{k+1}(X) = U_k(X) \cup \{Y \mid Z \to Y, Z \in U_k(X)\}$

stabilizace po maximálně $n=|V_N|$ krocích

pravidla A→B odstraníme z gramatiky

přidáme pravidla A→u,
kde B→u je přípustné pravidlo gramatiky G a $B \in U(A)$

$L(G) \supseteq L(G')$

pokud derivace obsahuje nové pravidlo A→u nahradíme ho sekvencí $A \Rightarrow^* B \Rightarrow u$

$L(G) \subseteq L(G')$

najdeme nejdelší posloupnosti $A \Rightarrow^* B \Rightarrow u$ složené z nepřípustných pravidel a nahradíme je $A \Rightarrow u$

Převod na Chomského NF

Povoleno pouze $X \to YZ$ nebo $X \to a$, kde $a \in V_T$, $X,Y,Z \in V_N$.

Víme:

pravidla A→B lze odstranit (ekvivalentní gramatiky)

pravidla A→λ lze odstranit (maximálně přijdeme o λ)

Dále:

pravidla tvaru $X \to a$, kde $a \in V_T$ jsou v pořádku

zbývají pravidla tvaru $X \to B_1 \dots B_k$, $k \geq 2$, $B_i \in V_N \cup V_T$

vytvoříme pravidlo $X \to C_1 \dots C_k$, kde:

$C_i = B_i$ je-li $B_i \in V_N$
$\quad = B'_i$ je-li $B_i \in V_T$ (B'_i je nový neterminál)
\+ přidáme pravidla $B'_i \to B_i$

pravidlo $X \to C_1 \dots C_k$, $k \geq 3$ nahradíme pravidly:

$X \to C_1D_1$, $D_1 \to C_2D_2$, …, $D_{k-2} \to C_{k-1}C_k$,
D_i jsou nové neterminály

Příklad převodu na Chomského NF

Původní gramatika

A → B | C
B → 0B1 | 01
C → D | E
D → 1D0 | 1
E → 0E | 0

Po odstranění X → Y

A → 0B1 | 01 | 1D0 | 1 | 0E | 0
B → 0B1 | 01
~~C → 1D0 | 1 | 0E | 0~~
D → 1D0 | 1
E → 0E | 0

Po nahrazení terminálů

A → NBJ | NJ | JDN | 1 | NE | 0
B → NBJ | NJ
D → JDN | 1
E → NE | 0
N → 0
J → 1

Chomského normální forma

$A \to NA_1 \mid NJ \mid JA_2 \mid 1 \mid NE \mid 0$
$A_1 \to BJ$
$A_2 \to DN$
$B \to NB_1 \mid NJ$
$B_1 \to BJ$
$D \to JD_1 \mid 1$
$D_1 \to DN$
E → NE | 0
N → 0
J → 1