Automaty a gramatiky

9

Roman Barták, KTIML

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

Od automatu ke gramatice

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

Pro v

–

–

Převo

pr ... y

ne

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

Automat	*Gramatika*
$\delta(q, 0, Z) = \{(q,ZJ)\}$	$Z \rightarrow 0ZJ$
$\delta(q, \lambda, Z) = \{(q, \lambda)\}$	$Z \rightarrow \lambda$
$\delta(q, 1, J) = \{(q, \lambda)\}$	$J \rightarrow 1$

p je stav automatu, když začínáme počítat pod Z

pravidla gramatiky:

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

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

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

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

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

Od automatu ke gramatice

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

Pro více-stavový ZA:

- převést na jednostavový ZA
- přímo na gramatiku

Převod více-stavového 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:

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

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

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

$q_2,\ldots, q_m,p$ 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 $[q,A,p] \rightarrow w$

k>1 (pro výpočty kratší než k platí)

$(q,au_1\ldots u_l,A) \vdash (q_1, u_1\ldots u_l, B_1\ldots B_l)$, $l \geq 1$ první krok výpočtu

dle přechodu $\delta(q,a,A) \ni (q_1, B_1 B_2\ldots B_l)$

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 $(q_{l+1}=p)$

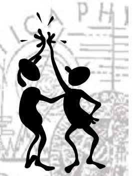
tyto výpočty jsou nutně kratší než *k*

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

dohromady:

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

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



Derivace → výpočet automatu

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

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)

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

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

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

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

Příklad převodu automatu na gramatiku

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

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

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

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

Deterministické zásobníkové automaty

Jak je to s nedeterminismem zásobníkových automatů?

– Je nutný!
– Pro rozpoznání ww^R potřebujeme nedeterministicky uhádnout střed.

Kde je skryt nedeterminismus zásobníkového automatu?

– množina možných přechodů
– opracování zásobníku bez čtení vstupu (λ-přechod)

Definice: Říkáme, že *zásobníkový automat* $M=(Q,X,Y,\delta,q_0,Z_0,F)$, je *deterministický*, jestliže platí:

– $\forall p \in Q,\ \forall a \in X \cup \{\lambda\},\ \forall Z \in Y \qquad |\delta(p,a,Z)| \leq 1$
– $\forall p \in Q,\ \forall Z \in Y\ (\ \delta(p,\lambda,Z) \neq \emptyset \Rightarrow \forall a \in X\ \delta(p,a,Z)=\emptyset\ )$

Každý krok výpočtu je přesně určen.

Příklady zásobníkových automatů

Deterministický zásobníkový automat (prázdný zásobník)

$L = \{0^n1^n \mid n>0\}$

$\delta(p,0,Z) = \{(p,A)\}$ … 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

(Klasický) zásobníkový automat (prázdný zásobník)

$L = \{ww^R \mid w \in \{0,1\}^* \}$

$\delta(p,0,X) = \{(p,NX)\}$ … čte 0 v první půlce (uschovává na zásobníku)
$\delta(p,1,X) = \{(p,JX)\}$ … čte 1 v první půlce (uschovává na zásobníku)
$\delta(p,\lambda,X) = \{(q,X)\}$ … překlopení do druhé půlky (nedeterminismus)
$X \in \{Z,N,J\}$
$\delta(q,0,N) = \{(q,\lambda)\}$ … čte 0 symetricky v druhé půlce
$\delta(q,1,J) = \{(q,\lambda)\}$ … čte 1 symetricky v druhé půlce
$\delta(q,\lambda,Z) = \{(q,\lambda)\}$ … ukončuje výpočet (není nedeterminismus)

Deterministické a bezprefixové jazyky

Již „víme“, že determinismus je u ZA slabší než nedeterminismus (ww^R).

Jak je to s přijímáním slov?

Deterministické bezkontextové jazyky **jsou jazyky rozpoznávané DZA koncovým stavem.**

Tvrzení: Regulární jazyk je deterministický BKJ.

Zásobník nemusíme využívat.

Bezprefixové bezkontextové jazyky **jsou jazyky rozpoznávané DZA prázdným zásobníkem.**

Pozorování: M je DZA, $u \in N(M) \Rightarrow \forall w \in X^+ \; uw \notin N(M)$

Jakmile jednou vyprázdníme zásobník (po přečtení *u*, které je přijato), nemůže výpočet pokračovat (čtením *w*).

Tvrzení: Bezprefixový BKJ je deterministický BKJ.

Převod nepřidává nedeterminismus.

Koncové stavy vs. prázdný zásobník

Je také každý deterministický jazyk bezprefixový?

NE!

Z vlastnosti M je DZA, $u \in N(M) \Rightarrow \forall w \in X^+ \; uw \notin N(M)$ snadno sestrojíme příslušný jazyk.

Potřebujeme:

základní deterministický jazyk, který

obsahuje slovo, jež je prefixem jiného přijímaného slova

Například $\{0^n1^m \mid 0<n\leq m\}$ (uděláme DZA, q_F je koncový stav).

$\delta(p,0,Z) = \{(p,AZ)\}$ … 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

$\delta(q,\lambda,Z) = \{(q_F,Z)\}$ … nyní se počet 0 a 1 vyrovnal

$\delta(q_F,1,Z) = \{(q_F,Z)\}$ … dále čteme jen 1

to už „prázdný zásobník“ nemůže

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

$\delta(p,0,Z) = \{(p,AZ)\}$ … čte první symbol 0

$\delta(p,0,A) = \{(p,AA)\}$ … čte další symboly 0

$\delta(p,1,A) = \{(q,\lambda)\}$ … čte první symbol 1

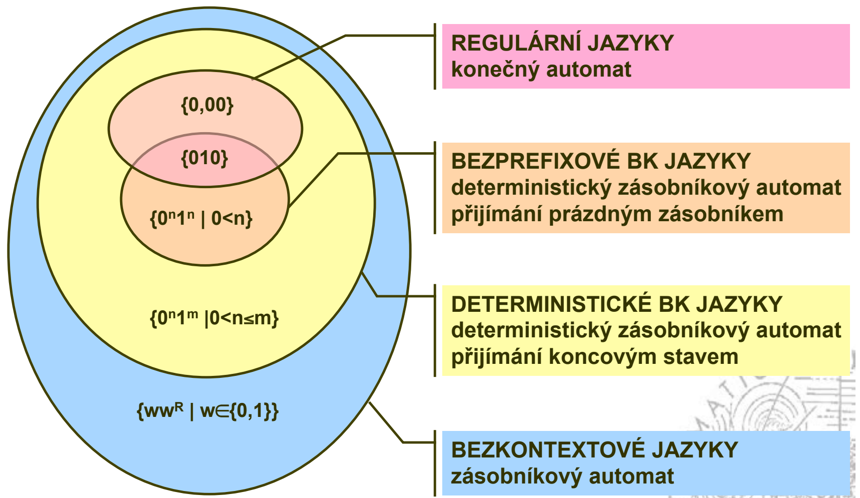
$\delta(q,1,A) = \{(q,\lambda)\}$ … čte další symboly 1

$\delta(q,\lambda,Z) = \{(q_{OK},Z)\}$ … 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



Greibachové normální forma

Co nám vadí při analýza slova?

Nevíme, jaké pravidlo vybrat!

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

Definice: **Říkáme, že gramatika je v *Greibachové normální formě* (tvaru), jestliže všechna pravidla mají tvar: $A \rightarrow au$, 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 $A \rightarrow uBv$ je pravidlo gramatiky G a $B \rightarrow w_1, \ldots, B \rightarrow w_k$ jsou všechna pravidla pro B. Potom nahrazením pravidla $A \rightarrow uBv$ pravidly $A \rightarrow uw_1v, \ldots, A \rightarrow uw_kv$ 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 $A \rightarrow Au_1, \ldots, A \rightarrow Au_k$ jsou všechna levě rekurzivní pravidla gramatiky G pro A a $A \rightarrow v_1, \ldots, A \rightarrow v_m$ jsou všechna ostatní pravidla pro A. Potom nahrazením všech těchto pravidel pravidly:

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

(Z je nový neterminál) dostaneme ekvivalentní gramatiku.

Důkaz:

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

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

$A \Rightarrow v_jZ \Rightarrow v_ju_{i_1}Z \Rightarrow \ldots \Rightarrow v_ju_{i_1}\ldots u_{i_n}Z \Rightarrow v_ju_{i_1}\ldots u_{i_n}$ (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 libovolně očíslujeme $\{A_1, \ldots, A_n\}$

2) povolíme rekurzivní pravidla pouze tvaru $A_i \rightarrow A_ju$, kde $i<j$

postupnou iterací od 1 do n

$A_i \rightarrow A_ju$ pro $j<i$ odstraníme spojováním pravidel; pro $j=i$ odstraníme levou rekurzi

získáme pravidla tvaru $A_i \rightarrow A_ju$ ($i<j$), $A_i \rightarrow au$ ($a \in V_T$), $Z_i \rightarrow u$

3) pravidla s A_i (původní neterminály) pouze tvaru $A_i \rightarrow au$

postupným spojováním pravidel od n do 1 (pro n již platí)

4) pravidla s Z_i (nové neterminály) pouze tvaru $Z_i \rightarrow au$

žádné pravidlo Z_i pro nezačíná vpravo Z_j

buď je v požadovaném tvaru nebo se spojí z pravidlem $A_j \rightarrow au$

5) odstranění terminálů uvnitř pravidel

Příklad převodu na Greibachové NF

Původní gramatika

$E \rightarrow E+T \mid T$

$T \rightarrow T*F \mid F$

$F \rightarrow (E) \mid a$

Odstranění levé rekurze

$E \rightarrow T \mid TE'$

$E' \rightarrow +T \mid +TE'$

$T \rightarrow F \mid FT'$

$T' \rightarrow *F \mid *FT'$

$F \rightarrow (E) \mid a$

(téměř) Greibachové normální forma

$E \rightarrow (E) \mid a \mid (E)T' \mid aT' \mid (E)E' \mid aE' \mid (E)T'E' \mid aT'E'$

$E' \rightarrow +T \mid +TE'$

$T \rightarrow (E) \mid a \mid (E)T' \mid aT'$

$T' \rightarrow *F \mid *FT'$

$F \rightarrow (E) \mid a$