Automaty a gramatiky

10

Roman Barták, KTIML

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

Co bylo minule

Od determinismu k bezprefixovosti
- koncový symbol zajišťuje bezprefixovost

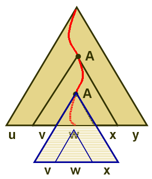
Zásobníkové automaty a gramatiky
- gramatika modeluje výpočty automatu

Normální formy
- *Greibachové normální forma* ($A \rightarrow au$)
  - odstranění levé rekurze
  - konstrukce analyzátorů jazyka
- *Chomského normální forma* ($X \rightarrow YZ$ nebo $X \rightarrow a$)
  - binární derivační strom
  - ???

Pumping lemma pro bezkontexové jazyky

Lemma o vkládání: Nechť L je bezkontextový jazyk. Potom existují přirozená čísla p, q taková, že každé slovo $z \in L$, $|z|>p$ lze psát ve tvaru $z = uvwxy$ a platí:
1) $|vwx| \leq q$ (pumpovací část není moc dlouhá)
2) buď $v \neq \lambda$ nebo $x \neq \lambda$ (lze psát $vx \neq \lambda$)
3) $\forall i \geq 0\ uv^iwx^iy \in L$

Idea důkazu:



- Vezmeme derivační strom pro slovo z
- V derivačním stromu najdeme nejdelší cestu
- Na této cestě najdeme dva stejné neterminály
- Tyto neterminály určí dva podstromy
- Podstromy definují rozklad slova
- Nyní můžeme větší podstrom posunout (i>1)
- nebo nahradit menším podstromem (i=0)

Důkaz lemma o vkládání pro BKJ

$|z|>p$: $z = uvwxy$, $|vwx| \leq q$, $vx \neq \lambda$, $\forall i \geq 0\ uv^iwx^iy \in L$

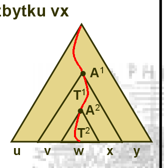
vezmeme gramatiku v Chomského NF (slova λ nevadí)
$|V_N|=k$, položme $p = 2^{k-1}$, $q = 2^k$
$|z| > 2^{k-1}$, v libovolném derivačním stromu je cesta délky >k
na této cestě musí ležet dva stejné neterminály a terminál *t*
vezmeme dvojici A^1, A^2 nejblíže k *t* (určuje podstromy T^1 a T^2)
cesta z A^1 do *t* je nejdelší v podstromu T^1 a má délku maximálně k+1
- tedy slovo dané stromem T^1 není delší než 2^k ($|vwx| \leq q$))

z A^1 vedou dvě cesty (ChNF), jedno do T^2 druhá do zbytku vx
- ChNF je nevypouštějící, tedy $vx \neq \lambda$

derivace slova ($A^1 \Rightarrow^* vA^2x$, $A^2 \Rightarrow^* w$)
- $S \Rightarrow^* uA^1y \Rightarrow^* uvA^2xy \Rightarrow^* uvwxy$

posuneme-li A^2 do A^1 (i=0)
- $S \Rightarrow^* uA^2y \Rightarrow^* uwy$

posuneme-li A^1 do A^2 (i=2,…)
- $S \Rightarrow^* uA^1y \Rightarrow^* uvA^1xy \Rightarrow^* uvvA^2xxy \Rightarrow^* uvvwxxy$



Použití lemma o vkládání

Jak ukázat, že daný jazyk není bezkontextový?

Příklad 1: $L = \{a^nb^nc^n \mid n \geq 1\}$ není bezkontextový jazyk
- sporem
  - zvolme k=max(p,q), potom $|a^kb^kc^k|>p$
  - pumpovací slovo není delší než q
  - tj. vždy lze pumpovat maximálně dva různé symboly
  - poruší se rovnost počtu symbolů - SPOR

Příklad 2: $L = \{a^ib^jc^k \mid 0 \leq i \leq j \leq k\}$ není bezkontextový jazyk
- sporem
  - zvolme n=max(p,q), potom $|a^nb^nc^n|>p$
  - pumpovací slovo není delší než q
  - tj. vždy lze pumpovat maximálně dva různé symboly
  - pokud pumpujeme *a* (případně *b*), pumpujeme nahoru
  - pokud pumpujeme *c* (případně *b*), pumpujeme dolu
  - potom i>k ($a\uparrow$, $c\downarrow$) nebo j>k ($b\uparrow$, $c\downarrow$) nebo i>j ($a\uparrow$, $b\downarrow$) - SPOR

Nekonečnost bezkontextových jazyků

Pro každý BKJ L existují přirozená čísla m, n taková, že: L je nekonečný $\Leftrightarrow \exists z \in L\ m<|z| \leq n$.

Důkaz:
- z lemmatu o vkládání máme *p* a *q*, položme: m = p, n = p+q
- „$\Leftarrow$“
  - p<|z|, tedy *z* lze pumpovat $\Rightarrow$ jazyk je nekonečný
- „$\Rightarrow$ “
  - jazyk je nekonečný $\Rightarrow \exists z \in L$ p=m < |z|
  - vezmeme nejkratší takové *z* a potom $|z| \leq n=p+q$
  - *sporem:* nechť p+q < |z|, lze pumpovat dolu, tj. |z'| < |z| odstraňujeme část o max. velikosti q, tedy p < |z'| - SPOR

Rychlejší algoritmus:
- vezmeme <u>redukovanou</u> gramatiku G v ChNF tž. L=L(G)
- uděláme orientovaný graf
  - vrcholy = neterminály, hrany = {(A,B), (A,C) pro $(A \rightarrow BC) \in P(G)$}
- hledáme orientovaný cyklus (existuje $\Rightarrow$ jazyk je nekonečný)

Algoritmická rozhodnutelnost u BKJ

Pro bezkontextové jazyky je algoritmicky rozhodnutelné, zda dané slovo patří či nepatří do jazyka.

- umíme $\lambda \in L(G)$ ($S \Rightarrow^* \lambda$)
- pro ostatní slova použijeme ChNF ($X \to YZ$, $X \to a$)
  - v každé derivaci se délka slova zvětšuje nebo roste počet terminálních symbolů (tj. v derivaci není cyklus!)
  - na terminální slovo délky n použijeme právě (2n-1) pravidel
  - derivací pro všechna slova délky n je konečně
  - můžeme postupně vyzkoušet všechny derivace vedoucí ke slovům dané délky, například prohledáváním do hloubky

Pro bezkontextové jazyky je algoritmicky rozhodnutelné, zda je jazyk prázdný.

umíme zjistit, zda z S lze generovat terminální slovo (algoritmus redukce)

Kdy lemma o vkládání nezabere

Pozor! Lemma o vkládání je pouze implikace!

BKJ $\Rightarrow$ lze pumpovat (nutná podmínka bezkontextovosti)

nejedná se o podmínku postačující

Příklad:

$L = \{a^i b^j c^k d^l \mid i=0 \vee j=k=l\}$ není bezkontextový jazyk

přesto lze pumpovat

i=0: $b^j c^k d^l$ lze pumpovat v libovolném písmenu

i>0: $a^i b^n c^n d^n$ lze pumpovat v části obsahující *a*

Jak na to?

- zobecnění pumping lemmatu (Ogdenovo lemma)
  - pumpování vyznačených symbolů
- uzávěrové vlastnosti

Automaty a gramatiky, Roman Barták

Průnik bezkontextových jazyků

Bezkontextové jazyky nejsou uzavřené na průnik.

Důkaz:

stačí najít dva BKG, jejichž průnik není BKG

$L_1 = \{a^i b^i c^j \mid 0 \leq i,j\}$ $\{S \to AC,\ A \to aAb \mid \lambda,\ C \to cC \mid \lambda\}$

$L_2 = \{a^i b^j c^j \mid 0 \leq i,j\}$ $\{S \to AB,\ A \to aA \mid \lambda,\ B \to bBc \mid \lambda\}$

$L_1 \cap L_2 = \{a^i b^i c^i \mid 0 \leq i\}$ není BKJ (víme z pumping lemmatu)

Pozorování:

paralelní běh dvou zásobníkových automatů

řídící jednotky umíme spojit (viz konečné automaty)

čtení umíme spojit (jeden automat může čekat)

bohužel dva zásobníky nelze obecně spojit do jednoho

dva zásobníky = Turingův stroj

= rekurzivně spočetné jazyky

Automaty a gramatiky, Roman Barták

Průnik bezkontextového a regulárního jazyka

(Deterministické) bezkontextové jazyky jsou uzavřené na průnik s regulárním jazykem.

Důkaz:

zásobníkový a konečný automat můžeme spojit

konečný automat $A_1 = (Q_1, X, \delta_1, q_1, F_1)$

zásobníkový automat (přijímání stavem) $M_2 = (Q_2, X, Y, \delta_2, q_2, Z_0, F_2)$

nový automat $M = (Q_1 \times Q_2, X, Y, \delta, (q_1,q_2), Z_0, F_1 \times F_2)$

$((p',q'), u) \in \delta((p,q),a,Z)$ právě když

- $a \neq \lambda$: $p' \in \delta_1(p,a)$ & $(q',u) \in \delta_2(q,a,Z)$ … automaty čtou vstup
- $a = \lambda$: $(q',u) \in \delta_2(q,\lambda,Z)$ … ZA mění zásobník
  $p' = p$ … KA stojí

zřejmě $L(M) = L(A_1) \cap L(M_2)$

paralelní běh automatů

Automaty a gramatiky, Roman Barták

Použití uzavřenosti průniku BKJ a RJ

$L = \{a^i b^j c^k d^l \mid i=0 \vee j=k=l\}$ není bezkontextový jazyk

SPOREM:

nechť L je bezkontextový jazyk

$L_1 = \{a b^i c^j d^k \mid 0 \leq i,j,k\}$ je regulární jazyk

$S \to aB,\ B \to bB \mid C,\ C \to cC \mid D,\ D \to dD \mid \lambda$

$L \cap L_1 = \{a b^i c^i d^i \mid 0 \leq i\}$ není bezkontextový jazyk - SPOR

L je kontextový jazyk

$S \to B' \mid aA$

$B' \to bB' \mid C',\ C' \to cC' \mid D',\ D' \to dD' \mid \lambda$

$A \to aA \mid P$

$P \to bPCD \mid \lambda$

$DC \to CD$ $\{DC \to XC,\ XC \to XD,\ XD \to CD\}$

$bC \to bc,\ cC \to cc,\ cD \to cd,\ dD \to dd$

Automaty a gramatiky, Roman Barták

Sjednocení a doplněk BKJ

Bezkontextové jazyky jsou uzavřené na sjednocení.

použijeme gramatiky (pro ZA kostrbaté)

$L_1 = L(G_1)$ $G_1 = (V_{N1}, V_{T1}, S_1, P_1)$

$L_2 = L(G_2)$ $G_2 = (V_{N2}, V_{T2}, S_2, P_2)$

můžeme předpokládat, že $V_{N1} \cap V_{N2} = \emptyset$ (jinak přejmenuj)

uděláme gramatiku:

$G = (V_{N1} \cup V_{N2} \cup \{S\}, V_{T1} \cup V_{T2}, S, P_1 \cup P_2 \cup \{S \to S_1 \mid S_2\})$

zřejmě $L(G) = L(G_1) \cup L(G_2)$

„počítá“ jedna nebo druhá gramatika

Bezkontextové jazyky nejsou uzavřené na doplněk.

sporem podle de Morganových pravidel

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

bezkontextové jazyky nejsou uzavřené na průnik - SPOR

v ZA nestačí prohodit koncové a nekoncové stavy!

Automaty a gramatiky, Roman Barták

Substituce a homomorfismus

Substituce σ převádí slova na jazyky

$\sigma(\lambda) = \lambda$, $\sigma(x)$ = jazyk

$\sigma(uv) = \sigma(u).\sigma(v)$ $\quad \sigma: X^* \to P(Y^*)$

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

Třída T jazyků je **uzavřena na substituci**, když:

$\forall a\in X\ \sigma(a)\in T\ \&\ L\in T \Rightarrow \sigma(L)\in T$

Homomorfismus převádí slova na slova

$h(\lambda) = \lambda$, $h(x)$ = slovo

$h(uv) = h(u) . h(v)$ $\quad h: X^* \to Y^*$

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

Inverzní homomorfismus převádí slova zpět

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

Uzavřenost BKJ na substituci

Bezkontextové jazyky jsou uzavřeny na substituci.

intuitivně: listy v derivačním stromu generují další stromy

formálně:

máme bezkontextový jazyk L_0, tj. gramatiku $G_0 = (V_{N0},V_{T0},S_0,P_0)$,

pro každý terminál a_i z V_{T0}, $\sigma(a_i)$ je bezkontextový jazyk - gramatika G_i

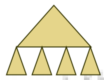
předpokládejme, že množiny neterminálů jsou navzájem disjunktní a že žádný terminál není v jiné gramatice neterminálem

definujme $G = (\cup_{i\geq 0}V_{Ni}, \cup_{i\geq 1}V_{Ti}, S_0, P)$, kde

$P = \cup_{i\geq 1}P_i \cup P'$ a P' = {P, kde každý terminál a_i je nahrazen S_i}

zřejmě: $L(G) = \sigma(L_0)$

Příklad:



$L_0 = \{a^ib^j \mid 0\leq i\leq j\}$ $\quad S_0 \to aS_0b \mid S_0b \mid \lambda$

$\sigma(a) = L_1 = \{c^id^i \mid 0\leq i\}$ $\quad S_1 \to cS_1d \mid \lambda$

$\sigma(b) = L_2 = \{c^i \mid 0\leq i\}$ $\quad S_2 \to cS_2 \mid \lambda$

$\sigma(L_0)$: $\quad S_0 \to S_1S_0S_2 \mid S_0S_2 \mid \lambda,\ \ S_1 \to cS_1d \mid \lambda,\ \ S_2 \to cS_2 \mid \lambda$

Bezkontextové jazyky jsou uzavřeny na homomorfismus.

přímý důsledek předchozí věty (terminál nahradíme slovem)

Inverzní homomorfismus

Bezkontextové jazyky jsou uzavřeny na inverzní homomorfismus.

$h^{-1}(L) = \{ w \mid h(w)\in L\}$ - máme zásobníkový automat M pro L a čteme w

idea:

- přečteme písmeno x a do vnitřního bufferu dáme h(x)
- simulujeme výpočet M, kdy vstup bereme z bufferu
- po vyprázdnění bufferu načteme další písmeno ze vstupu
- slovo je přijato, když je buffer prázdný a M je v koncovém stavu

formálně:

buffer je konečný, můžeme ho tedy modelovat ve stavu

pro L máme $M = (Q,X,Y,\delta,q_0,Z_0,F)$ (přijímání koncovým stavem)

$h: A^* \to X^*$

definujme $M' = (Q', A, Y, \delta',[q_0,\lambda], Z_0, F\times\{\lambda\})$, kde

$Q' = \{[q,u] \mid q\in Q, u\in X^*, \exists a\in A\ \exists v\in X^*\ h(a)=vu\}$, $\quad$ u je buffer

$\delta'([q,u], \lambda, Z) = \{ ([p,u],\gamma) \mid (p,\gamma) \in \delta(q, \lambda, Z) \}$

$\quad \cup \{ ([p,v],\gamma) \mid (p,\gamma) \in \delta(q, b, Z) \}$ $\quad$ … u=bv (čte buffer)

$\delta'([q, \lambda], a, Z) = \{ ([q,h(a)],Z) \}$ $\quad$ … naplňuje buffer

Kvocienty s regulárním jazykem

Bezkontextové jazyky jsou uzavřené na levý (pravý) kvocient s regulárním jazykem.

$R\backslash L = \{ w \mid \exists u\in R\ uw\in L\}$, $L/R = \{ u \mid \exists w\in R\ uw\in L\}$

idea:

- ZA beží na prázdno (nečte vstup) paralelně s KA
- je-li KA v koncovém stavu, můžeme začít číst vstup

formálně:

konečný automat $A_1 = (Q_1,X, \delta_1,q_1,F_1)$

zásobníkový automat $M_2 = (Q_2,X,Y,\delta_2,q_2,Z_0,F_2)$ (přijímání koncovým stavem)

definujme nový automat $M = (Q', X, Y, \delta, (q_1,q_2), Z_0, F_2)$, kde

$Q' = (Q_1\times Q_2) \cup Q_2$ $\quad$ (dvojice stavů pro paralelní běh ZA a KA)

$\delta'((p,q), \lambda, Z) = \{ ((p',q'),u) \mid \exists a\in X\ p'\in\delta_1(p, a)\ \&\ (q',u)\in\delta_2(q, a, Z)\}$

$\quad \cup \{ ((p,q'),u) \mid (q',u) \in \delta_2(q, \lambda, Z) \}$

$\quad \cup \{ (q',u) \mid \exists a\in X\ (q',u)\in\delta_2(q, a, Z)\ \&\ p'\in\delta_1(p, a)\ \&\ p'\in F_1\}$

$\delta'(q, a, Z) = \delta_2(q, a, Z)$ $\quad a\in X \cup\{\lambda\},\ q\in Q_2$

zřejmě $L(M) = L(A_1)\backslash L(M_2)$