Vztah jazyků Chomskeho hierarchie a jazyků TS

Jan Konečný; (přednáší Lukáš Havrlant)

15. října 2013

Rychlé (průběžné) opáčko KMI/FJAA

Definice

Formální gramatika (typu 0, bez omezení) je struktura $\langle \Sigma, N, S, P \rangle$, kde

- N – abeceda neterminálních symbolů,
- Σ – abeceda terminálních symbolů,
- $S \in N$ – startovní symbol,
- P – konečná množina odvozovacích pravidel ve tvaru

$$(N \cup \Sigma)^* N (N \cup \Sigma)^* \to (N \cup \Sigma)^*.$$

(tj. generativních pravidel)

Definice

Uvažujme odvozovací pravidlo $\pi = x \to y$ nad $\Sigma \cup N$. Pak řekneme, že řetězec $v \in (\Sigma \cup N)^*$ je přímo odvozen z řetězce $u \in (\Sigma \cup N)^*$ pomocí pravidla π, pokud existují řetězce $p, q \in (\Sigma \cup N)^*$ tak, že $u = pxq$ a $v = pyq$; označujeme $u \Rightarrow_\pi v$.

Píšeme $u \Rightarrow_P v$ pokud existuje pravidlo $\pi \in P$ tak, že $u \Rightarrow_\pi v$.

Definice

Posloupnost řetězců $x_0, \ldots, x_k$ $(k \geq 0)$, kde $\{x_0, \ldots, x_k\} \subseteq (\Sigma \cup N)^*$ se nazývá P-*derivace* (délky k), pokud $x_{i-1} \Rightarrow_P x_i$.

Definice

Pokud pro $u, v \in (N \cup \Sigma)^*$ existuje P-derivace $u = x_0, \ldots, x_k = v$, pak říkáme, že v je odvozen z u pomocí pravidel z Pa píšeme $u \Rightarrow_P^* v$.

Definice

Větná forma je jakýkoli řetězec $x \in (N \cup \Sigma)^*$, pro který $S \Rightarrow_P^* x$.

Větné formy, které se skládají jen z terminálů, se nazývají *věty* gramatiky.

Jazyk $L(G)$ gramatiky G je množina všech vět gramatiky.

Věta

Třída jazyků generovaných gramatikami typu 0 = částečně rekurzivní jazyky.

Důkaz.

Ukážeme tak, že

a) ke každému TS T sestavíme gramatiku G_T, tak že $L(T) = L(G)$.
b) ke každé gramatice G sestavíme TS T_G, tak že $L(T) = L(G)$.

ad b) snažší část. Sestrojíme nedeterministický TS T_G k lib. gramatice G:

(dvojpáskový) NTS T_G pro w:

1. (na první pásce nechá w), na druhou pásku napíše S.
2. pokud na druhé pásce není žádný neterminál, srovná obsah první a druhé pásky: pokud jsou stejné, přijme.
3. nedeterministicky zvolí pravidlo z P nedeterministicky zvolí výskyt jeho pravé strany na druhé pásce (pokud ho nenajde zamítne).
4. přepíše obsah druhé pásky podle zvoleného pravidla, pokrač. bodem 2.

Důkaz, pokračování.

ad a) sestrojíme gramatiku G_T k lib. TS T:

Víme že k libovolnému TS existuje ekvivalentní TS, který po sobě uklízí. Můžeme tedy předpokládat, že T po sobě uklízí a že

$$T = \langle Q, \Sigma, \Gamma, \delta, q_0, q_+, q_- \rangle.$$

Gramatika bude

$$\langle Q \cup \{\triangleleft\} \cup \{N_a \mid a \in \Gamma\}, \Sigma, S, P \rangle$$

s pravidly P:

- pro $\delta(q, a) = (q', a', L)$ budou v P přechody

  $$q' N_b N_{a'} \Rightarrow N_b q N_a \quad \text{pro všechna } b \in \Gamma,$$

- pro $\delta(q, a) = (q', a', R)$ bude v P přechod

  $$N_{a'} q' \Rightarrow q N_a.$$

Důkaz, pokračování.

- pravidlo

$$S \Rightarrow q_+ \triangleleft$$

Neterminál $\triangleleft$ reprezentuje konec uvažovaného úseku pásky, bude zajišťovat přidávání a odebírání prázdných políček (symbolů); pomocí následujících pravidel:

- pravidla na přidávání a odebírání volných políček

$$\triangleleft \Rightarrow N_{\textvisiblespace} \triangleleft$$

$$N_{\textvisiblespace} \triangleleft \Rightarrow \triangleleft$$

- pravidla pro finalizaci

$$N_a \Rightarrow a \quad \text{pro všechna } a \in \Sigma \qquad \triangleleft \Rightarrow \varepsilon \qquad q_0 \Rightarrow \varepsilon.$$

Tato gramatika G_T bude generovat stejný jazyk jako T; sleduje výpočet pozpátku □

Rychlé (průběžné) opáčko KMI/FJAA

Definice

Kontextově závislá gramatika je formální gramatika, pro jejíž množinu pravidel P platí:

- P – konečná množina odvozovacích pravidel ve tvaru

$$\alpha N \beta \to \alpha \gamma \beta,$$

kde $|\gamma| \geq 1$ s vyjímkou, že je možno mít v P i pravidlo

$$S \Rightarrow \varepsilon,$$

pokud se S nevyskytuje na pravé straně žádného pravidla.

Definice

Nezkracující gramatika je formální gramatika, pro jejíž množinu pravidel P platí:

- P – konečná množina odvozovacích pravidel ve tvaru

$$\alpha \to \beta$$

kde $|\alpha| \leq |\beta|$ s vyjímkou, že je možno mít v P i pravidlo

$$S \Rightarrow \varepsilon,$$

pokud se S nevyskytuje na pravé straně žádného pravidla.

Věta

Každá kontextová gramatika je nezkracující.

Věta

Ke každé nezkracující gramatice existuje ekvivalentní kontextově závislá gramatika.

Věta

Jazyky generované kontextové závislými gramatikami = jazyky přijímané LBA.

Idea důkazu.

a) ke každému LBA T sestavíme kontextovou gramatiku G_T, tak že $L(T) = L(G)$.

b) ke každé kontextové gramatice G sestavíme LBA T_G, tak že $L(T) = L(G)$.

V podstatě totožné s předchozím. Využijeme toho, že jsou nezkracující. □

Důkaz.

ad a) sestrojíme gramatiku G_T k lib. TS T:

Víme že k libovolnému TS existuje ekvivalentní TS, který po sobě uklízí. Můžeme tedy předpokládat, že T po sobě uklízí a že

$$T = \langle Q, \Sigma, \Gamma, \delta, q_0, q_+, q_- \rangle.$$

Gramatika bude

$$\langle Q \cup \{\triangleleft\} \cup \{N_a \mid a \in \Gamma\}, \Sigma, S, P \rangle$$

s pravidly P:

- pro $\delta(q, a) = (q', a', L)$ budou v P přechody

$$q' N_b N_{a'} \Rightarrow N_b q N_a \quad \text{pro všechna } b \in \Gamma,$$

- pro $\delta(q, a) = (q', a', R)$ bude v P přechod

$$N_{a'} q' \Rightarrow q N_a.$$

Důkaz, pokračování.

- pravidlo

$$S \Rightarrow q_+ \lhd$$

Neterminál $\lhd$ reprezentuje konec uvažovaného úseku pásky, bude zajišťovat přidávání a odebírání prázdných políček (symbolů); pomocí následujících pravidel:

- pravidla na přidávání a odebírání volných políček

$$\lhd \Rightarrow N_{\textvisiblespace} \lhd$$

- pravidla pro finalizaci

$$N_a \Rightarrow a \quad \text{pro všechna } a \in \Sigma \qquad \lhd \Rightarrow \varepsilon \qquad q_0 \Rightarrow \rhd.$$

Tato gramatika G_T bude generovat stejný jazyk jako T. $\square$

Všimněme si, že pokud je gramatika G nezkracující (resp. kontextově závislá), tak T_G z důkazy první věty nepotřebuje víc políček než $|w|$, klidně by to mohl být LBA.

Důkaz.

(dvojpáskový) NTS T_G pro w:

1. (na první pásce nechá w), na druhou pásku napíše S.
2. pokud na druhé pásce není žádný neterminál, srovná obsah první a druhé pásky: pokud jsou stejné, přijme.
3. nedeterministicky zvolí pravidlo z P nedeterministicky zvolí výskyt jeho pravé strany na druhé pásce (pokud ho nenajde zamítne).
4. přepíše obsah druhé pásky podle zvoleného pravidla, pokrač. bodem 2.

□

Věta

Existuje rekurzivní jazyk, který není generovaný kontextovou gramatikou.

Poznámka

Neboli Typ 1 $\subset$ R.

Idea důkazu.

Ukážeme, že

$$L_{Gd} = \{[G] \mid \ G \text{ je kontextová gramatika, která negeneruje } [G]\}$$

je takový jazyk. Viz následující dvě lemmata. $\square$

Lemma

L_{Gd} *je rekurzivní.*

Idea důkazu.

Procházíme do šířky strom derivací gramatiky G a hledáme, jestli negeneruje $[G]$. Stačí jít jen do určité hloubky, protože díky nezkracujícímu charakteru gramatiky nemá smysl procházet větve, kde je větná forma delší, než $[G]$. Navíc větných forem s délkou $\leq |[G]|$ je jen konečně mnoho. $\square$

Lemma

L_{Gd} není generován kontextově závislou gramatikou.

Důkaz je nápadně podobný důkazu toho, že L_d není přijímán TS.

Důkaz.

Protože gramatiky lze zakódovat do řetězců a řetězce lze očíslovat,, můžu seřadit gramatiky podle jejich kódů.
Můžu tedy uvažovat tabulku, jejíž řádky budou gramatiky, sloupce budou kódy gramatik, tak aby když je v i-tém řádku G_i, tak v i-tém sloupci $[G_i]$.

	$[G_1]$	$[G_2]$	$[G_3]$	$[G_4]$	$\dots$
G_1	1				
G_2			1		
G_3		1			
G_4				1	
$\vdots$					

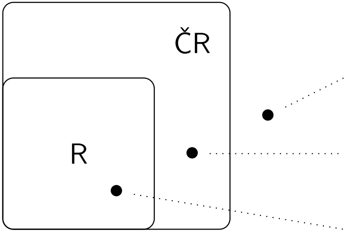
Tabuka obsahuje 1 na pozici $\langle i, j\rangle$ pokud G_i generuje $[G_j]$. Pokud existuje TS G_d generující L_{Gd}, musí být někde v té tabulce, řekněme na řádku x.
Co ale bude na pozici $\langle x, x\rangle$?
Ta hodnota se musí sama od sebe lišit. **SPOR** □

Shrnutí

$$\text{Typ 3} \subset \text{Typ 2} \subset \text{Typ 1} \subset \text{R} \subset \text{Typ 0} = \text{ČR}$$

Problémy z PŘEDNÁŠKY 1

Z PŘEDNÁŠKY 1 známe pár příkladů (zatím tomu jenom věříme).



Ekvivalence bezkontextových gramatik

Nonekvivalence bezkontextových gramatik

Ekvivalence KA

Ekvivalence KA

Věta

Jazyk

$$L_{eqKA} = \{[A_1, A_2] \mid\ A_1, A_2 \text{ jsou KA}, L(A_1) = L(A_2)\}$$

je rekurzivní.

Pomocné tvrzení:

Lemma

Jazyk $L_{KA\emptyset} = \{[A] \mid\ A \text{ je KA}, L(A) = \emptyset\}$ *je rekurzivní.*

Idea důkazu.

Stačí zjistit jestli je aspoň jeden koncový stav dostupný z počátečního. □

Důkaz.

Z KMI/FJAA víme, že regulární jazyky jsou uzavřené na průnik, sjednocení, doplněk (mj.). A umíme sestavit příslušné KA (nedeterministické, s ε-přechody.

Máme $L_1 = L_2$ p.k. $(L_1 \cap \neg L_2) \cup (\neg L_1 \cap L_2) = \emptyset$.

Sestavíme TS M, který rozhoduje L_{eqKA}.

TS M pro $[A_1, A_2]$

1. Sestaví KA A, který rozhoduje $(L_1 \cap \neg L_2) \cup (\neg L_1 \cap L_2)$.
2. Zjistí, jestli $[A] \in L_{KA\emptyset}$.

□

Ekvivalenci a nonekvivalenci gramatik ještě odložíme, potřebujeme Postův problém přiřazení (PŘEDNÁŠKA 6)

Další problémy/jazyky

Věta

Jazyk $REG_{TS} = \{\langle T\rangle \mid$ *T je TS a $L(T)$ je regulární*$\}$*. není rekurzivní.*

Důkaz.

(sporem), Nechť R je TS, který rozhoduje REG_{TS} a zkonstruujeme TS U, který rozhoduje L_U.

TS U pro vstup $[T, w]$, kde T je TS a w je vstup

1. Sestrojí následující stroj M

   TS M pro x:

   - *pokud x má tvar 0^n1^n, přijmi*
   - *pokud x nemá tento tvar, spust M pro w, přijmi, pokud M přijme.*

2. Spustí R pro $[M_2]$.
3. pokud R přijme, U přijme; pokud R zamítne, U zamítne.

□

Věta

Jazyk

$$L_{LBA\emptyset} = \{[B] \mid\ B \text{ je LBA a } L(B) = \emptyset\ \}$$

není rekurzivní.

Idea důkazu.

(sporem) Ukážeme, že pokud by $L_{LBA\emptyset}$ byl rekurzivní tak L_U bude taky rekurzivní. Pro TS T a slovo w sestavíme LBA B, jehož jazykem bude jazyk všech přijímajících historií výpočtu TS T nad slovem w. □

DODELAT JAKO CVIČENÍ.