Automaty a gramatiky

Tomáš Matoušek

Formální jazyky

Definice.

• *Abeceda je libovolná konečná množina. Abecedu budeme značit Σ, její prvky nazveme symboly.*

• *Slovem nad abecedou Σ nazveme konečnou posloupnost jejích symbolů, píšeme $u = a_1 a_2 \dots a_n$. Posloupnost, která neobsahuje žádný symbol nazveme prázdným slovem a označíme ji λ.*

• *Množinu všech slov nad abecedou Σ označíme Σ^*. Množinu všech slov bez prázdného slova označíme Σ^+.*

• *Jazyk L nad abecedou Σ je libovolná podmnožina Σ^*.*

Příklad.

abeceda	$\Sigma = \{0, 1\}$.
symboly	$0, 1$.
množina slov	$\Sigma^* = \{\lambda, 0, 1, 00, 01, \dots, 1000101010011, \dots\}$.
konečný jazyk	$L_1 = \{\lambda, 1, 01, 001\}$.
nekonečný jazyk	$L_2 = \{0, 00, 000, \dots\}$.

Definice. *Nechť $u = a_1 \dots a_n, v = b_1 \dots b_m$ jsou slova.*

• *Délka slova u je $|u| = n$.*

• *Zřetězením slov u, v nazveme posloupnost $uv = a_1 \dots a_n b_1 \dots b_m$.*

• *Otočením slova u nazveme slovo $u^R = a_n a_{n-1} \dots a_1$.*

• *Podslovem slova u nazveme každé slovo $a_i a_{i+1} \dots a_j$, kde $1 \le i \le j \le n$.*

Dále definujeme mocninu slova $u^0 = \lambda, u^k = \underbrace{uu \dots u}_{k}$, $k \in \mathbb{N}$.

Gramatiky – generování jazyků

Definice. *Gramatikou rozumíme čtveřici* $(\Sigma_T, \Sigma_N, S, P)$, *kde*

- Σ_T *je abeceda terminálních symbolů.*
- Σ_N *je abeceda neterminálních symbolů,* $\Sigma_N \cap \Sigma_T = \emptyset$.
- S *je počáteční symbol,* $S \in \Sigma_N$.
- P *je konečná množina pravidel, což jsou dvojice slov z abecedy* $\Sigma = \Sigma_N \cup \Sigma_T$, *přičemž první slovo obsahuje aspoň jeden neterminál. Pravidla píšeme ve tvaru* $x \to y$.

Gramatika nám umožňuje vytvářet slova splňující předem definovaná pravidla. Začneme vždy jedním symbolem S a použitím pravidel z P podle následující definice vytváříme další slova. Skončíme, když už není žádné pravidlo, které bychom mohli použít.

Definice. *Nechť* G *je gramatika,* u, v *jsou slova abecedy* Σ. *Řekneme, že slovo* v *je přímo odvozeno z* u *použitím pravidla* $x \to y \in P$ *(značíme* $u \Rightarrow v$*), pokud existují slova* $u_1, u_2, v_1, v_2 \in \Sigma^*$ *taková, že* $u = u_1 x u_2$ *a* $v = v_1 y v_2$.

Řekneme, že slovo v *je odvozeno z* u *(značíme* $u \Rightarrow^* v$*), pokud existuje posloupnost slov* $u = w_1, w_2, \ldots, w_n = v$ *taková, že* $w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n$.

Příklad. Gramatika $(\{a, b\},\ \{S, X, Y\},\ S,\ \{S \to aXb,\ X \to aXY,\ aX \to b,\ YY \to \lambda\})$.

Odvozování: $\underline{S} \Rightarrow a\underline{X}b \Rightarrow aa\underline{X}YB \Rightarrow aa\underline{aX}YYb \Rightarrow aab\underline{YY}b \Rightarrow aabb$

Odvození slova v ze slova u tedy znamená vybrání libovolného pravidla, jehož levá strana je podslovem u, a nahrazení odpovídajícího podslova slova u pravou stranou pravidla.

Definice. *Množina terminálních slov odvoditelných v dané gramatice* G *z počátečního symbolu* S *je jazyk generovaný gramatikou* G. *Budeme ho značit* $L(G)$. *Formálně je* $L(G) = \{w \in \Sigma_T^* | S \Rightarrow^* w\}$.

Pravidla gramatik mohou být různě omezena a tímto omezením ovlivníme, jaké jazyky budou gramatiky generovat. Základním rozdělením gramatik podle tvaru přepisovacích pravidel je tzv. Chomského hierarchie.

Definice. *Chomského hierarchií nazýváme rozdělení gramatik na následující typy:*
(0) *obecné gramatiky – bez omezení pravidel.*
(1) *kontextové gramatiky – pravidla tvaru* $uXv \to uxv, x \in \Sigma^+$ *nebo* $S \to \lambda$, *pokud* S *není na pravé straně žádného pravidla.*
(2) *bezkontextové gramatiky – pravidla tvaru* $X \to u$

(3) *regulární gramatiky – pravidla tvaru* $X \to tY, X \to t, t \in \Sigma_T^*$.
Přičemž $X, Y \in \Sigma_N, u, v \in \Sigma^*$.
Třídu jazyků generovaných gramatikou typu i *označíme* $\mathcal{L}_i$.

Věta. *Chomského hierarchie definuje uspořádání:* $\mathcal{L}_0 \supset \mathcal{L}_1 \supset \mathcal{L}_2 \supset \mathcal{L}_3$.

Automaty – rozpoznávání jazyků

Zatím jsme se zabývali gramatikami s jejichž pomocí jsme generovali různé jazyky. Nyní si ukážeme, jak zjistit, zda dané slovo patří do jazyka.

Definice. *Deterministický konečný automat (DKA) je pětice* $(Q, \Sigma, \delta, q_0, Q_F)$, *kde*

- Q *je konečná množina stavů automatu.*
- Σ *je abeceda automatu.*
- $\delta : Q \times \Sigma \to Q$ *je přechodová funkce.*
- $q_0 \in Q$ *je počáteční stav.*
- $Q_F \subseteq Q$ *je množina koncových stavů.*

Dále budeme pod pojmem konečný automat myslet DKA, pokud nebude řečeno jinak.

Definice. *Definujeme rozšířenou přechodovou funkci* $\delta^* : Q \times \Sigma^* \to Q$ *tak, že*
(i) $\delta^*(q, \lambda) = q, q \in Q$.
(ii) $\delta^*(q, aw) = \delta^*(\delta(q, a), w), a \in \Sigma, w \in \Sigma^*, q \in Q$.
Jelikož $\Sigma \subset \Sigma^*$, *můžeme rozšířenou přechodovou funkci značit také* δ.

Definice. *Konečný automat přijímá slovo* $w \in \Sigma$, *pokud* $\delta(q_0, w) \in Q_F$.
Jazyk přijímaný (rozpoznávaný) konečným automatem A *je množina slov, jež automat přijímá, značíme ho* $L(A)$.

„Vstupem“ konečného automatu je slovo z abecedy Σ a „výstupem“ je přijetí nebo nepřijetí tohoto slova. Konečný automat lze chápat jako stroj, který postupně čte symboly ze vstupu, podle přechodové funkce mění svůj vnitřní stav a po načtení celého slova řekne, zda toto slovo přijímá.

Jak poznat, zda je daný jazyk rozpoznatelný nějakým konečným automatem, lze vhodným užitím Dirichletova principu nebo pomocí následujícího lemmatu.

Lemma. (Pumpovací) *Nechť* L *je jazyk rozpoznatelný konečným automatem. Pak existuje* $n \in \mathbb{N}$ *takové, že libovolné slovo* $z \in L, |z| \geq n$ *lze rozdělit na tři části* $z = uvw$, *přičemž* $|uv| \leq n$, *a* $av^iw \in L$ *pro každé* $i \in \mathbb{N}_0$.

Pokud chceme sestavit automat rozpoznávající daný jazyk, můžeme někdy s výhodou použít nedeterministické konečné automaty.

Definice. *Nedeterministický konečný automat (NKA) je $(Q, \Sigma, \delta, Q_0, Q_F)$, kde*
- *Q je konečná množina stavů automatu.*
- *Σ je abeceda automatu.*
- *$\delta : Q \times \Sigma \to 2^Q$ je přechodová funkce.*
- *$Q_0 \subseteq Q$ je množina počátečních stavů.*
- *$Q_F \subseteq Q$ je množina koncových stavů.*

Poznámka. NKA se od DKA liší v definici přechodové funkce a v počtu možných počátečních stavů.

Definice. *Analogicky DKA můžeme definovat*
- *rozšířenou přechodovou funkci $\delta^* : Q \times \Sigma^* \to 2^Q$ (opět můžeme značit δ),*
- *přijímání slova $w \in \Sigma$, je-li $\delta(q_0, w) \cap Q_F \neq \emptyset$.*

Věta. *Ke každému NKA lze sestrojit DKA přijímající stejný jazyk.*

Definice. *Buď Σ abeceda, L, L_1, L_2 jazyky nad Σ.*
- *Zřetězení jazyků L_1, L_2 je jazyk $L_1.L_2 = \{uv | u \in L_1, v \in L_2\}$.*
- *Mocnina jazyka L je jazyk $L^0 = \{\lambda\}$ resp. $L^{n+1} = L.L^n, n \in \mathbb{N}$.*
- *Otočením (reverzí) jazyka rozumíme jazyk $L^R = \{u^R | u \in L\}$.*

Poznámka. Jelikož jsou jazyky množiny slov, lze na ně provádět také množinové operace: průnik, sjednocení, rozdíl, doplněk.

Věta. *Třída regulárních jazyků $\mathcal{L}_3$ je uzavřena na zřetězení, mocninu, otočení, průnik, sjednoceni, rozdíl i doplněk.*

Věta. *Ke každé regulární gramatice lze sestrojit konečný automat přijímající jazyk touto gramatikou generovaný a naopak.*

Poznámka. Rozšířením konečných automatů jsou postupně zásobníkové automaty, lineárně omezené automaty a Turingovy stroje. Ty pak přijímají jazyky generované gramatikami kontextovými, bezkontextovými a gramatikami typu 0.