2.7 Další uzávěrové vlastnosti třídy regulárních jazyků

Z předchozích přednášek víme, že třída regulárních jazyků je uzavřena na sjednocení, průnik, doplněk, zřetězení, Kleeneho operaci $\star$ a reverzi. Ukážeme ještě další operace s jazyky, na které je třída regulárních jazyků uzavřena.

2.7.1 Homomorfismus. Jsou dány dvě abecedy Σ, Γ a zobrazení h, které každému písmenu $a \in \Sigma$ přiřadí slovo $h(a)$ nad abecedou Γ.

Zobrazení h rozšíříme na zobrazení, které každému slovu $u \in \Sigma^\star$ přiřazuje slovo nad Γ takto:

1. $h(\varepsilon) = \varepsilon$,
2. $h(u\,a) = h(u)\,h(a)$.

Poznamenejme, že se jedná o rozšíření zobrazení $h\colon \Sigma \to \Gamma^\star$ na $h\colon \Sigma^\star \to \Gamma^\star$ tak, že obraz slova je zřetězením obrazů jednotlivých písmen slova.

Definice. *Obraz jazyka* L (nad Σ) *v homomorfismu* h je

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

□

Příklad: Uvažujme abecedy $\Sigma = \{0,1\}$, $\Gamma = \{a,b\}$ a zobrazení $h(0) = ab^2$, $h(1) = bab$. Pak $h(010) = ab^2babab^2 = ab^2(ba)^2b^2$. Homomorfní obraz jazyka $L = \{10^k \mid k \geq 0\}$ je $h(L) = \{bab(ab^2)^k \mid k \geq 0\}$.

2.7.2 Substituce. Obecnější pojem než homomorfismus je tzv. substituce. Jsou dány dvě abecedy Σ, Γ a zobrazení σ, které každému písmenu $a \in \Sigma$ přiřadí jazyk nad abecedou Γ.

Analogicky jako pro homomorfismus zobrazení σ rozšíříme na zobrazení, které každému slovu $u \in \Sigma^\star$ přiřazuje jazyk nad Γ takto:

1. $\sigma(\varepsilon) = \{\varepsilon\}$,
2. $\sigma(u\,a) = \sigma(u)\,\sigma(a)$.

I zde se jedná o rozšíření zobrazení σ takové, že obrazem slova (zřetězení písmen) je zřetězení obrazů písmen.

Definice. *Obraz jazyka* L (nad Σ) *v substituci* σ je

$$\sigma(L) = \bigcup \{\sigma(w) \mid w \in L\}.$$

□

Příklad: Uvažujme abecedy $\Sigma = \{0,1\}$, $\Gamma = \{a,b\}$ a zobrazení $\sigma(0) = L_1 = \{a^n \mid n \geq 0\}$, $\sigma(1) = L_2 = \{b^n \mid n \geq 0\}$. Pak $\sigma(01) = L_1\,L_2 = \{a^nb^m \mid n,m \geq 0\}$.

2.7.3 Inverzní homomorfismus. Je dán homomorfismus h, $h\colon \Sigma^\star \to \Gamma^\star$. Pak *inverzní homomorfismus* h^{-1} je $h^{-1}\colon \Gamma^\star \to \Sigma^\star$, kde $h^{-1}(L) = \{u \in \Sigma^\star \mid h(u) \in L\}$. □

Příklad. Uvažujme jazyk L nad abecedou $\Gamma = \{0,1\}$ popsaný regulárním výrazem $(\mathbf{00} + \mathbf{1})^\star$ a homomorfismus h určený $h(a) = 01$ a $h(b) = 10$.

Pak $h^{-1}(L)$ je jazyk nad abecedou $\Sigma = \{a,b\}$ popsaný regulárním výrazem $(\mathbf{ba})^\star$. (Ověřte si.)

2.7.4 Věta. Je dána substituce σ z $\Sigma^\star$ do jazyků nad abecedou Γ. Jestliže každý jazyk $\sigma(a)$ je regulární a jestliže L je regulární jazyk nad abecedou Σ, pak jazyk $\sigma(L)$ je regulární jazyk nad abecedou Γ. □

Myšlenka důkazu. Předpokládejme, že jazyk L je popsán regulárním výrazem $\mathbf{r}$ a každý z jazyků $\sigma(a)$ regulárním výrazem $\mathbf{r_a}$ pro každé $a \in \Sigma$. Pak regulární výraz pro jazyk $\sigma(L)$ dostaneme tak, že za každé $\mathbf{a}$, $a \in \Sigma$, v regulárním výrazu $\mathbf{r}$ „dosadíme“ výraz $\mathbf{r}_a$.

Důsledek. Je-li h homomorfismus a L regulární jazyk, pak jazyk $h(L)$ je také regulární jazyk. □

Důkaz. Tvrzení vyplývá z předchozí věty, protože jednoprvkový jazyk je vždy regulární a navíc homomorfismus je zvláštní případ substituce.

2.7.5 Věta. Jestliže h je homomorfismus, $h\colon \Sigma^\star \to \Gamma^\star$ a L je regulární jazyk nad abecedou Γ, pak inversní obraz $h^{-1}(L)$ je také regulární (nad Σ).

Myšlenka důkazu. Máme konečný automat $M_1 = (Q_1, \Gamma, \delta_1, q_1, F_1)$, který přijímá jazyk L. Zkonstruujeme konečný automat M, který přijme $h^{-1}(L)$. M má stejnou množinu stavů, stejný počáteční stav i stejnou množinu koncových stavů. Přechodová funkce δ automatu M zobrazí stav p při vstupním symbolu a do takového stavu, kam v M_1 přejde stav p při přečtení slova $h(a)$.

Přesněji pro $a \in \Sigma$, $p \in Q$ je

$$\delta(p, a) = \delta_1^\star(p, h(a)).$$

Není těžké ukázat, že M přijme přesně ta slova $w \in \Sigma^\star$, pro která M_1 přijme $h(w)$.

2.8 Algoritmická řešitelnost úloh pro regulární jazyky

Pro následující otázky týkající se konečných automatů a jimi přijímaných jazyků existují algoritmy, které dají správnou odpověď.

1. Pro daný konečný automat M (ať deterministický nebo nedeterministický) a slovo $w \in \Sigma^\star$ rozhodnout, zda $w \in L(M)$.

   Zdůvodnění. Jedná se o jednoduché zjištění, zda ve stavovém diagramu existuje sled označený slovem w, který začíná v počátečním stavu a končí v koncovém stavu.

2. Pro daný konečný automat M (ať deterministický nebo nedeterministický) rozhodnout, zda $L(M) \neq \emptyset$.

   Zdůvodnění. Jedná se o jednoduché zjištění, zda ve stavovém diagramu je některý koncový stav dosažitelný z počátečního stavu.

3. Pro daný konečný automat M rozhodnout, zda $L(M) = \Sigma^\star$.

   Zdůvodnění. Platí $L(M) = \Sigma^\star$ právě tehdy, když redukovaný automat k automatu M se skládá z jediného stavu, který je současně počáteční i koncový.

4. Pro dva konečné automaty M_1 a M_2 rozhodnout, zda $L(M_1) = L(M_2)$.

   Zdůvodnění. Stačí zredukovat oba automaty a (v tomto případě jednoduchým algoritmem) zjistit, zda jsou isomorfní.

Pro „obecnější“ třídy jazyků už některé z výše položených otázek algoritmicky řešitelné nejsou, jak uvidíme později.

2.8.1 Tvrzení. Je dán deterministický konečný automat M s n stavy. Pak

1. Jazyk $L(M)$ je neprázdný právě tehdy, když M přijímá slovo w délky $|w| < n$.
2. Jazyk $L(M)$ je nekonečný právě tehdy, když M přijímá slovo v délky $n \leq |v| < 2n$.

□

Důkaz. 1. Jestliže M přijímá slovo w délky $|w| < n$, pak je $L(M)$ jistě neprázdný.

Předpokládejme, že $L(M)$ je neprázdný. Pak ve stavovém diagramu existuje orientovaný sled označený w, který začíná v počátečním stavu a končí v koncovém stavu. Každý orientovaný sled obsahuje orientovanou cestu se stejným počátečním i koncovým stavem. Označme u slovo, které označuje tuto orientovanou cestu. Pak $u \in L(M)$ a délka u je nejvýše $n-1$ (ano, každá cesta v grafu s n vrcholy má nejvýše $n-1$ hran).

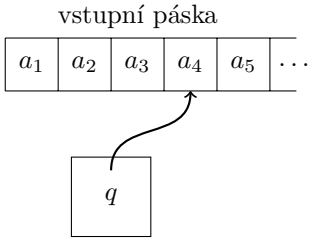
2. Předpokládejme, že jazyk $L(M)$ je nekonečný. Pak obsahuje slovo w délky větší než $n-1$. Jestliže $|w| < 2n$, máme hledané slovo. V opačném případě uvažujme orientovaný sled z počátečního stavu do koncového stavu označený w. Tento sled musí obsahovat cyklus. Označme C_1 první cyklus na tomto sledu. Odstraňme tento cyklus a dostaneme orientovaný sled kratší délky, který má stejný počáteční i koncový vrchol. Jestliže tento sled má délku menší než $2n$, slovo, které tento sled označuje, je hledané slovo v. Jestliže je stále příliš dlouhé, pokračujeme s odstraněním (třeba) prvního cyklu. Jakmile má slovo délku menší než $2n$, postup ukončíme. Protože každý cyklus má

délku maximálně n, tímto postupem dostaneme slovo délky maximálně $n-1+n$, tj. menší než $2n$ a větší než n.

Jestliže existuje $v \in L(M)$ s $n \leq |v| < 2n$, pak orientovaný sled určený tímto slovem obsahuje cyklus. Položme x slovo, které označuje tento cyklus. Pak $v = wxy$ a pro každé $i = 2, 3, \ldots$ platí $wx^iy \in L(M)$, proto je $L(M)$ nekonečný.

2.9 Dvousměrné automaty

Pouze informativně uvedeme ještě jeden typ konečných automatů, který, podobně jako nedeterministické automaty, přijímá pouze regulární jazyky. Pro lepší názornost připomeneme ještě jeden způsob, jak se dívat na konečné automaty. Tento způsob byl již zmíněn v první přednášce a je znázorněn na následujícím obrázku. (Poznamenejme, že ho využijeme později při zavedení zásobníkových automatů.)



Pro deterministické automaty víme, že jejich chování je určeno přechodovou funkcí δ, která každému stavu q a každému vstupnímu symbolu a přiřazuje následující stav $\delta(q, a)$. Práce DFA M nad slovem $w = a_1a_2 \ldots a_k$ spočívá v tom, že M ve stavu q_0 přečte první symbol a_1, přejde do stavu $q_1 = \delta(q_0, a_1)$, ve stavu q_1 přečte symbol a_2, přejde do stavu $q_2 = \delta(q_1, a_2) = \delta^\star(q_0, a_1a_2)$, atd. až do té doby, než "přečte" celé slovo $w = a_1a_2 \ldots a_k$.

Můžeme si proto představit DFA jako zařízení, které se skládá z řídící jednotky nacházející se v jednom ze stavů, pásky, na které je zapsáno vstupní slovo $w = a_1a_2 \ldots a_k$ a hlavy, která v každém okamžiku čte obsah jednoho políčka pásky. Je-li automat ve stavu q a čte vstupní symbol a_i, tak se řídící jednotka přesune do stavu $p = \delta(q, a_i)$ a hlava se posune o jedno políčko doprava a tudíž čte vstupní symbol a_{i+1}. Zda je slovo w přijato se rozhodne podle toho, zda řídící jednotka po přečtení celého slova skončí v některém koncovém stavu nebo ne. Obdobně si můžeme představit takto i práci NFA.

Dvousměrné automaty se od DFA liší tím, že hlava se po pásce nemusí pohybovat jen doprava, ale může též doleva (přitom ale nemění obsah pásky).

2.9.1 Dvousměrné automaty. Dvousměrný deterministický automat je pětice $(Q, \Sigma, \delta, q_0, F)$, kde Q, Σ, q_0 a F mají stejný význam jako pro deterministické konečné automaty a přechodová funkce δ je zobrazení

$$\delta\colon Q \times \Sigma \longrightarrow Q \times \{R, L\},$$

kde

- $\delta(q, a) = (p, R)$ znamená, že po přečtení vstupního písmene a ve stavu q se automat přesune do stavu p a hlava se posune na vstupní pásce o jedno políčko doprava;
- $\delta(q, a) = (p, L)$ znamená, že po přečtení vstupního písmene a ve stavu q se automat přesune do stavu p a hlava se posune na vstupní pásce o jedno políčko doleva.

Slovo w je přijato dvousměrným automatem právě tehdy, když automat začne práci v počátečním stavu q_0, hlava čte první písmeno vstupního slova w a automat při práci opustí pravý okraj vstupního slova a je při tom v některém z koncových stavů. □

2.9.2 Věta. Dvousměrné automaty přijímají pouze regulární jazyky. □

Větu nedokazujeme, důkaz je technicky náročný. Jeho myšlenka spočívá v tom, že se ukáže, že ke každému dvousměrnému automatu existuje DFA, který přijímá stejný jazyk; jedna z možností, jak to udělat, je použít Nerodovu větu.

Poznamenejme ale, že má-li dvousměrný automat n stavů, pak jemu odpovídající deterministický automat může mít až n^n.

Kapitola 3

Gramatiky

V předcházejících kapitolách jsme se zabývali konečnými automaty a jazyky, které tyto automaty přijímají. Nyní zavedeme nový „nástroj“, který nám umožní popsat větší třídy jazyků než jen regulární jazyky. Tímto nástrojem bude gramatika.

3.1 Hierarchie gramatik

Nejprve uvedeme příklad:

Příklad. V programovacích jazycích se často vyskytují definice podobné definici typu číslo v Backus-Naurově formě:

- ⟨číslo⟩ ::= ⟨číslo bez zn.⟩ | + ⟨číslo bez zn.⟩ | − ⟨číslo bez zn.⟩
- ⟨číslo bez zn.⟩ ::= ⟨číslice⟩ | ⟨číslo bez zn.⟩⟨číslice⟩
- ⟨číslice⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Jedná se o speciální příklad gramatiky: Symbolům ⟨číslo⟩, ⟨číslo bez zn.⟩ a ⟨číslice⟩ říkáme neterminály a symbolům $\{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ terminály. Cílem je podle „pravidel“ výše „vygenerovat“ čísla, ta představují terminální slova.

3.1.1 Definice. *Gramatika* je uspořádaná čtveřice $\mathcal{G} = (N, \Sigma, S, P)$, kde

- N je konečná množina tzv. *neterminálů*;
- Σ je konečná neprázdná množina tzv. *terminálů*, platí $N \cap \Sigma = \emptyset$;
- $S \in N$ je *startovací symbol*;
- P je konečná množina pravidel typu $\alpha \to \beta$, kde α a β jsou slova nad $N \cup \Sigma$ taková, že α obsahuje alespoň jeden neterminál.

□

Označme v předchozím příkladě $S = \langle\text{číslo}\rangle$, $A = \langle\text{číslo bez zn.}\rangle$ a $B = \langle\text{číslice}\rangle$, pak se jedná o gramatiku, kde

$$N = \{S, A, B\}, \quad \Sigma = \{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

a pravidla P jsou

- $S \to A \mid +A \mid -A$
- $A \to B \mid BA$
- $B \to 0 \mid 1 \mid \ldots \mid 9$.

Použili jsme zkrácený zápis 15 pravidel

- $S \to A, \quad S \to +A, \quad S \to -A,$
- $A \to B, \quad A \to BA,$
- $B \to 0, \quad B \to 1, \quad \ldots, \quad B \to 9.$

3.1.2 Přímé odvození. Zhruba řečeno, ze slova γ přímo odvodíme nějaké slovo δ jestliže existuje pravidlo $\alpha \to \beta$ z P takové, že ve slově γ najdeme některý výskyt podslova α a slovo δ vzniklo z γ nahrazením α slovem β. Formálně:

Definice. Je dána gramatika $\mathcal{G} = (N, \Sigma, S, P)$. Řekneme, že δ se *přímo odvodí* (též *přímo přepíše*) z γ, jestliže existuje v P pravidlo $\alpha \to \beta$ a slova $\varphi, \psi \in (N \cup \Sigma)^\star$ taková, že $\gamma = \varphi\,\alpha\,\psi$ a $\delta = \varphi\,\beta\,\psi$.

Tento fakt zapisujeme $\gamma \Rightarrow_{\mathcal{G}} \delta$. □

Například: máme přímé odvození $+BA \Rightarrow +1A$, protože jsme ve slově $+BA$ použili pravidlo $B \to 1$ a slovem 1 jsme nahradili neterminál B.

3.1.3 Odvození, derivace. Odvození je nyní posloupnost přímých odvození. Jedná se vlastně o reflexivní a tranzitivní uzávěr relace $\Rightarrow_{\mathcal{G}}$. Formálně:

Definice. Je dána gramatika $\mathcal{G} = (N, \Sigma, S, P)$. Řekneme, že δ se *odvodí* z γ, jestliže

- buď $\gamma = \delta$,
- nebo existuje posloupnost přímých odvození

$$\gamma = \gamma_1 \;\Rightarrow_{\mathcal{G}}\; \gamma_2 \;\Rightarrow_{\mathcal{G}} \ldots \Rightarrow_{\mathcal{G}}\; \gamma_k = \delta.$$

Tento fakt značíme $\gamma \Rightarrow^\star_{\mathcal{G}} \delta$ a této konečné posloupnosti říkáme *derivace*. □

3.1.4 Jazyk generovaný gramatikou. Definice. Řekneme, že slovo $w \in \Sigma^\star$ je *generováno* gramatikou $\mathcal{G}$, jestliže existuje derivace $S \;\Rightarrow^\star_{\mathcal{G}}\; w$.

Jazyk $L(\mathcal{G})$ *generovaný* gramatikou $\mathcal{G}$ se skládá ze všech slov generovaných gramatikou $\mathcal{G}$, tj.

$$L(\mathcal{G}) = \{w \in \Sigma^\star \mid S \;\Rightarrow^\star_{\mathcal{G}}\; w\}.$$

□

3.1.5 Konvence.

- Neterminály značíme obvykle velkými písmeny $A, B, X, Y, \ldots$.
- Terminály značíme obvykle malými písmeny ze začátku abecedy $a, b, c, d, \ldots$.
- Slova z $(N \cup \Sigma)^\star$ obvykle značíme řeckými písmeny α, β, ...
- Terminální slova, tj. slova z $\Sigma^\star$, značíme malými písmeny z konce abecedy $u, w, x, y, \ldots$.

Obvykle v textu vynecháváme jméno gramatiky, je-li z kontextu jasné, o jakou gramatiku se jedná. Píšeme proto $\Rightarrow$ a $\Rightarrow^\star$ místo $\Rightarrow_{\mathcal{G}}$ a $\Rightarrow^\star_{\mathcal{G}}$.

3.1.6 Chomského hierarchie. Podle podmínek, které klademe na pravidla dané gramatiky, rozlišujeme gramatiky a jimi generované jazyky takto:

- *Gramatiky typu 0* jsou gramatiky tak, jak jsme je zavedli v odstavci 3.1.1. Jazyky generované gramatikami typu 0 se nazývají *jazyky typu 0*.
- *Gramatiky typu 1*, též *kontextové gramatiky*, jsou takové gramatiky, kde každé pravidlo v P je tvaru
$$\alpha\,A\,\beta \to \alpha\,\gamma\,\beta,$$
kde $\alpha, \beta, \gamma \in (N \cup \Sigma)^\star$, A je neterminál a $\gamma \neq \varepsilon$. Jedinou výjimku tvoří pravidlo $S \to \varepsilon$, pak se ale S nevyskytuje na pravé straně žádného pravidla.

  Jazyky generované gramatikami typu 1 se nazývají *jazyky typu 1*, též *kontextové jazyky*.
- *Gramatiky typu 2*, též *bezkontextové gramatiky*, (což zkracujeme na CF gramatiky) jsou takové gramatiky, kde každé pravidlo v P je tvaru
$$A \to \gamma,$$
kde $\gamma \in (N \cup \Sigma)^\star$ a A je neterminál.

  Jazyky generované gramatikami typu 2 se nazývají *bezkontextové jazyky* nebo *jazyky typu 2*.

- *Gramatiky typu 3* neboli *regulární gramatiky* (též *pravé lineární gramatiky*) jsou takové gramatiky, kde každé pravidlo v P je tvaru

  $$A \to w\,B,\ A \to w,$$

  kde A, B jsou neterminály a w je terminální slovo.

  Jazyky generované gramatikami typu 3 se nazývají *regulární jazyky* nebo *jazyky typu 3*.

□

Poznamenejme, že regulární jazyky již byly definovány jako ty jazyky, které jsou přijímány konečnými automaty — později ukážeme, že je to správně, totiž, že každý jazyk typu 3 je přijímán konečným automatem, a naopak, každý regulární jazyk je generován gramatikou typu 3.