Týden 5

Přednáška

Redukce bezkontextových gramatik

Připomněli jsme si relaci $\Rightarrow$ na množině $(\Pi \cup \Sigma)^*$ pro danou bezkontextovou gramatiku $G = (\Pi, \Sigma, S, P)$. Také jsme si obecně připomněli *reflexivní a tranzitivní uzávěr* relace, konkrétně pak relaci $\Rightarrow^*$.

Víme tedy, že (pro danou G) je $\Rightarrow^*$ nejmenší množinou dvojic $(\alpha, \beta) \in (\Pi \cup \Sigma)^* \times (\Pi \cup \Sigma)^*$ splňující následující tři podmínky (pro každé $\alpha, \beta, \gamma \in (\Pi \cup \Sigma)^*$):

- $\alpha \Rightarrow^* \alpha$;
- $(\alpha \Rightarrow \beta,\ \beta \Rightarrow^* \gamma) \Longrightarrow \alpha \Rightarrow^* \gamma$.

Na konkrétním příkladu

$$
\begin{aligned}
S &\longrightarrow aSb \,|\, aAbb \,|\, \varepsilon \\
A &\longrightarrow aAB \,|\, bB \\
B &\longrightarrow aAb \,|\, BB \\
C &\longrightarrow CC \,|\, cS
\end{aligned}
$$

jsme se pak zabývali redukcí bezkontextové gramatiky.

Přitom jsme obecně pro gramatiku $G = (\Pi, \Sigma, S, P)$ definovali množinu $\mathcal{D}_G$ (značenou $\mathcal{D}$, když G je z kontextu jasná) následující induktivní definicí. (Tedy $\mathcal{D}$ je nejmenší množina splňující následující podmínky.)

- $S \in \mathcal{D}$;
- $(\, X \in \mathcal{D}, (X \longrightarrow \alpha Y \beta) \in P, Y \in \Pi \,) \Longrightarrow Y \in \mathcal{D}$.

Je vidět, že $\mathcal{D}$ obsahuje všechny dosažitelné neterminály, tedy takové, které se mohou objevit v nějakém odvození z počátečního S (byť to odvození nemusí třeba skončit terminálním slovem). Tedy $\mathcal{D} = \{\, Z \in \Pi \mid \exists \alpha, \beta : S \Rightarrow^* \alpha Z \beta \,\}$.

Množinu $\mathcal{T}_G$ (či $\mathcal{T}$) všech neterminálů, z nichž lze odvodit alespoň jedno terminální slovo, tedy množinu $\mathcal{T} = \{\, Z \in \Pi \mid \exists u \in \Sigma^* : Z \Rightarrow^* u \,\}$, můžeme definovat induktivní definicí s jedinou podmínkou:

- $(\, (X \longrightarrow \alpha) \in P, \alpha \in (\Sigma \cup \mathcal{T})^* \,) \Longrightarrow X \in \mathcal{T}$.

Obě definice dávají očividné návody k algoritmické konstrukci příslušných množin; ty jsme si také ujasnili a provedli.

(Poznámka. Na minulém cvičení jsme u definice $\mathcal{T}$ uvedli dvě podmínky. Vypíchli jsme totiž zvlášť případ $\alpha \in \Sigma^*$, na jehož základě zjistí algoritmus ty neterminály, které odvodí terminální slovo jedním krokem.)

Definice. Gramatika $G = (\Pi, \Sigma, S, P)$ je *redukovaná*, jestliže $\mathcal{D}_G = \mathcal{T}_G = \Pi$.

Zjistili jsme, že pro zredukování gramatiky G je vhodné nejdříve zjistit množinu $\mathcal{T}_G$ a odstranit z G všechna pravidla, která obsahují alespoň jeden neterminál, který není v $\mathcal{T}_G$. Výsledná gramatika G' očividně generuje tentýž jazyk jako G.

> (Co to znamená, když nic nezbude, tedy ani S nepatří do $\mathcal{T}_G$? Samozřejmě $L(G) = \emptyset$.)

Pro G' nyní zkonstruujeme množinu $\mathcal{D}_{G'}$ a odstraníme všechna pravidla, která obsahují neterminál(y), jež nejsou v $\mathcal{D}_{G'}$. Výsledná gramatika G'' je již redukovaná.

De facto jsme tak ukázali následující větu.

Věta. Ke každé BG G, pro niž $L(G) \neq \emptyset$, lze sestrojit redukovanou gramatiku G' tž. $L(G') = L(G)$.

Poznámka. Opačný postup, tj. odstranění nedosažitelných neterminálů a poté odstranění těch, které neodvodí terminální slovo, k redukované gramatice vést nemusí. To ilustruje např. gramatika s pravidly $S \longrightarrow a$, $S \longrightarrow XY$, $X \longrightarrow bX$, $Y \longrightarrow a$.

Speciální formy bezkontextových gramatik

Připomněli jsme si, co jsou *nevypouštějící gramatiky*; tyto gramatiky nemají pravidla typu $X \longrightarrow \varepsilon$. Algoritmus převodu obecné bezkontextové gramatiky G na nevypouštějící gramatiku G' takovou, že $L(G') = L(G) - \{\varepsilon\}$, navrhnete na cvičení. Tím bude jasná následující věta.

Věta. Ke každé BG G lze sestrojit nevypouštějící gramatiku G' tž. $L(G') = L(G) - \{\varepsilon\}$.

Poté jsme si připomněli gramatiky v *Chomského normální formě*. Taková gramatika má výhradně pravidla ve tvarech $X \longrightarrow YZ$ a $X \to a$. Tedy na pravé straně každého pravidla je buď jeden terminál nebo dva neterminály. Ilustrovali jsme si konstrukci, která dokazuje následující větu.

Věta. Ke každé BG G lze sestrojit BG G' v ChNF tž. $L(G') = L(G) - \{\varepsilon\}$.

Převod gramatiky G do ChNF se dá rozdělit do čtyř kroků:

- převod na nevypouštějící gramatiku,
- odstranění pravidel typu $X \to Y$,
- pro každý terminál a přidáme nový neterminál A_a a pravidlo $A_a \to a$; na každé pravé straně delší než 1 nahradíme a neterminálem A_a,
- každé pravidlo typu $X \to Y_1 Y_2 \ldots Y_n$, kde $n \geq 3$, nahradíme pravidly

$$\begin{aligned}
&X \to Y_1 Z_1,\\
&Z_1 \to Y_2 Z_2,\\
&\ldots\ldots\\
&Z_{n-3} \to Y_{n-2} Z_{n-2},\\
&Z_{n-2} \to Y_{n-1} Y_n
\end{aligned}$$

kde $Z_1, Z_2, \ldots, Z_{n-2}$ jsou nově přidané neterminály.

Některé myšlenky převodu jsme načrtli na příkladu následující gramatiky.

$$\begin{aligned}
&S \longrightarrow (E) \mid E\\
&E \longrightarrow F + F \mid F \times F\\
&F \longrightarrow a \mid S
\end{aligned}$$

Připomeňme ještě gramatiky v *Greibachové normální formě* (GNF). Taková gramatika má výhradně pravidla ve tvarech $X \longrightarrow a\alpha$, kde a je terminál a α je řetězec neterminálů (třeba prázdný). Platí také následující věta.
Věta. Ke každé BG G lze sestrojit BG G' v GNF tž. $L(G') = L(G) - \{\varepsilon\}$.

Zásobníkové automaty

Uvedli jsme neformálně model „zásobníkový automat“ (ZA) a zkonstruovali jsme konkrétní ZA M přijímající jazyk

$$L = \{wcw^R \mid w \in \{a,b\}^*\}\,.$$

(Sestavili jsme patřičnou množinu instrukcí typu $(q, a, X) \to (q', \alpha)$.)
Zároveň jsme uvedli formální definici zásobníkového automatu jako struktury

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)\,,$$

kde Q je konečná množina *stavů*, Σ je konečná *vstupní abeceda*, Γ je konečná *zásobníková abeceda*, $q_0 \in Q$ je *počáteční stav*, $Z_0 \in \Gamma$ je *počáteční zásobníkový symbol* a

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_{fin}(Q \times \Gamma^*)$$

je přechodová funkce (neboli konečná množina instrukcí).
Definovali jsme *konfiguraci ZA M* jako trojici (q, w, α), kde $q \in Q$, $w \in \Sigma^*$, $\alpha \in \Gamma^*$.
Na množině takových konfigurací přirozeně definujeme relaci $\vdash_M$ (odvození v jednom kroku):

$$(q, aw, X\beta) \vdash_M (q', w, \alpha\beta) \Leftrightarrow \delta(q, a, X) \ni (q', \alpha)$$

(kde $a \in (\Sigma \cup \{\varepsilon\})$, $w \in \Sigma^*$, $\beta \in \Gamma^*$).
Relace $\vdash_M^*$ (odvození v konečně mnoha krocích) je reflexivním a tranzitivním uzávěrem relace $\vdash_M$.
Jazykem rozpoznávaným (nebo též *přijímaným*) zásobníkovým automatem M rozumíme jazyk

$$L(M) = \{\, w \in \Sigma^* \mid (q_0, w, Z_0) \vdash_M^* (q, \varepsilon, \varepsilon) \text{ pro nějaký } q \in Q\,\}\,.$$

Připomněli jsme, že tedy jako základní bereme přijímání prázdným zásobníkem, a speciálně jsme zdůraznili, že jako základní verze se u zásobníkových automatů berou *nedeterministické* zásobníkové automaty.
Všimli jsme si, že náš výše sestrojený automat byl deterministický, což znamená

- $\delta(q, a, X)$ je vždy nejvýše jednoprvková množina (pro $a \in \Sigma \cup \{\varepsilon\}$) (tedy neexistují dvě různé instrukce se stejnou levou stranou),
- je-li $\delta(q, \varepsilon, X) \neq \emptyset$, pak $\delta(q, a, X) = \emptyset$ pro vš. $a \in \Sigma$ (může-li automat udělat ε-krok, nemůže mu jiná instrukce zároveň umožňovat přečtení vstupního symbolu).

Pak jsme náš konkrétní ZA upravili tak, aby rozpoznával jazyk

$$L' = \{ww^R \mid w \in \{a, b\}^*\}$$

a uvědomili jsme si, že vzniklý automat už není deterministický. Uvedli jsme si, že L' nelze rozpoznat žádným deterministickým ZA, tedy že L' nepatří do třídy DCFL. (Intuitivně to není těžké nahlédnout, exaktní důkaz je ovšem komplikovaný.)
Pak jsme si uvedli větu

> *Věta.* (Nedeterministické) zásobníkové automaty rozpoznávají právě bezkontextové jazyky, tedy jazyky z třídy CFL.

Ukažme ji v jednom směru (k bezkontextové gramatice G lze sestrojit [dokonce jen jednostavový] ZA M tak, že $L(G) = L(M)$); ilustrujme na příkladu gramatiky

1/ $A \longrightarrow A + B$
2/ $A \longrightarrow B$
3/ $B \longrightarrow B * C$
4/ $B \longrightarrow C$
5/ $C \longrightarrow (A)$
6/ $C \longrightarrow a$

Použijeme obecný postup

> Pro BG $G = (\Pi, \Sigma, S, P)$ sestrojme $M = (\{q_0\}, \Sigma, \Pi \cup \Sigma, \delta, q_0, S)$, kde
> pro $X \in \Pi$: $\delta(q_0, \varepsilon, X) = \{\, (q_0, \alpha) \mid (X \to \alpha) \in P \,\}$
> a pro $a \in \Sigma$: $\delta(q_0, a, a) = \{\, (q_0, \varepsilon) \,\}$.
> (Všude jinde δ přiřazuje $\emptyset$.)

Ilustrujeme-li běh sestrojeného ('velmi' nedeterministického) ZA např. na slově $a * (a + a)$, všimneme si, že tento běh odpovídá konstrukci příslušného derivačního stromu metodou „shora dolů“.

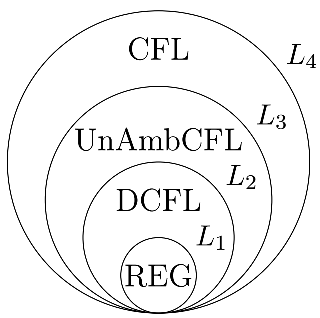
Vztah mezi přijímáním jazyka prázdným zásobníkem a přijímajícími stavy

Načrtli jsme, proč v tom není zásadní rozdíl; třída přijímaných jazyků je v obou případech stejná (jedná se o třídu bezkontextových jazyků).

V případě nedeterministických automatů lze tyto verze mezi sebou snadno převádět, využitím přidaného speciálního symbolu na dně zásobníku. V případě deterministických zásobníkových automatů je vhodné brát jako základní verzi tu s přijímacími stavy; lze ji ovšem nahradit přijímáním prázdným zásobníkem, zakončíme-li každé vstupní slovo speciálním koncovým symbolem. (Tedy když $L \subseteq \Sigma^*$ je přijímán DZA M s přijímacími stavy, tak jazyk $L \cdot \{\dashv\}$, kde $\dashv \not\in \Sigma$, je přijímán DZA M' prázdným zásobníkem.)

Hierarchie jazyků

Probrali jsme si načrtnutou hierarchii jazyků.



$$L_4 = \{a^n b^n c^n \mid n \geq 0\}$$
$$L_3 = \{a^i b^j c^k \mid i = j \vee j = k\}$$
$$L_2 = \{ww^R \mid w \in \{a,b\}^*\}$$
$$L_1 = \{a^n b^n \mid n \geq 1\}$$

Uzávěrové vlastnosti třídy CFL (tedy třídy bezkontextových jazyků)

Uvědomili jsme si, že snadno umíme ukázat, že CFL je uzavřena vůči sjednocení. Pohovořili jsme o dalších uzávěrových vlastnostech.

Partie textu k prostudování

Části 4.1. - 4.4. (bezkontextové gramatiky), část 5.1. (speciální bezkontextové gramatiky). Zásobníkové automaty (část 4.5.) a jejich varianty. Uzávěrové vlastnosti CFL (část 5.2.).

(Máte si udělat přinejmenším dobrou první představu a zamyslet se nad příklady, speciálně těmi plánovanými na cvičení, ať se můžete na cvičení aktivně účastnit a případné problémy si tam objasnit.)

Cvičení

Příklad 5.1

Připomeňte si správný postup redukce gramatiky a proveďte jej na příkladu.

$S \longrightarrow aBC \mid aCa \mid bBCa$

$B \longrightarrow bBa \mid bab \mid SS$

$C \longrightarrow BS \mid aCaa \mid bSSc$

Příklad 5.2

Pro bezkontextovou gramatiku $G = (\Pi, \Sigma, S, P)$ definujte množinu všech neterminálů, z nichž lze odvodit prázdné slovo, tedy množinu $\mathcal{E} = \{ Z \in \Pi \mid Z \Rightarrow^* \varepsilon \}$, induktivní definicí. $\mathcal{E} \subseteq \Pi$ je tedy nejmenší množina splňující podmínky .... (podobně jako u obdobných definic v přednášce).

Na základě této induktivní definice navrhněte algoritmus konstruující $\mathcal{E}$ a aplikujte jej na následující gramatiku G.

$$
\begin{aligned}
S &\longrightarrow AB \,|\, \varepsilon \\
A &\longrightarrow aAAb \,|\, BS \,|\, CA \\
B &\longrightarrow BbA \,|\, CaC \,|\, \varepsilon \\
C &\longrightarrow aBB \,|\, bS
\end{aligned}
$$

Teď bychom chtěli zkonstruovat gramatiku G', která nebude obsahovat tzv. ε-pravidla, tedy pravidla s pravou stranou ε, ale bude generovat stejný jazyk jako G, s případnou výjimkou ε (tedy $L(G') = L(G) - \{\varepsilon\}$).

Můžeme prostě ε-pravidla z G vypustit a prohlásit výslednou gramatiku za kýženou G' ? Vysvětlete, proč ne; tedy ukažte (jednoduchý) případ, kdy tento postup nefunguje.

Zkuste přijít na (obecně platný) postup, jak G' sestrojit, když známe množinu $\mathcal{E}$. (Nápověda. Pravidlo s pravou stranou obsahující neterminály z $\mathcal{E}$ je potřeba nahradit několika pravidly, v nichž jsou některé výskyty neterminálů z $\mathcal{E}$ případně vypuštěny; musíme přitom zaručit, že nedojde k ztrátě možnosti odvození nějakého neprázdného slova.)

Příklad 5.3

Domyslete podrobnosti převodu gramatiky do Chomského normální formy a aplikujte tento převod na následující gramatiku.

$$
\begin{aligned}
S &\longrightarrow (E) \,|\, E \\
E &\longrightarrow F + F \,|\, F \times F \\
F &\longrightarrow a \,|\, S
\end{aligned}
$$

Příklad 5.4

Promyslete si následující postup, který k bezkontextové gramatice G sestrojí jednostavový ZA M tak, že $L(G) = L(M)$):

Pro BG $G = (\Pi, \Sigma, S, P)$ sestrojme $M = (\{q_0\}, \Sigma, \Pi \cup \Sigma, \delta, q_0, S)$, kde

pro $X \in \Pi$: $\delta(q_0, \varepsilon, X) = \{ (q_0, \alpha) \mid (X \to \alpha) \in P \}$

a pro $a \in \Sigma$: $\delta(q_0, a, a) = \{ (q_0, \varepsilon) \}$.

(Všude jinde δ přiřazuje $\emptyset$.)

Aplikujte jej na příkladu gramatiky

1/ $A \longrightarrow A + B$
2/ $A \longrightarrow B$
3/ $B \longrightarrow B * C$
4/ $B \longrightarrow C$
5/ $C \longrightarrow (A)$
6/ $C \longrightarrow a$

Ilustrujte úspěšný běh sestrojeného (‘velmi’ nedeterministického) ZA např. na slově
$a * (a + a)$;
konstruujte tedy posloupnost konfigurací
$(q_0, a * (a + a), A) \vdash (q_0, a * (a + a), B) \vdash (q_0, a * (a + a), B * C) \vdash (q_0, a * (a + a), C * C) \vdash$
$\vdash (q_0, a * (a + a), a * C) \vdash (q_0, *(a + a), *C) \vdash (q_0, (a + a), C) \vdash \cdots \vdash (q_0, \varepsilon, \varepsilon)$
a všimněte si, že tento běh odpovídá konstrukci příslušného derivačního stromu metodou „shora dolů“.

Příklad 5.5

V jednom dřívějším příkladu jste zkonstruovali gramatiku generující jazyk

$$L = \{ w \in \{a, b\}^* \mid |w| \geq 1 \text{ a } |w|_a = |w|_b \}.$$

Připomeňte si ji a pak na ni aplikujte obecný postup, který k bezkontextové gramatice sestrojí ekvivalentní (jednostavový) zásobníkový automat (realizující metodu shora dolů). Takto sestrojíte (nedeterministický) ZA M.

Pak zkuste (co nejefektivnější) přímý návrh ZA M'. Podaří se vám navrhnout M' deterministický? Jestli ano, podaří se vám takový deterministický automat s jedním stavem?

Příklad 5.6

Připomeňte si, co to znamená, že CFL (třída bezkontextových jazyků) je uzavřena vůči sjednocení, zřetězení, iteraci a zrcadlovému obrazu. Pak toto prokažte (jednoduchými konstrukcemi, které např. k dvěma gramatikám G_1, G_2 sestrojí gramatiku G tž. $L(G) = L(G_1) \cup L(G_2)$, atd.).
(Nápověda pro zrcadlový obraz: připomeňte si, že
$(L_1 \cdot L_2 \cdot \cdots \cdot L_m)^R = (L_m)^R \cdot (L_{m-1})^R \cdot \cdots \cdot (L_1)^R$.)

Příklad 5.7

Prokažte, že CFL je uzavřena vůči průniku s regulárním jazykem, tedy, že pro každý bezkontextový jazyk L a každý regulární jazyk R platí, že $L \cap R$ je bezkontextový. (Využijte charakterizaci CFL pomocí zásobníkových automatů a vzpomeňte si na konstrukci automatu pro průnik dvou jazyků zadaných konečnými automaty.)