Týden 7

Přednáška-pondělí

Poznámky k příkladu syntaktické analýzy z minulého týdne

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 $\alpha, \beta, \gamma \in (\Pi \cup \Sigma)^*$):

1. když $\alpha \Rightarrow \beta$, tak $\alpha \Rightarrow^* \beta$;
2. $\alpha \Rightarrow^* \alpha$;
3. když $\alpha \Rightarrow^* \beta$ a $\beta \Rightarrow^* \gamma$, tak $\alpha \Rightarrow^* \gamma$.

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, jako nejmenší množinu splňující následující podmínky:

1. $S \in \mathcal{D}$;
2. když $X \in \mathcal{D}$ a $(X \longrightarrow \alpha Y \beta) \in P$, kde $Y \in \Pi$, tak $Y \in \mathcal{D}$.

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).
Množinu $\mathcal{T}_G$, stručněji $\mathcal{T}$, všech neterminálů, z nichž lze odvodit alespoň jedno terminální slovo, jsme definovali jako nejmenší množinu splňující následující podmínky:

1. když $(X \longrightarrow w) \in P$, kde $w \in \Sigma^*$, tak $X \in \mathcal{T}$;
2. když $(X \longrightarrow \alpha) \in P$, kde $\alpha \in (\Sigma \cup \mathcal{T})^*$, tak $X \in \mathcal{T}$.

Obě definice dávají očividné návody k algoritmické konstrukci příslušných množin; ty jsme si také ujasnili a provedli.
(Můžeme si všimnout, že v naší definici $\mathcal{T}$ je první podmínka zahrnuta v druhé a je tedy možné ji vypustit. Uvedli jsme ji proto, že explicitně vypichuje základní případ, tedy ty neterminály, u nichž se příslušnost k $\mathcal{T}$ zjistí hned v počáteční fázi algoritmu.)
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. (Opačný postup, tj. odstranění nedosažitelných neterminálů a poté odstranění těch, které neodvodí terminální slovo, k redukované gramatice vést nemusí.)

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 $X \to Y_1 Y_2 \dots Y_n$, kde $n \geq 3$, nahradíme pravidly

  $X \to Y_1 Z_1$,
  $Z_1 \to Y_2 Z_2$,
  .....
  $Z_{n-3} \to Y_{n-2} Z_{n-2}$,
  $Z_{n-2} \to Y_{n-1} Y_n$

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

Prvními dvěma kroky se zabýváte na cvičení (druhým v referátu). Na přednášce jsme si (druhý a) zbývající kroky ilustrovali na příkladu následující gramatiky.

$$S \longrightarrow (E) \mid E$$
$$E \longrightarrow F + F \mid F \times F$$
$$F \longrightarrow a \mid S$$

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 přijímaní 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.
Pak jsme si uvedli větu

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

Ukázali jsme si ji jen v jednom směru (k bezkontextové gramatice G lze sestrojit [dokonce jen jednostavový] ZA M tak, že $L(G) = L(M)$) a ilustrovali jsme na příkladu gramatiky

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

Použili jsme tedy 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 $\emptyset$.)

Ilustrovali jsme běh sestrojeného ('velmi' nedeterministického) ZA na slově $a * (a + a)$ a také fakt, že tento běh odpovídá konstrukci příslušného derivačního stromu metodou „shora dolů".

Přednáška-čtvrtek

Budeme se věnovat pumping lemmatu pro bezkontextové jazyky. (Poznámka. Je možné, že pojednáme i o něčem dalším.)

Věta. ($uvwxy$-teorém)
Nechť L je bezkontextový jazyk. Pak existují přirozená čísla p, q tž. každé slovo $z \in L$ delší než p lze psát ve tvaru $z = uvwxy$, přičemž platí $vx \neq \varepsilon$, $|vwx| \leq q$ a $uv^iwx^iy \in L$ pro vš. $i \geq 0$.

Vezmeme-li $n = \max\{p+1, q\}$, lze přeformulovat takto:

Věta. Nechť L je bezkontextový jazyk. Pak existuje přirozené číslo n tž. každé slovo $z \in L$, $|z| \geq n$, lze psát ve tvaru $z = uvwxy$, přičemž platí $vx \neq \varepsilon$, $|vwx| \leq n$ a $uv^iwx^iy \in L$ pro vš. $i \geq 0$.

Precizace „obrázkového důkazu“.

Nechť $L = L(G)$ pro BG $G = (\Pi, \Sigma, S, P)$ v Chomského NF, tedy s pravidly typu $X \longrightarrow YZ$, $X \longrightarrow a$.

Všimněme si: když na větvi deriv. stromu pro $z \in L$ jsou dva výskyty téhož neterminálu, řekněme A, pak

$S \Rightarrow^* uAy \Rightarrow^* uvAxy \Rightarrow^* uvwxy = z$

pro nějaké $u, v, w, x, y \in \Sigma^*$, $vx \neq \varepsilon$.

Nechť $|\Pi| = k$. Všimněme si dále:

- na větvi délky alespoň $k+1$ (tedy s alespoň $k+2$ vrcholy) jsou jistě alespoň dva výskyty téhož neterminálu (poslední vrchol už může být označen terminálem);

- má-li derivační strom pro $z \in \Sigma^*$ všechny větve kratší než $k+1$, pak nutně $|z| \leq 2^{k-1}$ (počet listů binárního stromu hloubky $k-1$; v poslední úrovni, při přepisu neterminálu na terminál, už k zdvojnásobení nedochází);

- v deriv. stromě pro $z \in L$, $|z| > 2^{k-1}$, určitě existují dva různé vrcholy v_1, v_2 na stejné větvi (v_1 blíž ke kořeni) označené stejným neterminálem, přičemž podstrom s kořenem v_1 má hloubku nejvýš $k+1$ [podstrom s kořenem v_1 už nemá vlastní podstrom, u nějž by došlo k opakování neterminálu na jedné větvi] – tedy má nejvýše 2^k listů.

Můžeme proto vzít: $p = 2^{k-1}$, $q = 2^k$.

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.

(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

Prezentace referátů

Referát č. 11 (Redukce bezkontextové gramatiky)
Vysvětlete, co dělá algoritmus popsaný v sekci 3 publikace

J. Esparza, P. Rossmanith, and S. Schwoon. A uniform framework for problems on context-free grammars. EATCS Bulletin, 72:169-177, October 2000,

která by měla být přístupná na
`http://www7.in.tum.de/um/bibdb/author-esparza.shtml`.
(Ilustrujte na vhodném příkladu.)

Pak stručně vysvětlete, jak je možné tento algoritmus využít při redukci gramatiky (tj. při "Identifying useless variables" na str. 8 zmíněné publikace).

Referát č. 12 (Chomského normální forma bezkontextových gramatik)

Při převodu bezkontextové gramatiky do Chomského normální formy (který je naznačen např. v textu
`http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.pdf`)
se po odstranění pravidel typu $A \to \varepsilon$ provádí krok, který odstraňuje pravidla typu $X \to Y$ (neterminál se přepisuje na neterminál). (Pak tedy dostaneme jen pravidla typu $A \to \alpha$, kde α je terminál nebo $|\alpha| \geq 2$, přičemž generovaný jazyk se nezměnil.)

Na vhodném příkladu ilustrujte algoritmus odstraňující ona pravidla $X \to Y$.

Příklady

Příklad 7.1

Pro bezkontextovou gramatiku $G = (\Pi, \Sigma, S, P)$ definujte množinu všech neterminálů, z nichž lze odvodit prázdné slovo (tedy ε), jako nejmenší množinu $\mathcal{E} \subseteq \Pi$ splňující ... (podobně jako u obdobných definic v přednášce).

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

$$S \longrightarrow AB \,|\, \varepsilon$$
$$A \longrightarrow aAAb \,|\, BS \,|\, CA$$
$$B \longrightarrow BbA \,|\, CaC \,|\, \varepsilon$$
$$C \longrightarrow aBB \,|\, bS$$

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.

Zkuste přijít na (obecně platný) postup, jak G' sestrojit, když známe množinu $\mathcal{E}$.

Příklad 7.2

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. 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 7.3

Připomeňte si, co to znamená, že CFL (třída bezkontextových jazyků) je uzavřena vůči sjednocení, zřetězení, iteraci a zrcadlovému obrazu. Pak toto prokažte (jednoduchými konstrukcemi).

Příklad 7.4

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. (Vzpomeňte si na konstrukci automatu pro průnik dvou jazyků zadaných konečnými automaty.)