Algoritmus CYK

Peter Kostolányi

16. apríla 2024

Cockeov-Youngerov-Kasamiho algoritmus, známy predovšetkým ako *algoritmus CYK*, umožňuje pre danú bezkontextovú gramatiku G a slovo w pomocou dynamického programovania zistiť, či slovo w patrí do jazyka $L(G)$. Ak považujeme veľkosť gramatiky G za konštantu, pracuje tento algoritmus v čase $O(|w|^3)$.

Vstupom algoritmu CYK je bezkontextová gramatika v tzv. prísnom Chomského normálnom tvare. Najprv sa preto zameriame na prevod gramatík do tohto normálneho tvaru, s ktorým je spojená aj potreba odstránenia reťazových pravidiel typu $\xi \to \eta$. Až následne opíšeme samotný algoritmus CYK.

1 Odstránenie reťazových pravidiel

Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. *Reťazovým pravidlom* (angl. „*chain rule*“) gramatiky G rozumieme ľubovoľné pravidlo $\xi \to \eta \in P$ s práve jedným neterminálom na jeho pravej strane; ide teda o ľubovoľné pravidlo z $P \cap (N \times N)$.

Ukážeme, že reťazových pravidiel sa dá „zbaviť“ – každú bezkontextovú gramatiku G možno previesť do normálneho tvaru bez reťazových pravidiel. Budeme sa pritom snažiť prísť s konštrukciou, ktorá zachováva „bezepsilonovosť“ pôvodnej gramatiky; bez tejto požiadavky by išlo o triviálnu záležitosť.

Veta 1. *Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. Potom existuje bezkontextová gramatika $G' = (N, T, P', \sigma)$ taká, že $L(G') = L(G)$ a $P' \cap (N \times N) = \emptyset$.*

Vstupom nasledujúceho algoritmu na odstránenie reťazových pravidiel je ľubovoľná bezkontextová gramatika $G = (N, T, P, \sigma)$. Výstupom je ekvivalentná gramatika $G' = (N, T, P', \sigma)$ bez reťazových pravidiel, ktorá sa od gramatiky G líši iba v množine prepisovacích pravidiel P'. Každá pravá strana pravidla z P' ale bude aj pravou stranou pravidla z P, čo okrem iného zaručí splnenie spomínanej požiadavky na zachovanie „bezepsilonovosti“ gramatiky.

1. Inicializuj $P' := \emptyset$.
2. Zostroj orientovaný graf D s množinou vrcholov N, v ktorom z vrcholu ξ do vrcholu η vedie hrana práve vtedy, keď $\xi \to \eta \in P$.
3. Opakuj pre každý neterminál $\xi \in N$:
   - 3.1 Nájdi množinu vrcholov $A(\xi)$ dosiahnuteľných v grafe D z vrcholu ξ.
   - 3.2 Pre každý neterminál $\eta \in A(\xi)$ a pre každé pravidlo $\eta \to x \in P$ také, že $x \notin N$ pridaj do P' pravidlo $\xi \to x$.

Pre každé $\xi \in N$ je $\xi \in A(\xi)$, preto množina pravidiel P' obsahuje okrem iného aj všetky pravidlá z P, ktoré nie sú reťazové. Ak gramatika G obsahuje pre nejaký neterminál ξ pravidlo $\xi \to \xi$, obsahuje graf D slučku vo vrchole ξ. Výsledná množina pravidiel P' je ale rovnaká ako v prípade, keď graf D takúto slučku neobsahuje – pravidlá $\xi \to \xi$ teda možno pri implementácii algoritmu ignorovať.

Príklad 1. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika, kde $N = \{\sigma, \alpha, \beta, \gamma\}$, $T = \{a, b\}$ a

$$\begin{aligned} P = \{&\sigma \to a\alpha \mid \alpha \mid aa \\ &\alpha \to \sigma \mid \beta \mid b \\ &\beta \to a\sigma \mid b\beta \mid \gamma a \\ &\gamma \to b\alpha \mid \varepsilon\}. \end{aligned}$$

Aplikujme na túto gramatiku algoritmus CYK a nájdime ekvivalentnú gramatiku $G' = (N, T, P', \sigma)$ bez reťazových pravidiel.

Graf D zodpovedajúci gramatike G je znázornený na obrázku 1. Zrejme je $A(\sigma) = \{\sigma, \alpha, \beta\}$, $A(\alpha) = \{\sigma, \alpha, \beta\}$, $A(\beta) = \{\beta\}$ a $A(\gamma) = \{\gamma\}$.

Obr. 1: Orientovaný graf D zodpovedajúci gramatike G.

Výsledná gramatika $G' = (N, T, P', \sigma)$ má preto množinu pravidiel P' danú ako

$$\begin{aligned} P' = \{&\sigma \to a\alpha \mid aa \mid b \mid a\sigma \mid b\beta \mid \gamma a \\ &\alpha \to a\alpha \mid aa \mid b \mid a\sigma \mid b\beta \mid \gamma a \\ &\beta \to a\sigma \mid b\beta \mid \gamma a \\ &\gamma \to b\alpha \mid \varepsilon\}. \end{aligned}$$

2 Prísny Chomského normálny tvar

Prísny Chomského normálny tvar sa od „obyčajného“ Chomského normálneho tvaru líši tým, že sú navyše zakázané pravidlá typu $\xi \to \varepsilon$.

Definícia 1. Bezkontextová gramatika $G = (N, T, P, \sigma)$ je v *prísnom Chomského normálnom tvare*, ak $P \subseteq N \times (N^2 \cup T)$.

Poznámka 1. Gramatika v prísnom Chomského normálnom tvare zjavne nemôže generovať prázdne slovo ε. Pôjde teda o normálny tvar „až na ε“. Niekedy sa, rovnako ako napríklad pri „bezepsilonovom“ normálnom tvare alebo Greibachovej normálnom tvare, povoľuje pravidlo $\sigma \to \varepsilon$ v prípade, že sa počiatočný neterminál σ nevyskytuje na pravej strane žiadneho pravidla.

Veta 2. *Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. Potom existuje gramatika G' v prísnom Chomského normálnom tvare taká, že $L(G') = L(G) - \{\varepsilon\}$.*

Vstupom nasledujúceho algoritmu je bezkontextová gramatika $G = (N, T, P, \sigma)$. Výstupom je gramatika $G' = (N', T, P', \sigma)$ v prísnom Chomského normálnom tvare taká, že $L(G') = L(G) - \{\varepsilon\}$.

1. Pomocou štandardného algoritmu preveď gramatiku G do „bezepsilonového“ normálneho tvaru. Nech G_1 je výsledná gramatika.
2. Pomocou štandardného algoritmu zbav gramatiku G_1 reťazových pravidiel. Nech G_2 je výsledná gramatika.
3. Pre každý terminál $c \in T$ zaveď nový neterminál ξ_c a pravidlo $\xi_c \to c$.
4. Vo všetkých pravidlách *gramatiky G_2 dĺžky aspoň dva* nahraď všetky výskyty každého terminálu c príslušným neterminálom ξ_c. Nech $G_3 = (N_3, T, P_3, \sigma)$ je výsledná gramatika.
5. Pre každé pravidlo $\pi\colon \alpha \to \beta_1\beta_2 \dots \beta_k \in P_3$, kde $k \in \mathbb{N} - \{0, 1, 2\}$ a $\beta_1, \dots, \beta_k \in N_3$:
   - 5.1 Zaveď nové neterminály $\psi_{\pi,1}, \dots, \psi_{\pi,k-2}$.
   - 5.2 Odober pravidlo $\alpha \to \beta_1\beta_2 \dots \beta_k$.
   - 5.3 Pridaj pravidlá $\alpha \to \beta_1\psi_{\pi,1}, \psi_{\pi,1} \to \beta_2\psi_{\pi,2}, \dots, \psi_{\pi,k-2} \to \beta_{k-1}\beta_{k-2}$.

   Vráť na výstupe výslednú gramatiku $G' = (N', T, P', \sigma)$.

V kroku 1 uvedeného algoritmu sa gramatika G „odepsilonuje“, v dôsledku čoho je na pravej strane každého pravidla gramatiky G_1 neprázdne slovo. V kroku 2 sa gramatika G_1 zbaví reťazových pravidiel, pričom nepribudnú žiadne pravidlá s ε na pravej strane. Gramatika G_2 teda obsahuje na pravej strane každého pravidla buď terminál, alebo slovo dĺžky aspoň dva. V krokoch 3 a 4 sa táto gramatika prerobí na gramatiku G_3, kde na pravej strane každého pravidla je buď terminál, alebo slovo dĺžky aspoň dva pozostávajúce iba z neterminálov. Obzvlášť treba upozorniť na skutočnosť, že v kroku 4 sa na rozdiel od prevodu do „obyčajného“ Chomského normálneho tvaru nerobí nič s pravidlami typu $\xi \to c$. Príliš dlhé pravé strany sa nakoniec skrátia v kroku 5 rovnako ako pri prevode do „obyčajného“ Chomského normálneho tvaru.

Príklad 2. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika, kde $N = \{\sigma, \alpha, \beta\}$, $T = \{a, b\}$ a

$$\begin{aligned} P = \{&\sigma \to \alpha a \alpha \mid aa \\ &\alpha \to \beta \mid \beta bb \\ &\beta \to a\sigma \mid ab \mid \varepsilon\}. \end{aligned}$$

Preveďme gramatiku G do prísneho Chomského normálneho tvaru.

Začnime krokom 1 spočívajúcim v „odepsilonovaní“ gramatiky G – štandardná konštrukcia si tu vyžaduje nájsť množinu E všetkých vymazávajúcich neterminálov tejto gramatiky, čo zrealizujeme známym iteratívnym algoritmom:

$$\begin{aligned} E_0 &= \{\beta\}, \\ E_1 &= \{\alpha, \beta\}, \\ E_2 &= \{\alpha, \beta\} = E. \end{aligned}$$

Následne do množiny prepisovacích pravidiel pridáme pravidlá zodpovedajúce všetkým možným vypusteniam neterminálov z E na pravých stranách pôvodných pravidiel a odoberieme pravidlá s prázdnym slovom na pravej strane. Dostávame tak gramatiku s pravidlami

$$\begin{aligned} &\sigma \to \alpha a \alpha \mid aa \mid a\alpha \mid \alpha a \mid a, \\ &\alpha \to \beta \mid \beta bb \mid bb, \\ &\beta \to a\sigma \mid ab. \end{aligned}$$

Z tejto gramatiky teraz v rámci kroku 2 odstránime reťazové pravidlá. Po použití algoritmu z predchádzajúceho oddielu dostávame gramatiku s prepisovacími pravidlami

$$\begin{aligned} &\sigma \to \alpha a \alpha \mid aa \mid a\alpha \mid \alpha a \mid a, \\ &\alpha \to \beta bb \mid bb \mid a\sigma \mid ab, \\ &\beta \to a\sigma \mid ab. \end{aligned}$$

V krokoch 3 a 4 uvedeného algoritmu sa gramatika G_2 transformuje na gramatiku s prepisovacími pravidlami

$$\begin{aligned} \sigma &\to \alpha \xi_a \alpha \mid \xi_a \xi_a \mid \xi_a \alpha \mid \alpha \xi_a \mid a, \\ \alpha &\to \beta \xi_b \xi_b \mid \xi_b \xi_b \mid \xi_a \sigma \mid \xi_a \xi_b, \\ \beta &\to \xi_a \sigma \mid \xi_a \xi_b, \\ \xi_a &\to a, \\ \xi_b &\to b. \end{aligned}$$

V kroku 5 sa napokon ošetria príliš dlhé pravé strany pravidiel. Zaveďme označenie pravidiel $\pi_1 := \sigma \to \alpha\xi_a\alpha$ a $\pi_2 := \alpha \to \beta\xi_b\xi_b$. Pre výslednú gramatiku $G' = (N', T, P', \sigma)$ potom dostávame $N' = \{\sigma, \alpha, \beta, \xi_a, \xi_b, \psi_{\pi_1,1}, \psi_{\pi_2,1}\}$ a

$$\begin{aligned} P' = \{&\sigma \to \alpha\psi_{\pi_1,1} \mid \xi_a\xi_a \mid \xi_a\alpha \mid \alpha\xi_a \mid a \\ &\alpha \to \beta\psi_{\pi_2,1} \mid \xi_b\xi_b \mid \xi_a\sigma \mid \xi_a\xi_b \\ &\beta \to \xi_a\sigma \mid \xi_a\xi_b \\ &\xi_a \to a \\ &\xi_b \to b \\ &\psi_{\pi_1,1} \to \xi_a\alpha \\ &\psi_{\pi_2,1} \to \xi_b\xi_b\}. \end{aligned}$$

3 Algoritmus CYK

Vstupom *Cockeovho-Youngerovho-Kasamiho algoritmu* (alebo *algoritmu CYK*) je bezkontextová gramatika $G = (N, T, P, \sigma)$ v prísnom Chomského normálnom tvare a slovo w. Výstupom je „áno“ alebo „nie“ podľa toho, či $w \in L(G)$ alebo $w \notin L(G)$.

V algoritme CYK sa postupne konštruuje množina $M \subseteq N$ neterminálov ξ takých, že $\xi \Rightarrow^* w$. Je zrejmé, že $w \in L(G)$ práve vtedy, keď $\sigma \in M$.

Samotná konštrukcia množiny neterminálov M je založená na nasledujúcom kľúčovom pozorovaní. Nech $w = a_1 \dots a_n$, kde $n \in \mathbb{N} \setminus \{0\}$ a $a_1, \dots, a_n \in T$.[1] Pre $i = 1, \dots, n$ a $j = 1, \dots, n - i + 1$ označme ako $M_{i,j}$ množinu neterminálov ξ takých, že $\xi \Rightarrow^* a_i \dots a_{i+j-1}$ – neterminál ξ teda dokáže vygenerovať podslovo dĺžky j slova w začínajúce na jeho i-tej pozícii. Pre $i = 1, \dots, n$ potom zrejme

$$M_{i,1} = \{\xi \in N \mid \xi \to a_i \in P\}. \tag{1}$$

Ak má totiž pre neterminál ξ existovať odvodenie $\xi \Rightarrow^* a_i$, musí byť toto odvodenie vďaka prísnemu Chomského normálnemu tvaru dĺžky jedna. Podobne pre $i = 1, \dots, n$ a $j = 2, \dots, n - i + 1$ je

$$M_{i,j} = \{\xi \in N \mid \exists k \in [j-1]\ \exists \eta \in M_{i,k}\ \exists \nu \in M_{i+k,j-k} \colon \xi \to \eta\nu \in P\}. \tag{2}$$

Odvodenie $\xi \Rightarrow^* a_i \dots a_{i+j-1}$ sa totiž vďaka prísnemu Chomského normálnemu tvaru musí začať použitím nejakého pravidla $\xi \to \eta\nu$. Ak má ale byť $\xi \Rightarrow \eta\nu \Rightarrow^* a_i \dots a_{i+j-1}$, musí nutne pre nejaké $k \in \{i, \dots, j-1\}$ byť aj $\eta \Rightarrow^* a_i \dots a_{i+k-1}$ a $\nu \Rightarrow^* a_{i+k} \dots a_{i+j-1}$.

Konštrukcia množín $M_{i,1}$ je teda triviálnou záležitosťou. Ak teraz poznáme všetky množiny $M_{i,j}$ pre $j \leq r$, ľahko na základe (2) nájdeme aj množiny $M_{i,j}$ pre $j = r+1$. Rovnice (1) a (2) tak umožňujú skonštruovať všetky množiny $M_{i,j}$ pomocou dynamického programovania, postupne pre rastúce j. Množina M je potom daná ako $M = M_{1,n}$.

Príklad 3. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika v prísnom Chomského normálnom tvare, kde $N = \{\sigma, \alpha, \beta, \gamma\}$, $T = \{a, b, c\}$ a

$$\begin{aligned} P = \{&\sigma \to \alpha\alpha \mid \beta\gamma \\ &\alpha \to \beta\alpha \mid \gamma\gamma \mid a \\ &\beta \to \sigma\alpha \mid b \\ &\gamma \to \beta\sigma \mid c\}. \end{aligned}$$

Pomocou algoritmu CYK zistíme, či slovo $w = bcacca$ patrí do jazyka $L(G)$. Je teda $n = |w| = 6$.

[1] Prípad $w = \varepsilon$ je triviálny, keďže gramatika v prísnom Chomského normálnom tvare nikdy nemôže vygenerovať prázdne slovo. Pri rozšírenej definícii prísneho Chomského normálneho tvaru, umožňujúcej pravidlo $\sigma \to \varepsilon$ za predpokladu, že sa σ nevyskytuje na pravej strane žiadneho pravidla, treba prázdne slovo riešiť osobitne (stačí sa ale „pozrieť“, či gramatika obsahuje pravidlo $\sigma \to \varepsilon$).

S použitím vzťahu (1) najprv vypočítame množiny $M_{i,1}$ pre $i = 1, \ldots, 6$:

$$M_{1,1} = \{\beta\}, \quad M_{2,1} = \{\gamma\}, \quad M_{3,1} = \{\alpha\}, \quad M_{4,1} = \{\gamma\}, \quad M_{5,1} = \{\gamma\}, \quad M_{6,1} = \{\alpha\}.$$

Vypočítajme teraz množiny $M_{i,2}$ pre $i = 1, \ldots, 5$. Napríklad $M_{1,2}$ pozostáva podľa (2) z neterminálov ξ takých, že existuje pravidlo $\xi \to \eta\nu$, kde $\eta \in M_{1,1}$ a $\nu \in M_{2,1}$. Jediným takýmto neterminálom je neterminál σ, pre ktorý existuje pravidlo $\sigma \to \beta\gamma$. Rovnakým spôsobom vypočítame aj množiny $M_{2,2}, \ldots, M_{5,2}$:

$$M_{1,2} = \{\sigma\}, \quad M_{2,2} = \emptyset, \quad M_{3,2} = \emptyset, \quad M_{4,2} = \{\alpha\}, \quad M_{5,2} = \emptyset.$$

Následne môžeme pokračovať s množinami $M_{i,3}$ pre $i = 1, \ldots, 4$. Uvažujme množinu $M_{1,3}$, ktorá podľa (2) obsahuje neterminály ξ také, že existuje pravidlo $\xi \to \eta\nu$, kde buď $\eta \in M_{1,1}$ a $\nu \in M_{2,2}$ (ak $k = 1$), alebo $\eta \in M_{1,2}$ a $\nu \in M_{3,1}$ (ak $k = 2$). V prvom prípade je množina $M_{2,2}$ prázdna, a teda nedostávame žiaden neterminál. V druhom prípade dostávame neterminál β, keďže preň existuje pravidlo $\beta \to \sigma\alpha$, kde $\sigma \in M_{1,2}$ a $\alpha \in M_{3,1}$. Ako možno ľahko overiť, žiaden iný neterminál do $M_{1,3}$ nepatrí. Podobne vypočítame aj množiny $M_{2,3}, \ldots, M_{4,3}$:

$$M_{1,3} = \{\beta\}, \quad M_{2,3} = \emptyset, \quad M_{3,3} = \{\sigma\}, \quad M_{4,3} = \{\sigma\}.$$

Podobne môžeme pokračovať aj ďalej, pričom pre $j = 4$ dostávame

$$M_{1,4} = \{\sigma\}, \quad M_{2,4} = \emptyset, \quad M_{3,4} = \{\beta\},$$

pre $j = 5$ dostávame

$$M_{1,5} = \{\alpha\}, \quad M_{2,5} = \emptyset,$$

a pre $j = 6$ napokon dostávame

$$M = M_{1,6} = \{\sigma, \gamma\}.$$

Keďže $\sigma \in M$, môžeme uzavrieť, že $w \in L(G)$. Celý tento proces je zhrnutý v tabuľke 1.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
$j = 1$	β	γ	α	γ	γ	α
$j = 2$	σ	$\emptyset$	$\emptyset$	α	$\emptyset$	—
$j = 3$	β	$\emptyset$	σ	σ	—	—
$j = 4$	σ	$\emptyset$	β	—	—	—
$j = 5$	α	$\emptyset$	—	—	—	—
$j = 6$	σ, γ	—	—	—	—	—

Tabuľka 1: Beh algoritmu CYK pre gramatiku G a slovo *bcacca*.