

# Algoritmus CYK

Peter Kostolányi

28. marca 2017

*Cockeov-Youngerov-Kasamiho algoritmus*, známy predovšetkým ako *algoritmus CYK*, umožňuje pre danú bezkontextovú gramatiku  $G$  a slovo  $w$  zistiť, či slovo  $w$  patrí do jazyka  $L(G)$ . Algoritmus pracuje v čase<sup>1</sup>  $O(|w|^3)$  a je založený na dynamickom programovaní.

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  $\xi \rightarrow \eta$ . Až následne popíšeme samotný algoritmus CYK.

## Odstránenie reťazových pravidiel

Nech  $G = (N, T, P, \sigma)$  je bezkontextová gramatika. *Reťazovým pravidlom* (angl. „chain rule“) gramatiky  $G$  nazveme ľubovoľné pravidlo  $\xi \rightarrow \eta \in P$  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 (ako je zrejmé z použitej notácie) líši od gramatiky  $G$  iba v množine prepisovacích pravidiel.

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

Pre každé  $\xi \in N$  platí  $\xi \in A(\xi)$ , preto množina pravidiel  $P'$  obsahuje okrem iného aj všetky pravidlá z  $P$ , ktoré nie sú reťazové. Ak gramatika  $G$  obsahuje pre nejaký neterminál  $\xi$  pravidlo  $\xi \rightarrow \xi$ , graf  $D$  obsahuje slučku vo vrchole  $\xi$ . Výsledná množina pravidiel  $P'$  je ale rovnaká ako v prípade, keď graf  $D$  takúto slučku neobsahuje – algoritmus teda možno optimalizovať tak, že pravidlá  $\xi \rightarrow \xi$  nebude vôbec uvažovať.

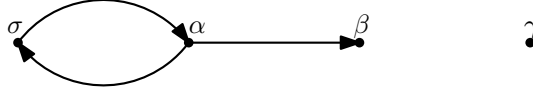
*Príklad 1.* Nech  $G = (N, T, P, \sigma)$  je bezkontextová gramatika, kde  $N = \{\sigma, \alpha, \beta, \gamma\}$ ,  $T = \{a, b\}$  a

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

<sup>1</sup>Ak „veľkosť“ gramatiky  $G$  považujeme za konštantu.

Aplikujeme horeuvedený algoritmus na zostrojenie ekvivalentnej gramatiky  $G' = (N, T, P', \sigma)$  bez reťazových pravidiel.

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

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

### Prísny Chomského normálny tvar

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

**Definícia 1.** Nech  $G = (N, T, P, \sigma)$  je bezkontextová gramatika. Gramatika  $G$  je v prísnom Chomského normálnom tvare, ak  $P \subseteq N \times (NN \cup T)$ .

*Poznámka 1.* Gramatika v prísnom Chomského normálnom tvare zjavne nemôže generovať prázdne slovo  $\varepsilon$ . Pôjde teda o normálny tvar „až na  $\varepsilon$ “. Niekedy sa tento drobný nedostatok, rovnako ako pri „bezepsilonovom“ normálnom tvare alebo pri Greibachovej normálnom tvare, rieši povolením pravidla  $\sigma \rightarrow \varepsilon$  v prípade, že sa počiatočný neterminál  $\sigma$  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 preved' 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  $c \in T$  zaved' nový neterminál  $\xi_c$  a pravidlo  $\xi_c \rightarrow c$ .
4. Vo všetkých pravidlách gramatiky  $G_2$  dĺžky aspoň dva nahrad' všetky výskyty každého terminálu  $c$  zodpovedajúcim neterminálom  $\xi_c$ . Nech  $G_3 = (N_3, T, P_3, \sigma)$  je výsledná gramatika.
5. Pre každé pravidlo  $\pi: \alpha \rightarrow \beta_1\beta_2 \dots \beta_k \in P_3$ ,  $k > 2$ ,  $\beta_1, \dots, \beta_k \in N_3$ :
  - 5.1 Zaved' nové neterminály  $\psi_{\pi,1}, \dots, \psi_{\pi,k-2}$ .
  - 5.2 Odober pravidlo  $\alpha \rightarrow \beta_1\beta_2 \dots \beta_k$ .
  - 5.3 Pridaj pravidlá  $\alpha \rightarrow \beta_1\psi_{\pi,1}, \psi_{\pi,1} \rightarrow \beta_2\psi_{\pi,2}, \dots, \psi_{\pi,k-2} \rightarrow \beta_{k-1}\beta_k$ .

Označ výslednú gramatiku ako  $G' = (N', T, P', \sigma)$ .

*Poznámka 2.* V kroku 1 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  $\varepsilon$  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  $\xi \rightarrow c$ . 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

$$P = \{\sigma \rightarrow \alpha\alpha\alpha \mid aa \\ \alpha \rightarrow \beta \mid \beta\beta\beta \\ \beta \rightarrow a\sigma \mid ab \mid \varepsilon\}.$$

Prevedme gramatiku  $G$  do prísneho Chomského normálneho tvaru.

Krokom 1 horeuvedeného algoritmu je prevod do „bezepsilonového normálneho tvaru“. Ten začína nájdením množiny vymazávajúcich neterminálov  $E$ :

$$E_0 = \{\beta\}, \\ E_1 = \{\alpha, \beta\}, \\ E_2 = \{\alpha, \beta\} = E.$$

Následne sa z množiny prepisovacích pravidiel odoberie pravidlo  $\beta \rightarrow \varepsilon$  a pridajú sa pravidlá umožňujúce „vypustenie“ neterminálov z  $E$  (za predpokladu, že takýmto „vypustením“ nevznikne prázdne slovo). Množina prepisovacích pravidiel  $P_1$  gramatiky  $G_1$  je potom nasledovná:

$$P_1 = \{\sigma \rightarrow \alpha\alpha\alpha \mid aa \mid a\alpha \mid \alpha a \mid a \\ \alpha \rightarrow \beta \mid \beta\beta\beta \mid bb \\ \beta \rightarrow a\sigma \mid ab\}.$$

V kroku 2 sa z gramatiky  $G_1$  odstránia reťazové pravidlá. Po použití štandardného postupu z predchádzajúceho oddielu dostávame gramatiku  $G_2$  s množinou prepisovacích pravidiel

$$P_2 = \{\sigma \rightarrow \alpha\alpha\alpha \mid aa \mid a\alpha \mid \alpha a \mid a \\ \alpha \rightarrow \beta\beta\beta \mid bb \mid a\sigma \mid ab \\ \beta \rightarrow a\sigma \mid ab\}.$$

V krokoch 3 a 4 sa gramatika  $G_2$  transformuje na gramatiku  $G_3$  s množinou prepisovacích pravidiel

$$P_3 = \{\sigma \rightarrow \alpha\xi_a\alpha \mid \xi_a\xi_a \mid \xi_a\alpha \mid \alpha\xi_a \mid a \\ \alpha \rightarrow \beta\xi_b\xi_b \mid \xi_b\xi_b \mid \xi_a\sigma \mid \xi_a\xi_b \\ \beta \rightarrow \xi_a\sigma \mid \xi_a\xi_b \\ \xi_a \rightarrow a \\ \xi_b \rightarrow b\}.$$

V kroku 5 sa nakoniec ošetrí príliš dlhé pravé strany pravidiel. Zavedme označenie pravidiel  $\pi_1 := \sigma \rightarrow \alpha\xi_a\alpha$  a  $\pi_2 := \alpha \rightarrow \beta\xi_b\xi_b$ . Pre výslednú gramatiku  $G' = (N', T, P', \sigma)$  potom dostávame

$$N' = \{\sigma, \alpha, \beta, \xi_a, \xi_b, \psi_{\pi_1,1}, \psi_{\pi_2,1}\} \text{ a}$$

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

### 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  $\xi$  takých, že  $\xi \Rightarrow^* w$ . Je zrejmé, že  $w \in L(G)$  práve vtedy, keď  $\sigma \in M$ .

Samotná konštrukcia množiny  $M$  je založená na nasledujúcom kľúčovom pozorovaní. Nech  $w = a_1 \dots a_n$ , kde  $n \geq 1$ .<sup>2</sup> Pre  $1 \leq i \leq n$  a  $1 \leq j \leq n - i + 1$  označme symbolom  $M_{i,j}$  množinu neterminálov  $\xi$  takých, že  $\xi \Rightarrow a_i \dots a_{i+j-1}$  – neterminál  $\xi$  teda dokáže vygenerovať podslovo slova  $w$  začínajúce na  $i$ -tej pozícii a dĺžky  $j$ . Pre všetky  $1 \leq i \leq n$  potom zrejme platí

$$M_{i,1} = \{\xi \in N \mid \xi \rightarrow a_i \in P\}. \quad (1)$$

Ak má totiž pre neterminál  $\xi$  platiť  $\xi \Rightarrow^* a_i$ , musí byť toto odvodenie vďaka prísnemu Chomského normálnemu tvaru dĺžky jedna. Podobne, pre všetky  $1 \leq i \leq n$  a  $1 < j \leq n - i + 1$  platí

$$M_{i,j} = \{\xi \in N \mid \exists k \in \{1, \dots, j-1\}, \eta \in M_{i,k}, \nu \in M_{i+k,j-k} : \xi \rightarrow \eta\nu \in P\}. \quad (2)$$

Ovodenie  $\xi \Rightarrow^* a_i \dots a_{i+j-1}$  sa totiž vďaka prísnemu Chomského normálnemu tvaru musí začať použitím nejakého pravidla  $\xi \rightarrow \eta\nu$ . Ak má ale platiť  $\xi \Rightarrow \eta\nu \Rightarrow^* a_i \dots a_{i+j-1}$ , musí nutne pre nejaké  $k \in \{i, \dots, j-1\}$  platiť 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 zjavne 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  $j$ . Pre množinu  $M$  potom platí  $M = M_{1,n}$ .

*Príklad 3.* Algoritmus CYK ilustrujeme na príklade. 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

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

Nech  $w = bcacca$  (a teda  $n = |w| = 6$ ). Pomocou (1) najprv vypočítame množiny  $M_{i,1}$  pre  $i = 1, \dots, 6$ :

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

<sup>2</sup>Prípad  $n = 0$  je triviálny, keďže gramatika v prísnom Chomského normálnom tvare nemôže generovať  $\varepsilon$ . Pri rozšírenej definícii prísneho Chomského normálneho tvaru, umožňujúcej pravidlo  $\sigma \rightarrow \varepsilon$  za predpokladu, že sa  $\sigma$  nevyskytuje na pravej strane žiadneho pravidla, treba prázdne slovo riešiť osobitne (stačí sa ale „pozrieť“, či gramatika obsahuje pravidlo  $\sigma \rightarrow \varepsilon$ ).

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

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

Následne môžeme pokračovať s množinami  $M_{i,3}$  pre  $i = 1, \dots, 4$ . Uvažujme množinu  $M_{1,3}$ , ktorá podľa (2) obsahuje neterminály  $\xi$  také, že existuje pravidlo  $\xi \rightarrow \eta\nu$ , 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  $\beta$ , keďže preň existuje pravidlo  $\beta \rightarrow \sigma\alpha$ , 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 zvyšné množiny:

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

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 nakoniec, pre  $j = 6$  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.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
$j = 1$	$\beta$	$\gamma$	$\alpha$	$\gamma$	$\gamma$	$\alpha$
$j = 2$	$\sigma$	$\emptyset$	$\emptyset$	$\alpha$	$\emptyset$	—
$j = 3$	$\beta$	$\emptyset$	$\sigma$	$\sigma$	—	—
$j = 4$	$\sigma$	$\emptyset$	$\beta$	—	—	—
$j = 5$	$\alpha$	$\emptyset$	—	—	—	—
$j = 6$	$\sigma, \gamma$	—	—	—	—	—

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