

# KONTEKSTNO-NEODVISNE GRAMATIKE ZA KODIRANJE IN STISKANJE PODATKOV

JANEZ PODLOGAR

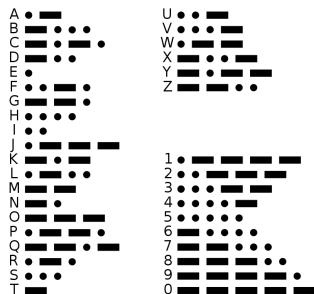
POVZETEK. V delu podamo definicije kodiranja, dekodiranja in stiskanja podatkov ter predstavimo primere, ki motivirajo stiskanje podatkov s kontekstno-neodvisnimi gramatikami.

## 1. KODIRANJE PODATKOV

Zapis informacije v neki obliki ni primeren za vsakršno rabo. Besedilo, zapisano z pismenkami, je neberljivo za slepe osebe, saj je komunikacijski kanal v tem primeru vid. Prav tako pisanega besedila v prvotni obliki ni mogoče poslati s telegrafom. V tem primeru je komunikacijski kanal žica in pismenke se po njej ne morejo sprehoditi. V obeh primerih je informacija, ki bi jo radi prenesli, zapisana v neprimerni obliki. V prvem primeru je potrebno besedilo zapisati z Braillovo pisavo. V drugem primeru pa je potrebno besedilo pretvoriti v električni signal. Spreminjanje zapisa sporočila imenujemo *kodiranje*, sistemu pravil, po katerem se kodiranje opravi, pa *kod*.

**Primer 1.1.** *Morsejeva abeceda* je kodiranje črk, števil in ločil s pomočjo zaporedja kratkih in dolgih signalov:

- Dolžina kratkega signala je ena enota.
- Dolgi signal je trikrat daljši od kratkega signala.
- Razmik med signali znotraj črke je tišina dolžine kratkega signala.
- Razmik med črkami je tišina dolga tri kratke signale oziroma en dolgi signal.
- Presledek med besedami je tišina dolga sedmih kratkih signalov.



SLIKA 1. Mednarodna Morsejeva abeceda

Prvotni namen Morsejeve abecede je komunikacija preko telegrama, saj komunikacijski kanal dovoljuje le električne signale in tišino med njimi. Kodiranje črk je takšno, da imajo črke z višjo frekvenco (v angleškem jeziku) krajši zapis. Tako se dolžina kodiranega sporočila skrajša in posledično tudi čas njegovega prenosa.

◇

$$\Sigma^* = \{a_1 a_2 a_3 \cdots a_n \mid n \in \mathbb{N}_0 \wedge \forall i : a_i \in \Sigma\},$$

**Opomba 1.3.** *Kleenejeva zvezdica* ali *Kleenejevo zaprtje* je operacija, ki abecedi  $\Sigma$  priredi najmanjšo podmnožico  $\Sigma^*$ , ki vsebuje *prazen niz*  $\varepsilon$  in je zaprta za konkatenacijo oziroma veriženje. Z drugimi besedami,  $\Sigma^*$  je množica vseh končnih nizov, ki jih lahko generiramo z veriženjem črk abecede  $\Sigma$ .

$$\Sigma^0 = \{\varepsilon\}, \quad \Sigma^1 = \Sigma$$
$$\Sigma^{i+1} = \{wa \mid w \in \Sigma^i \text{ in } a \in \Sigma\}.$$
$$\Sigma^* = \bigcup_{i \geq 0} \Sigma^i.$$

$$\forall w \in \Sigma^*: \kappa^{-1}(\kappa(w)) = w.$$
$$\Sigma = \{A, B, \dots, Z\} \cup \{0, 1, \dots, 9\} \cup \{\_ \}, \quad \Sigma_c = \{., -, \_ \}.$$
$$\kappa(\underline{\quad}) = \underline{\quad}$$
$$K(w) = \kappa(a_1)_{-} \kappa(a_2)_{-} \cdots \kappa(a_n)_{-}.$$
$$K(\text{SOS}) = \cdots - - - \cdots,$$

$$K(\text{AD HOC}) = \cdot - \quad - \cdot \quad \quad \cdot \dots \quad - - - \quad - \cdot - \cdot$$

$$K^{-1}(\text{---} \cdot \text{---} \cdot \text{---} \cdot \text{---}) = \text{QED},$$

— — • — — • — • •



## 2. STISKANJE PODATKOV

Eden izmed namenov kodiranja je tudi doseči čim večjo ekonomičnost zapisa. Želimo, da bi bilo naše sporočilo čim krajše. Kodiranje, ki skrajša zapis podatkov, imenujemo *stiskanje podatkov*.

**Definicija 2.1.** *Stiskanje* je kodiranje  $K$  za katerega velja

$$\exists n \in \mathbb{N} \forall w \in \Sigma^* : |w| \geq n \implies |K(w)| \ll |w|.$$

**Opomba 2.2.** Ločimo *kodiranje brez izgube*, kjer velja

$$\forall w \in \Sigma^* : \kappa^{-1}(\kappa(w)) = w$$

in *kodiranje z izgubo*, kjer kodiranje ni reverzibilen proces in v grobem velja

$$\forall w \in \Sigma^* : \kappa^{-1}(\kappa(w)) \approx w.$$

**Primer 2.3.** Za abecedo vzemimo  $\Sigma = \{a, b, c\}$  in pogledimo niz

$$w = cababcccababcccab.$$

Opazimo, da se nam v nizu  $w$  večkrat ponovita vzorca  $ab$  in  $ccc$ . Zato uvedemo novi oznaki  $A = ab$  in  $B = ccc$ . Sedaj lahko zapišemo  $w$  kot

$$w = cAABAABA.$$

Ponovno se nam pojavi vzorec, tokrat  $AAB$ . Uvedemo novo oznako  $C = AAB$  in zapišemo  $w$  kot

$$w = cCCA.$$

Prvotni niz smo z novimi oznakami skrajšali. Kot bomo videli, smo pretvorili niz  $w$  v kontekstno neodvisno gramatiko  $G_w$  s produkcijskimi pravili

$$S \rightarrow cCCA,$$

$$A \rightarrow ab,$$

$$B \rightarrow ccc,$$

$$C \rightarrow AAB.$$

◇

## 3. KONTEKSTNO-NEODVISNE GRAMATIKE

**Definicija 3.1.** *Formalna gramatika*  $G$  so pravila, ki nam iz abecede  $\Sigma$  tvorijo jezik, označimo ga z  $L(G)$ .

**Definicija 3.2.** *Kontekstno-neodvisna gramatika* je četverica  $G = (V, \Sigma, P, S)$ , kjer je  $V$  končna množica *spremenljivk*, abeceda  $\Sigma$  množica *končnih simbolov* tako, da  $\Sigma \cap V = \emptyset$ ,  $P \subseteq V \times (V \cup \Sigma)^*$  relacija, ki ji pravimo *produkcijsko pravilo* in  $S \in V$  *začetna spremenljivka*.

**Definicija 3.3.** Naj bo  $G = (V, \Sigma, P, S)$  kontekstno-neodvisna gramatika. Naj bodo  $\alpha, \beta, \gamma \in (V \cup \Sigma)^*$  nizi spremenljivk in končnih simbolov,  $A \in V$  spremenljivka ter naj bo  $(A, \beta) \in P$  produkcijsko pravilo, označimo ga z  $A \rightarrow \beta$ . Pravimo, da se  $\alpha A \gamma$  *prepiše s pravilom*  $A$  v  $\alpha \beta \gamma$ , pišemo  $\alpha A \gamma \Rightarrow \alpha \beta \gamma$ . Pravimo, da  $\alpha$  *porodi*  $\beta$ , če je  $\alpha = \beta$  ali če za  $k \geq 0$  obstaja zaporedje  $\alpha_1, \alpha_2, \dots, \alpha_n \in (V \cup \Sigma)^*$  tako, da

$$\alpha \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n \Rightarrow \beta$$

in pišemo  $\alpha \xRightarrow{*} \beta$ .

**Posledica 3.4.** *Jezik kontekstno neodvisne gramatike*  $G$  je

$$L(G) = \{w \in \Sigma^* \mid S \xRightarrow{*} w\}.$$

**Opomba 3.5.** Ime kontekstno-neodvisna gramatika izvira iz oblike produkcijskih pravil. Na levi strani produkcijskega pravila mora vedno stati samo spremenljivka. Torej vsebuje samo pravila oblike

$$A \rightarrow \alpha,$$

kjer je  $A \in V$  in  $\alpha \in (V \cup \Sigma)^*$ . Ne sme pa vsebovati pravila oblike

$$\alpha A \gamma \rightarrow \alpha \beta \gamma,$$

kjer je  $A \in V$  in so  $\alpha, \beta, \gamma \in (V \cup \Sigma)^*$ , saj je pravilo odvisno od predhodnega konteksta. Odvisno je od tega ali pred njim stoji  $\alpha$  in za njim  $\beta$ .

**Primer 3.6.** Formalizirajmo gramatiko iz Primera 2.3, ki smo jo generirali z nizom  $w = cababcccababcccab$ . Označimo jo z  $G_w = (V, \Sigma, P, S)$ , kjer je

$$\begin{aligned} V &= \{S, A, B, C\}, \\ \Sigma &= \{a, b, c\}, \\ P &= \{S \rightarrow cCCA, A \rightarrow ab, B \rightarrow ccc, C \rightarrow AAB\}, \\ S &= S. \end{aligned}$$

Vidimo, da  $G_w$  ustreza naši definiciji kontekstno-neodvisne gramatike in res kodira  $w$ , saj je

$$L(G_w) = \{w\}.$$

Dolžina gramatike  $G_w$  je enaka številu črk abecede  $\{S, A, B, C, \rightarrow\}$ , ki smo jih porabili za opis gramatike in je enaka  $|P| = 20$ . Vidimo, da niza  $w$  nismo skrajšali, saj je  $|w| = 17$ . Niz  $w$  je bil prekratek, da bi ga lahko zares stisnili. Naj bo sedaj

$$w = cababcccababcccabababccc.$$

Gramatika, ki jo generira novi niz se od prejšnje gramatike razlikuje le v

$$P = \{S \rightarrow cCCAC, A \rightarrow ab, B \rightarrow ccc, C \rightarrow AAB\}.$$

Sedaj smo stisnili  $w$  z  $G_w$ , saj je

$$|w| = 24 > 21 = |P|.$$

◇

Za razliko od klasičnih metod stiskanja, naša metoda ne izvaja procesa stiskanja direktno na nizu  $w$ , temveč poiščemo gramatiko  $G_w$ , ki generira enojec  $\{w\}$  za svoj jezik. Med njimi poiščemo “najmanjšo”, ki jo nato kodiramo. Ker gramatike  $G_w$  generira  $w$  in ker je gramatika “majhna” jo bomo kodirali v “kratko” kodo. Tako bomo preko gramatike “dobro” stisnili niz  $w$ .