UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika – 1. stopnja

Janez Podlogar

KONTEKSTNO-NEODVISNE GRAMATIKE ZA KODIRANJE IN STISKANJE PODATKOV

Delo diplomskega seminarja

Mentor: prof. dr. Ljupčo Todorovski

Ljubljana, 2024

Kazalo

Kontekstno-neodvisne gramatike za kodiranje in stiskanje podatkov

Povzetek

Definiramo podrazred kontekstno-neodvisnih gramatik imenovan dopustne gramatike. Nizu w iz abecede priredimo dopustno gramatiko G_w, katere jezik je $\{w\}$. Predstavimo dva razreda prirejanj dopustne gramatike nizu in za vsak razred podamo primer prirejanja. Binarno kodiranje prepisovalnih pravil dopustne gramatike G_w skupaj s predstavljenim prirejanjem zagotavlja dobro zgornjo mejo odvečnosti. Takšno stiskanje podatkov je univerzalen kod.

Context-Free Grammars for Data Encoding and Compression

Abstract

We define a subclass of context-free grammars, called admissable grammars. For each string w of an alphabet, we assign an admissable grammar G_w, such that its language is $\{w\}$. We introduce two classes of assigning an admissable grammars to a string and provide an example for each class. Binary encoding of the production rules of an admissable grammar G_w with the underlying admissable grammar assignment results in good redundancy bound. Such data Compression constitutes a universal code.

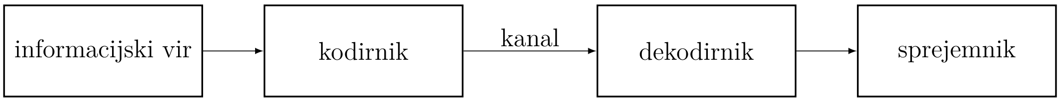
Math. Subj. Class. (2020): 68P30, 68Q42, 94A15

Ključne besede: kontekstno-neodvisna gramatika, stiskanje podatkov, teorija informacij, stiskanje brez izgube, univerzalen kod

Keywords: context-free grammar, data compression, information theory, lossless compression, universal code

1 Uvod

Temeljni problem sporazumevanja je prenos sporočila od informacijskega vira do sprejemnika. Privzemimo komunikacijski model prikazan na spodnji sliki, ki je prirejna po [14].



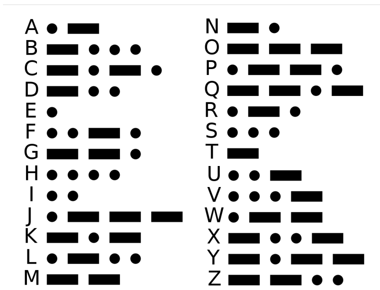
Slika 1: Komunikacijski model

Informacijski vir izbere želeno sporočilo iz množice vseh možnih sporočil. *Kodirnik* pretvori sporočilo v primereno obliko za prenos po *kanalu* do *dekodirnika*, ki sporočilo pretvori v primerno obliko za *sprejemnik*. Spreminjanje zapisa sporočila imenujemo *kodiranje*, sistemu pravil, po katerem se kodiranje opravi, pa *kod*.

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. Najpomembnejši namen kodirnika je pretvorba sporočila v obliko, ki jo lahko pošljemo po kanalu. V opisanih primerih je sporočilo, ki bi ga radi prenesli, zapisana v neprimerni obliki. V primeru slepe osebe je potrebno besedilo zapisati z Braillovo pisavo, v primeru telegrafa pa je besedilo potrebno pretvoriti v električni signal, kot ilustrira naslednji primer.

Primer 1.1. *Morsejeva abeceda* je kodiranje črk, številk in ločil s pomočjo zaporedja kratkih in dolgih signalov. Določajo jo pravila:

- 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.
- Med besedami je tišina, dolga sedem kratkih signalov.



Slika 2: Mednarodna Morsejeva abeceda, vzeta iz [15].

Namen Morsejeve abecede je komunikacija preko telegrafa, saj komunikacijski kanal dovoljuje le električne signale in tišino med njimi. ♢

Od kodirnika zahtevamo več kot le pretvorbo v ustrezno obliko za prenos po kanalu. Če imamo nezanesljiv kanal, se med prenosom po kanalu lahko pojavijo napake v sporočilu. Kod za popravljanje napak omogoča, da za ceno daljšega sporočila popravimo napake, ki se pojavijo ob motnjah pri prenosu. Morda pa tudi nekoga tretjega zanima vsebina našega sporočila. Tedaj sporočilo kodiramo tako, da ga lahko dekodirajo le pooblaščene osebe. Takšnemu kodiranju pravimo šifriranje. Šifriranje ne preprečuje dostopa do kodiranega sporočila, ampak onemogoča pravilno dešifiranje oziroma razumevanje vsebine prestrezniku.

Eden izmed namenov kodiranja je tudi doseči jedernatost sporočila. Stiskanje podatkov je zapis informacij sporočila v zgoščeni obliki. Z uporabo lepih lastnosti infromacijskega vira in posameznega sporočila, lahko sporočilo učinkoviteje prenesemo in porabimo manj prostora. Ena od takih lastnosti je statistična struktura jezika. Uporabimo jo v Morsejevi abecedi, kjer imajo črke z višjo frekvenco (pojavitve v angleškem jeziku) krajši zapis.

V delu bomo preučevali metode kodiranja, ki izkoriščajo prisotnost ponavljajočih se vzorcev v sporočilu. Zanimalo nas bo stiskanje preko *gramatike*, več o drugih metodah pa najdemo v [13]. Gramatika ali slovnica je nabor pravil, ki jih mora stavek upoštevati, da je "pravilen". Pri stiskanju podatkov nas zanimajo *formalne gramatike*, ki jih razumemo kot nabor pravil za generiranje zaporedja črk. Zaporedje črk želimo stisniti preko zgoščenega nabora pravil, ki generirajo dano zaporedje.

Primer 1.2. Poglejmo niz $w = cababcccababcccab$. Opazimo, da se v nizu ponovita vzorca ab in ccc. Uvedemo novi oznaki $A = ab$ in $B = ccc$. Sedaj zapišemo niz w kot

$$w = cAABAABA.$$

Uvedemo novo oznako $C = AAB$ in zapišemo w kot

$$w = cCCA.$$

Prvotni niz smo z novimi oznakami skrajšali. Kot bomo videli, smo niz w pretvorili v formalno gramatiko s pravili

$$\begin{aligned} S &\to cCCA, \\ A &\to ab, \\ B &\to ccc, \\ C &\to AAB. \end{aligned}$$

$\diamondsuit$

2 Osnovni pojmi

2.1 Jezik na abecedi

Gradnik besedila je abeceda. To je množica črk, ki ponavadi predstavljajo zvoke v govorjenem jeziku. Za nas bo abeceda množica veljavnih črk jezika.

Definicija 2.1. *Abeceda* je končna neprazna množica. Elementom abecede pravimo *črke.* Za vsak $\ell > 0$ rekurzivno definiramo *množico vseh nizov abecede* $\mathcal{A}$ *dolžine* $\ell + 1$

$$\mathcal{A}^0 = \{\varepsilon\},$$
$$\mathcal{A}^{\ell+1} = \{wa \mid w \in \mathcal{A}^\ell \text{ in } a \in \mathcal{A}\},$$

kjer ε imenujemo *prazen niz. Množica vseh končnih nizov abecede* $\mathcal{A}$ je

$$\mathcal{A}^* = \bigcup_{\ell \geq 0} \mathcal{A}^\ell$$

in *množica vseh končnih nizov abecede* $\mathcal{A}$ *brez praznega niza* je

$$\mathcal{A}^+ = \mathcal{A}^* \setminus \{\varepsilon\}.$$

Jezik na abecedi $\mathcal{A}$ je poljubna podmnožica množice $\mathcal{A}^*$. *Dolžino niza* w označimo z $|w|$ in je enaka številu črk v nizu $w \in \mathcal{A}^*$.

Definicija 2.2. Naj bo $\mathcal{A}$ abeceda in $*$ binarna operacija na $\mathcal{A}^*$ za katero je ε nevtralni element in za niza $w, u \in \mathcal{A}^*$ velja

$$w * u = w_1 w_2 \cdots w_n u_1 u_2 \cdots u_m,$$

kjer sta $w_1 w_2 \cdots w_n$ in $u_1 u_2 \cdots u_m$ predstavitvi nizov w in u s črkami abecede $\mathcal{A}$. Operacijo $*$ imenujemo *stikanje.* Simbol $*$ spustimo in krajše pišemo wu.

Opomba 2.3. Stikanje je asociativna operacija. Torej je $(\mathcal{A}^*, *)$ monoid in $(\mathcal{A}^+, *)$ polgrupa.

Definicija 2.4. Naj bo $w \in \mathcal{A}^*$ in $w = w_1 w_2 \cdots w_n$ predstavitev niza w s črkami abecede $\mathcal{A}$. *Frekvenca črke* s *v nizu* w je

$$f(s|w) = \left|\left\{i \in \{1, 2 \cdots, n\} \mid w_i = s\right\}\right|.$$

Definicija 2.5. Niz $u \in \mathcal{A}^+$ je *predpona* niza $w \in \mathcal{A}^+$, če $\exists v \in \mathcal{A}^*$, da je $uv = w$.

Primer 2.6. Za abecedo $\mathcal{A} = \{a, b, c\}$ so $ab \in \mathcal{A}^2$, $abccc \in \mathcal{A}^5$, $cababcccababcccab \in \mathcal{A}^{17}$ končni nizi abecede $\mathcal{A}$ in ab je predpona $abccc$. ♢

2.2 Kodiranje sporočila

Kodiranje sporočila je spreminjanje njegovega zapisa po sistemu pravil, ki ga imenujemo kod. Predstavimo dve kodiranji, ki ju bomo uporabili pri dokazu binarnega kodiranja dopustne gramatike 4.27.

Definicija 2.7. *Kodna preslikava* je injektivna preslikava $\kappa\colon \mathcal{A}_s^* \to \mathcal{A}_c^*$, kjer imenujemo $\mathcal{A}_s$ *izvorna abeceda*, $\mathcal{A}_c$ *kodna abeceda* in $\kappa(w)$ *koda niza* w. *Dekodna preslikava* je preslikava $\delta\colon C \subseteq \mathcal{A}_c^* \to \mathcal{A}_s^*$, da velja

$$\forall w \in \mathcal{A}_s^*\colon \delta(\kappa(w)) = w.$$

Primer 2.8. Formalizirajmo Morsejevo abecedo iz primera 1.1. Abecedi sta

$$\mathcal{A}_s = \{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}\} \cup \{\text{␣}\}, \quad \mathcal{A}_c = \{\cdot, -, \square\},$$

kjer je ␣ presledek, $\cdot$ kratki signal, $-$ dolgi signal in $\square$ kratka enota tišine. Definiramo kodno preslikavo črk abecede $\kappa_s \colon \mathcal{A}_s \to \mathcal{A}_c^*$, ki vsaki črki iz abecede $\mathcal{A}_s$ priredi niz črk kodne abecede $\mathcal{A}_c$. Predpis preslikave κ_s je določen na sliki 2. Presledek ␣ kodiramo v eno kratko enoto tišine

$$\kappa_s(\text{␣}) = \square.$$

Za niz $w = a_1 a_2 \ldots, a_n \in \mathcal{A}_s^*$ kodno preslikavo κ definiramo po črkah

$$\kappa(w) = \kappa_s(a_1)\square\square\square\kappa_s(a_2)\square\square\square \cdots \square\square\square\kappa_s(a_n).$$

Poglejmo dve kodiranji sporočil

$$\begin{aligned} \kappa(\mathrm{SOS}) &= \ \cdot\square\cdot\square\cdot\square\square\square - \square - \square - \square\square\square\cdot\square\cdot\square\cdot\,, \\ \kappa(\mathrm{ET\text{␣}TU}) &= \ \cdot\square\square\square - \square\square\square\square\square\square\square - \square\square\square\cdot\square\cdot\square - . \end{aligned}$$

Recimo, da smo prejeli sporočilo, a se je pošiljatelj zmotil in je namesto kode, ki bi se dekodirala v

$$\delta(\, - \square - \square \cdot \square - \square\square\square \cdot \square\square\square - \square \cdot \square \cdot \,) = \mathrm{QED},$$

poslal kodo

$$- \square - \square \cdot \square - \square - \square\square\square \cdot \square\square\square - \square \cdot \square \cdot .$$

Sporočila ne znamo dekodirati, saj se ne nahaja v domeni C dekodne preslikave δ. $\diamondsuit$

Definicija 2.9. *Stiskanje* je kodna preslikava κ za katero velja

$$\exists n \in \mathbb{N}\ \forall w \in \mathcal{A}^* \colon\ |w| \geq n \implies |\kappa(w)| \ll |w|\,.$$

Razmerju $\frac{|\kappa(w)|}{|w|}$ pravimo *razmerje stisljivosti.* Kodo $\kappa(w)$ imenujemo *stisnjen niz* w in $\delta(\kappa(w))$ *rekonstrukcija niza* w.

Stiskanje podatkov razdelimo v dve kategoriji. *Stiskanje brez izgube*, ki omogoča natančno rekonstrukcijo izvirnega sporočila iz stisnjenih podatkov, in *stiskanje z izgubo*, za katero je značilna nepovratna izguba informacije.

Definicija 2.10. Stiskanje je *brez izgube*, če velja

$$\forall w \in \mathcal{A}^* \colon \delta(\kappa(w)) = w.$$

Stiskanje je *z izgubo*, če kodna preslikava nima levega inverza, a velja

$$\forall w \in \mathcal{A}^* \colon \delta(\kappa(w)) \doteq w.$$

Stiskanje brez izgube uporabljamo pri stiskanju besedl, saj je pomembno da je rekonstrukcija besedila enaka izvirnemu besedilu. Majhne razlike med rekonstrukcijo in izvirnim besedilom lahko povzročijo velike pomenske razlike. Primer tega so bančni zapiski.

V nekaterih primerih pa lahko toleriramo izgubo informacije. Na primer, pri zvočnih posnetkih, slikah in videoposnetkih je lahko rekonstrukcija drugačna od izvirnika, saj so razlike za človeka neopazne. V zameno za izgubo informacije dosežemo boljšo razmerje stisljivosti kot pri stiskanju brez izgube.

Spoznajmo dve kodiranji, ki ju bomo uporabili v dokazu 4.27.

2.2.1 Eniško kodiranje

Definicija 2.11. *Eniško kodiranje naravnih števil* je kodna preslikava

$$\begin{aligned} \eta\colon \mathbb{N} &\to \{0,1\}^+, \\ n &\mapsto \underbrace{0\cdots 0}_{n-1}1. \end{aligned}$$

Primer 2.12. Naj bo η eniška kodna preslikava. Potem je

$$\begin{aligned} \eta(1) &= 1, \\ \eta(2) &= 01, \\ \eta(9) &= 000000001. \end{aligned}$$

$\diamondsuit$

2.2.2 Leksikografsko kodiranje

Definicija 2.13. Naj bo $S \subseteq \{1,2,\ldots,M\}^n$ za in $w_1, w_2, \ldots, w_k \in \{1,2,\ldots,M\}$. Z $n_s(w_1w_2\cdots w_k)$ označimo *število nizov v S za katere je $w_1w_2\cdots w_k$ predpona.*

Definicija 2.14. Naj bo $S \subseteq \{1,2,\ldots,M\}^n$. *Leksikografksi indeks S* je preslikava

$$i_s\colon S \to \{0,1\ldots,n-1\},$$

da je $i_s(w_1w_2\cdots w_n) < i_s(u_1u_2\cdots u_n)$ natanko takrat, ko za najmanjši tak k, da je $w_k \neq u_k$, velja $w_k < u_k$.

Primer 2.15. Naj bo $S = \{1101, 1111, 1100, 0101, 0110, 0001\}$. Velja

$$0001 < 0101 < 0110 < 1100 < 1101 < 1111.$$

in $n_s(0) = 3$, $n_s(10) = 0$, $n_s(110) = 2$, $n_s(1100) = 1$ $\diamondsuit$

Trditev 2.16. *Leksikografski indeks $S \subseteq \{0,1\}^n$ je podan s predpisom*

$$i_s(w) = \sum_{j=1}^{n} w_j \cdot n_s(w_1w_2\cdots w_{j-1}0).$$

Sledeči algoritem je inverz funkcije i_s. Torej za $i \in \{0,1,\ldots,n-1\}$ poišče tak $w \in S$, da je $i_s(w) = i$. Za $j = 1,2,\ldots,n$ naredi: Če je $i > n_s(w_1w_2\cdots w_{j-1}0)$, je $x_j = 1$ in nastavi $i \coloneqq i - n_s(w_1w_2\cdots w_{j-1}0)$; sicer je $x_j = 0$.

Dokaz. Velja $w_1w_2\cdots w_{j-1}0 < w_1w_2\cdots w_{j-1}1$. Za vsak j takšen, da je $w_j = 1$, preštejemo število nizov v S, ki se prvič razlikujejo od niza w na j-tem mestu. To so nizi, ki imajo manjši leksikografski indeks od w. Število teh nizov je po definiciji enako številu $n_s(w_1w_2\cdots w_{j-1}0)$.

Ko seštejemo $n_s(w_1w_2\cdots w_{j-1}0)$ za vsak $j = 1,2,\cdots,n$, preštejemo vse elemente S, katerih leksikografski indeks je manjši od leksikografskega indeksa od w. $\square$

Primer 2.17. Naj bo $S = \{1101, 1111, 1100, 0101, 0110, 0001\}$ kot v primeru 2.15. Potem je

$$i_s(1101) = 1 \cdot n_s(0) + 1 \cdot n_s(10) + 0 \cdot n_s(110) + 1 \cdot n_s(1100) = 4$$

Poglejmo si inverzni algoritem. Naj bo $i = 4$. Poiščimo tak $w \in S$, da je $i_s(w) = 4$.

- $j = 1$: $i = 4 > 3 = n_s(0) \implies x_1 = 1$ in $i \coloneqq 4 - 3 = 1$;
- $j = 2$: $i = 1 > 0 = n_s(10) \implies x_2 = 1$ in $i \coloneqq 1 - 0 = 1$;
- $j = 3$: $i = 1 < 2 = n_s(110) \implies x_3 = 0$;
- $j = 4$: $i = 1 > 1 = n_s(1100) \implies x_4 = 1$ in $i \coloneqq 1 - 1 = 0$.

Torej je $i_s(1101) = 4$. ♢

Zgornjo trditev razširimo na poljubno abecedo $\{1, 2, \ldots, M\}$. Dokaz spustimo, saj je skoraj enak dokazu prejšnje trditve.

Trditev 2.18. *Leksikografski indeks od $w \in S \subseteq \{1, 2, \ldots, M\}^n$ je podan s predpisom*

$$i_s(w) = \sum_{j=1}^{n} \sum_{k=1}^{w_j - 1} n_s(w_1 w_2 \cdots w_{j-1} k).$$

Sledeči algoritem je inverz funkcije i_s. Torej za $i \in \{0, 1, \ldots, n-1\}$ poišče tak $w \in S$, da je $i_s(w) = i$. Za $j = 1, 2, \ldots, n$ naredi: Poišči najmanjši $m \in \{1, 2, \ldots, M\}$, da je

$$i < \sum_{k=1}^{m} n_s(w_1 w_2 \cdots w_{j-1} k),$$

potem je $x_j = m$ in in nastavi $i \coloneqq i - \sum_{k=1}^{m-1} n_s(w_1 w_2 \cdots w_{j-1} k)$.

Formulo za leksikografski indeks lahko posplošimo na poljubno *linearno urejeno* abecedo. Za abecedo $\mathcal{A}$ velikosti M podamo bijektivno preslikavo ξ_s v $\{1, 2, \ldots, M\}$ in z njo *induciramo linearno urejenost na $\mathcal{A}$* tako, da za $a, b \in \mathcal{A}$ velja

$$a < b \iff \xi_s(a) < \xi_s(b).$$

Definicija 2.19. Naj bo $\mathcal{A}$ abeceda velikosti n. *Abecedni vrstni red $\mathcal{A}$*, je zaporedje črk $a_1, a_2, \ldots, a_n \in \mathcal{A}$, da je $a_1 < a_2 < \ldots < a_n$.

Primer 2.20. Naj bo $\mathcal{A} = \{a, b, c, d, e\}$ in $\xi_s(a) = 1$, $\xi_s(b) = 2$, $\xi_s(c) = 3$, $\xi_s(d) = 4$, $\xi_s(e) = 5$. Potem je abecedni vrstni red $a < b < c < d < e$, saj je $\xi_s(a) < \xi_s(b) < \xi_s(c) < \xi_s(d) < \xi_s(e)$. Niz $abaabbc \in \mathcal{A}$ preslikamo v niz abecede $\{1, 2, 3, 4, 5\}$ po črkah

$$abaabbc \mapsto 1211223.$$

♢

Trditev 2.21. *Naj bo $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$ in $c_1, c_2, \ldots, c_M \in \mathbb{N}$. Naj bo*

$$S = \Big\{u \in \mathcal{A}^* \mid \forall i = 1, 2, \ldots, M \colon f(a_i|u) = c_i\Big\}$$

množica vseh nizov abecede $\mathcal{A}$, kjer se za $i = 1, 2, \ldots, M$ črka a_i pojavi c_i-krat. Potem je

$$n_s(w_1 w_2 \cdots w_{j-1} k) = \begin{cases} \binom{n-j}{r_1, r_2, \ldots, r_k} & ; \text{ če je } w_1 w_2 \cdots w_{j-1} k \in S \\ 0 & ; \text{ sicer} \end{cases},$$

kjer je $n = \sum_{i=1}^{M} c_i$ in $r_i = c_i - f(a_i | w_1 w_2 \cdots w_{j-1} k)$ za $i = 1, 2, \ldots, M$.

Dokaz. Vrednost $n - j$ določi število prostih mest po predponi. Z r_i označimo kolikokrat se mora v_i pojaviti po predponi. Multinomski koeficient pove na koliko načinov lahko uredimo preostale črke. Če $w_1 w_2 \cdots w_{j-1} k \notin S$, potem ni predpona nobenega niza iz S. □

Definicija 2.22. Naj bo $w \in \mathcal{A}^*$ in $\{a_1, a_2, \ldots, a_k\} \subseteq \mathcal{A}$ množica črk, ki nastopajo v w. Definiramo *podmnožico vseh nizov abecede $\mathcal{A}$, ki ima enako frekvenco znakov kot niz w*

$$S(w) = \Big\{u \in \mathcal{A}^* \mid \forall i = 1, 2, \ldots, k \colon f(a_i|u) = f(a_i|w)\Big\}.$$

Leksikografsko kodiranje niza w je kodna preslikava

$$\begin{aligned} \lambda \colon \mathcal{A} &\to \{0,1\}^*, \\ w &\mapsto B_1 B_2 B_3, \end{aligned}$$

kjer je:

- B_1 koda znakov abecede $\mathcal{A}$, ki nastopajo v nizu w. To je niz, kjer za vsak element iz $\mathcal{A}$ v abecednem vrstnem redu z enico označimo ali je vsebovan v $\{a_1, a_2, \ldots, a_k\}$ in z ničlo, če ni;
- B_2 eniška koda vsake frekvenc $f(a_i|w)$ v enakem redu kot nastopajo črke v B_1;
- B_3 binarni zapis $i_{s(w)}(w)$.

Abeceda $\mathcal{A}$ in njen abecedeni vrstni red sta poznana tako kodirniku kot dekodirniku.

Primer 2.23. Naj bo $\mathcal{A} = \{1, 2, 3, 4, 5\}$. Leksikografsko kodiramo $w = 1211223$. Dobimo $B_1 = 11100$, saj črke $1, 2, 3$ nastopajo v w, medtem ko $4, 5$ ne. Ker je

$$\eta\left(f(1|w)\right) = 001,\ \eta\left(f(2|w)\right) = 001,\ \eta\left(f(3|w)\right) = 1,$$

sledi, $B_2 = 0010011$. Izračun

$$\log_2(|S(w)|) = \log_2\left(\binom{7}{3, 3, 1}\right) = \log_2(140) \doteq 7.13,$$

pove, da za zapis poljubnega indeksa niza iz S potrebujemo najmanj 8 bitov. Indeks od w je

$$\begin{aligned} i_{s(w)}(1211223) &= n_{s(w)}(11) + n_{s(w)}(12111) + n_{s(w)}(121121) \\ &\quad + n_{s(w)}(1211221) + n_{s(w)}(1211222) \\ &= 20, \end{aligned}$$

saj je

$$n_{s(w)}(11) = \binom{5}{1, 3, 1} = \frac{5!}{1! \cdot 3! \cdot 1!} = 20,$$
$$n_{s(w)}(12111) = n_{s(w)}(121121) = n_{s(w)}(1211221) = n_{s(w)}(1211222) = 0.$$

Torej je $B_3 = 00010100$. Zaključimo, da je

$$\lambda(1211223) = 1110000010011000101 00.$$

$\diamondsuit$

Primer 2.24. Naj bo $\mathcal{A} = \{a, b, c, d, e\}$ in $\xi_s(a) = 1$, $\xi_s(b) = 2$, $\xi_s(c) = 3$, $\xi_s(d) = 4$, $\xi_s(e) = 5$. Leksikografsko kodiramo niz $w = abaabbc \in \mathcal{A}^*$. Ker imamo bijektivno preslikavo med abecedo $\mathcal{A}$ in abecedo $\{1, 2, 3, 4, 5\}$, imamo bijektino preslikavo med $\mathcal{A}^+$ in $\{1, 2, 3, 4, 5\}^+$. Niz w po črkah z ξ_s preslikamo v niz $1211223 \in \{1, 2, 3, 4, 5\}^*$. Niz *abaabbc* je le preimenovanje niza 1211223, torej velja

$$\lambda(abaabbc) = \lambda(1211223).$$

$\diamondsuit$

Več o kodiranju nizov iz $S \subseteq \{1, 2, \ldots, M\}^n$, ki uporabljajo strukturo S, najdemo v [4].

2.3 Teorija informacij

Teorija informacij preučuje prenos, obdelavo, pridobivanje in uporabo informacij. Temelje področja je postavil Claude Shannon v [14]. Koncept informacije je preširok, da bi ga lahko zajeli z eno samo definicijo. V razdelku predstavimo dva pogosta modela informacijskega vira in definiramo *entropijo*. Entropija ima številne lastnosti, ki se strinjajo z intuitivnim merilom količine informacij.

Recimo, da imamo dogodek A in $\mathbb{P}(A)$ verjetnost, da se ta dogodek zgodi. Koliko informacij dobimo, če se ta dogodek zgodi? Intuitivno bi rekli, da:

- je gotov dogodek popolnoma pričakovan in ne nosi nobene informacije;
- manj verjetni kot je dogodek, bolj presenetljiv je in nosi več informacij;
- je informacija neodvisnih dogodkov enaka vsoti informacij dogodkov.

Da se preveriti, da spodnja definicija informacije dogodka izpolnjuje vse tri intuitivne zahteve.

Definicija 2.25. Naj bo X diskretna slučajna spremeljivka. Označimo $p(x) = \mathbb{P}(X = x)$. *Shannonova informacija dogodka* A je

$$i(A) = -\log_b p(x),$$

kjer je $b > 1$. V najpogostejšem primer, ko je $b = 2$, merimo Shannonovo informacijo v *bitih*.

Primer 2.26. Mečemo pošten kovanec. Z G označimo dogodek da pade glava in z C dogodek da pade cifra. Smiselno je, da oba dogodka nosita enako informacij.

$$i(G) = i(C) = 1 \text{ bit}.$$

Recimo, da kovanec ni pošten in, da je $p(G) = \frac{1}{8}$ ter $p(C) = \frac{7}{8}$. Potem ima dogodek, da je padla glava, več informacij.

$$\begin{aligned} i(G) &= -\log_2\left(\frac{1}{8}\right) = 8 \text{ bitov}, \\ i(C) &= -\log_2\left(\frac{7}{8}\right) \doteq 0.2 \text{ bitov}. \end{aligned}$$

$\diamondsuit$

Shannonova informacija je le alternativni način izražanja verjetnosti dogodka. Predstavljamo si jo kot mero "presenečenja", da se je dogodek zgodil. Malo verjetni dogodki so zelo presenetljivi in bodo zelo vplivali na naša dejanja, medtem ko nas skoraj gotovi dogodki ne presenetijo.

Primer 2.27. Verjetnost, da Andreja zadane na loteriji, je ena proti milijon. Če njen mož Bojan izve, da je zadela na loteriji, bo prejel več informacij kot če izve, da ni zadela.

$$\begin{aligned} i(\text{Andreja zadane}) &= -\log_2(0.000001) \doteq 20 \text{ bitov}, \\ i(\text{Andreja ne zadane}) &= -\log_2(0.999999) \doteq 1.4 \cdot 10^{-6} \text{ bitov}. \end{aligned}$$

Če Andreja ne zadane, bo Bojan nadaljeval z svojim dnem, kot da se ni nič zgodilo. Če pa Andreja zadane, se bo Bojanovo življenje zelo spremenilo. $\diamondsuit$

Entropija je pričakovana vrednost Shannonove informacije naključne spremenljivke in pove, kako presenetljiva je naključna spremenljivka "v povprečju".

Definicija 2.28. Naj bo X diskretna slučajna spremeljivka. *Entropija slučajne spremenljivke* X je

$$\begin{aligned} H(X) &= \sum_{x \in X} -p(x) \log_b p(x) \\ &= \sum_{x \in X} p(x) i(x) \\ &= \mathbb{E}[i(X)]. \end{aligned}$$

Definicija 2.29. *Informacijski vir abecede* $\mathcal{A}$ je preslikava $\mu\colon \mathcal{A}^+ \to [0,1]$, da je

$$\begin{aligned} \sum_{a \in \mathcal{A}} \mu(a) = 1,& \\ \mu(w) = \sum_{a \in \mathcal{A}} \mu(wa) &\text{ za vsak } w \in \mathcal{A}^+ \end{aligned}$$

Z $\Lambda(\mathcal{A})$ označimo družino vseh informacijskih virov abecede $\mathcal{A}$.

Definicija 2.30. Naj bo X diskretna slučajna spremeljivka z zalogo vrednosti v abecedi $\mathcal{A}$. *Vir X brez spomina abecede $\mathcal{A}$* je preslikava

$$\begin{aligned} \mu\colon \mathcal{A}^+ &\to [0,1] \\ w &\mapsto p(w_1)p(w_2)\cdots p(w_n), \end{aligned}$$

kjer je $w_1w_2\cdots w_n$ predstavitev niza w s črkami abecede $\mathcal{A}$. *Niz generiran z virom X brez spomina abecede $\mathcal{A}$ dolžine n* je niz, ki ga dobimo tako, da staknemo n realziacij diskretna slučajna spremeljivka X.

Trditev 2.31. *Vir X brez spomina abecede $\mathcal{A}$ je informacijski vir abecede $\mathcal{A}$.*

Dokaz.

$$\begin{aligned} \sum_{a\in\mathcal{A}} \mu(a) &= \sum_{a\in\mathcal{A}} p(a) = 1, \\ \mu(w) &= \mu(w)\cdot 1 \\ &= \mu(w)\cdot\left(\sum_{a\in\mathcal{A}} \mu(a)\right) \\ &= \sum_{a\in\mathcal{A}} \mu(w)\mu(a) \\ &= \sum_{a\in\mathcal{A}} p(w_1)p(w_2)\cdots p(w_n)p(a) \\ &= \sum_{a\in\mathcal{A}} \mu(wa). \end{aligned}$$

□

Vir brez spomina je najpreprostejši model informacijskega vira, saj je vsaka črka v nizu neodvisna od vseh ostalih. V splošnem ne poznamo porazdelitve slučajne spremenljivke, ki definira vir brez spomina, zato porazdelitev ocenimo. Slovenščino modeliramo z virom brez spomina tako, da porazdelitev ocenimo preko frekvenc črk v slovenskem besedilu.

Definicija 2.32. Imamo vir X brez spomina abecede $\mathcal{A}$, kjer ne poznamo porazdelitve slučajne spremenljivke X in w in w niz generiran z virom X brez spomina abecede $\mathcal{A}$ dolžine n. *Entropija niza w* je

$$H(w) = \sum_{s\in\{w_1,\dots,w_n\}} -\frac{f(s|w)}{|w|}\log_2\left(\frac{f(s|w)}{|w|}\right).$$

Primer 2.33. Niz $w = 1211223$ iz primera 2.23 ima entropijo

$$H(w) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{1}{7}\log_2\left(\frac{1}{7}\right) \doteq 1.45 \text{ bitov}.$$

♢

Sledeča lema, dokazana v [5, Lema 2.3], pojasni entropijo s kombinatoričnega vidika. Uporabimo jo v dokazu 4.27.

Lema 2.34. *Naj bo w niz abecede $\mathcal{A}$ dolžine n. Potem velja*

$$|S(w)| \leq 2^{n \cdot H(w)}.$$

Primer 2.35. Za niz $w = 1211223$ iz primera 2.23 velja ocena

$$140 = |S(w)| \leq 2^{7 \cdot H(w)} \doteq 1130.$$

$\diamondsuit$

Shannon v [14] postavi statistično spodnjo mejo za stiskanje vira brez izgube. Izrek je podan verbalno za vir brez spomina v [10, 1. poglavje, 4. razdelek] na sledeč način.

Izrek 2.36. *Niz generiran z virom X brez spomina $\mathcal{A}$ dolžine n, se lahko stisne z zanemarljivim tveganjem izgube informacij v $n \cdot H(X)$ bitov, ko gre $n \to \infty$. Če niz stisknemo v manj kot $n \cdot H(X)$ bitov, skoraj gotovo pride do izgube informacij.*

Primer 2.37. Imamo vir X brez spomina abecede $\mathbb{N}$, da velja $p(n) = 2^{-n}$ za $n \in \mathbb{N}$. Potem je eniško kodiranje po črkah optimalno v smislu zgornjega izreka, saj je v tedaj eniško kodiranje enako *Huffmanovemu kodiranju* [6]. Huffmanovo kodiranje je optimalno kodiranje po črkah pri znani verjetnostni porazdelitvi vira brez spomina.

$\diamondsuit$

Posledica izreka 2.36 je, da za optimalno stiskanje κ brez izgube X vira brez spomina, velja $\mathbb{E}[|\kappa(w)|] \geq H(X)$. Torej v povprečju velja $|\kappa(w)| = -\log_2 \mu(w)$. Več o tem najdemo v [10, 1. poglavje, 5. razdelek]. *Odvečnost* stiskanja κ je razlika med dolžino kodiranega niza in optimalno dolžino. Odvečnost torej pove kako blizu je κ optimalnemu stiskanju. Sledeča definicija odvečnosti meri kako blizu smo v najslabšem primeru glede na dolžino niza.

Definicija 2.38. *Maksimalna točkovna odvečnost reda n kodne preslikave κ glede na družino informacijskih virov abecede $\mathcal{A}$* je

$$\mathrm{odv}_n\left(\kappa, \Lambda(\mathcal{A})\right) = n^{-1} \max_{w \in \mathcal{A}^n} \sup_{\mu \in \Lambda(\mathcal{A})} \left(|\kappa(w)| + \log_2 \mu(w)\right).$$

Niz, generiran z virom brez spomina, ne modelira dobro sporočil, ki jih srečamo v praksi. Zato predstavimo sledeči kompleksnejši model informacijskega vira, ki ga bomo uporabili v poglavju 4.3.

Definicija 2.39. Naj bo $m \in \mathbb{N}$. Informacijski vir abecede $\mathcal{A}$ je *končni vir abecede stopnje m*, če obstaja množica stanj S velikosti m, začetno stanje $s_0 \in S$ in množica *verjetnosti prehoda s črko iz stanja v stanje*

$$\{p(a, s|s') \in \mathbb{R}_{\geq 0} \mid s, s' \in S, a \in \mathcal{A}\}$$

tako, da za vsak $s' \in S$ velja

$$\sum_{a \in \mathcal{A}} \sum_{s \in S} p(a, s|s') = 1$$

in za vsak $w_1 w_2 \cdots w_n \in \mathcal{A}^+$ velja

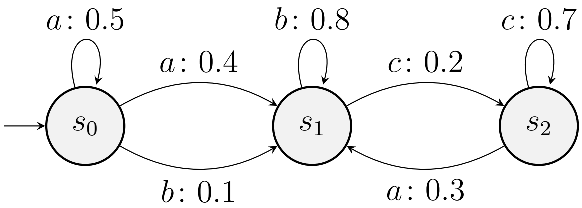
$$\mu(w_1 w_2 \cdots w_n) = \sum_{s_1, s_2, \ldots, s_n} \left(p(w_1, s_1|s_0) \prod_{i=2}^{n} p(w_i, s_i|s_{i-1})\right).$$

Z $\Lambda_{kv}^m(\mathcal{A})$ označimo družino končnih virov abecede $\mathcal{A}$ stopnje m.

Primer 2.40. Naj bo $\mathcal{A} = \{a, b, c\}$ in$S = \{s_0, s_1, s_2\}$. Napišemo le neničelne verjetnosti prehoda s črko iz stanja v stanje

$$p(a, s_0|s_0) = 0.5,\ p(a, s_1|s_0) = 0.4,\ p(b, s_1|s_0) = 0.1,$$
$$p(b, s_1|s_1) = 0.8,\ p(c, s_2|s_1) = 0.2,$$
$$p(c, s_2|s_2) = 0.7,\ p(c, s_1|s_2) = 0.3.$$

Ker velja $\sum_{a\in\mathcal{A}}\sum_{s\in S} p(a, s|s') = 1$, imamo končni vir abecede $\{a, b, c\}$ stopnje 3. Shematično ga predstavimo na sledeč način. Vsako stanje je vozlišče. Začetno vozlišče označimo z vhodno puščico. Vozlišči sta povezani, če obstaja neničelna verjetnost prehoda med njima. Nad vsako povezavo napišemo črko s katero se premaknemo iz enega stanja v drugega in verjetnost prehoda.



Slika 3: Shema končnega vira

Določimo verjetnost, da naš informacijski vir generira nize *cba*, *abcab*, *aabcabc*. Vedno začnemo v začetnem stanju s_0. Ker se iz začetnega stanja ne moremo premakniti s črko c v nobeno drugo stanje, je

$$\mu(cba) = 0.$$

Niz *acca* lahko generiramo samo z enim zaporedjem vozlišč s_0, s_1, s_2, s_2, s_1, zato je

$$\begin{aligned}\mu(acca) &= p(a, s_1|s_0) \cdot p(c, s_2|s_1) \cdot p(c, s_2|s_2) \cdot p(a, s_1|s_2)\\ &= 0.4 \cdot 0.2 \cdot 0.7 \cdot 0.3\\ &\doteq 0.0168.\end{aligned}$$

Niz *abc* lahko generiramo z zaporedjem s_0, s_0, s_1, s_2 ali pa s s_0, s_1, s_1, s_2, zato je

$$\begin{aligned}\mu(abc) &= p(a, s_0|s_0) \cdot p(b, s_1|s_0) \cdot p(c, s_2|s_1) + p(a, s_1|s_0) \cdot p(b, s_1|s_1) \cdot p(c, s_2|s_1)\\ &= 0.5 \cdot 0.1 \cdot 0.2 + 0.4 \cdot 0.8 \cdot 0.2\\ &\doteq 0.74.\end{aligned}$$

$\diamondsuit$

Več o odvečnost v povezavi s končnimi viri abecede $\mathcal{A}$ stopnje m, najdemo v [11].

3 Kontekstno-neodvisne gramatike

Pravopis določa pravila o rabi črk in ločil. Slovnica je sistem pravil za tvorjenje povedi in sestavljanje besedil. Slovenska slovnica, Slovenski pravopis in Slovar slovenskega knjižnega jezika določajo slovenski knjižni jezik, ki je poglavitno sredstvo javnega in uradnega sporazumevanja v Sloveniji. Podobno je *formalna gramatika* sistem pravil, ki pove kako iz dane abecede tvorimo nize oziroma kateri nizi so "pravilni". Veljavne nize imenujemo *formalni jezik*. Formalne gramatike in formalni jeziki imajo široko uporabo. Uporabljajo se za modeliranje naravnih jezikov, kompresijo podatkov, so osnova programskih jezikov ter formalizirajo matematično logiko in sisteme aksiomov.

Definicijo formalne gramatike poda Chomsky v [2] in jih razdeli v štiri razrede z postopnim povečevanjem omejitev [3, 1]. V delu bomo spoznali le en razred formalnih gramatik, in sicer *kontekstno-neodvisne gramatike*.

Definicija 3.1. Relacija $P \subseteq A \times B$ je *celovita*, če velja $\forall x \in A \; \exists y \in B \colon (x, y) \in P$.

Definicija 3.2. *Kontekstno-neodvisna gramatika*, je četverica $G = (V, \Sigma, P, S)$, kjer je

- V abeceda *nekončnih simbolov*;
- Σ abeceda *končnih simbolov* taka, da $\Sigma \cap V = \emptyset$;
- $P \subseteq V \times (V \cup \Sigma)^*$ celovita relacija, elementom relacije pravimo *prepisovalna pravila*;
- $S \in V$ je *začetni simbol*.

Definicija 3.3. Naj bo G kontekstno-neodvisna gramatika in α, β, $\gamma \in (V \cup \Sigma)^*$, $A \in V$ ter naj bo $(A, \beta) \in P$, pišemo $A \to \beta$. *Levi člen prepisovalnega pravila* $A \to \beta$ je A in *desni člen prepisovalnega pravila* je β. Pravimo, da se $\alpha A \gamma$ *prepiše s pravilom* $A \to \beta$ v $\alpha\beta\gamma$, pišemo $\alpha A \gamma \Rightarrow \alpha\beta\gamma$. Pravimo, da α *izpelje* β, če je $\alpha = \beta$ ali če za $k \geq 0$ obstaja zaporedje $\alpha_1, \alpha_2, \ldots, \alpha_n \in (V \cup \Sigma)^+$, da

$$\alpha = \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_n = \beta,$$

kar krajše pišemo $\alpha \overset{*}{\Rightarrow} \beta$. Kontekstno-neodvisno gramatiko okrajšamo s KNG.

Primer 3.4. Naj bo $V = \{S\}$, $\Sigma = \{a, b\}$, $P = \{S \to aSb, S \to \epsilon\}$ in $S = S$. To je res KNG, saj sta množici V in Σ končni, disjunktni in je P celovita. Izpeljemo nize

$$\begin{gathered} S \Rightarrow \epsilon, \\ S \Rightarrow aSb \Rightarrow ab, \\ S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb, \\ \vdots \end{gathered}$$

$\diamondsuit$

Ime kontekstno-neodvisna gramatika izvira iz oblike prepisovalnih pravil. Na levi strani pravila mora vedno stati samo en nekončni simbol. Torej ne sme vsebovati pravil oblike $\alpha A \gamma \to \alpha\beta\gamma$, saj je uporaba tega pravila odvisna od *konteksta* nekončnega simbola A. Kontekst določata niza $\alpha, \beta \in (V \cup \Sigma)^*$, ki se nahajata neposredno pred in po nekončnim simbolom.

Standardno z velikimi tiskanimi črkami označujemo nekončne simbole, z malimi tiskanimi črkami označujemo končne simbole in z grškimi črkami označujemo končne nize nekončnih in končnih simbolov. Ko govorimo o poljubnem simbolu, ga označimo z y.

Definicija 3.5. Jezik KNG G je $L(G) = \{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$. Jezik so torej nizi, ki jih lahko izpeljemo s pravili iz začetnega simbola in vsebujejo le končne simbole.

Primer 3.6. Jezik KNG iz primera 3.5 je $\{a^n b^n \mid n \geq 0\}$. ♢

Primer 3.7. Formalizirajmo formalno gramatiko iz primera 1.2. Dobili smo jo z nizom $w = cababcccababcccab$. Označimo jo z $G_w = (V, \Sigma, P, S)$, kjer je

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

Vidimo, da je G_w KNG in $L(G_w) = \{w\}$. ♢

3.1 Dopustne gramatike

Da lahko stistnemo niz w s pomočjo KNG mora biti njen jezik enojec $\{w\}$, saj lahko iz KNG enolično rekonstruramo niz w. V razdelku predstavimo poseben primer KNG, to so *dopustne gramatike* [7].

Definicija 3.8. KNG G je *deterministična*, če vsak nekončen simbol $A \in V$ nastopa natanko enkrat kot levi člen nekega prepisovalnega pravila. KNG, ki ni deterministična, je *nedeterministična*.

Determinističnost zagotovi, da je prepisovalno pravilo natanko določeno z njegovim levim členom.

Trditev 3.9. *Naj bo G deterministična KNG. Potem je jezik G enojec ali pa prazna množica.*

Dokaz. Recimo, da je $L(G) \neq \emptyset$ in da vsebuje več kot en niz. Naj bosta $w, u \in L(G)$ in $w \neq u$. Potem obstajata različni izpeljavi $S \overset{*}{\Rightarrow} w$ in $S \overset{*}{\Rightarrow} u$. Ker sta različni, smo v zaporedju izpeljave izbirali med dvema različnima prepisovalnima praviloma. To je v protislovju z determinističnostjo. Torej je $w = u$ in obstaja le ena izpeljava niza iz S. □

Determinizem sam po sebi ni dovolj močan, da prepreči praznost jezika, kot pokažeta sledeča primera, kjer se v izpeljavi "zaciklamo". Zato bomo od dopustnih gramatik zahtevali, da je njihov jezik neprazen.

Primer 3.10. Naj bo G KNG, $V = \{S\}$, $\Sigma = \{a\}$, $P = \{S \to S\}$ in $S = S$. KNG je deterministična. Jezik je prazen, saj ne moremo izpeljati niza, ki bi vseboval le končne simbole. ♢

Primer 3.11. Naj bo G KNG, $V = \{S, A, B\}$, $\Sigma = \{a\}$, $P = \{S \to Aa, A \to Ba, B \to Aa\}$ in $S = S$. KNG je deterministična. Jezik je prazen, saj ne moremo izpeljati niza, ki bi vseboval le končne simbole. Z uporabo končno mnogo prepisovalnih pravil se le ciklamo med A in B, pri tem število ponovitev končnega simbola a pove kolikokrat smo uporabili pravila.

$$S \Rightarrow Aa \Rightarrow Baa \Rightarrow Aaaa \Rightarrow Baaa \Rightarrow \dots$$

♢

Ker bomo KNG stisnili, želimo da ne vsebuje odvečnih simbolov. Simbol je odvečen, če se ne pojavi v nobenei izpeljavi niza, ki je v jeziku KNG.

Definicija 3.12. Pravimo, da KNG G *ne vsebuje neuporabnih simbolov*, ko za vsak simbol $y \in V \cup \Sigma,\ y \neq S$, obstajajo nizi $\alpha_1, \alpha_2, \ldots, \alpha_n \in (V \cup \Sigma)^+$, da je y vsebovan vsaj v enem izmed nizov in velja

$$S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n \in L(G).$$

Zgornje zahteve sedaj združimo v nov podrazred KNG.

Definicija 3.13. KNG G je *dopustna gramatika*, če je deterministična, ne vsebuje neuporabnih simbolov, ima prazen jezik in prazen niz ne nastopa kot desni član kateregakoli prepisovalnega pravila v P.

Posledica 3.14. *Jezik dopustne gramatike je enojec.*

Dokaz. Direktno sledi iz trditve 3.9 in zahteve po nepraznosti jezika dopustne gramatike. □

Če je G dopustna gramatika, obstaja enolično določen $w \in \Sigma(G)^+$, da je $L(G) = \{w\}$. Zato jo bomo pogosto označili kar z G_w in rekli, da generira niz w.

Prepisovalna pravila točno določajo KNG, saj lahko iz njih enolično določimo V, Σ in S. Nekončni simboli so levi člani prepisovalnih pravil, končni simboli so desni člani prepisovalnih pravil, ki niso tudi levi člani kateregakoli prepisovalnega pravila in začetni simbol je nekončni simbol, ki ne nastopa kot desni član kateregakoli prepisovalnega pravila. Torej je dovolj, da stisnemo le prepisovalna pravila.

Primer 3.15. Podana so prepisovalna pravila

$$P = \{A \to aBCD, B \to ab, C \to Bb, D \to Cb\}.$$

Sledimo zgornjemu razmisleku in dobimo $V = \{A, B, C, D\}$ in $\Sigma = \{a, b\}$. $S = A$. Vidimo, da je $L(G) = aababbabbb$ in zlahka preverimo, da je KNG. Da je dopustna, bomo preverili v primeru 3.32. ♢

3.2 D0L-sistem

Kot nas je pri uvedbi formalne gramatike motivirala slovnica, nas sedaj motivira biologija. Procese, ki potekajo istočasno, na primer razmnoževanje bakterij ali rast rastlin, lahko opišemo z *Lindenmayerjevim sistemom*, krajše *L-sistemom.* Ker se ukvarjamo z dopustnimi gramatikami, se omejimo na *deterministične kontekstno-neodvisne L-sisteme.* Več o splošnih L-sistemih najdemo v [12].

Definicija 3.16. Naj bo Σ abeceda. *Endomorfizem na* Σ^* je preslikava $f\colon \Sigma^* \to \Sigma^*$, da je

$$f(\varepsilon) = \varepsilon,$$
$$\forall w, u \in \Sigma^*\colon f(wu) = f(w)f(u).$$

Induktivno definiramo

$$\begin{aligned}
f^0(w) &= w, \\
f^1(w) &= f(w), \\
f^k(w) &= f(f^{k-1}(w)),
\end{aligned}$$

kjer je $w \in \Sigma^*$ in $k \geq 2$ celo število.

Opomba 3.17. Endomorfizem f na Σ^* je natanko določen, ko za vsako črko $a \in \Sigma$ podamo njeno preslikavo $f(a) \in \Sigma^*$.

Definicija 3.18. *Deterministični kontekstno-neodvisni L-sistem*, na kratko *D0L-sistem*, je trojica $D = (\Sigma, f, w)$, kjer je Σ abeceda, f endomorfizem na Σ^*, in $w \in \Sigma^*$. Sistem ima *fiksno točko* w^*, če za zaporedje $\{f^k(w) \mid k = 0, 1, 2 \ldots\}$ velja

$$\begin{gathered}
w^* \in \{f^k(w) \mid k = 0, 1, 2, \ldots\}, \\
f(w^*) = w^*.
\end{gathered}$$

Definicija 3.19. Naj bo G deterministična KNG v kateri prazen niz ne nastopa kot desni člen kateregakoli prepisovalnega pravila. Na $(V \cup \Sigma)^*$ definiramo endomorfizem f_G, da velja

$$\begin{gathered}
\forall a \in \Sigma \colon f_G(a) = a; \\
\text{če je } A \to \alpha \text{ prepisovalno pravilo, potem je } f_G(A) = \alpha.
\end{gathered}$$

D0L-sistem $(V \cup \Sigma, f_G, S)$ označimo z D0L(G) in ga imenujemo *D0L-sistem prirejen* G.

Če uporabimo f_G na nekem nizu nekončnih in končnih simbolov, uporabimo v enem koraku vsa prepisovalna pravila, ki jih lahko. Medtem, ko pri KNG v vsaki iteraciji uporabimo le eno prepisovalno pravilo naenkrat.

Primer 3.20. Spomnimo se KNG G iz primera 3.15 in ji priredimo D0L-sistem. Prepisovalna pravila so

$$P = \{A \to aBCD, B \to ab, C \to Bb, D \to Cb\}.$$

Ker je G deterministična in prazni niz ne nastopa kot desni člen kateregakoli prepisovalnega pravila, ji lahko priredimo D0L(G). Potem je $S = A$, $f_G(a) = a$, $f_G(b) = b$, $f_G(A) = aBCD$, $f_G(B) = ab$, $f_G(C) = Bb$ in $f_G(D) = Cb$. Izračunajmo še fiksno točko:

$$\begin{aligned}
f_G^0(A) &= A, \\
f_G^1(A) &= aBCD, \\
f_G^2(A) &= aabBbCb, \\
f_G^3(A) &= aababbBbb, \\
f_G^4(A) &= aababbabbb.
\end{aligned}$$

Vidimo, da je fiksna točka enaka nizu, ki ga izpelje G. ♢

Trditev 3.21. *Naj bo G dopustna gramatika. Potem je $w \in L(G)$ fiksna točka D0L(G).*

Dokaz. Ker je G dopustna, je deterministična in prazen niz ne nastopa kot desni član kateregakoli prepisovalnega pravila. Torej ji lahko priredimo D0L(G). Prav tako velja, da je jezik enojec, torej $L(G) = \{w\}$.

Imamo prepisovalno pravilo $S \to \alpha$, kjer je $\alpha \in (V \cup \Sigma)^*$. Naj bo $y_1 y_2 \cdots y_n$ predstavitev niza α s črkami. Potem je

$$f_G^i(S) = f_G^i(y_1 y_2 \cdots y_n) = f_G^i(y_1) f_G^i(y_2) \cdots f_G^i(y_n).$$

Če je $y_j \in \Sigma$, je $f_G^i(y_j) = y_j$, sicer pa $y_j \in V$ in zaradi determinističnosti obstaja prepisovalno pravilo $y_j \to z$, kjer je $z \in (V \cup \Sigma)^*$. Torej je $f_G^i(y_i) = f_G^{i-1}(z)$.

Ker so množice V, Σ, P končne in S izpelje w, obstaja tak $i \in N$, da je $f_G^i(S) = w$. Niz $w \in \Sigma^+$ je res fiksna točka, saj je

$$f_G(w) = f_G(w_1 w_2 \cdots w_n) = f_G(w_1) f_G(w_2) \cdots f_G(w_n) = w_1 w_2 \cdots w_n = w,$$

kjer so $w_1, w_2, \ldots, w_n \in \Sigma$. □

3.3 Izpeljevalni graf

Definicija 3.22. Naj bo G KNG v kateri prazen niz ne nastopa kot desni član kateregakoli prepisovalnega pravila. *Izpeljevalni graf* G, označimo ga z $\Gamma(G)$, je usmerjen graf z vozlišči $V \cup \Sigma$. Za prepisovalno pravilo

$$A \to y_1 y_2 \cdots y_n,$$

iz vozlišča A izvirajo usmerjene povezave v vozlišča $y_1, y_2, \ldots, y_n \in V \cup \Sigma$.

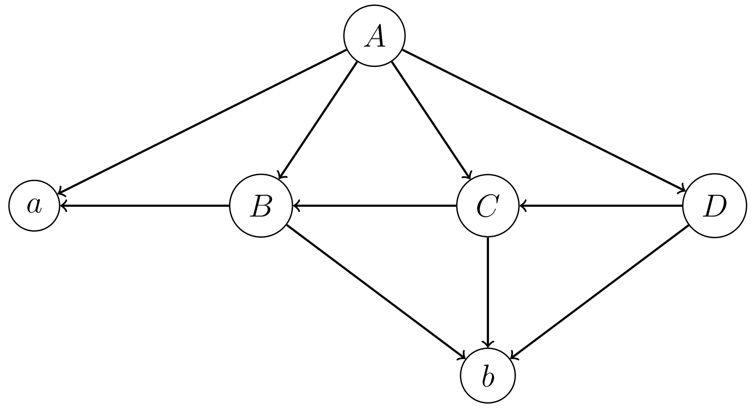
Osvežimo osnovni definiciji iz teorije grafov.

Definicija 3.23. *Pot dolžine n* je zaporedje vozlišč $(v_1, v_2, \ldots, v_{n+1})$, da za vsak $i = 1, 2, \ldots, n$ velja, da je (v_i, v_{i+1}) povezava v grafu. Pravimo, da je pot *cikel*, če je $v_1 = v_{i+1}$. Graf brez ciklov je *acikličen*.

Definicija 3.24. Vozlišče v je *koren usmerjenega grafa*, če je za vsako vozlišče $u \neq v$ obstaja pot od v do u.

Primer 3.25. Poglejmo izpeljevalni graf KNG G iz primera 3.15. Prepisovalna pravila so

$$P = \{A \to aBCD, B \to ab, C \to Bb, D \to Cb\}.$$



Slika 4: Izpeljevalni graf $\Gamma(G)$.

$\diamondsuit$

Lema 3.26. *Naj bo G dopustna gramatika. Potem ima $\Gamma(G)$ koren S.*

Dokaz. Naj bo $y \in V \cup \Sigma$ različen od S. Poiščemo pot od S do y. Ker je G dopustna, ne vsebuje neuporabnih simbolov. Po definiciji za $y \in V \cup \Sigma$, $y \neq S$, obstajajo $\alpha_1, \alpha_2, \ldots, \alpha_n \in (V \cup \Sigma)^+$, da je y vsebovan v vsaj enem izmed njih in velja $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n \in L(G)$.

Recimo da y prvič nastopi v α_k. Velja, da

$$y \text{ nastopa v } \alpha_k;$$
$$S \to \alpha_1;$$
$$\text{če je } k > 1, \text{ potem } \forall i = 1, 2, \ldots, k-1\colon \alpha_i \Rightarrow \alpha_{i+1}.$$

Zgradimo pot z indukcijo na k. Za $k = 1$, je y vsebovan v α_1, torej je pot kar povezava med S in y. Naj bo sedaj $k > 1$ in predposatvimo, da obstaja pot od S do vsakega simbola v α_{k-1}. Izberemo tisti $A \in V$, ki iz α_{k-1} izpelje α_k. Ker y nastopa v α_k, obstaja povezava med A in y. Po predpostavki obstaja pot med S in A, zato A nastopa v α_{k-1}. Našli smo pot od S do y. $\square$

Lema 3.27. *Naj bo G dopustna gramatika. Potem je $\Gamma(G)$ acikličen.*

Dokaz. Po trditevi 3.21 obstaja fiksna točka w endomorfizma f_G. Torej, obstaja $i = 1, 2, \ldots$, da je $f_G^i(w) = w \in \Sigma^+$. To pomeni, da

$$\text{končno mnogo členov zaporedja } \{f_G^i(S) \mid i = 1, 2, \ldots\} \text{ ni niz v } \Sigma^+. \tag{3.1}$$

Naj bosta $A, B \in V$ med katerima obstaja pot od A do B. Ker je G dopustna, ne vsebuje neuporabnih simbolov. Za A obstajajo $\alpha_1, \alpha_2, \ldots, \alpha_n \in (V \cup \Sigma)^+$, da A nastopa v vvsaj v enem izmed njih in $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n \in L(G)$. To pomeni, da A za nek $i = 1, 2, \ldots$ nastopa v $f_G^i(S)$. Ker obstaja pot od A do B, potem B nastopa za nek $j = i+1, i+2, \ldots$ v $f_G^j(S)$.

Recimo da ima graf $\Gamma(G)$ cikel, torej obstaja neka pot od A do A. Potem se A pojavi neskončnokrat v zaporedju $\{f_G^i(S) \mid i = 1, 2, \ldots\}$, to pa je v protislovju z izjavo 3.1, ki sledi iz predpostavke. $\square$

3.4 Karakterizacija dopustne gramatike

Preko izpeljevalnega grafa gramatike lahko ugotovimo ali je podana gramatika dopustna. Tudi preko D0L-sistema gramatike lahko preizkusimo dopustnost gramatike. Še več, D0L-sistem gramatike poda algoritem za izračun niza, ki ga generira dopustna gramatika.

Izrek 3.28. *Naj bo G KNG v kateri prazen niz ne nastopa kot desni član kateregakoli prepisovalnega pravila. Potem so naslednje trditve ekvivalentne:*

1. *G je dopustna gramatika.*

2. *$\Gamma(G)$ je acikličen in ima koren S.*

3. *Za D0L(G) je $f_G^{|V|}(S) \in \Sigma^+$ in vsak simbol iz $V \cup \Sigma$ nastopa v vsaj enem izmed nizov $f_G^i(S)$ za $i = 0, 1, \ldots, |V|$.*

Pred dokazom zgornjega izreka potrebujemo še tri leme, ki bodo pomagale pri dokazu.

Lema 3.29. *Naj bo G KNG v kateri prazen niz ne nastopa kot desni član kateregakoli prepisovalnega pravila in je $\Gamma(G)$ acikličen. Vzemimo $\alpha \in (V \cup \Sigma)^+ \setminus \Sigma^+$. Potem obstaja nekončni simbol $A \in V$, ki nastopa v nizu α in za $\forall i = 1, 2, \ldots$ velja*

$$A \text{ ne nastopa v nizu } f_G^i(\alpha).$$

Dokaz. Dokažimo s protislovjem. Predpostavimo, da zaključek leme ne velja. Z H označimo vse nekončne simbole V, ki nastopajo v α. Po predpostavki je H neprazna. Za vsak $B \in H$ definiramo

$$H(B) = \{C \in V \mid C \text{ nastopa v enem izmed nizov } f_G^i(B),\ i = 1, 2, \ldots\}.$$

Vsak nekončen simbol iz V, ki nastopa v enem izmed $f_G^i(\alpha)$, kjer je $i = 1, 2, \ldots$, leži v $\cup_{B \in H} H(B)$. Po predpostavki

$$\forall A \in V \colon A \notin H \vee A \text{ nastopa v enem izmed nizov } f_G^i(\alpha),\ i = 1, 2, \ldots$$

sledi, da za vsak $A \in H$ obstaja $B \in H$ tako, da je $A \in H(B)$.

Sedaj izberemo takšno neskončno zaporedje $A_1, A_2, \ldots$ elementov H, da za vsak $i = 1, 2, \ldots$ velja $A_i \in H(A_{i+1})$. Ker je množica H končna, obstaja nek A in naravni števili $i < j$, da je $A_i = A_j = A$.

Opomnimo, da za $A \in H(B)$ obstaja pot od B do A v grafu $\Gamma(G)$. Sledi, da v zaporedju $A_1, A_2, \ldots$ obstaja pot $(A_j, A_{j-1}, \ldots, A_{i+1}, A_i)$. Ker je ta pot cikel smo prišli do protisolovja. □

Lema 3.30. *Naj bo G KNG v kateri prazen niz ne nastopa kot desni član kateregakoli prepisovalnega pravila in je $\Gamma(G)$ acikličen. Potem je*

$$\forall \alpha \in (V \cup \Sigma)^+ \colon f_G^{|V|}(\alpha) \in \Sigma^+.$$

Dokaz. Naj bo $\alpha \in (V \cup \Sigma)^+$ poljuben. Predpostavimo $f_G^{|V|}(\alpha) \notin \Sigma^+$ in pokažimo, da nas to vodi v protislovje.

Po predpostavki in ker je $\forall a \in \Sigma$ velja $f_G(a) = a$, sklepamo da niz $f_G^i(\alpha)$, kjer je $i = 0, 1, \ldots, |V|$, vsebuje vsaj en nekončen simbol iz V. Na vsakem nizu $f_G^i(\alpha)$ uporabimo lemo 3.29 in dobimo zaporedje $A_0, A_1, \ldots, A_{|V|}$ nekončnih simbolov iz V, da za $i = 0, 1, \ldots, |V|$ velja

$$A_i \text{ nastopa v nizu } f_G^i(\alpha), \tag{3.2}$$

$$\forall j = i+1, i+2, \ldots, |V| : A_i \text{ ne nastopa v nizu } f_G^j(\alpha). \tag{3.3}$$

Ker je zaporedje $A_0, A_1, \ldots, A_{|V|}$ simbolov iz V, daljše od $|V|$, se vsaj en simbol ponovi. Torej, obstaja $A \in V$ in $i, j \in \{0, 1, \ldots, |V|\}$, $i < j$, da je $A_i = A_j = A$.

Po 3.2 A_i nastopa v $f_G^i(\alpha)$ ter po 3.3 ne nastopa v $f_G^j(\alpha)$. Vendar po 3.2 tudi A_j nastopa v $f_G^j(\alpha)$. Ker je $A_i = A_j = A$, smo prišli do protislovja. □

Lema 3.31. *Naj bo G KNG v kateri prazen niz ne nastopa kot desni član kateregakoli prepisovalnega pravila in naj bo $\Gamma(G)$ acikličen ter ima koren S. Potem vsak simbol iz $V \cup \Sigma$ nastopi v vsaj enem izmed nizov $f_G^i(S)$ za $i = 0, 1, \ldots, |V|$.*

Dokaz. Če v grafu $\Gamma(G)$ obstaja pot dolžine i od $A \in V$ do $y \in V \cup \Sigma$, potem simbol y nastopa v $f_G^i(A)$. Koren S očitno nastopa v nizu $f_G^0(S)$. Izberemo poljuben $y \in V \cup \Sigma$ različen od S. Ker je S koren grafa, obstaja pot dolžine i od S do y.

Pot je oblike $(S, A_2, A_3, \ldots, A_i, y)$, kjer so $S, A_2, A_3, \ldots, A_i \in V$. Ker je graf brez ciklov, so vsa vozlišča $S, A_2, A_3, \ldots, A_i$ paroma različna. Res je $i < |V|$. □

Sedaj lahko dokažemo izrek.

Dokaz izreka 3.28.

$1 \Rightarrow 2$ Sledi po lemah 3.26 in 3.27.

$2 \Rightarrow 3$ Sledi po lemah 3.30 in 3.31.

$3 \Rightarrow 1$ Obstoj D0L(G) zagotovi determinističnost in da prazen niz ne nastopa kot desni član kateregakoli prepisovalnega pravila.

Jezik je neprazen, saj $f_G^{|V|}(S) \in \Sigma^+$.

KNG G ne vsebuje neuporabnih simbolov, saj vsak simbol iz $V \cup \Sigma$ nastopa v vsaj enem izmed nizov $f_G^i(S)$ za $i = 0, 1, \ldots, |V|$.

□

Primer 3.32. Preverimo dopustnost gramatike iz primera 3.15. V primeru 3.25 smo narisali izpeljevalni graf te gramatike. Ker je graf acikličen in koren S, je gramatika dopustna po 2. točki izreka 3.28. V primeru 3.20 smo tej gramatiki priredili D0L-sistem. Do niza, ki vsebuje same končne simbole, smo res prišli v $|V| = 4$ iteracijah. Tudi vsak od simbolov A_0, A_1, A_2, A_3, a, b se pojavi v izračunanih nizih. Torej je gramatika dopustna po 3. točki izreka 3.28. ♢

Kot smo omenili na začetku razdelka, bomo podali algoritem za izračun niza generiranega z dopustno gramatiko.

Posledica 3.33. *Niz generiran z* G_w *je* $w = f_G^{|V|}(S)$.

Dokaz. Direktno sledi po 3. točki izreka 3.28. □

Prav tako posplošimo trditev 3.21.

Posledica 3.34. *Naj bo* G *dopustna gramatika in* $\alpha \in (V \cup \Sigma)^+$. *Potem ima D0L sistem* $(V \cup \Sigma, f_G, \alpha)$ *fiksno točko* $w^* \in \Sigma^+$, *ki je*

$$w^* = f_G^{|V|}(\alpha).$$

Dokaz. Z lemo 3.30 razširimo 3. točko izreka 3.28 iz S na vse $\alpha \in (V \cup \Sigma)^+$. □

Sledeči endomorfizem bo prišel prav v naslednjem poglavju.

Definicija 3.35. Naj bo G dopustna gramatika. Definiramo preslikavo

$$\begin{aligned} f_G^\infty \colon (V \cup \Sigma)^* &\to (V \cup \Sigma)^*, \\ \alpha &\mapsto w^*, \end{aligned}$$

kjer je w^* fiksna točka D0L-sistema $(V \cup \Sigma, f_G^\infty, \alpha)$.

Trditev 3.36. *Naj bo* G *dopustna gramatika. Potem veljajo naslednje trditve:*

1. f_G^∞ *je endomorifzem na* $(V \cup \Sigma)^*$.
2. $\forall \alpha \in (V \cup \Sigma)^+ \colon f_G^\infty(\alpha) \in \Sigma^+$.
3. $\forall \alpha \in (V \cup \Sigma)^+ \colon f_G^\infty(\alpha) = f_G^{|V|}(\alpha)$.
4. *Če je* $A \to \alpha$, *prepisovalno pravilo, potem je* $f_G^\infty(A) = f_G^\infty(\alpha)$.

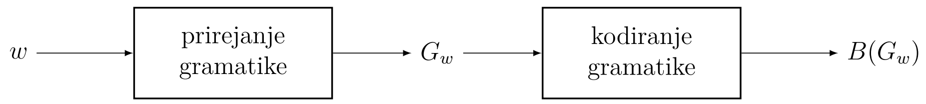
Dokaz.

1. Rutinsko preverimo, da ustreza definiciji.
2. Velja po definiciji fiksne točke endomorfizma.
3. Sledi direktno iz posledice 3.34.
4. Ker imamo prepisovalno pravilo, $A \to \alpha$ zaporedje $\{f^k(\alpha) \mid k = 1, 2, 3 \ldots\}$ dobimo tako, da odstraimo prvi člen zaporedja $\{f^k(A) \mid k = 0, 1, 2 \ldots\}$. Zaporedji imata enako fiksno točko.

□

4 Prirejanje in kodiranje dopustne gramatike

Spodnja slika prikazuje glavno idejo stiskanja niza s pomočjo KNG. Nizu w abecede najprej priredimo KNG G_w in jo nato binarno kodiramo.



Slika 5: Kodirnik niza w

Od tu naprej naj bo $\mathcal{A}$ poljubna abeceda, $|\mathcal{A}| \geq 2$ iz katere bomo generirali niz. Naj bo $\{A_0, A_1, \ldots\}$ končna abeceda in predpostavimo da je $\mathcal{A} \cap \{A_0, A_1, \ldots\} = \emptyset$. Simbole $A_0, A_1, \ldots$ uporabljamo za nekončne simbole, njihov *naravni vrstni red* je določen z naraščanjem indeksa.

4.1 Prirejanje gramatike

Namen dopustne gramatike je generirati niz abecede $\mathcal{A}$, zato ni pomembno, kako poimenujemo nekončne simbole

Definicija 4.1. Naj bo $\mathcal{G}(\mathcal{A})$ množica vseh KNG G, ki zadostujejo:

1. G je dopustna gramatika;
2. $\Sigma \subseteq \mathcal{A}$;
3. $V = \{A_0, A_1, \ldots, A_{|V|-1}\}$;
4. $S = A_0$;
5. Če naštejemo nekončne simbole V v vrstnem redu prve pojavitve od leve proti desni v nizu
$$f_G^0(A_0) f_G^1(A_0) \cdots f_G^{|V|-1}(A_0),$$
dobimo zaporedje $A_0, A_1, A_2, \ldots, A_{|V|-1}$.

Z zahtevo 5. so nekončni simboli poimenovani po edinstvem vrstnem redu. To je vrstni red, ki ga inducira iskanje v globino v izpeljevalnem grafu $\Gamma(G)$, pri katerem so otroci obiskani v vrstnem redu od leve proti desni. Ta vrstni red bo dekodirniku v dokazu izreka 4.27 omogočil pravilno določiti naslednji nekončni simbol.

KNG $G \notin \mathcal{G}(\mathcal{A})$, ki izpolnjuje zahtevi 1. in 2., preimenujmo nekončne simbole tako, da izpolnimo 5. zahtevo. Potem izpolnimo tudi 3. in 4. zahtevi. Dobimo $[G] \in \mathcal{G}(\mathcal{A})$, ki jo imenujemo *kanonično oblika* G, in velja $L([G]) = L(G)$.

Primer 4.2. Naj bo $\mathcal{A} = \{a, b\}$. Podano imamo dopustno gramatiko G z začetnim simbolov S in prepisovalnimi pravili

$$P = \{S \to BaA, A \to aC, B \to Db, C \to bB, D \to ab\}.$$

Zahtevi 1. in 2. sta izpolnjeni. Izračnajmo $f_G^i(S)$ za $i = 0, 1 \ldots, |V| - 1$.

$$\begin{aligned} f_G^0(S) &= S, \\ f_G^1(S) &= BaA, \\ f_G^2(S) &= DbaaC, \\ f_G^3(S) &= abbaabB, \\ f_G^4(S) &= abbaabDb. \end{aligned}$$

Skupaj staknemo zgornje nize, da dobimo niz iz 5. zahteve,

$$SBaADbaaCabbaabBabbaabDb.$$

Naštejemo nekončne simbole v vrstnem redu prve pojavitve od leve proti desni v zgornjem nizu: S, B, A, C, D.

Glede na ta seznam nekončne simbole ustrezno preimenujemo

$$\begin{aligned} S &\rightsquigarrow A_0, \\ B &\rightsquigarrow A_1, \\ A &\rightsquigarrow A_2, \\ C &\rightsquigarrow A_3, \\ D &\rightsquigarrow A_4. \end{aligned}$$

Dobimo $[G] \in \mathcal{G}(\mathcal{A})$ s prepisovalnimi pravili

$$P = \{A_0 \to A_1 a A_2, A_1 \to A_3 b, A_2 \to a A_4, A_3 \to ab, A_4 \to b A_1\}.$$

$\diamondsuit$

Ključni del stiskanja niza na sliki 5 je prirejanje gramatike. Formalno je to preslikava, ki vsakemu nizu abecede $\mathcal{A}$ dodeli dopustno gramatiko iz $\mathcal{G}(\mathcal{A})$, ki ta niz generira.

Definicija 4.3. *Prirejanje gramatike nizu abecede* $\mathcal{A}$ je preslikava

$$\begin{aligned} \pi \colon \mathcal{A}^+ &\to \mathcal{G}(\mathcal{A}), \\ w &\mapsto G_w. \end{aligned}$$

Predstavimo dva razreda prirejanj dopustne gramatike: asimptotsko kompaktno in neskrčljivo.

Definicija 4.4. Z $\mathcal{G}^*(\mathcal{A})$ označimo pravo podmnožico množice $\mathcal{G}(\mathcal{A})$, da za vsak $G \in \mathcal{G}^*(\mathcal{A})$ velja

$$\forall A, B \in V,\ A \neq B \colon f_G^\infty(A) \neq f_G^\infty(B).$$

Definicija 4.5. Z $|G|$ označimo vsoto dolžin desnih členov prepisovalnih pravil KNG G.

4.1.1 asimptotsko kompaktno prirejanje gramatike

Definicija 4.6. Prirejanje gramatike nizu abecede $\mathcal{A}$ je *asimptotsko kompaktno*, če za vsak niz $w \in \mathcal{A}^+$ velja $G_w \in \mathcal{G}^*(\mathcal{A})$ in je

$$\lim_{n\to\infty} \max_{w\in\mathcal{A}^n} \frac{|G_w|}{|w|} = 0.$$

Predstavimo dve asimptotsko kompaktni prirejanji gramatike, *Lempel-Ziv prirejanje gramatike* in *bisekcijsko prirejanje gramatike.*

Definicija 4.7. Naj bo $w_1 w_2 \cdots w_n$ predstavitev niza w. *Lempel-Ziv členitev niza* w je množica podnizov $\sigma_{\mathrm{lz}}(w)$, ki jo induktivno gradimo na sledeč način:

- $\sigma_{\mathrm{lz}}(w) = \emptyset$.
- Poiščemo najmanjši $i_1 \in \mathbb{N}$, da $u_1 = w_1 \cdots w_{i_1}$ ne nastopa v $\sigma_{\mathrm{lz}}(w)$. Dodamo u_1 v $\sigma_{\mathrm{lz}}(w)$.
- Poiščemo najmanjši $i_2 \in \mathbb{N}$, da $u_2 = w_{i_1} \cdots w_{i_2}$ ne nastopa v $\sigma_{\mathrm{lz}}(w)$. Dodamo u_2 v $\sigma_{\mathrm{lz}}(w)$.

$$\vdots$$

- Poiščemo najmanjši $i_m \in \mathbb{N}$, da $u_{i_m} = w_{i_{m-1}} \cdots w_{i_m}$ ne nastopa v $\sigma_{\mathrm{lz}}(w)$. Dodamo u_m v $\sigma_{\mathrm{lz}}(w)$. Ker je $i_m = n$, zaključimo.

Primer 4.8. Naj bo $w = 010010000001$. Podčratmo $u_1, u_2, \ldots, u_m$.

$$w = \underline{0}\,\underline{1}\,\underline{00}\,\underline{10}\,\underline{000}\,\underline{001}.$$

Lempel-Ziv členitev niza w je $\sigma_{\mathrm{lz}}(w) = \{0, 1, 00, 10, 000, 001\}$ ◇

Definicija 4.9. Naj bo $w \in \mathcal{A}^+$, $w_1 w_2 \cdots w_n$ predstavitev niza w s črkami abecede in $\sigma_{\mathrm{lz}}(w) = \{u_1, u_2, \ldots, u_m\}$. *Lempel-Ziv dopustno gramatiko* G_w^{lz} je

- $\Sigma = \{u \in \sigma_{\mathrm{lz}}(w) \mid |u| = 1\}$;
- $V = \{A_w\} \cup \{A_u \mid u \in \sigma_{\mathrm{lz}}(w)\}$;
- $S = A_w$ in imamo prepisovalno pravilo

  $$A_w \to A_{u_1} A_{u_2} \cdots A_{u_m};$$

- Za vsak $u \in \sigma_{\mathrm{bis}}(w)$ imamo prepisovalno pravilo

  $$A_u \to A_\alpha b,$$

  kjer je $u = \alpha b$, $\alpha \in \Sigma^*$ in $b \in \Sigma$.

Da dobimo kanonično obliko $[G_w^{\mathrm{lz}}]$ ustrezno preimenujemo nekončne simbole po postopku opisanem v definiciji 4.1. Prirejanju $w \mapsto [G_w^{\mathrm{lz}}]$ pravimo *Lempel-Ziv prirejanje gramatike.*

Trditev 4.10. *Lempel-Ziv prirejanje gramatike je asimptotsko kompaktno prirejanje gramatike.*

Dokaz. Lempel in Ziv sta v [9] pokazala, da velja

$$\max_{w \in \mathcal{A}^n} |\sigma_{\mathrm{lz}}(w)| = \mathcal{O}\left(\frac{n}{\log_2(n)}\right).$$

Preprosto preverimo, da je $\left|G_w^{\mathrm{lz}}\right| = 3 \cdot |\sigma_{\mathrm{lz}}(w)|$, zato

$$\lim_{n\to\infty} \max_{w \in \mathcal{A}^n} \frac{\left|G_w^{\mathrm{lz}}\right|}{|w|} = 0.$$

□

Primer 4.11. Nizu $w = 010010000001$ priredimo Lempel-Ziv gramatiko. V prejšnjem primeru 4.8 smo izračunali $\sigma_{\mathrm{lz}}(w) = \{0, 1, 00, 10, 000, 001\}$. Prepisovalna pravila dopustne gramatike G_w^{lz} so

$$\begin{aligned}
A_w &\to A_0 A_1 A_{00} A_{10} A_{000} A_{001},\\
A_0 &\to 0,\\
A_1 &\to 1,\\
A_{00} &\to A_0 0,\\
A_{10} &\to A_1 0,\\
A_{000} &\to A_{00} 0,\\
A_{001} &\to A_{00} 1.
\end{aligned}$$

Da dobimo kanonično obliko ustrezno preimenujemo nekončne simbole, kar je v tem primeru zelo lahko. Prepisovalna pravila $[G_w^{\mathrm{lz}}] \in \mathcal{G}^*(\{0,1\})$ so

$$\begin{aligned}
A_0 &\to A_1 A_2 A_3 A_4 A_5 A_6,\\
A_1 &\to 0,\\
A_2 &\to 1,\\
A_3 &\to A_1 0,\\
A_4 &\to A_2 0,\\
A_5 &\to A_3 0,\\
A_6 &\to A_3 1.
\end{aligned}$$

◇

Definicija 4.12. Naj bo $w \in \mathcal{A}^+$, $w_1 w_2 \cdots w_n$ predstavitev niza w s črkami abecede. *Bisekcijska členitev niza* w je

$$\sigma_{\mathrm{bis}}(w) = \{w\} \cup \left\{ w_i w_{i+1} \cdots w_j \;\middle|\; \log_2(j-i-1) \in \mathbb{N}_0 \text{ in } \frac{i-1}{j-i-1} \in \mathbb{N}_0 \right\}.$$

Pogoj $\log_2(j-i-1) \in \mathbb{N}_0$ pove, da vsebuje le podnize dolžine 2^n za $n \in \mathbb{N}_0$. Pogoj $\frac{i-1}{j-i-1} \in \mathbb{N}_0$ pove, da se po nizu w "premikamo" s korakom dolžine podniza.

Primer 4.13. Naj bo $w = 0001010$. Poiščimo podmnožico nizov $\sigma_{\text{bis}}(w)$. Po definiciji je $0001010 \in \sigma_{\text{bis}}(w)$. Ker je $|w| = 7$, zaradi prvega pogoja gledamo podnize dolžine $1, 2, 4$. Vsi nizi dolžine 1 so trivialno vsebovani. Poglejmo podnize dolžine 2. Zaradi drugega pogoja se po nizu "premikamo" s korakom dolžine 2. Množica $\sigma_{\text{bis}}(w)$ vsebuje podčrtane podnize:

$$\underline{00}\,\underline{01}\,\underline{01}\,0.$$

Množica $\sigma_{\text{bis}}(w)$ vsebuje podčrtan podniz dolžine 4:

$$\underline{0001}\,010.$$

Če podčratmo vse nize, ki so vsebovani v $\sigma_{\text{bis}}(w)$, vidimo zakaj jo imenujemo Bisekcijska dopustna gramatika:

$$\underline{\underline{\underline{\underline{0}\,\underline{0}}\,\underline{\underline{0}\,\underline{1}}}\,\underline{\underline{0}\,\underline{1}}\,\underline{0}}.$$

Torej je $\sigma_{\text{bis}}(w) = \{0001010, 0001, 01, 00, 1, 0\}$. $\diamondsuit$

Definicija 4.14. Naj bo $w \in \mathcal{A}^+$, $w_1 w_2 \cdots w_n$ predstavitev niza w s črkami abecede. *Bisekcijska dopustna gramatika* G_w^{bis} je

- $\Sigma = \{u \in \sigma_{\text{bis}}(w) \mid |u| = 1\}$;
- $V = \{A_u \mid u \in \sigma_{\text{bis}}(w)\}$;
- $S = A_w$;
- Naj bo $u \in \sigma_{\text{bis}}(w)$.

  Če je $|u| = 1$, je prepisovalno pravilo oblike

  $$A_u \to u.$$

  Če je $\log_2(|u|) \in \mathbb{N}$, niz u zapišemo kot stik dveh enako dolgih nizov l in d. Prepisovalno pravilo je oblike

  $$A_u \to A_l A_d.$$

  Sicer je $\log_2(|u|) \notin \mathbb{N}$. Sledi $u = w$. Vzemimo $u_1, u_2, \ldots, u_m \in \sigma_{\text{bis}}(w)$, da je $|w| > |u_1| > |u_2| > \cdots > |u_m|$ in velja $u_1 u_2 \cdots u_m = w$. Nizi $u_1, u_2, \ldots, u_m$ so enolično določeni in zato tudi prepisovalno pravilo, ki je oblike

  $$A_u \to A_{u_1} A_{u_2} \cdots A_{u_m}.$$

Da dobimo kanonično obliko $[G_w^{\text{bis}}]$ ustrezno preimenujemo nekončne simbole. Prirejanju $w \mapsto [G_w^{\text{bis}}]$ pravimo *bisekcijsko prirejanje gramatike.*

V [8] je pokazano, da je bisekcijsko prirejanje gramatike asimptotsko kompaktno prirejanje gramatike.

Primer 4.15. Nizu $w = 0001010$ priredimo bisekcijsko gramatiko. V prejšnjem primeru 4.13 smo izračunali $\sigma_{\text{bis}}(w) = \{0001010, 0001, 01, 00, 1, 0\}$. Prepisovalna pravila dopustne gramatike G_w^{bis} so

$$\begin{aligned}
A_w &\to A_{0001}A_{01}A_0,\\
A_{0001} &\to A_{00}A_{01},\\
A_{01} &\to A_0A_1,\\
A_{00} &\to A_0A_0,\\
A_1 &\to 1,\\
A_0 &\to 0.
\end{aligned}$$

Da dobimo kanonično obliko ustrezno preimenujemo nekončne simbole. Prepisovalna pravila $[G_w^{\text{bis}}] \in \mathcal{G}^*(\{0,1\})$ so

$$\begin{aligned}
A_0 &\to A_1A_2A_3,\\
A_1 &\to A_4A_2,\\
A_2 &\to A_3A_5,\\
A_3 &\to 0,\\
A_4 &\to A_3A_3,\\
A_5 &\to 1.
\end{aligned}$$

$\diamondsuit$

4.1.2 Neskrčljivo prirejanje gramatike

Definicija 4.16. Pravimo, da je $G \in \mathcal{G}^*(\mathcal{A})$ *neskrčljiva gramatika*, če:

1. Vsak $A \in V$, $A \neq S$ nastopa vsaj dvakrat kot desni člen prepisovalnih pravil;
2. Ne obstajata $y_1, y_2 \in V \cup \Sigma$, da niz y_1y_2 nastopa kot podniz desnega člana kateregali prepisovalnega pravila več kot enkrat na neprekrivajočih se mestih.

Definicija 4.17. *Neskrčljivo prirejanje gramatike nizu abecede* $\mathcal{A}$ vsakemu nizu abecede $\mathcal{A}$ priredi neskrčljivo gramatiko.

Različne neskrčjiva prirejanja gramatike dobimo tako, da izvajamo različne sisteme "skrčitev". V nadaljevanju si bomo ogledali pet pravil po katerih iz poljubne dopustne gramatike v končno mnogo korakih pridemo do neskrčljive gramatike.

Skrčitveno pravilo 1. Naj bo G dopustna gramatika in $A \in V$ nekončni simbol, ki se pojavi samo enkrat kot desni člen prepisovalnega pravila. Torej, obstajata prepisovalni pravili $B \to \alpha A\gamma$ in $A \to \beta$, kjer so $\alpha, \beta, \gamma \in (V \cup \Sigma)^*$.

Če v prepisovalnih pravilih dopustne gramatike G v enem koraku

- nadomestimo pravilo $B \to \alpha A\gamma$ z $B \to \alpha\beta\gamma$;
- odstranimo pravilo $A \to \beta$,

dobimo dopustno gramatiko G', da velja $L(G') = L(G)$. Dopustna gramatika G' je "bližje" 1. zahtevi definicije neskrčljive gramatike.

Skrčitveno pravilo 2. Naj bo G dopustna gramatika in recimo, da obstaja prepisovalno pravilo

$$A \to \alpha_1\beta\alpha_2\beta\alpha_3,$$

kjer so $\alpha_1, \alpha_2, \alpha_3, \beta \in (V \cup \Sigma)^*$ in $|\beta| \geq 2$.

Izberemo "svež" $B \notin \{A_0, A_1, \ldots\} \setminus V \cup \Sigma$. Če v prepisovalnih pravilih dopustne gramatike G v enem koraku

- dodamo pravilo $B \to \beta$;
- nadomestimo pravilo $A \to \alpha_1\beta\alpha_2\beta\alpha_3$ z $A \to \alpha_1 B\alpha_2 B\alpha_3$,

dobimo dopustno gramatiko G', da velja $L(G') = L(G)$. Dopustna gramatika G' je "bližje" zadostitvi 2. zahtevi definicije neskrčljive gramatike.

Skrčitveno pravilo 3. Naj bo G dopustna gramatika in recimo, da obstajata dve različni prepisovalni pravili

$$A \to \alpha_1\beta\alpha_2,$$
$$B \to \alpha_3\beta\alpha_4,$$

kjer so $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta \in (V \cup \Sigma)^*$, $|\beta| \geq 2$, $(\alpha_1 \neq \varepsilon \vee \alpha_2 \neq \varepsilon)$ in $(\alpha_3 \neq \varepsilon \vee \alpha_4 \neq \varepsilon)$.

Izberemo "svež" $C \notin \{A_0, A_1, \ldots\} \setminus (V \cup \Sigma)$. Če v prepisovalnih pravilih dopustne gramatike G v enem koraku

- dodamo pravilo $C \to \beta$;
- nadomestimo pravilo $A \to \alpha_1\beta\alpha_2$ z $A \to \alpha_1 C\alpha_2$;
- nadomestimo pravilo $A \to \alpha_3\beta\alpha_4$ z $A \to \alpha_3 C\alpha_4$,

dobimo dopustno gramatiko G', da velja $L(G') = L(G)$. Dopustna gramatika G' je "bližje" zadostitvi 2. zahtevi definicije neskrčljive gramatike.

Skrčitveno pravilo 4. Naj bo G dopustna gramatika in recimo, da obstajata prepisovalni pravili

$$A \to \alpha_1\beta\alpha_2,$$
$$B \to \beta,$$

kjer so $\alpha_1, \alpha_2, \beta \in (V \cup \Sigma)^*$, $|\beta| \geq 2$ in $\alpha_1 \neq \varepsilon \vee \alpha_2 \neq \varepsilon$.

Če v prepisovalnih pravilih dopustne gramatike G

- nadomestimo pravilo $A \to \alpha_1\beta\alpha_2$ z $A \to \alpha_1 B\alpha_2$,

dobimo dopustno gramatiko G', da velja $L(G') = L(G)$. Dopustna gramatika G' je "bližje" zadostitvi 1. zahtevi definicije neskrčljive gramatike.

Skrčitveno pravilo 5. Naj bo G dopustna gramatika in recimo, da obstajata $A, B \in V \cup \Sigma$, $A \neq B$ tako, da $f_G^\infty(A) = f_G^\infty(B)$.

Če v prepisovalnih pravilih dopustne gramatike G

- zamenjamo vse B, ki nastopajo kot desni člani pravil, z A;

- odstranimo vsa prepisovalna pravila, katerih levi član je neuporaben simbol;
- odstranimo vse neuporabne simbole,

dobimo dopustno gramatiko G', da velja $L(G') = L(G)$. Dopustna gramatika G' je "bližje" vsebovanosti v $\mathcal{G}^*(\mathcal{A})$.

Kako smo lahko prepričani, da iz dopustne gramatike G z uporabo končno mnogo skrčitvenih pravil pridemo do neskrčljive gramatike G'?

Definicija 4.18. Za dopustno gramatiko G definiramo $C(G) = 2\,|G| - |V|$.

Trditev 4.19. *Za vsako dopustno gramatiko G je $C(G) > 0$.*

Dokaz. Ker je G dopustna gramatika, brez škode za splošnost nekončne simbole premenujemo v $A_0, A_1, \dots, A_{|V|-1}$ tako, da zadostimo 5. zahtevi definicije 4.1. Niz vseh desnih članov prepisovalnih pravil je

$$f_G(A_0) f_G(A_1) \cdots f_G(A_{|V|-1})$$

in je po definiciji dolžine $|G|$. Od tod takoj vidimo, da je $|V|$ natančna spodnja meja za $|G|$. Enakost je dosežena, ko so prepisovalna pravila oblike $A_0 \to A_1, A_1 \to A_2, \dots, A_{|V|-1} \to a$, kjer je $a \in \Sigma$. Iz $|G| \geq |V|$ sledi $2\,|G| > |V|$. □

Trditev 4.20. *Naj bo G dopustna gramatika in G' dopustna gramatika dobljena z uporabno nekegega skrčitvenega pravila na G. Potem je $C(G') < C(G)$.*

Dokaz. Z V označimo nekončne simbole od G in z V' nekončne simbole od G'. Poglejmo si $C(G')$ za vsako skrčitveno pravilo:

1. $|V'| = |V| - 1$ in $|G'| = |G| - 1$.
   Sledi $C(G') = C(G) - 1$.
2. $|V'| = |V| - 1$ in $2 + |\beta| = |G'| \leq |G| = 2\,|\beta|$, saj je $|\beta| \geq 2$.
   Sledi $C(G') \leq C(G) - 1$.
3. Enak izračun kot prejšnja točka.
   Sledi $C(G') < C(G)$.
4. $|V'| = |V|$ in $1 + |\beta| = |G'| < |G| = 2\,|\beta|$, saj je $|\beta| \geq 2$.
   Sledi $C(G') < C(G)$.
5. Ker zamenjamo vse B, ki nastopajo kot desni člani prepisovalnih pravil, z A, postane B neuporaben simbol. Torej je $|V'| < |V|$, saj odstranimo B in tudi $|G'| < |G|$, saj odstranimo prepisovalno pravilo v katerem B nastopa kot levi član.
   Sledi $C(G') < C(G)$.

□

Izrek 4.21. *Iz dopustne gramatike G dobimo neskrčljivo gramatiko G' z uporabo največ $C(G) - 1$ skrčitvenih previl.*

Dokaz. Sledi direktno iz trditve 4.19 in trditve 4.20 □

Z skrčitvenimi pravili zasnujemo različna neskrčljiva prirejanja gramatike. Poglejmo si dve taki prirejanji.

Definicija 4.22. Za podani niz w začnemo z trivialno dopustno gramatiko $S \to w$. Dokler lahko, uporabi eno izmed skrčitvenih pravil $2, 3, 4$. Dobimo neskrčljivo dopustno gramatiko G_w^{nup}

Da dobimo kanonično obliko ustrezno preimenujemo nekončne simbole. Prirejanje $w \mapsto [G_w^{\text{nup}}]$ imenujemo *metoda najdaljšega ujemajočega podniza.*

Primer 4.23. Z metodo najdaljšega ujemajočega podniza priredimo nizu

$$w = 0110111001100111000111011011 0111.$$

neskrčljivo gramatiko.

1. Začnemo z trivialno dopustno gramatiko G_w:

$$S \to 01101110011001110001110110110111.$$

Izračunamo $C(G_w) = 63$. Podčrtamo najdaljši podniz v prepisovalnem pravilu, ki se pojavi dvakrat na neprekriavjočih se mestih:

$$S \to \underline{0110111}\,0011001110001110 11\,\underline{0110111}.$$

Uporabimo skrčitveno pravilo 2. Dobimo prepisovalni pravili

$$S \to A001100111000111011A,$$
$$A \to 0110111.$$

2. Podčrtamo najdaljši podniz v prepisovalnem pravilu, ki se pojavi dvakrat na neprekriavjočih se mestih:

$$S \to A0011\,\underline{001110}\,\underline{001110}\,11A,$$
$$A \to 0110111.$$

Uporabimo skrčitveno pravilo 2. Dobimo prepisovalna pravila

$$S \to A0011BB11A,$$
$$A \to 0110111,$$
$$B \to 001110.$$

3. Podčrtamo najdaljši podniz v prepisovalnem pravilu, ki se pojavi dvakrat na neprekriavjočih se mestih:

$$S \to A\,\underline{0011}\,BB11A,$$
$$A \to 0110111,$$
$$B \to \underline{0011}\,10.$$

Uporabimo skrčitveno pravilo 3. Dobimo prepisovalna pravila

$$\begin{aligned}
S &\to ACBB11A,\\
A &\to 0110111,\\
B &\to C10,\\
C &\to 0011.
\end{aligned}$$

4. Podčrtamo najdaljši podniz v prepisovalnem pravilu, ki se pojavi dvakrat na neprekriavjočih se mestih:

$$\begin{aligned}
S &\to ACBB11A,\\
A &\to \underline{011}\,\underline{011}\,1,\\
B &\to C10,\\
C &\to 0\,\underline{011}.
\end{aligned}$$

Uporabimo skrčitveno pravilo 2 (lahko bi uporabili tudi skrčitveno pravilo 3). Dobimo prepisovalna pravila

$$\begin{aligned}
S &\to ACBB11A,\\
A &\to DD1,\\
B &\to C10,\\
C &\to 0011,\\
D &\to 011.
\end{aligned}$$

5. Podčrtamo najdaljši podniz v prepisovalnem pravilu, ki se pojavi dvakrat na neprekriavjočih se mestih:

$$\begin{aligned}
S &\to ACBB11A,\\
A &\to DD1,\\
B &\to C10,\\
C &\to 0\,\underline{011},\\
D &\to \underline{011}.
\end{aligned}$$

Uporabimo skrčitveno pravilo 4. Dobimo prepisovalna pravila

$$\begin{aligned}
S &\to ACBB11A,\\
A &\to DD1,\\
B &\to C10,\\
C &\to 0D,\\
D &\to 011.
\end{aligned}$$

6. Podčrtamo najdaljši podniz v prepisovalnem pravilu, ki se pojavi dvakrat na neprekriavjočih se mestih:

$$
\begin{aligned}
S &\to ACBB\,\underline{11}\,A,\\
A &\to DD1,\\
B &\to C10,\\
C &\to 0D,\\
D &\to 0\,\underline{11}.
\end{aligned}
$$

   Uporabimo skrčitveno pravilo 3. Dobimo prepisovalna pravila

$$
\begin{aligned}
S &\to ACBBEA,\\
A &\to DD1,\\
B &\to C10,\\
C &\to 0D,\\
D &\to 0E,\\
E &\to 11.
\end{aligned}
$$

Ne obstaja podniz dolžine vsaj 2, ki bi se pojavil vsaj dvakrat, zato zaključimo. Opomnimo, da je $C(G_w^{\mathrm{nup}}) = 30 < C(G_w) = 63$ in da smo do neskrčljive gramatike G_w^{nup} prišli z uporabo 6 skrčitvenih pravil.

Da dobimo kanonično obliko neskrčljive gramatike ustrezno preimenujemo nekončne simbole. Prepisovalna pravila $[G_w^{\mathrm{nup}}] \in \mathcal{G}^*(\{0,1\})$ so

$$
\begin{aligned}
A_0 &\to A_1A_2A_3A_3A_4A_1,\\
A_1 &\to A_5A_51,\\
A_2 &\to 0A_5,\\
A_3 &\to A_210,\\
A_4 &\to 11,\\
A_5 &\to 0A_4.
\end{aligned}
$$

$\diamondsuit$

4.2 Binarno kodiranje dopustne gramatike

Drugi del stiskanja niza na sliki 5 je binarno kodiranje pravil dopustne gramatike, ki jo priredimo nizu. V razdelku predstavimo glavni izrek dela in sicer eksplicitno podamo binarni kod prepisovalnih pravil dopustne gramatike, ki ga lahko enolično dekodiramo.

Definicija 4.24. *Binarno kodiranje dopustne gramatike* je preslikava

$$
\begin{aligned}
B\colon \mathcal{G}(\mathcal{A}) &\to \{0,1\}^+,\\
G &\mapsto B(G).
\end{aligned}
$$

Definicija 4.25. Naj bo $G \in \mathcal{G}(\mathcal{A})$. Naj bo

$$\rho_G = f_G(A_0) f_G(A_1) \cdots f_G(A_{|V|-1})$$

niz vseh desnih članov prepisovalnih pravil. Niz ρ_G brez prve pojavitve nekončnih simbolov $A_1, \ldots, A_{|V|-1}$ označimo z ω_G. *Entropija gramatike G* je

$$H(G) = |\omega_G| \cdot H(\omega_G).$$

Primer 4.26. Spomnimo se dopustne gramatike $G \in \mathcal{G}(\{a, b\})$ iz primera 4.2. Prepisovalna pravila so

$$P = \{A_0 \to A_1 a A_2, A_1 \to A_3 b, A_2 \to a A_4, A_3 \to ab, A_4 \to b A_1\}.$$

Potem je

$$\rho_G = A_1 a A_2 A_3 b a A_4 a b b A_1,$$
$$\omega_G = abaabbA_1,$$
$$H(G) = 7 \cdot \left(-\frac{3}{7} \log_2 \left(\frac{3}{7}\right) - \frac{3}{7} \log_2 \left(\frac{3}{7}\right) - \frac{1}{7} \log_2 \left(\frac{1}{7}\right)\right) \doteq 10.14.$$

$\diamondsuit$

Izrek 4.27. *Obstaja bijektivno binarno kodiranje dopustne gramatike, da*

1. $\forall G_1, G_2 \in \mathcal{G}(\mathcal{A}), G_1 \neq G_2$ *niz* $B(G_1)$ *ni predpona niza* $B(G_2)$,
2. $\forall G \in \mathcal{G}(\mathcal{A})\colon\ |B(G)| \leq |\mathcal{A}| + 4\,|G| + \lceil H(G) \rceil$.

Dokaz. Abeceda $\mathcal{A}$ in njen vrstni red sta poznana tako kodirniku kot dekodirniku. Spomnimo se definicije eniške kode 2.11 in leksikografskega kodiranja niza 2.22. Vsakemu $G \in \mathcal{G}(\mathcal{A})$ priredimo kodo $B(G) = B_1 B_2 B_3 B_4 B_5 B_5$, kjer je:

- B_1 eniška koda $|V|$. Sledi $|B_1| = |V| \leq |G|$.
- B_2 je stik vseh eniških kod dolžin desnih članov prepisovalnih pravil. Sledi $|B_2| = |G|$.
- B_3 dobimo tako, da se sprehodimo čez ρ_G in prvo pojavitev vsakega nekončnega simbola V zamenjamo z 1, ostale pa z 0. Sledi $|B_3| = |G|$.
- B_4 je niz, kjer za vsak element iz $\mathcal{A}$ v abecednem vrstnem redu z 1 označimo, če je vsebovan v Σ in z 0, če ni. Sledi $|B_4| = |\mathcal{A}|$;
- B_5 je stik eniških kod frekvenc $f(y|\rho_G)$ za vsak $y \in (V \cup \Sigma) \setminus S$. Najprej staknemo frekvence končnih simbolov v abecednem vrstnem redu nato pa nekončne simbole v naravnem abecednem vrstnem redu. Sledi $|B_5| = |G|$.
- B_6 je binarni zapis $i_{s(\omega_G)}(\omega_G)$. Sledi $|B_6| = \lceil \log_2(S(\omega_G)) \rceil$.

Lema 2.34 pove, da je $|S(\omega_G)| \leq 2^{|\omega_G| \cdot H(\omega_G)}$. Od tod sledi, da je

$$|B(G)| \leq |\mathcal{A}| + 4\,|G| + \lceil H(G) \rceil.$$

$\square$

Opomba 4.28. Koda $B_4B_5B_6$ je skoraj enaka kot leksikografkso kodiranje niza ω_G. Razlika je le v B_5, saj sedaj ne kodiramo le frekvenc simbolov, ki nastopajo v ω_G, temveč vse frekvence simbolov, ki nastopajo v ρ_G.

Primer 4.29. Naj bo $\mathcal{A} = \{a, b, c\}$ z abecednim vrstnim redom $a < b < c$. Binarno kodirajmo dopustno gramatiko $G \in \mathcal{G}(\mathcal{A})$ iz primera 4.2. Prepisovalna pravila so

$$\begin{aligned} A_0 &\to A_1aA_2, \\ A_1 &\to A_3b, \\ A_2 &\to aA_4, \\ A_3 &\to ab, \\ A_4 &\to bA_1. \end{aligned}$$

Torej je $V = \{A_0, A_1, A_2, A_3, A_4\}$, $\Sigma = \{a, b\}$ in $S = A_0$. V prejšnjem primeru 4.26 smo izračunali

$$\begin{aligned} \rho_G &= A_1aA_2A_3baA_4abbA_1, \\ \omega_G &= abaabbA_1. \end{aligned}$$

Posameze kode so:

- $B_1 = 00001$;
- $B_2 = 00101010101$;
- $B_3 = 10110010000$;
- $B_4 = 110$;
- $B_5 = 00100101111$;
- $B_6 = 00010100$, saj je niz ω_G le preimenovanje niza $w = 1211223$ kot v primeru 2.24. V primeru 2.23 smo nizu w že izračunali leksikografski indeks.

Staknemo in dobimo

$$B(G) = 0000100101010101101100100001100010010111100010100.$$

Iz $|\mathcal{A}| = 3$, $|G| = 11$ in $\lceil H(G) \rceil = 11$, kar smo izračunali v primeru 4.26, sledi

$$B(G) = 49 \leq 58.$$

Dekodirajmo isti niz. Iščemo $G \in \mathcal{G}$, da je

$$B(G) = 000010010101010110110010000110001001011110001010000.$$

1. Preberemo niz do prve enice in dobimo $B_1 = 00001$. Pove, da je $|V| = 5$. Natančneje, $V = \{A_0, A_1, A_2, A_3, A_4\}$.

2. Ker je $|V| = 5$, preberemo naslednjih 5 enic in dobimo $B_2 = 00101010101$. Skupaj z B_1 sporoči, da imamo prepisovalna pravila

$$\begin{aligned} A_0 &\to \alpha_1, \\ A_1 &\to \alpha_2, \\ A_2 &\to \alpha_3, \\ A_3 &\to \alpha_4, \\ A_4 &\to \alpha_5, \end{aligned}$$

kjer je $|\alpha_1| = 3$, $|\alpha_2| = 2$, $|\alpha_3| = 2$, $|\alpha_4| = 2$, $|\alpha_5| = 2$ in zato je $|G| = |\alpha_1| + |\alpha_2| + |\alpha_3| + |\alpha_4| + |\alpha_5| = 11$.

3. Preberemo naslednjih $|G|$ mest in dobimo je $B_3 = 10110010000$. Od tod sledi, da je

$$\rho_G = A_1 y_1 A_2 A_3 y_2 y_3 A_4 y_4 y_5 y_6 y_7,$$

kjer so $y_i \in V \cup \Sigma$ za $i = 1, 2, \cdots, 7$ še neznani.

4. Ker poznamo abecedo $\mathcal{A}$ preberemo naslednjih $|\mathcal{A}| = 3$ simbolov in dobimo $B_4 = 110$. Ker poznamo vrstni red, sledi $\Sigma = \{a, b\}$.

5. Preberemo naslednjih $|G|$ mest in dobimo $B_5 = 00100101111$. To sporoči, da je

$$\begin{aligned} f(a|\rho_G) &= 3, \\ f(b|\rho_G) &= 3, \\ f(A_1|\rho_G) &= 2, \\ f(A_2|\rho_G) &= 1, \\ f(A_3|\rho_G) &= 1, \\ f(A_4|\rho_G) &= 1. \end{aligned}$$

Izračunamo koliko ponovitev posameznega simbola je še neznanih v nizu $\rho_G = A_1 y_1 A_2 A_3 y_2 y_3 A_4 y_4 y_5 y_6 y_7$.

$$\begin{aligned} r_a &= 3, \\ r_b &= 3, \\ r_{A_1} &= 1, \\ r_{A_2} &= 0, \\ r_{A_3} &= 0, \\ r_{A_4} &= 0. \end{aligned}$$

To pove, da je $S(\omega_G) = \Big\{ u \in \Sigma^* \mid \forall y \in (V \cup \Sigma) \setminus S \colon f(y|u) = f(y|\omega_G) \Big\}$.

6. Še neprebrana mesta so $B_6 = 0001010000$, torej je $i_{s(\omega_G)}(\omega_G) = 20$. Z algoritmom iz definicije 2.22 poiščemo ω_G. To smo že storili v primeru 2.23 za niz $w = 1211223$. Sledi, da je

$$\omega_G = abaabbA_1.$$

Ker je $\omega_G = y_1y_2y_3y_4y_5y_6y_7$ je

$$\rho_G = A_1aA_2A_3baA_4abbA_1.$$

Ker poznamo dolžine desnih članov prepisovalnih pravil, iz niza ρ_G dopolnimo prepisovalna pravila iz 2. točke.

$$\begin{aligned}
A_0 &\to A_1aA_2,\\
A_1 &\to A_3b,\\
A_2 &\to aA_4,\\
A_3 &\to ab,\\
A_4 &\to bA_1.
\end{aligned}$$

Ker je gramatika dopustna, je dovolj da podamo le prepisovalna pravila. Uspešno smo dekodirali binarno kodo in rekonstruirali začetno dopustno gramatiko G.

$\diamondsuit$

4.3 Stiskanje niza

V prvem razdelku poglavja smo spoznali dve prirejanji gramatike: asimptotsko kompaktno in neskrčljivo. V drugem razdelku smo spoznali binarno kodiranje gramatike. Sedaj ti dve preslikavi združimo v *stiskanje niza abecede $\mathcal{A}$ z gramatikami $\mathcal{G}(\mathcal{A})$*

Definicija 4.30. *Stiskanje niza abecede $\mathcal{A}$ z gramatikami $\mathcal{G}(\mathcal{A})$* je par preslikav kodne in dekodne preslikave $\varPhi = (\kappa, \delta)$. Kodna preslikava je

$$\begin{aligned}
\kappa\colon A^+ &\to \{0,1\}^+,\\
w &\mapsto B(\pi(w)),
\end{aligned}$$

kjer je π prirejanje gramatike nizu abecede $\mathcal{A}$ in B binarno kodiranje dopustne gramatike iz izreka 4.27.

Definicija 4.31. S $\varPi_{as}(\mathcal{A})$ označimo vsa stiskanja niza abecede $\mathcal{A}$ z $\mathcal{G}(\mathcal{A})$, kjer je π asimptotsko kompaktno prirejanja gramatike. Elementom pravimo *stiskanje niza abecede $\mathcal{A}$ z asimptotsko kompaktnim prirejanjem.* S $\varPi_{nk}(\mathcal{A})$ označimo vsa stiskanja niza abecede $\mathcal{A}$ z $\mathcal{G}(\mathcal{A})$, kjer je π neskrčljivo prirejanje gramatike. Elementom pravimo *stiskanje niza abecede $\mathcal{A}$ z neskrčljivim prirejanjem.*

V sledečih dveh izrekih uporabimo masksimalno točkovno odvečnost reda n kodne preslikave κ glede na družino informacijskih virov abecede $\mathcal{A}$ iz definicije 2.38 in končni vir abecede $\mathcal{A}$ stopnje m iz definicije 2.39. Čeprav ju ne bomo dokazali, sta izreka zanimiva, saj podata zgornjo mejo maksimalne točkovne odvečnosti reda n stiskanja $\varPhi$ glede na družino informacijskih virov $\Lambda_{kv}^m(\mathcal{A})$ in postavita stiskanje niza z z gramatikami $\mathcal{G}(\mathcal{A})$ v širši kontekst stiskanja podatkov. Njuna dokaza najdemo v [7].

Izrek 4.32. *Naj bo $\Phi \in \Pi_{as}(\mathcal{A})$ in $\{\nu_n\}$ zaporedje pozitivnih števil, ki konvergira prot* 0 *tako, da je*

$$\max_{w \in \mathcal{A}^n} \frac{|G_w|}{|w|} = O(\nu_n).$$

Potem za vsak $m \in \mathbb{N}$

$$odv_n\left(\kappa, \Lambda_{kv}^m(\mathcal{A})\right) = O\left(\nu_n \cdot \log_2\left(\frac{1}{\nu_n}\right)\right).$$

Izrek 4.33. *Velja $\Pi_{nk}(\mathcal{A}) \subset \Pi_{as}(\mathcal{A})$ in za vsak $m \in \mathbb{N}$ velja*

$$\max_{\Phi \in \Pi_{nk}(\mathcal{A})} odv_n\left(\kappa, \Lambda_{kv}^m(\mathcal{A})\right) = O\left(\frac{\log_2 \log_2(n)}{\log_2(n)}\right).$$

Prvi izrek pove, da odvečnost stiskanja z asimptotsko kompaktnim prirejanjem konvergira proti 0 v odvisnosti od izbire kodiranja znotraj razreda. Drugi pa, da je stiskanje z neskrčljivim prirejanjem tudi stiskanje z asimptotsko kompaktnim prirejanjem. Pove tudi, da odvečnost konvergira enakomerno proti 0 za vsa stiskanja z neskrčljivim prirejanjem vsaj tako hitro kot $\frac{\log_2 \log_2(n)}{\log_2(n)}$ pomnožena z neko konstanto.

Lempel in Ziv sta predstavila pojem *univerzalnega koda* [17, 16, 18]. To je stiskanje brez izgube, ki zmore učinkovito obravnavati katerikoli informacijski vir, saj dinamično prepozna in izkoristi ponavljajoče vzorece v sporočilu. To zagotavlja skoraj optimalne stopnje stiskanja brez potrebe po predhodnem znanju informacijskega vira. Izkaže se, da iz izreka 4.33 sledi, da je vsako stiskanje $\Phi \in \Pi_{as}(\mathcal{A})$ univerzalen kod. Dokaz najdemo v [7].

Slovar strokovnih izrazov

admissable grammar dopustna gramatika
alphabet abeceda
channel kanal
code kod
concatenation stikanje
context-free grammar kontekstno-neodvisna gramatika
data compression stiskanje podatkov
decoder dekodirnik
discrete memoryless source vir brez spomina
encoder kodirnik
encoding kodiranje
encryption šifriranje
entropy entropija
enumerate encoding leksikografsko kodiranje
error correcting code kod za popravljanje napak
finite state source končni vir
formal grammar formalna gramatika
grammar transform prirejanje gramatike
information source informacijski vir

information theory teorija informacij
irreducible grammar neskrčljiva gramatika
language over an alphabet jezik na abecedi
left-total relation celovita relacija
lossless compression stiskanje brez izgube
lossy compression stiskanje z izgube
maximal pointwise redundancy maksimalna točkovna odvečnost
phrase structure grammar frazeološka strukturna gramatika
prefix predpona
production rules prepisovalno pravilo
reciever sprejemnik
redundancy odvečnost
total order linearna urejenost
unary coding eniško kodiranje
universal code univerzalen kod
unrestricted grammar neomejena gramatika

Literatura

[1] N. Chomsky in M. Schützenberger, *The algebraic theory of context-free languages**, v: Computer Programming and Formal Systems (ur. P. Braffort in D. Hirschberg), Studies in Logic and the Foundations of Mathematics **35**, Elsevier, 1963, str. 118–161, DOI: `https://doi.org/10.1016/S0049-237X(08)72023-8`, dostopno na `https://www.sciencedirect.com/science/article/pii/S0049237X08720238`.

[2] N. Chomsky, *Three models for the description of language*, IRE Transactions on Information Theory **2**(3) (1956) 113–124, DOI: `10.1109/TIT.1956.1056813`.

[3] N. Chomsky, *On certain formal properties of grammars*, Information and Control **2**(2) (1959) 137–167, DOI: `https://doi.org/10.1016/S0019-9958(59)90362-6`, dostopno na `https://www.sciencedirect.com/science/article/pii/S0019995859903626`.

[4] T. Cover, *Enumerative source encoding*, IEEE Transactions on Information Theory **19**(1) (1973) 73–77, DOI: `10.1109/TIT.1973.1054929`.

[5] I. Csiszár in J. Körner, *Information theory: coding theorems for discrete memoryless systems*, 2. izd., Cambridge University Press, 2011.

[6] D. A. Huffman, *A method for the construction of minimum-redundancy codes*, Proceedings of the IRE **40**(9) (1952) 1098–1101, DOI: `10.1109/JRPROC.1952.273898`.

[7] J. Kieffer in E.-H. Yang, *Grammar-based codes: a new class of universal lossless source codes*, IEEE Transactions on Information Theory **46**(3) (2000) 737–754, DOI: `10.1109/18.841160`.

[8] J. Kieffer in dr. *Universal lossless compression via multilevel pattern matching*, Information Theory, IEEE Transactions on **46** (2000) 1227–1245, DOI: `10.1109/18.850665`.

[9] A. Lempel in J. Ziv, *On the complexity of finite sequences*, IEEE Transactions on Information Theory **22**(1) (1976) 75–81, DOI: `10.1109/TIT.1976.1055501`.

[10] D. MacKay, *Information theory, inference and learning algorithms*, Cambridge University Press, 2003.

[11] E. Plotnik, M. Weinberger in J. Ziv, *Upper bounds on the probability of sequences emitted by finite-state sources and on the redundancy of the lempel-ziv algorithm*, IEEE Transactions on Information Theory **38**(1) (1992) 66–72, DOI: `10.1109/18.108250`.

[12] G. Rozenberg in A. Salomaa, *Lindenmayer systems: impacts on theoretical computer science, computer graphics, and developmental biology*, Springer Berlin Heidelberg, 2012.

[13] K. Sayood, *Introduction to data compression*, The Morgan Kaufmann Series in Multimedia Information and Systems, Elsevier Science, 2017.

[14] C. Shannon in W. Weaver, *The mathematical theory of communication*, Illini books, University of Illinois Press, 1949.

[15] SpinningSpark, *International Morse Code - letters*, 2008, [ogled 24. 6. 2024], dostopno na `https://commons.wikimedia.org/wiki/File:International_Morse_Code_-_letters.svg`.

[16] J. Ziv, *Coding theorems for individual sequences*, IEEE Transactions on Information Theory **24**(4) (1978) 405–412, DOI: `10.1109/TIT.1978.1055911`.

[17] J. Ziv in A. Lempel, *A universal algorithm for sequential data compression*, IEEE Transactions on Information Theory **23**(3) (1977) 337–343, DOI: `10.1109/TIT.1977.1055714`.

[18] J. Ziv in A. Lempel, *Compression of individual sequences via variable-rate coding*, IEEE Transactions on Information Theory **24**(5) (1978) 530–536, DOI: `10.1109/TIT.1978.1055934`.