

Kodiranje

V prvem koraku za vhodni niz (v našem primeru GEMMA) sestavimo tabelo verjetnosti (Tabela 1). To storimo tako, da preštejemo pojavitve znakov (frekvenca znaka), ki je prikazana v drugem stolpcu spodnje tabele. Seštevek vseh frekvenc znakov predstavlja kumulativno frekvenco ali dolžino vhodnega niza ($1+1+2+1 = 5$). V tabeli izračunamo tudi spodnjo in zgornjo mejo posameznega simbola. Spodnja meja simbola se vedno začne z 0, zgornjo mejo pa določimo tako, da spodnji meji prištejemo frekvenco trenutnega simbola ($spMeja(x) + Frekvenca(x)$).

Meje posameznega simbola $[a, b)$ so določene z dvema vrednostma, spodnjo mejo in zgornjo mejo. Notacija $[a, b)$ pomeni razpon števil od a pa do b , kjer je a vsebovan, b pa ne.

Znak	Frekvenca	Verjetnost = frekvenca/kumulativna frekvenca	Interval (celo št.)	
			spMeja	zgMeja
G	1	1/5	0	1
E	1	1/5	1	2
M	2	2/5	2	4
A	1	1/5	4	5
	5			

Tabela 1: Verjetnostna tabela

Ko imamo tabelo zgrajeno, inicializiramo kodirnik in pričnemo s kodiranjem niza. Kodiranje izvajamo znak po znak dokler ne pridemo do konca vhodnega niza.

Pri inicializaciji si določimo globalni interval s katerim bomo zakodirali vhodni niz. Tega določimo tako, da spodnjo mejo postavimo na 0, zgornjo mejo pa si izračunamo s pomočjo formule: $zgMeja = 2^{št.bitov} - 1$, kjer je št. bitov nastavev kodirnika in je lahko 8, 16, 32 ali 64. Ko imamo nastavljen globalni interval si izračunamo še pomožne četrtninske vrednosti globalnega intervala (prva četrtnina, druga četrtnina in tretja četrtnina). Nato lahko začnemo s kodiranjem prvega znaka.

Za kodiranje uporabljamo spodnje tri enačbe, ki spreminjajo globalni interval. Kot je bilo že omenjeno, se postopek začne z $[0, 2^{32-1} - 1)$, ob predpostavki, da uporabljamo 32 bitov (v primerih kodiranja in dekodiranja bomo uporabljali 8 bitov za lažjo predstavbo).

$$\begin{aligned}
 korak &= \frac{zgMeja - spMeja + 1}{kumulativnaFrekvenca} && \text{Enačba 1} \\
 zgMeja &= spMeja + korak * zgMeja(x) - 1 && \text{Enačba 2} \\
 spMeja &= spMeja + korak * spMeja(x) && \text{Enačba 3}
 \end{aligned}$$

Pri postopku kodiranja nastane težava, saj se spodnja in zgornja meja vse bolj pomikata ena proti drugi in takoj, ko se približata na interval, ki je manjši od števila različnih simbolov na vhodu, dekodiranje postane nemogoče. Za ta problem uvedemo preslikavi E1 in E2. Ko preračunamo nove meje (Enačba 1, Enačba 2 in Enačba 3), preverimo, če je zgornja meja manjša od polovice celotnega interval ($zgMeja < drugaCetrtnina$) in izvedemo preslikava E1, kjer se meji spremenita na naslednji način

$$\begin{aligned} spMeja &= spMeja \cdot 2 \\ zgMeja &= zgMeja \cdot 2 + 1, \end{aligned}$$

na izhod pa se pošlje bit **0**.

V nasprotnem primeru preverimo, če je spodnja meja v zgornji polovici celotnega intervala ($spMeja \geq drugaCetrtna$) ter izvedemo preslikavo E2, kjer se meji spremenita na naslednji način

$$\begin{aligned} spMeja &= 2 \cdot (spMeja - drugaCetrtna) \\ zgMeja &= 2 \cdot (zgMeja - drugaCetrtna) + 1, \end{aligned}$$

na izhod pa se pošlje bit **1**.

Postopek preverjanja E1 in E2 ponavljamo, dokler noben pogoj ni več izpolnjen.

Čeprav sta preslikavi E1 in E2 korak v pravo smer, ne odpravita težave v celoti. Problem nastane, ko se spodnja in zgornja meja gibljeta v bližini polovice intervala vendar vseeno ostaneta vsak na svoji polovici.

Tukaj izvedemo preslikavo E3. Uporabimo spremenljivki *prvaCetrtna* in *tretjaCetrtna*. Takoj, ko je $spMeja \geq prvaCetrtna$ in $zgMeja < tretjaCetrtna$, izvedemo preslikavo E3. Pri tem spremenimo obe meji na naslednji način:

$$\begin{aligned} spMeja &= 2 \cdot (spMeja - prvaCetrtna) \\ zgMeja &= 2 \cdot (zgMeja - prvaCetrtna) + 1; \end{aligned}$$

Vodimo si še števec E3_Counter, ki beleži kolikokrat se je preslikava E3 izvedla. V naslednji iteraciji kodiranja, ko nastopi preslikava E1 ali E2 upoštevamo števec E3_Counter, in sicer, če se izvede preslikava E1, damo na izhod bit **1** tolikokrat, kolikokrat se je izvedla preslikava E3 (E3_Counter). Če se izvede preslikava E2, pa damo na izhod bit **0** tolikokrat, kolikokrat se je izvedla preslikava E3 (E3_Counter). Ko zapišemo število enic oz. ničel pa ponastavimo E3_Counter na vrednost 0.

Po končanem kodiranju moremo na izhod poslati ostanek, ki ga določimo z naslednjim pogojem: če je *spMeja* manjša od *prvaCetrtna*, pošljemo na izhod bita **01** in E3_Counter krat bit **1** (če je E3_Counter = 4 bi poslali na izhod **01 1111**).

V nasprotnem primeru pošljemo na izhod bita **10** in E3_Counter krat bit **0** (če je E3_Counter = 4 bi poslali na izhod **10 0000**).

Primer kodiranja:

Kodirali bomo niz **GEMMA**.

Za lažjo predstavitev se bo v spodnjem primeru uporabljalo 8 bitov za predstavitev globalnega intervala.

Uporabljala se bo verjetnostna tabela (Tabela 1).

Inicializacija:

$$spMeja = 00000000[0]$$

$$zgMeja = 2^{n-1} - 1 = 2^{8-1-1} - 1 = 11111111[127]$$

$$drugaCetrtna = \left\lfloor \frac{zgMeja + 1}{2} \right\rfloor = 10000000[64]$$

$$prvaCetrtna = \left\lfloor \frac{drugaCetrtna}{2} \right\rfloor = 01000000[32]$$

$$tretjaCetrtna = \lfloor prvaCetrtna \cdot 3 \rfloor = 11000000[96]$$

Enačbe:

$$korak = \frac{zgMeja - spMeja + 1}{comulativnaFrekvenca}$$

$$zgMeja = spMeja + korak * zgMeja(x) - 1$$

$$spMeja = spMeja + korak * spMeja(x)$$

Znak	Meja	Korak	Meje	Meje ₍₂₎	Izhod	E1/E2	E3
G	L	$\frac{127 - 0 + 1}{5} = 25$	$0 + 25 * 0 = 0$	0000000 [0]	00	0	0
	H		$0 + 25 * 1 - 1 = 24$	0011000 [24]		99	99
E	L	$\frac{99 - 0 + 1}{5} = 20$	$0 + 20 * 1 = 20$	0010100 [20]	0	40	16
	H		$0 + 20 * 2 - 1 = 39$	0100111 [39]		79	95
M	L	$\frac{95 - 16 + 1}{5} = 16$	$16 + 16 * 2 = 48$	0110000 [48]		48	0
	H		$16 + 16 * 4 - 1 = 79$	1001111 [79]		79	127
M	L	$\frac{127 - 0 + 1}{5} = 25$	$0 + 25 * 2 = 50$	0110010 [50]		50	50
	H		$0 + 25 * 4 - 1 = 99$	1100011 [99]		99	99
A	L	$\frac{99 - 50 + 1}{5} = 10$	$50 + 10 * 4 = 90$	1011010 [90]	1 000	52	16
	H		$50 + 10 * 5 - 1 = 99$	1100011 [99]		71	95
Ostanek					01 11		

Rezultat: 00 0 1 000 01 11