

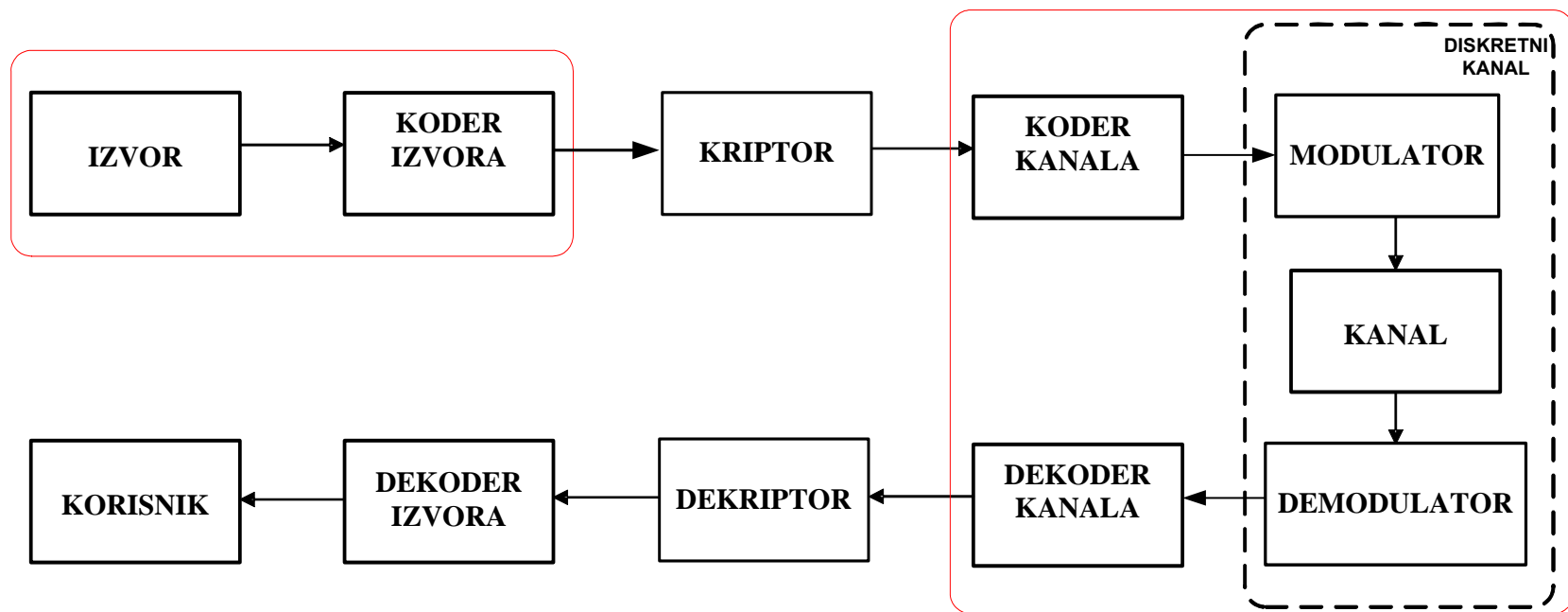
Univerzitet u Beogradu
Elektrotehnički fakultet

Principi modernih telekomunikacija

1v. Diskretni izvori, koder izvora

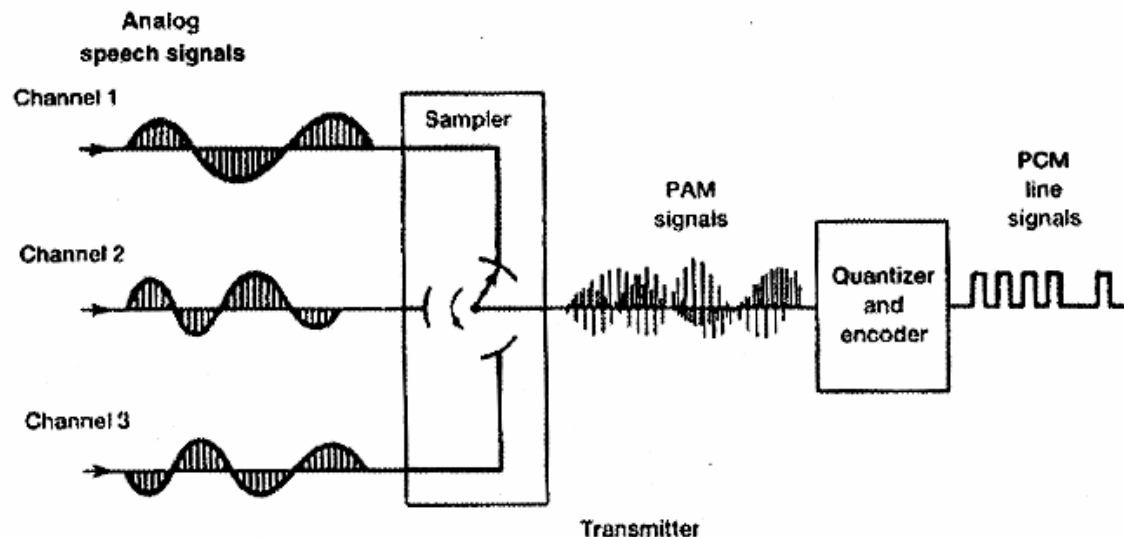
Opšta blok šema

- Za početak posmatramo prenos na nivou bita
 - Iz izvora izlaze biti
 - U diskretni kanal ulaze biti



Vrste izvora

- Da li je poznata priroda podataka koje emituje izvor – štampani tekst, govor, slika, video?
- Podaci se uvek mogu digitalizovati
 - Ovaj proces zavisi od vrste signala i širine spektra;
 - A/D konverzija, odabiranje, kvantizacija
 - Proces zavisi od prirode podataka koje emituje izvor – štampani tekst, govor, slika, video.



ASCII tabela – za štampani tekst

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Diskretni izvor bez memorije

- Opisuju se:
 - skupom mogućih poruka $S=\{s_1, s_2, \dots, s_N\}$
 - verovatnoćama pojavljivanja pojedinih simbola iz ovog skupa $P(s_i)$, $i=1, \dots, N$.
- Primer
 - Izvor emituje šest simbola – A, B, C, D, E, F
 - Verovatnoće pojavljivanja
 $P(A)=0.5, P(B)=0.2, P(C)=0.1, P(D)=0.1, P(E)=0.07, P(F)=0.03$
 - Primer sekvence
ADAABABCAEBAAACCBFAFADABADABAEAE

Količina informacija

- Najjednostavniji način – pomoću logaritma

$$Q(s_i) = \log[1/P(s_i)]$$

- Funkcija mora da zadovolji sledeće
 - Količina informacija ne može biti negativna.
 - Ako je verovatnoća pojave simbola ravna jedinici, događaj je siguran i simbol ne nosi nikakvu informaciju prijemniku $\log(1)=0$;
 - Ako su simboli nezavisni, količine informacija koju oni nose se sabiraju

$$Q(s_i s_k) = \log[1/P(s_i, s_k)] = \log[1/P(s_i)P(s_k)] = Q(s_i) + Q(s_k)$$

- Ako je baza logaritma 2 jedinica je Šenon (*Claude Shannon*).
- Prethodni primer:

$$Q(A) = \log_2(1/0.5) = \log_2(2) = 1[\text{Sh}], \quad Q(F) = \log_2(1/0.03) = 5.06[\text{Sh}],$$

Entropija

- Entropija predstavlja prosečnu “meru neizvesnosti (neopredeljenosti)” posmatrača o tome šta će izvor da emituje.
 - Emitovanjem pojedinih simbola izvor emituje u proseku tačno potrebnu količinu informacija i upravo potpuno razrešava ovu neizvesnost.

$$H(S) = \overline{Q(s_i)} = \sum_{i=1}^q P(s_i)Q(s_i) = \sum_{i=1}^q P(s_i) \lg \left(\frac{1}{P(s_i)} \right) = - \sum_{i=1}^q P(s_i) \lg P(s_i) \quad \left[\frac{\text{Sh}}{\text{simb}} \right].$$

- Primer
 - $H(S) = P(A)Q(A) + P(B)Q(B) + P(C)Q(C) + P(D)Q(D) + P(E)Q(E) + P(F)Q(F)$
 $= 0.5 * 1[\text{Sh}] + 0.2 * 2.32[\text{Sh}] + \dots + 0.07 * 5.06[\text{Sh}] = 2.05 [\text{Sh/simb}]$

Efikasan prenos

- Neka sekvencu koju emituje diskretni kanal želimo da predstavimo u diskretnom obliku
- ASCII kod – svaki simbol se predstavlja sa 7 bita
- Prethodni primer
 - Za prenos 30 slova iz prikazane sekvence potrebno je $30 \cdot 7 = 210$ bita.
 - Koliko god ima slova potrebno je sedam puta više bita za njihov prenos.
- Da li se poruka može predstaviti manjim brojem bita a da se ne naruši informacija koja se prenosi?
 - Želimo da pravilno rekonstruišemo svih 30 slova na strani prijema.

Algoritmi za nedestruktivnu kompresiju

- **Dva osnovna tipa:**
 - Kodovi zasnovani na verovatnoći pojavljivanja simbola (statički);
 - Kodovi zasnovani na strukturi rečnika (dinamički).
- **Statički algoritmi**
 - Šenon-Fanoov postupak;
 - Hafmenov algoritam;
 - Adaptivni Hafmen (dinamička verzija statičkog algoritma).
- **Dinamički algoritmi**
 - Lempel-Zivov (LZ) algoritam;
 - Njegove modifikacije – LZ77, LZ78, LZW.

Šenon-Fanoov postupak

- Simboli izvorne liste uredi se po nerastućim verovatnoćama (ako simboli imaju podjednake verovatnoće, njihov redosled u okviru toga podskupa nije važan)
- Zatim se skup simbola podeli na dva podjednako verovatna dela (ili bar približno podjednako verovatna dela).
 - Kodne reči za simbole jednog podskupa počinjaće sa 0, a za simbole drugog podskupa sa 1.
- Svaki od ovih podskupova se dalje deli na dva podjednako verovatna ili približno podjednako verovatna podskupa dodajući na odgovarajuće mesto u kodnoj reči 0 ili 1.
 - Kada u svim podskupovima ostane samo po jedan simbol kodovanje je završeno.
- Osnovna ideja je jednostavna – deljenjem na podjednako verovatne (ili približno podjednako verovatne) podskupove postiže se da je verovatnoća pojavljivanja 0 i 1 na odgovarajućem mestu u kodnoj reči ista (po 0,5) ili približna, pa će taj bit nositi maksimalnu (ili skoro maksimalnu) količinu informacija od 1 šenon.

Šenon-Fanoov postupak, primer

a)

S	P_i	I podela	II podela	III podela
s_1	0,5	<u>0</u>	<u>0</u>	<u>0</u>
s_2	0,25	1	<u>10</u>	<u>10</u>
s_3	0,125	1	11	<u>110</u>
s_4	0,125	1	11	111

b)

S	P_i	I podela	II podela	III podela	IV podela
s_1	0,6	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
s_2	0,2	1	<u>10</u>	<u>10</u>	<u>10</u>
s_3	0,1	1	11	<u>110</u>	<u>110</u>
s_4	0,07	1	11	111	<u>1110</u>
s_5	0,03	1	11	111	1111

c)

S	P_i	I podela	II podela	III podela
s_1	0,3	<u>0</u>	<u>00</u>	<u>00</u>
s_2	0,2	<u>0</u>	<u>01</u>	<u>01</u>
s_3	0,2	1	<u>10</u>	<u>10</u>
s_4	0,2	1	11	<u>110</u>
s_5	0,1	1	11	111

Hafmenov postupak

- Hafmenov kod koji odgovara izvoru koji:
 - emituje šest simbola
 - verovatnoće zadate u drugoj koloni tabele:
- Postupak
 - Poređati po opadajućim verovatnoćama
 - Sažimati po dva simbola i dati im isti prefiks

[illegible]

Srednja dužina kodne reči, efikasnost koda

- Srednja dužina kodne reči:

$$L_{sr} = 0.5 * 1 + 0.2 * 2 + 0.1 * 3 + 0.1 * 4 + 0.07 * 5 + 0.03 * 5 = 2.1 \text{ [} b / s \text{]}$$

- Entropija izvora

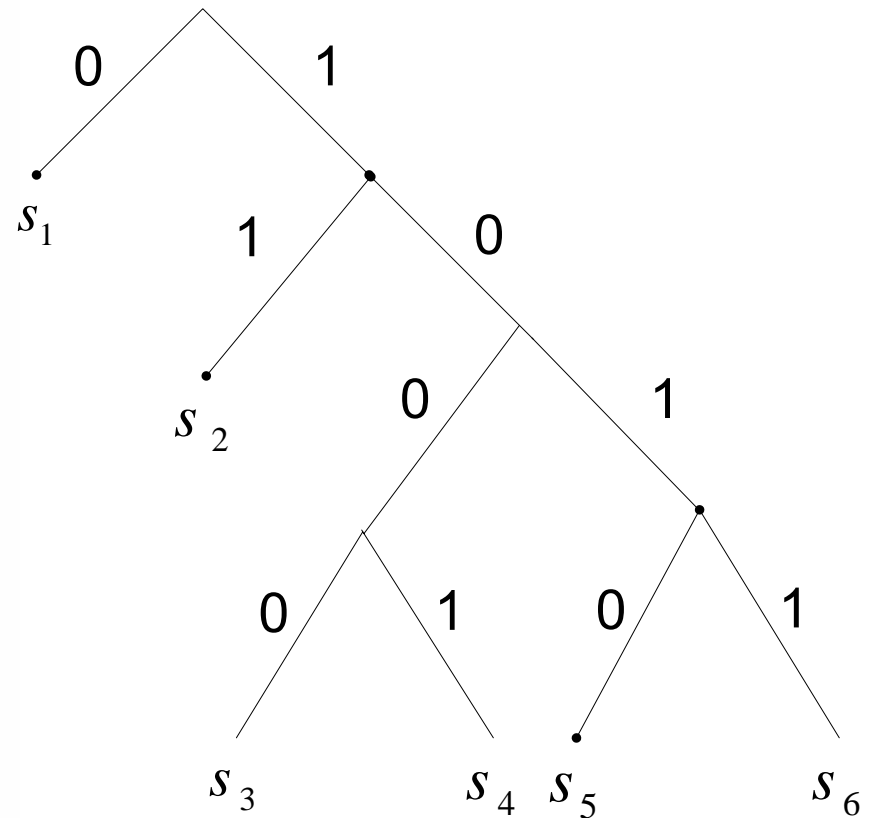
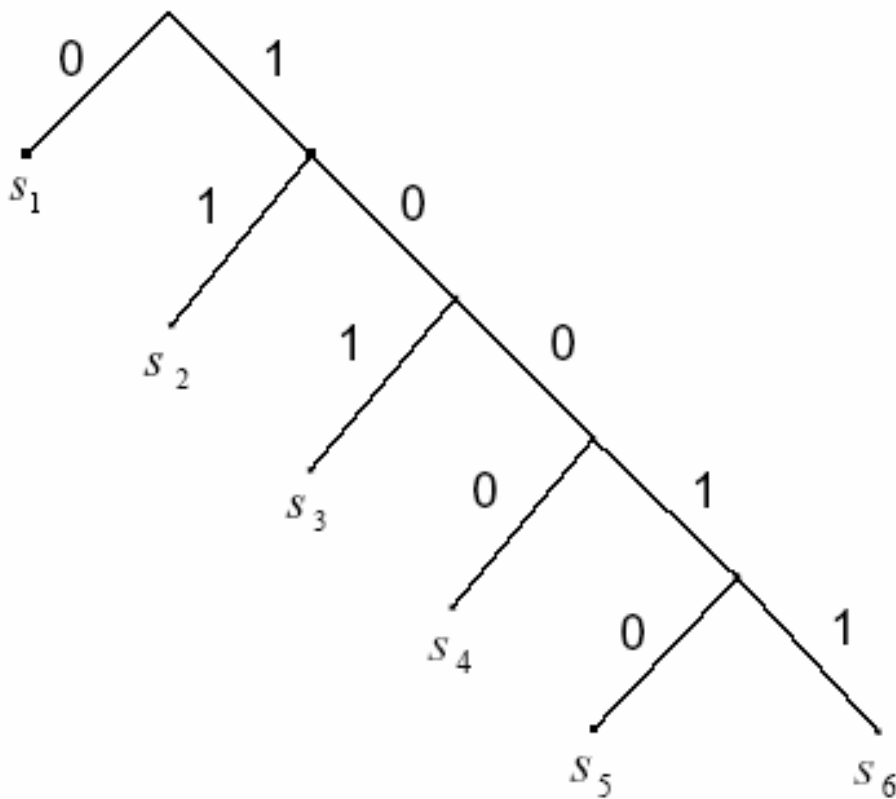
$$H(s) = \sum_{i=1}^6 P(s_i) \lg \frac{1}{P(s_i)} = 2.0502 \text{ [} Sh / simb \text{]}$$

- Efikasnost

$$\eta = \frac{H(s)}{L_{sr}} \cdot 100\% = 97.63\%$$

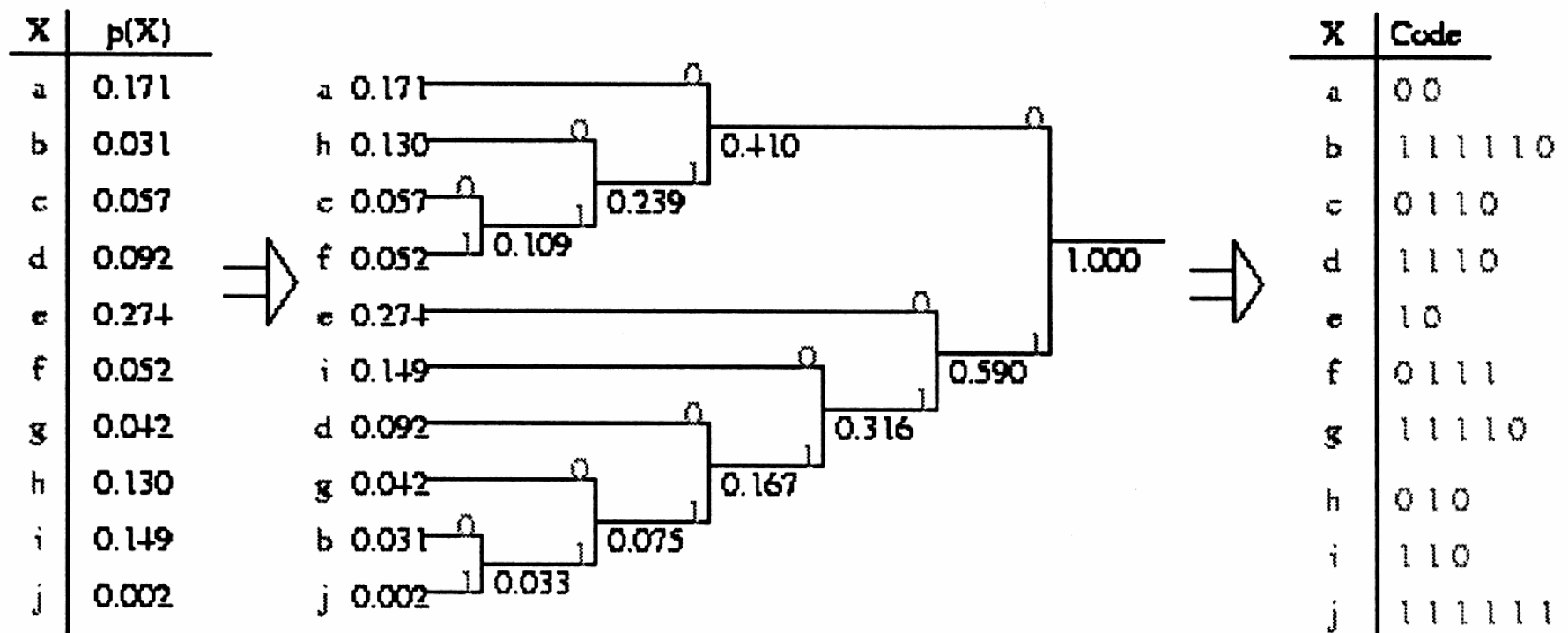
Predstava pomoću stabla

- Zavisno od redosleda sažimanja (ista dužina kodne reči!):



Predstava pomoću stabla, drugi primer

- Deset simbola izvorne liste
- Dužina kodne reči bitno zavisi od verovatnoće pojavljivanja simbola kome je reč pridružena.



Proširenje izvora

- Ako se umesto pojedinih simbola posmatraju sekvence od po 2, 3 ili više (n) sukcesivnih simbola, tada se kaže da se posmatra drugo, treće ili ***n-to proširenje izvora***.
 - Ono se obično obeležava sa S^n a broj njegovih simbola je upravo q^n .
 - Drugim rečima, n -to proširenje izvora je izvor čiji su simboli sekvence od po n simbola prvobitnog izvora.
- Primer
 - Originalni izvor emituje poruke iz skupa $S=\{A, B, C\}$ sa verovatnoćama $P(A)=1/2$, $P(B)=1/4$, $P(C)=1/4$;
 - Proširenje izvora “emituje” složene simbole iz sledećeg skupa:
 $S^2=\{AA, AB, AC, BA, BB, BC, CA, CB, CC\}$.

Entropija proširenja izvora

- Računa se na osnovu verovatnoća složenih simbola $\sigma_i = s_i s_k$, za izvore bez memorije važi $P(s_i, s_k) = P(s_i)P(s_k)$.
- Pritom važi relacija

$$H(S^2) = 2H(S).$$

- Prethodni primer:
 - $H(S) = 1/2 * 1 + 1/4 * 2 + 1/4 * 2 = 1.5$ [Sh/simb]
 - $P(AA) = 0.5 * 0.5 = 0.25, \dots, P(CC) = 0.25 * 0.25 = 0.0625$
 - Po definiciji:
 - $H(S^2) = 0.25 * \text{ld}(4) + \dots + 0.0625 * \text{ld}(16) = 3$ [Sh/simb]
 - Ispunjeno je $H(S^2) = 2H(S)$.

Izvori sa memorijom

- Govor, odnosno štampani tekst, je svakako jedan od najvažnijih primera diskretnog niza s memorijom.
- Entropije za neke jezike date su u tabeli:
 - H_{\max} – kad bi sva slova bila podjednako verovatna
 - H_0 – bez memorije, poznate verovatnoće pojavljivanja slova
 - H_1 – verovatnoća pojave svakog slova zavisi samo od prethodnog

Jezik (tekst)	Razmak	H_{\max}	H_0	H_1	H_2
srpski	da	4,95	4,24	3,41	–
hrvatski	da	4,76	4,19	3,59	3,10
ruski	da	5,00	4,05	3,52	3,01
engleski	ne	4,70	4,14	3,56	3,30
engleski	da	4,76	4,03	3,32	3,10
francuski	da	4,76	3,95	3,17	2,83
nemački	da	4,76	4,04	3,42	2,82

Suvišnost

- Suvišnost (redundansa)

$$R = \frac{H_{\max} - H}{H_{\max}} \cdot 100 = \left(1 - \frac{H}{H_{\max}}\right) 100 [\%],$$

- Procenjena entropija i suvišnost za neke jezike:
 - Veliki deo konstrukcija u svakom jeziku je predvidiv;
 - Suvišnost obično iznosi preko 70%.

Jezik	Entropija [Sh/simb]	Suvišnost [%]
engleski	1,30	72,7
ruski	1,37	72,6
francuski	1,40	70,6

Prva Šenonova teorema

- Prva Šenonova teorema – dovoljnim proširivanjem reda izvora može se postići proizvoljno visoka efikasnost:

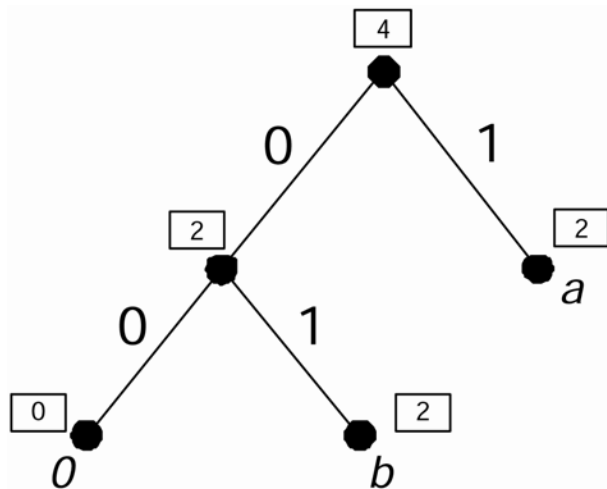
$$\lim_{n \rightarrow \infty} \frac{L_{sr,n}}{nH(s)} = 1$$

- Ovaj izraz važi i za izvore sa memorijom!
- Kompresija najviše može ići do nivoa gde se svaki simbol u proseku predstavlja sa onoliko bita koliko iznosi entropija izvora.

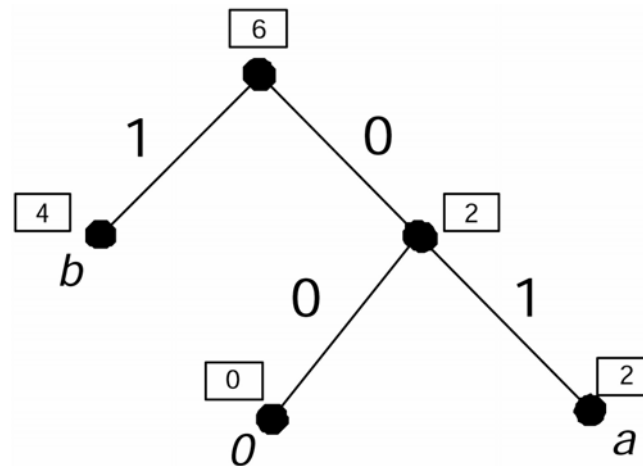
Adaptivni Hafmenov algoritam

- Struktura stabla se menja u zavisnosti od broja pojavljivanja pojedinih simbola na ulazu koderu.

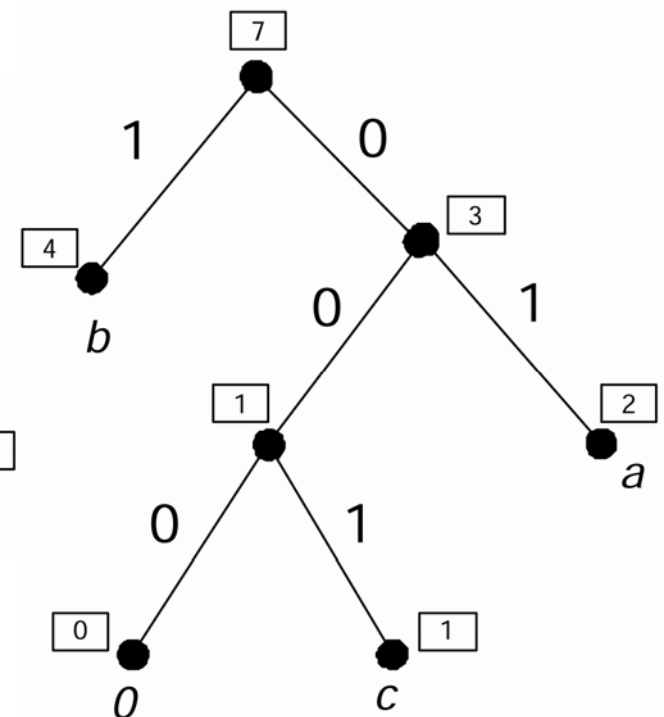
aabb



aabbbb



aabbbbc



Lempel Zivov (LZ) algoritam

- Dve faze:
 - Prvo se formira rečnik na osnovu dela sekvence koju emituje izvor;
 - Kada je rečnik jednom formiran, on se koristi za kompresiju ostalog dela sekvence koju emituje izvor.
- Obično se koristi za kompresiju teksta – na prvih 256 pozicija slova, brojevi i specijalni znaci (prošireni ASCII).
 - Poznavanje ovog (manjeg, standardnog i nezavisnog od statistike prenošene sekvence!) dela rečnika je potreban i dovoljan uslov da se rekonstruiše rečnik i izvrši dekompresija samo na osnovu presretnute sekvence!

LZ, kompresija

abbaabbaaba bb**abb**abb ->0110242 366 (tj. 000 001 001 000 010 100 010 011 110 110)

```

W=NIL;
loop
  read k;
  if wk u rečniku
    w=wk;
  else
    code of w->out
    wk-> tabela stringova
    w=k;
  end;
end loop;

```

rečnik						
adresa	sadržaj	w	k	wk	?	out
0	a					
1	b	nil	a	a	+	
		a	b	ab	-	0
2	ab	b	b	bb	-	1
3	bb	b	a	ba	-	1
4	ba	a	a	aa	-	0
5	aa	a	b	ab	+	
		ab	b	abb	-	2
6	abb	b	a	ba	+	
		ba	a	baa	-	4
7	baa	a	b	ab	+	
		ab	a	aba	-	2

LZ, dekompresija

- Na osnovu primljene sekvence može se jednostavno rekonstruisati rečnik:

0110242 366 -> abbaabbaab bbabbabb

```
W=nil;  
loop  
  read in;  
  out=sadrzaj(in)  
  sadrzaj=w+out(1);  
  w=out;  
end loop;
```

rečnik		w	in	out	w+out(1)
adresa	sadržaj				
0	a				
1	b	nil	0	a	a
		a	1	b	ab
2	ab	b	1	b	bb
3	bb	b	0	a	ba
4	ba	a	2	ab	aa
5	aa	ab	4	ba	abb
6	abb	ba	2	ab	baa
7	baa				

LZ vs. Hafmen

- Neka je sekvenca koju treba komprimovati
ababababababababa
 - Ukupno Ukupno
 - Dve podsekvence dužine 2 (ab,ba)
 - Dve podsekvence dužine 3 (aba,bab)
 - Dve podsekvence dužine 4 (abab,baba)
 - ...
 - Ako rečnik ima 16 adresa (kod sa 4 bita)
 - > na adr. 14. i 15. će biti sekvence dužine 8
 - Ako rečnik ima 4096 adresa (kod sa 12 bita)
 - > na adr. 4094. i 4095. će biti sekvence dužine **2048!**
 - U realnosti rečnik ima ukupno 4096 pozicija, prvih 256 su osnovni simboli (0-255), na preostalim (256-4095) su izvedeni simboli (kombinacije osnovnih).
 - Statistička zavisnost je znatno manja nego u navedenom primeru ali dovoljna da štampani tekst radi bolje nego Hafmen.
- | rečnik | |
|--------|---------|
| adresa | sadržaj |
| 0 | a |
| 1 | b |
| 2 | ab |
| 3 | ba |
| 4 | aba |
| 5 | abab |
| 6 | bab |
| 7 | baba |
| 8 | ababa |
| ... | ... |

rečnik	
adresa	sadržaj
0	a
1	b
2	ab
3	ba
4	aba
5	abab
6	bab
7	baba
8	ababa
...	...

Primene algoritama za kompresiju

PRIMENE:

- Nedestruktivna kompresija pisanog teksta ili slike obično se zasniva na LZ kodovima (LZ77, LZW) ili kombinaciji LZ/Hafmen.

<i>Utility</i>	<i>Format</i>	<i>Compression</i>
pkarc (DOS) arc (Unix, Mac, etc.)	.arc, .ark	LZW
arj (DOS)	.arj	LZ77 + hashing, secondary static Huffman
Compuserve GIF	.gif	LZW
gzip	.gz	LZ77 + hashing, secondary static Huffman
lha, lharc	.lha, .lhz	LZ77 + tries, secondary static Huffman
squeeze (DOS)	.sqz	LZ77 + hashing
pkzip (DOS) zip (Unix) WinZip (Windows)	.zip	LZ77 + hashing, secondary static Huffman
zoo (DOS/Mac/Unix)	.zoo	LHA
freeze (Unix)	.F	LZ77 + hashing, secondary adaptive Huffman
yabba (Unix)	.Y	LZ78 variant
compress (Unix)	.Z	LZW

JPEG:

1. do irreversible compression on colour channels
2. compute the *Discrete Cosine Transform* for
3. "reduce" the DCT output: more reduction
4. Huffman encode the reduced output

MPEG

1. do irreversible compression on colour channels (not on shade channel)
2. for each block of 16x16 in a frame, try to find a "similar" block in a previous (*or future*)
3. store the differences between blocks instead of storing entire blocks
4. Huffman encode the whole thing