Univerzitet Sinergija

Konvolucioni kodovi

Ciljevi

Nakon ove lekcije, znaćemo:

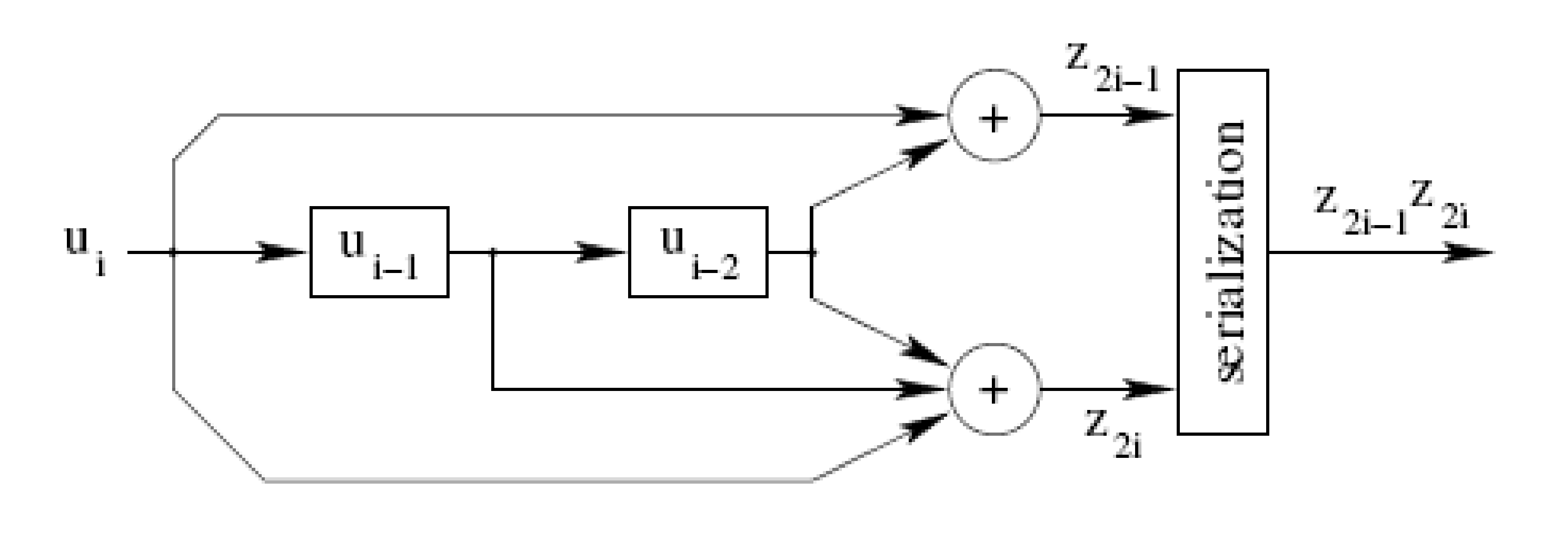
1. Šta su konvolucioni kodovi
2. Kako se vrši kodovanje konvolucionim kodovima
3. Šta je stanje i dijagram stanja
4. Šta je rešetkasta prezentacija vezana za predstavljanje ovih kodova
5. Kako koristiti Viterbijev algoritam na rešetkama, u cilju dekodovanja minimalnim rastojanjem,
6. Kako izračunati minimalno rastojanje konvolucionih kodova.

Uvod

- Konvolucioni kodovi ne pripadaju klasi blok kodova. Mehanizam kodovanja pamti prethodno kodovane simbole.
- U izvesnom smislu se mogu tretirati kao neograničeni blok kodovi.
- Proces kodovanja i dekodovanja se značajno razlikuje.
- Daleko su efikasniji od blok kodova.

Kodovanje konvolucionim kodovima

- Izlaganje o načinu kodovanja konvolucionim kodovima ćemo započeti jednim primerom kodovanja. Šema kodera za odabrani primer je data na sl.6.1. Radi se o binarnom kodu.



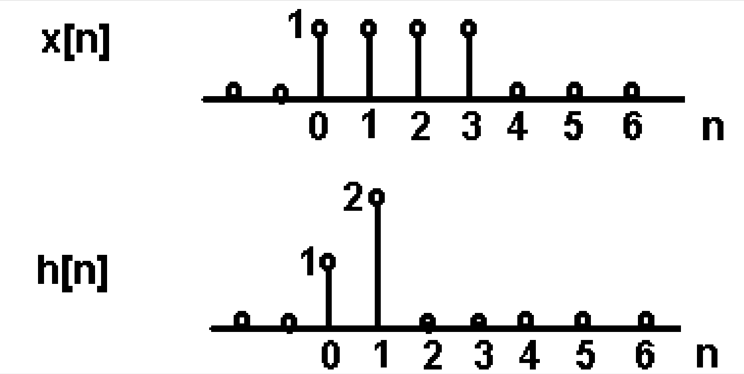
- Sl.6.1 Svaki simbol poruke u_i se koduje u dva simbola z_{2i-1} i z_{2i}

Kodovanje konvolucionim kodovima

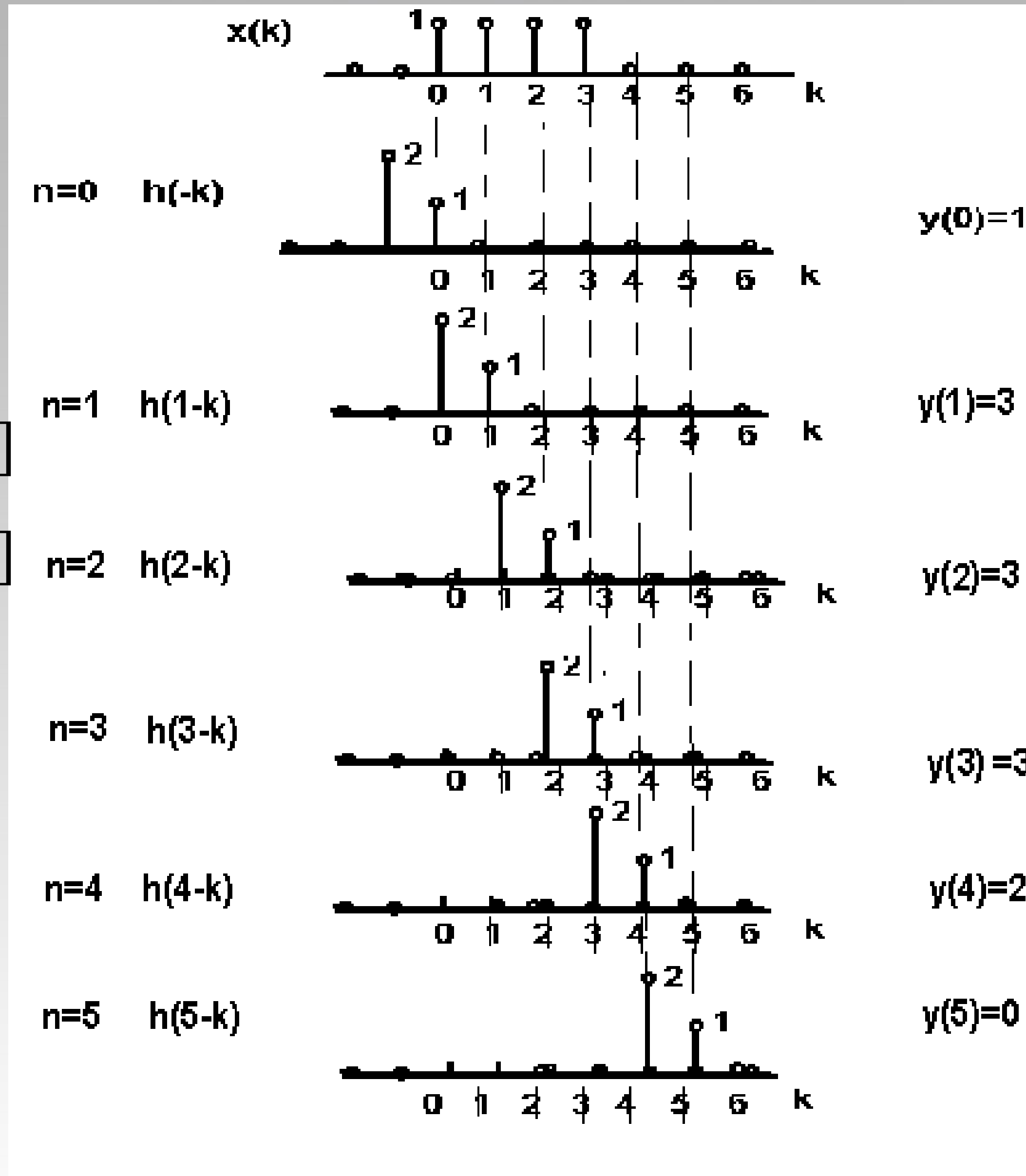
- U svakom vremenskom trenutku i , jedan simbol poruke u_i ulazi u koder, dok dva simbola kodne reči z_{2i-1} i z_{2i} se emituju, odnosno $u = (u_1, u_2, \dots, u_i, \dots)$ se koduje u $z = (z_1, z_2, \dots, z_{2i-1}, z_{2i}, \dots)$. Stoga je brzina ovog koda 1/2.
- Proces kodovanja za primer sa sl.6.1 se može opisati sa
- $z_{2i-1} = u_i + u_{i-2}$ (6.5)
- $z_{2i} = u_i + u_{i-1} + u_{i-2}$ (6.6)
- $u_i \to (u_{i-2} + u_i,\ u_{i-2} + u_{i-1} + u_i)$
- Ove jednačine se mogu posmatrati i kao diskretna konvolucija ulazne sekvence sa sekvencama 1,0,1,0,0... i 1,1,1,0,0....

Primer računanja konvolucije

- Izračunati niz *y*[*n*] generisan konvolucijom nizova *x*[*n*] i *h*[*n*] datih na donjim slikama



$$y[n] = \sum_{k=\infty}^{\infty} x[k]h[n-k]$$



$h(1-k)=h[-(k-1)]$

$h(2-k)=h[-(k-2)]$

$y(0)=x(0)h(0)$

$y(1)=x(0)h(1)$

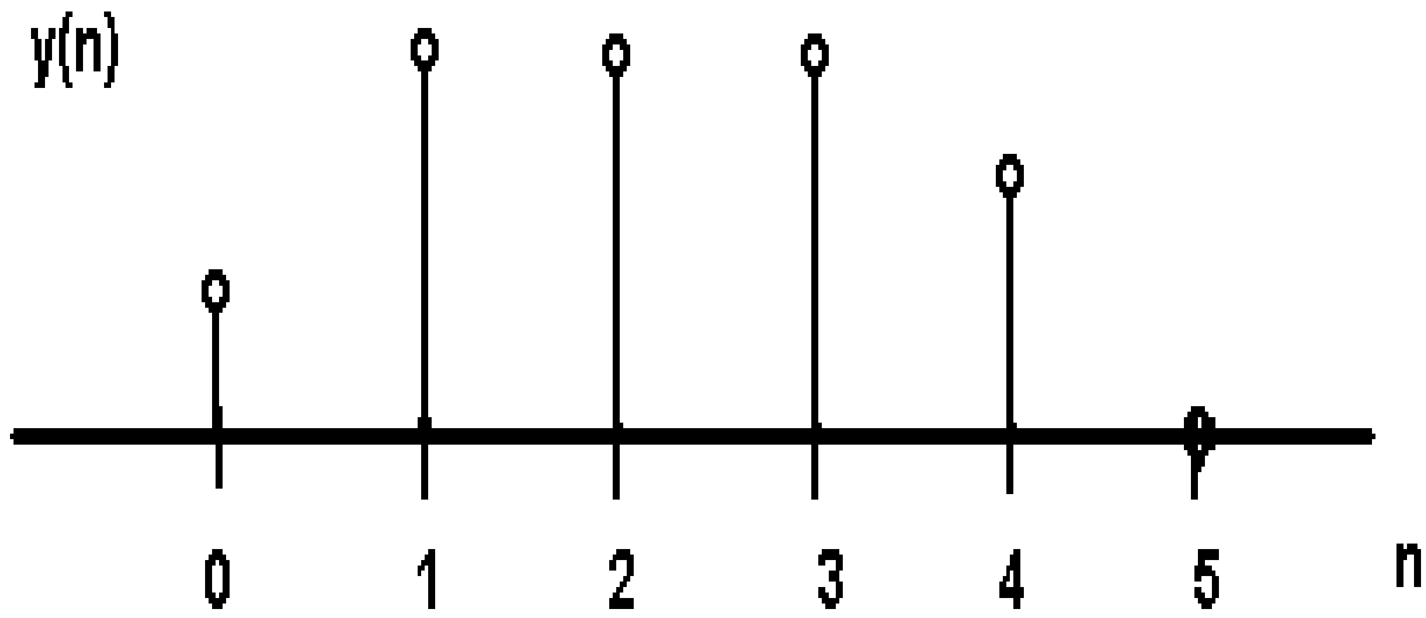
$+x(1)h(0)$

$y(2)=x(2)h(0)$

$+x(1)h(1)$

Primer računanja konvolucije

Primer računanja konvolucije



Kodovanje konvolucionim kodovima

- Ako se vratimo na prethodni primer zapazićemo da proces kodovanja nije u potpunosti odredjen samo konvolucijom, budući da nam nisu poznate prethodne vrednosti u_{i-2} i u_{i-1}. Odnosno, to je ekvivalentno pitanju šta su početna stanja sistema u trenutku t=0?
- Pogodno je usvojiti nulta početna stanja, dakle $u_{-1} = 0$ i $u_0 = 0$.
- Da bi se ovo obezbedilo i za svaku narednu kodnu reč, neophodno je da koder nakon kodovanja prethodne kodne reči predje u ovo nulto početno stanje. To znači da svaka poruka koja ulazi u koder mora na kraju da sadrži dovoljno nula da bi se sve memorije sistema vratile u nulta početna stanja.

Kodovanje konvolucionim kodovima

- U prethodnom primeru to znači da se kodovanje svake reči završava kodovanjem još dve nule.

Primer 6.26 *Kodovanje konvolucionim kodom*

Potrebno je kodovati poruku u=101 pomoću konvolucionog kodera prikazanog na sl.6.1.

Da bi smo sagledali proces kodovanja, izračunajmo sve relevantne veličine. One su date u donjoj tabeli.

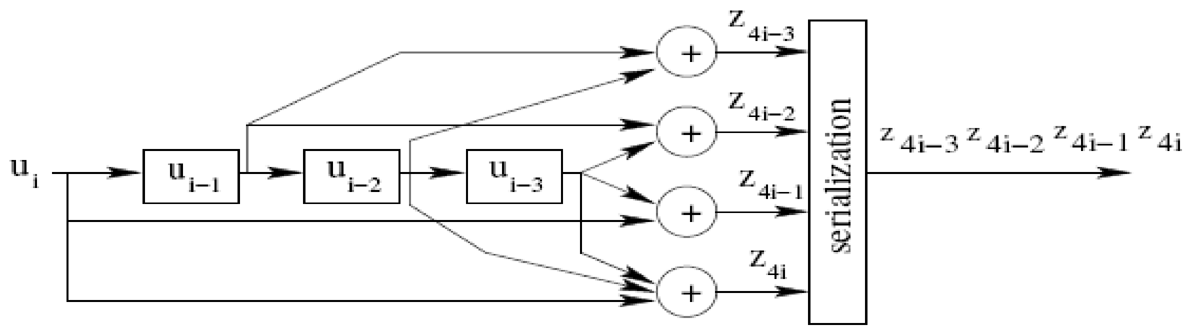
i	u_i	State $(u_{i-1}u_{i-2})$	$z_{2i-1}z_{2i}$
1	1	00	11
2	0	10	01
3	1	01	00
4	(0)	10	01
5	(0)	01	11

Ovo su dodati nulti simboli u cilju vraćanja kodera u nulto stanje

Stoga je odgovarajuća kodna reč 1101000111

Control Question 67

Consider the convolutional code, the encoder of which is described by the following diagram:



1. How many zeros must be added after each word to encode?

   (a) 1

   (b) 2

   (c) 3

   (d) 4

   (e) 5

2. How is 10110 encoded?

   (a) 0011110010101000100111100111

   (b) 10110

   (c) 00111100101010000101111001110000

   (d) 0010010110011010110001001001

   (e) 11110000111111110000000000000000

   (f) 11000011010100011010011111100000

Odgovaor

1. (c)

2. (c): 0011110010101000010111100111000

u_i	u_{i-1}	u_{i-2}	u_{i-3}	z_{4i-3}	z_{4i-2}	z_{4i-1}	z_{4i}
1	0	0	0	0	0	1	1
0	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
1	1	0	1	1	0	0	0
0	1	1	0	0	1	0	1
0	0	1	1	1	1	1	0
0	0	0	1	0	1	1	1
0	0	0	0	0	0	0	0

Kodovanje konvolucionim kodovima

- Razmotrimo sada opšti postupak generisanja koda pomoću kodera sa sl.6.1.
- Neka je poruka dužine 3 bita, odnosno neka je $u = (u_1, u_2, u_3)$, što znači da u postupku kodovanja stvarno učestvuje poruka $u = (u_1, u_2, u_3, 0,0)$. Dužina odgovarajuće kodne reči je $2 \cdot 5 = 10$. U matričnoj formi, jednačine (6.5) i (6.6) se mogu zapisati na sledeći način

$$(z_{2i-1}, z_{2i}) = (u_{i-2}, u_{i-1}, u_i) \cdot \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \qquad (6.7)$$

Kodovanje konvolucionim kodovima

Kompletna kodna reč z se dobija množenjem $u = (u_1, u_2, u_3, 0{,}0)$ sa matricom

$$G_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

Ili jednostavnije, množenjem $u = (u_1, u_2, u_3)$ sa matricom

$$G_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Dakle $z = u \cdot G_3$.

Kodovanje konvolucionim kodovima

- Za poruke dužine m, ova formula se može generalizovati na $z = u \cdot G_m$, gde je G_m m x $(2m+4)$ matrica napravljena tako da su redovi matrice iz izraza (6.7) složeni u jedan red, a zatim za svaki novi red pomereni u desno za dve kolone.

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{matrix}$$

$$G_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Opšta definicija konvolucionih kodova

Definicija 6.15

(n,k,r) D-arni konvolucioni kod je neograničeni linearni kod, čija generator matrica ima sledeću (beskonačnu) formu

$$G = \begin{bmatrix} F_0 & F_1 & F_2 & \cdots & F_r & [0] & [0] & [0] & \cdots \\ [0] & F_0 & F_1 & \cdots & F_{r-1} & F_r & [0] & [0] & \cdots \\ [0] & [0] & F_0 & \cdots & F_{r-2} & F_{r-1} & F_r & [0] & \cdots \\ \vdots & \vdots & \ddots & \ddots & & & & \ddots & \ddots \end{bmatrix}$$

Gde su F_i k x n matrice a [0] k x n nula matrice. Svaki skup od k redova matrice G je isti kao prethodni , pomeren za n mesta u desno.

Opšta definicija konvolucionih kodova

Definicija 6.15 *Nastavak*

Poruka u dužine m $u = (u_1, \dots., u_m)$ se koduje postupkom

$$z = \bar{u} \cdot G_{m'},$$

gde je $\bar{u}$ vektor dužine $m' = qk, q = \left\lceil \frac{m}{k} \right\rceil,$

$$\bar{u} = (u_1, \dots., u_m, 0, \dots, 0).$$

$G_{m'}$ je gornja leva podmatrica matrice G dimenzije qk x n(r+q).

Primetimo da je $\bar{u} = u$, odnosno $m' = m$, ako je m multipl od k (dakle ovo sigurno važi u posebnom slučaju k=1)

Opšta definicija konvolucionih kodova

U uvedenim notacijama u Definiciji 6.15, k odgovara broju simbola poruke koji ulaze u koder (k ulaznih linija), n odgovara broju simbola izlazne kodne reči po jednom ulazu (n izlaznih linija), dok je r maksimalni broj memorijskih elemenata (dužina registara), na jednoj ulaznoj liniji.

Primer 6.27

Kodera sa sl.6.1 implementira (2,1,2) konvolucioni kod:

k=1 ulaznih linija, r=2 memorija, dajući n=2 bita kodnih reči za svaki ulazni bit.

Kao što smo videli u prethodnom odeljku, za ulazne poruke od 3 simbola, generator matrica je 3 x 10 matrica

Opšta definicija konvolucionih kodova

$$G_3 = \begin{bmatrix} \boxed{1\ 1} & \boxed{0\ 1} & \boxed{1\ 1} & 0\ 0 & 0\ 0 \\ 0\ 0 & \boxed{1\ 1} & \boxed{0\ 1} & \boxed{1\ 1} & 0\ 0 \\ 0\ 0 & 0\ 0 & \boxed{1\ 1} & \boxed{0\ 1} & \boxed{1\ 1} \end{bmatrix}$$

U kojoj je zaista k=1 red, a pomeranje redova u desno se vrši za 2 pozicije. U odnosu na opštu definiciju generator matrice, za matricu G_3 važi

$F_0 = [1\ 1]$, što odgovara koeficijentima za u_i u jednačinama (6.5) i (6.6)

$F_1 = [0\ 1]$, odgovara koeficijentima za u_{i-1}

$F_2 = [1\ 1]$, odgovara koeficijentima u_{i-2}.

Opšta definicija konvolucionih kodova

- Primetimo da su konvolucioni kodovi linearni. Bilo koja linearna kombinacija kodnih reči je takodje kodna reč, uz konvenciju da se kraće kodne reči izjednačavaju prema dužoj, dodavanjem nula na kraju.

Opšta definicija konvolucionih kodova

Control Question 68

1. What are (n, k, r) of the convolutional code given in the last question?

   (a) $(1, 3, 4)$
   (b) $(7, 4, 1)$
   (c) $(3, 1, 4)$
   (d) $(4, 1, 3)$
   (e) $(7, 1, 4)$
   (f) $(1, 3, 7)$

2. What is the generator matrix of this code?

   (a) How many F blocks are there?
   (b) What is the size of each F block: $? \times ?$?
   (c) Give all the F blocks.

Predstava preko rešetki

- Stanje jednog sistema je skup internij parametara (sadržaja registara, memorija i sl.) koji su neophodni za izračunavanje izlaza za zadate ulaze. Na primer za koder sa sl.6.1, stanje u trenutku i je trenutni sadržaj dve memorije, odnosno $S = (u_{i-1}, u_{i-2})$.
- Rad kodera je odredjen stanjem i ulazom. Konvolucioni koder je mašina stanja (state machine). Rad takvih sistema je odredjen dijagramom stanja.

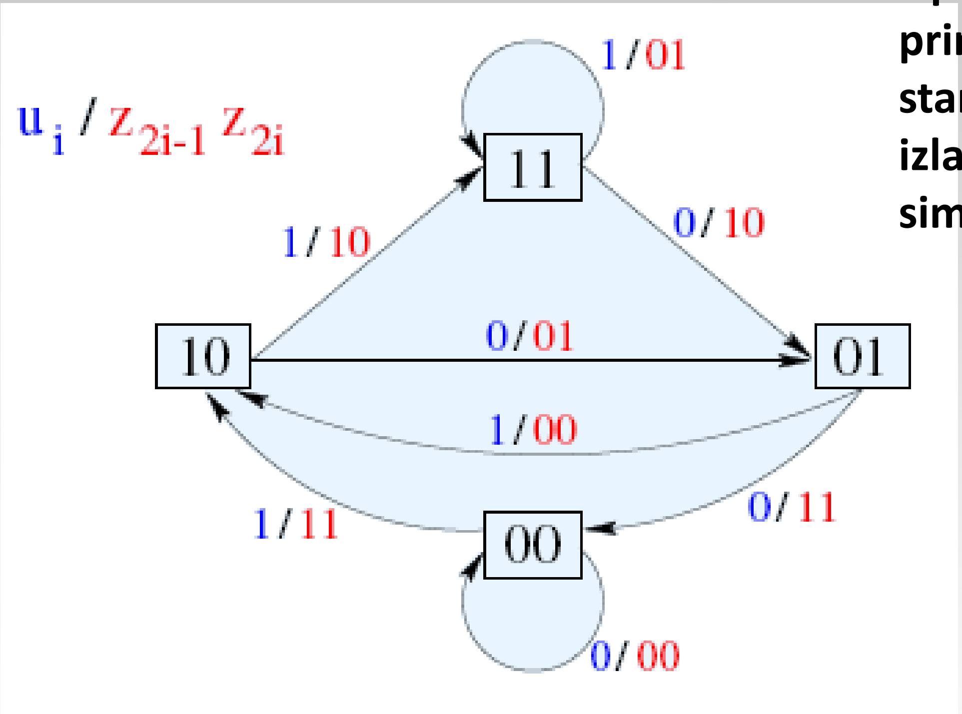
Definicija 6.16 *Dijagram stanja*

Dijagram stanja kodera je gtaf čiji su čvorovi sva moguća unutrašnja stanja kodera. Grana izmedju čvorova S_i i S_j u ovom grafu prezentuje činjenicu da postoji ulaz koji stanje S_i prevodi u S_j. Grane se po pravilu označavaju ulaznim simbolima i odgovarajućim izlaznim simbolima.

Predstava preko rešetki

Primer 6.28 *Dijagram stanja*

Za koder sa sl.6.1 dijagram stanja je dat sa



Npr. ako se u stanju 01 primi kao ulaz simbol 1 stanje postaje 10, a na izlazu se pojavljuju dva simbola 00.

Plave oznake su ulazi koji menjaju stanja, dok su crvene oznake odgovarjući simboli izlaza.

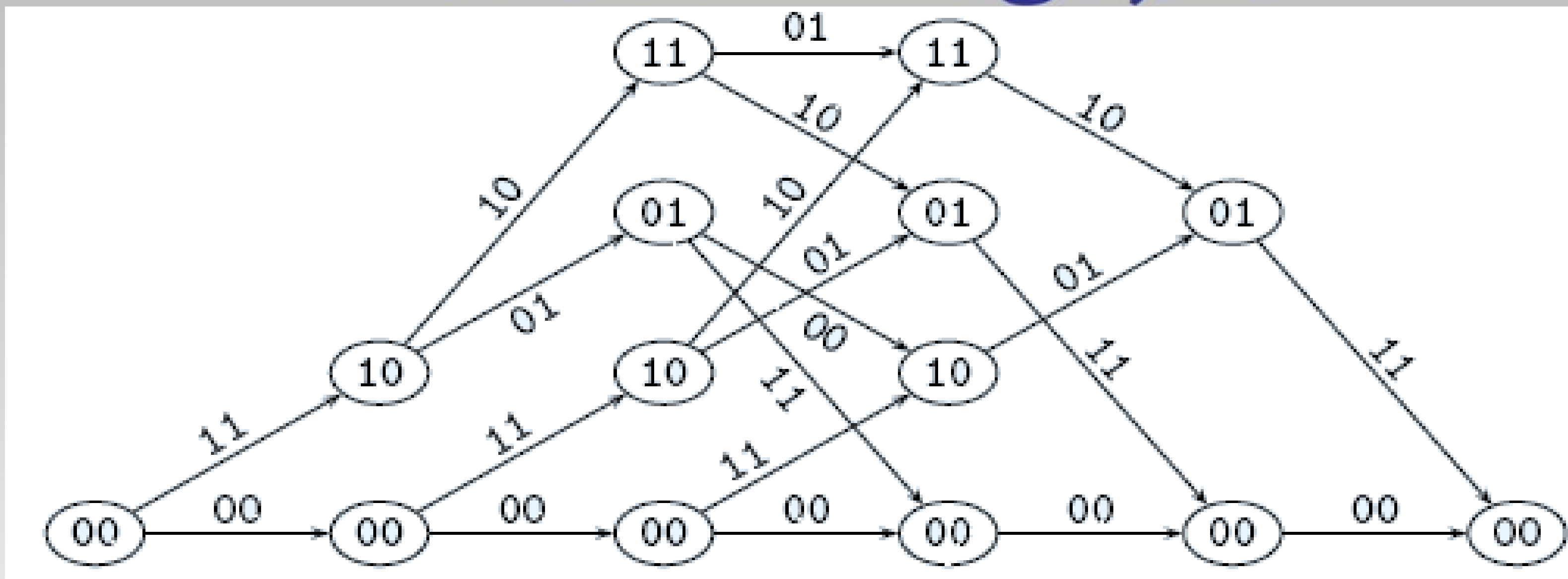
Predstava preko rešetki

- Skup kodnih reči konvolucionog koda (n,k,r) za sve moguće poruke dužine m bita odgovara svim mogućim putanjama dužine $n(r + \left\lceil\frac{m}{k}\right\rceil)$ u dijagramu stanja , koje startuju od nultog stanja i vraćaju se u to nulto stanje.
- Sve putanje u dijagramu stanja iste dužine $n(r + \left\lceil\frac{m}{k}\right\rceil)$ razvijene u vremenu, se nazivaju rešetka dimenzije m konvolucionog koda (n,k,r).

Primer 6.29 *Rešetka*

Za (2,1,2) konvolucioni kod iz prethodnog primera, rešetka dimenzije 3, koja reprezentuje kodovanje svih poruka dužine 3 bita je data na sledećem dijagramu:

Predstava preko rešetki



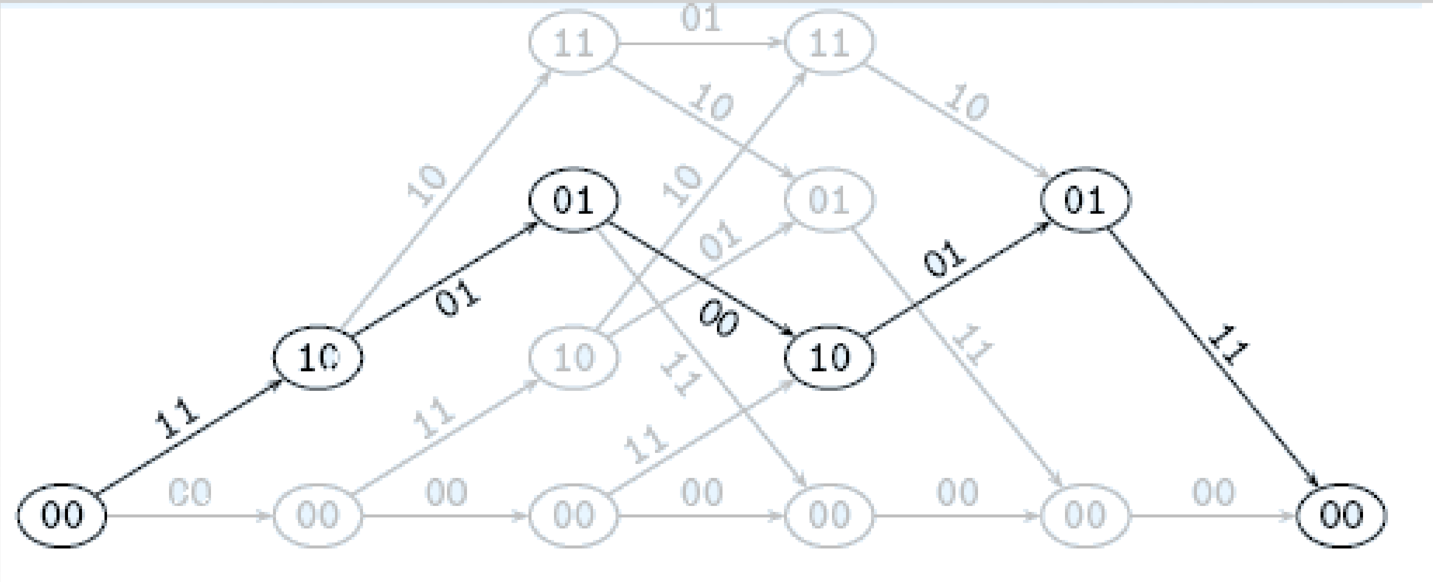
Gornja grana, za svaki čvor,odgovara ulaznom bitu 1, a donja grana ulaznom bitu 0.

Prve tri kolone grana odgovaraju kodovanju 3 bita poruke, a dve poslednje kolone grana odgovaraju završnim nulama, pa stoga imaju samo donje izlazne grane za odgovarjuće čvorove.

Predstava preko rešetki

Primer 6.30 *Kodovanje u rešetci*

Kodovanje poruke u=101, generiše sledeću putanju:



Odakle se vidi da je generisana kodna reč z=1101000111.

Predstava preko rešetki

Univerzitet Sinergija

Control Question 69

Consider the lattice corresponding to the encoding of a message of length 3 with the encoder of the former control question.

How many columns of states does it have?

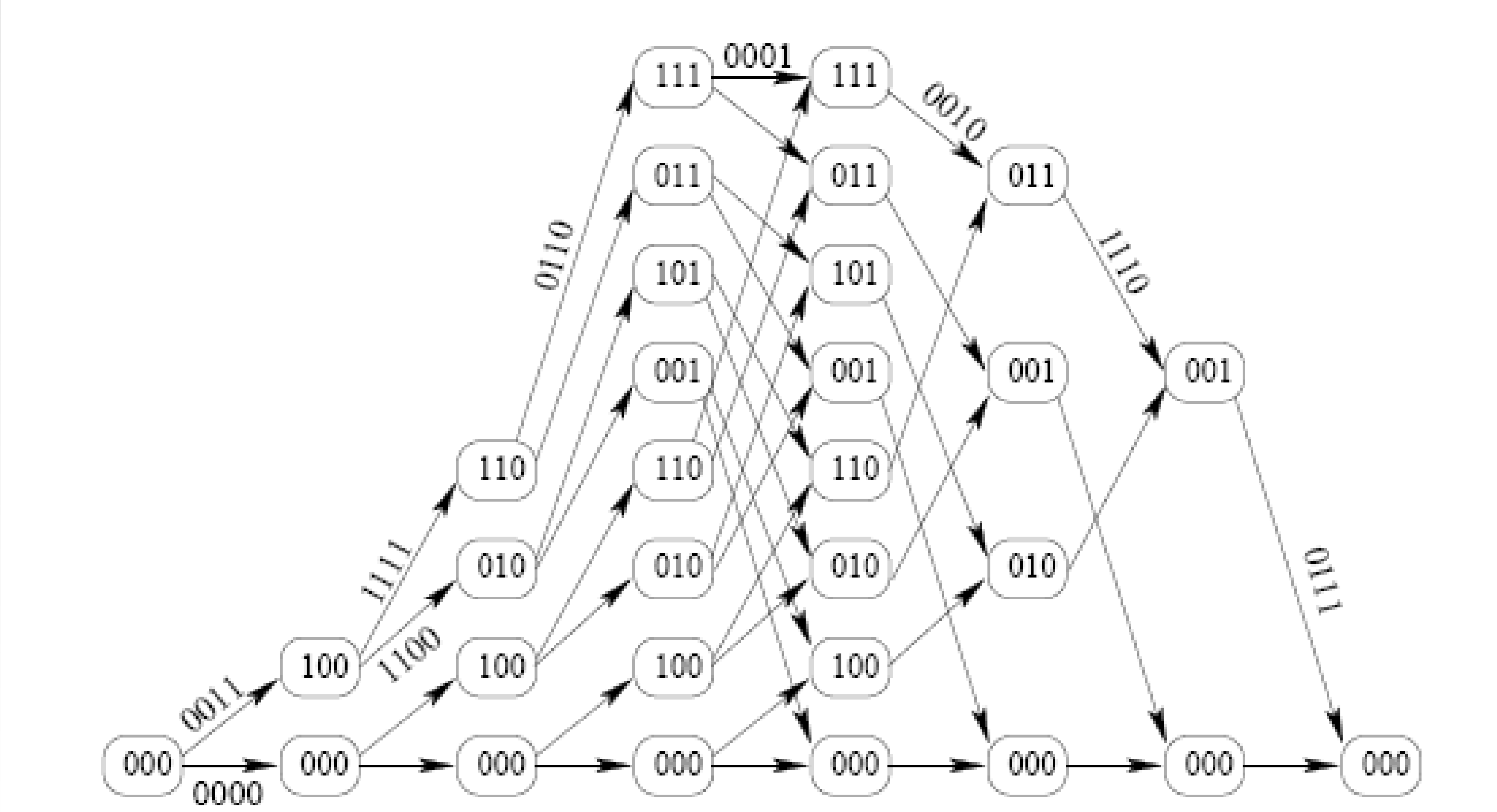
For each column, how many states are there?

Give the arc label for the following pairs of states. If no arc is present, answer ”no arc”:

from 100 to 010:
from 101 to 110:
from 001 to 111:
from 111 to 011:

Predstava preko rešetki

Navedimo još jedan primer rešetke (u kojoj nisu sve grane označene):



Predstava preko rešetki

U donjoj tabeli su date kodne reči emitovane u svakoj od mogućih situacija:

$S_i \backslash u_i$	0	1
000	0000	0011
001	0111	0100
010	1001	1010
011	1110	1101
100	1100	1111
101	1011	1000
110	0101	0110
111	0010	0001

Dekodovanje

- Kao što smo videli kodnim rečima odgovaraju putanje od startnog čvora do krajnjeg čvora u rešetci.
- Dekodovanje se sastoji u nalaženju *najpogodnije putanje* koja odgovara primljenoj kodnoj reči.
- U terminima dekodovanja minimalnim rastojanjem, odnosno dekodovanjem kodne reči sa minim brojem grešaka, *najpogodnija putanja* se definiše kao putanja sa minimalnim Hemingovim rastojanjem u odnosu na poruku koja se dekoduje.
- Nalaženje te najbliže putanje se može obaviti dinamičkim programiranjem, odnosno Viterbijevim algoritmom.

Dekodovanje

- Viterbijev algoritam dekoduje jedan za drugim blokove kodnih reči (odnosno m bita poruke nakon prethodnih blokova poruke), pamteći u svakom koraku samo lokalno optimalna rešenja.
- Ovo se svodi na to da se u svakom čvoru u koloni koja odgovara tekućem bloku dekodovanja, pamti najbolja putanju koja dolazi do tog čvora.
- Na kraju, najbolja putanja pronadjena za poslednji čvor je ujedno dekodovana poruka.
- Ono što je važno i korisno za Viterbijev algoritam, je svojstvo da je broj najboljih putanja za svaki vremenski korak, uvek manji ili jednak broju stanja kodera.
- Stoga je Viterbijev algoritam linearne umesto eksponencijalne kompleksnti naivnog algoritma dekodovanja koji bi se sastojao u poredjenju svih putanja sa porukom koju treba dekodovati.

Dekodovanje

- Predjimo na detaljniji opis Viterbijevog algoritma. Uvedimo prvo neke oznake.
- $\gamma_i(s)$ – najbolje dekodovanje (sa minimalnim mogućim greškama) dužine i koje se završava u stanju s

$$\gamma_i(s) = \min_{\substack{z_1,\ldots,z_{2i} \\ \text{ending in } s}} d(z_1 \ldots z_{2i}, \widehat{z}_1 \ldots \widehat{z}_{2i})$$

- Kompletno dekodovanje odgovara $\gamma_{|\hat{z}|}(00)$, $|\hat{z}|$=m+2, gde je $|\hat{z}|$ dužina poruke koja se dekoduje.

Dekodovanje

Lako je videti da za svaki par stanja (s, s') važi

$$\gamma_i(s) = \min_{\substack{z_{2i-1}z_{2i} \\ \text{from } s' \text{ to } s}} \left(d(z_{2i-1}z_{2i}, \widehat{z}_{2i-1}\widehat{z}_{2i}) + \gamma_{i-1}(s') \right).$$

Ovo vodi ka Viterbijevom algoritmu:

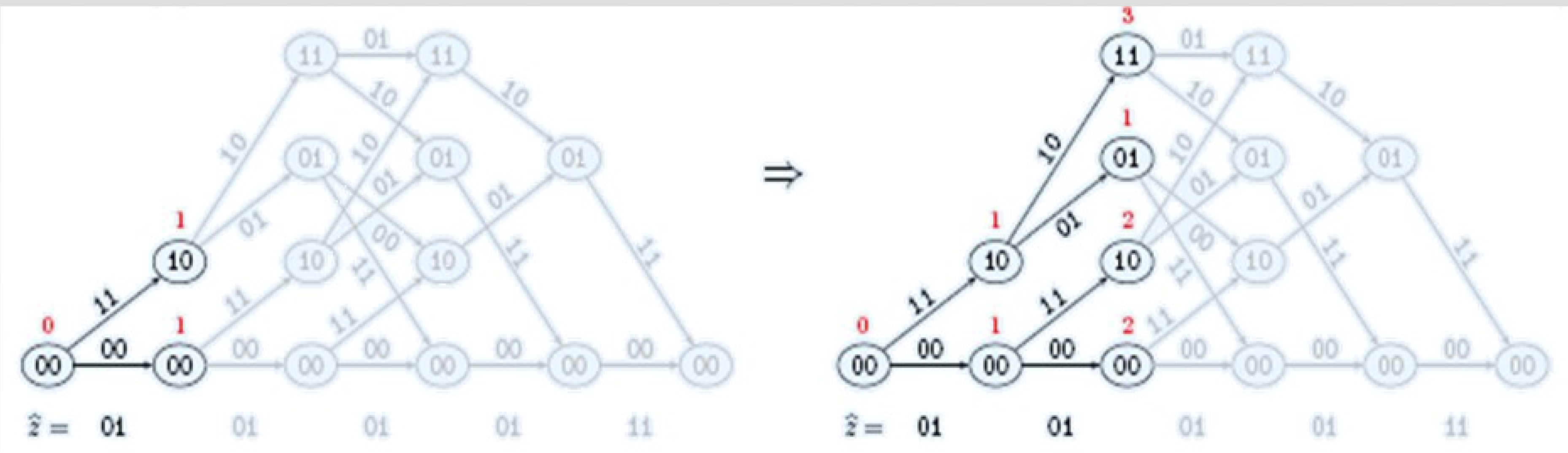
$\gamma_0(00) = 0$
for i from 1 to $|\widehat{z}|$ do
  for all s do
    $\gamma_i(s) = \min_{s' \to s} (d(z_{2i-1}z_{2i}, \widehat{z}_{2i-1}\widehat{z}_{2i}) + \gamma_{i-1}(s'))$
      označiti granu od s' do s za koju se dostiže minimum
  end for
end for

Rekonstrukcija optimalnog puta se dobija unazad, od krajnjeg do početnog stanja.

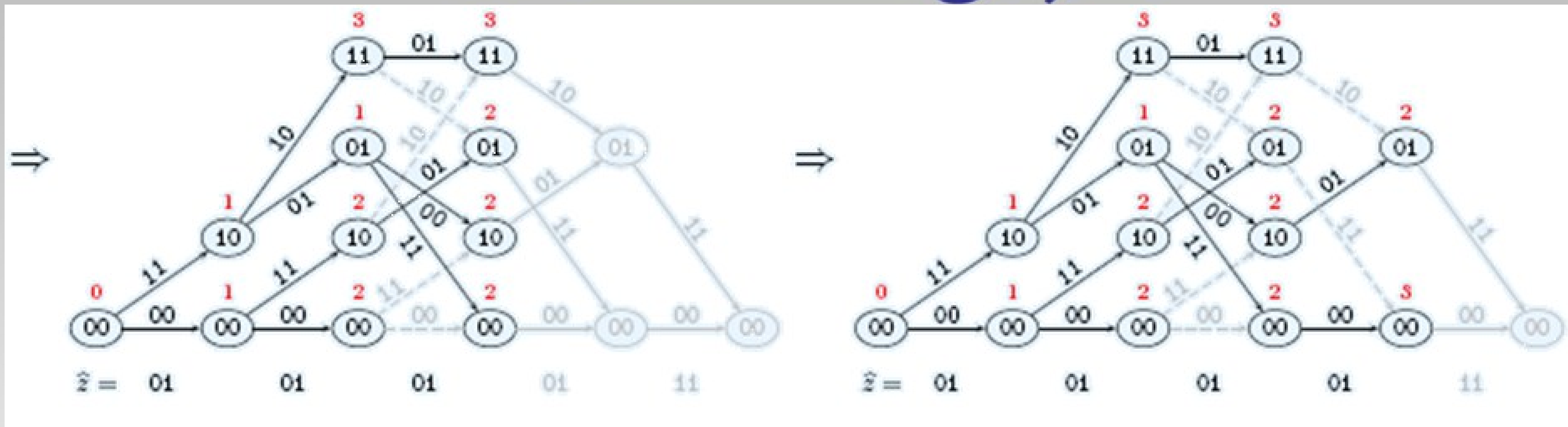
Dekodovanje

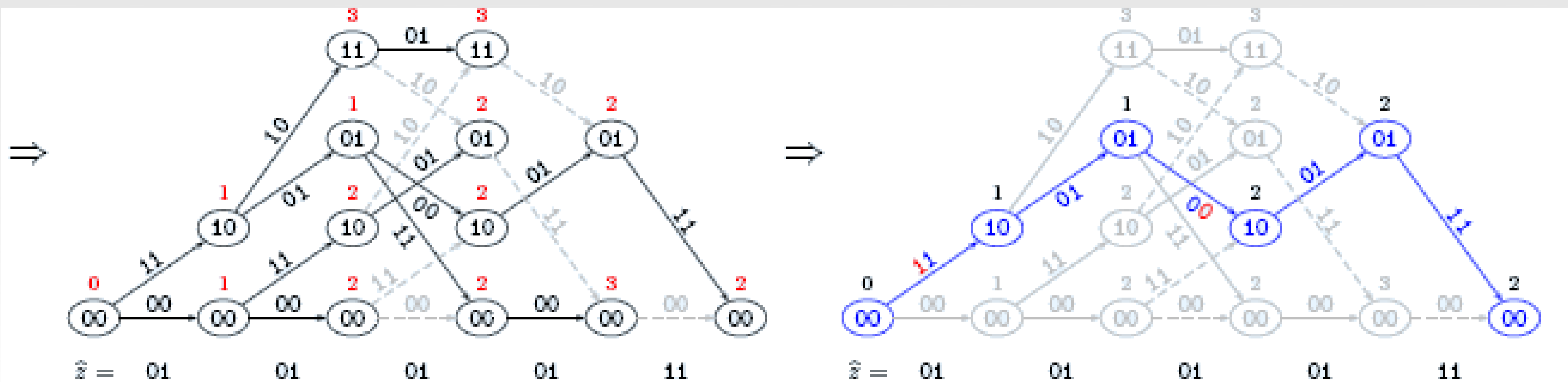
Primer 6.31

Pretpostavimo da je preko kanala sa šumom poslata kodna reč z=1101000111, dok je primljena reč $\hat{z}$=0101010111, odnosno desile su se dve greške. Tada će Viterbijev algoritam proći kroz sledeće korake



Dekodovanje





Dekodovanje

U prvom koraku rada Viterbijevog algoritma, razmatraju se samo dva moguća puta:

- Razmatrani prvi kodni simbol je 1, što vodi od stanja 00 ka stanju 01. Tada se emituje 11 i pravi se jedna greška, budući da je primljeno 01.
- Razmatrani prvi simbol je 0, što vodi od stanja oo ka stanju oo. Tada se emituje 00 i pravi se jedna greška, budući da je primljeno 01.

U drugom koraku, postoji samo jedan put za dostizanje svakog od stanja (i sada postoje 4 takva puta). Minimalan broj grešaka za svako od ova četiri stanja je jednak minimalnom broju grešaka prethodnih stanja plus broj grešaka odgovarajuće putanje (odnosno razlike izmedju emitovanih i primljenih simbola).

Dekodovanje

Na primer ako se razmatra put izmedju stanja 10 i 11, imamo dodatne dve greške, budući da se tada emituje 10, aprimljeno je 01. Ovo vodi ka minimalnom broju grešaka u stanju 11 u koraku 2 $\gamma_2(11) = 1 + 2 = 3$.

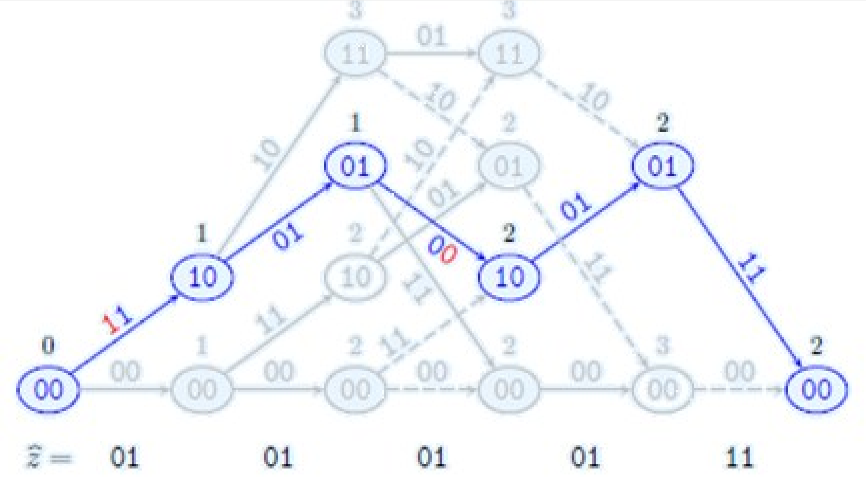
U koraku tri, imamo dve moguće putanje za svako stanje. Algoritam zadržava samo po jednu alternativu koja proizvodi manji broj grešaka. Ostale grane su označene isprekidanim bledjim linijama.

Algoritam radi na istovetnom principu sve do poslednjeg koraka, u kome se dostiže nulto stanje sa minimalnim brojem grešaka jednkom 2.

Poslednji korak je rekonstrukcija putanje unazad (plave linije). Ova putanja odgovara kodnoj reči 1101000111.

Dekodovanje

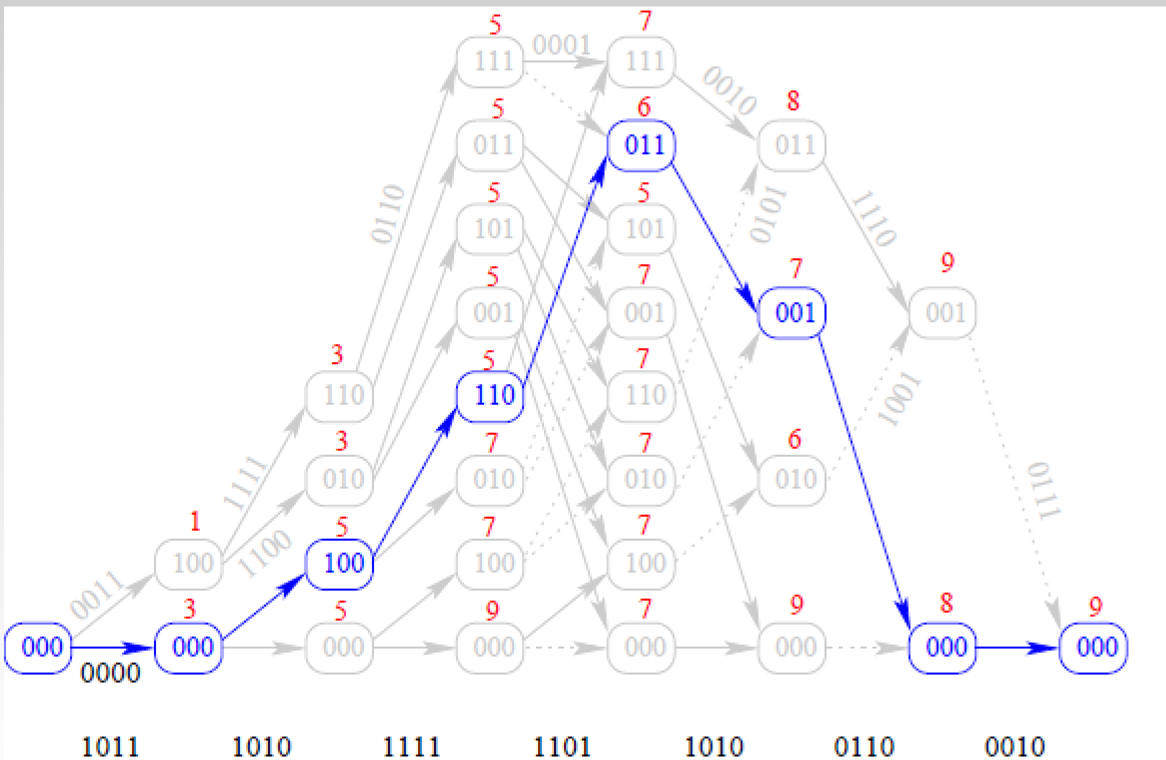
- Nakon dobijanja kodne reči, neophodno je rekonstruisati originalnu poruku. Ovo možemo obaviti prateći oznake grana (o ili 1) na nadjenoj optimalnoj putanji, ili uzimanjem prvog bita svakog stanja kroz koja prolazi optimalna putanja u redosledu od kraja putanje ka početku.
- Za ovaj primer i optimalnu putanju označenu plavom bojom, dobijamo niz 10100, i odbacivanjem dve zadnje nule koje nisu deo poruke, dobijamo konačno poruku 101, što i odgovara originalno poslatoj poruci.



Dekodovanje

Primer 6.31 a

Za koder iz prethodnog primera, dekodovati 1011101011111101101001100010.



Primljena poruka ima 9 grešaka. Emitovana kodna reč je 0000 0011 1111 0101 1110 0111 0000, što odgovara kodovanoj poruci 0110000, odnosno originalnoj poruci 0110.

Dekodovanje

- Budući da su konvolucioni kodovi linearni kodovi, primenljiva je Teorema 6.2 , prema kojoj je minimalno rastojanje jednako minimalnoj težini.

Osobina 6.17 *Minimalna težina konvolucionog koda*

Minimalna težina jednog konvolucionog koda jednaka je minimalnom broju nenultih simbola na putanji koja polazi i završava se u nultom stanju.

Dekodovanje

- Za koder iz prethodnog primera nalazimo da je $d_{min}(C) = 9$. Na donjoj slici je iz rešetke kodovanja izdvojena kodna reč koja ima težinu jednaku minimalnoj mogućoj težini 9 za ovaj konvolucioni kod.