Teorija informacije

Zaštitno kodiranje III

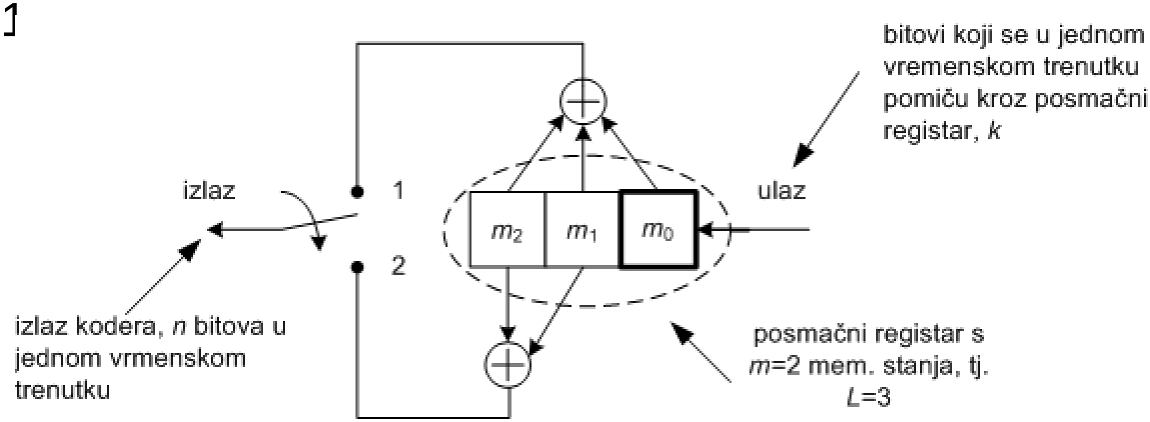
Sadržaj predavanja

- Uvod
  - Komunikacijski sustav; Cilj zašt. kodiranja; Podjela zaštitnih kodova.
- Blok kodovi
  - Uvod
  - Paritetno kodiranje
  - Linearno binarni blok kodovi
    - Generirajuća matrica **G** i njen standardni oblik
      - Kodiranje
      - Dekodiranje (dekodiranje preko sindroma)
      - Proračun vjerojatnosti ispravnog dekodiranja
    - Hammingovi kodovi
    - Ciklični kodovi
    - Konvolucijski i turbo kodovi

Konvolucijski i turbo kodovi

Konvolucijski kodovi

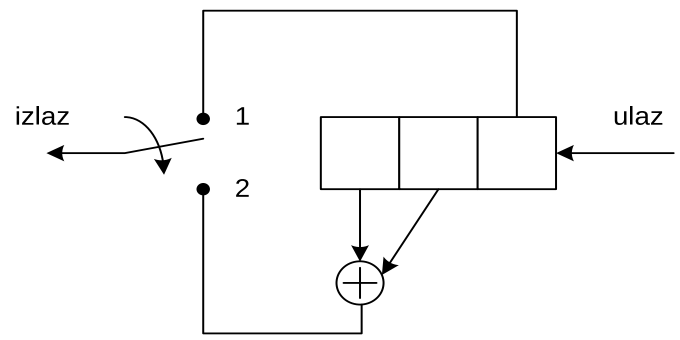
- Konvolucijski kodovi (engl. *Convolutional codes*) spadaju u grupu **memorijskih kodova**
  - Generiranje *i*-tog bita u kodnoj riječi ovisi ne samo o trenutačnom ulaznom bitu kodera nego i o određenom broju prethodnih ulaznih bitova
  - Blok kodovi su bezmemorijski kodovi
- Konvolucijski koder se sastoji od:
  - binarnih posmačnih registara (svaki registar uključuje *m* memorijskih stanja)
  - digitalnog logičkog sklopovlja, tj. binarnih zbrajala preko kojih se definiraju izlazi kodera (konvolucijski koder može imati jedan i više izlaza.)
- Konvolucijski koder opisujemo s tri parametra, i to: (n, k, L)
  - L – granična duljina kodera (engl. *Constraint length*) → broj bitova koji utječu na pojedini izlaz kodera, $L = m + 1$
  - k – ulazni bitovi
  - n – izlazni bitovi



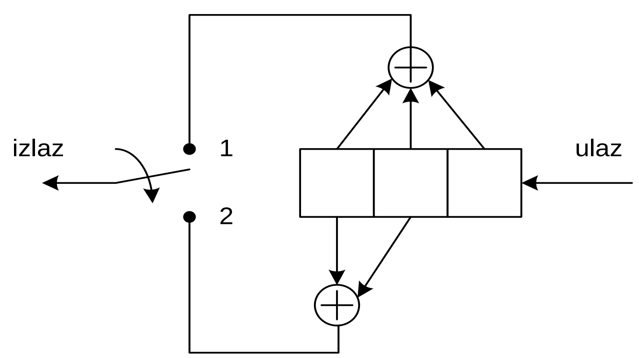
- **Stanje m_0 predstavlja ulaz kodera. Ulazni bit se iz njega na signal “clock” posmiče u stanje m_1 i prema modulo-2 zbrajalu.**

Tipovi konvolucijskih kodera

- Sistematski konvolucijski koder (k = 1, n = 2): u kodnoj riječi pored bitova zaštite pojavljuju se i bitovi izvorne poruke



- Nesistematski konvolucijski koder (k = 1, n = 2): u kodnoj riječi ne nalaze se bitovi izvorne poruke



- **Sistematski koder daje manju Hammingovu udaljenost između kodnih riječi jer se odbacuje jedno ili više binarnih zbrajala.**

Generirajuća matrica konvolucijskih kodova (1/3)

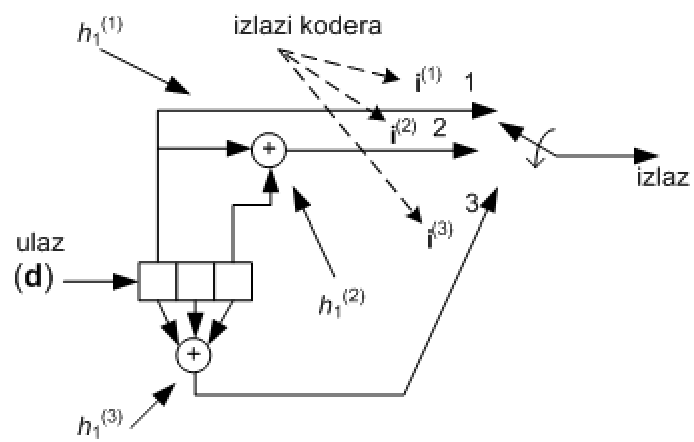
- Općeniti gledano, generirajuća matrica **G** je beskonačna jer ulazni slijed informacijskih bitova može teoretski u svojoj duljini biti beskonačan
- U praksi se uvijek kodira informacijski slijed konačne duljine → iz konvolucijskog koda nastaje blok kôd
- Matrica **G** se definira pomoću *n* vektora (tzv. funkcijski generatori), i to po jedan za svaki od *n* izlaza kodera
  - *Svaki funkcijski generator (oznaka "h") označava* ***veze*** *između izlaza i stanja posmačnih registara za pojedini izlaz kodera*
  - Postojanje "1"/"0" na nekom mjestu unutar vektora (npr. *i*-to mjesto) pokazuje da *i*-to stanje u posmačnom registru je ili nije spojeno na promatrani izlaz
- Primjer (3, 1, 3)/ funkc. generatori:

j-ti izlaz

$h_i^{(j)}=[h_{i,0}^{(j)}\ h_{i,1}^{(j)} \ldots h_{i,m}^{(j)}]$

i-ti ulaz

m-to mem. stanje pos. registra



$h_1^{(1)}=[100]$
$h_1^{(2)}=[101]$
$h_1^{(3)}=[111]$

Generirajuća matrica konvolucijskih kodova (2/3)

- Primjer (3, 1, 3)/ izlazi kodera (tzv. kodirani vektori):

$$\mathbf{i}^{(1)} = \mathbf{d} * h_1^{(1)} = [i_0^{(1)}\ i_1^{(1)} \ldots]$$

$$\mathbf{i}^{(2)} = \mathbf{d} * h_1^{(2)} = [i_0^{(2)}\ i_1^{(2)} \ldots]$$

$$\mathbf{i}^{(3)} = \mathbf{d} * h_1^{(3)} = [i_0^{(3)}\ i_1^{(3)} \ldots]$$

- Znak "*" predstavlja binarnu (modulo-2) diskretnu konvoluciju

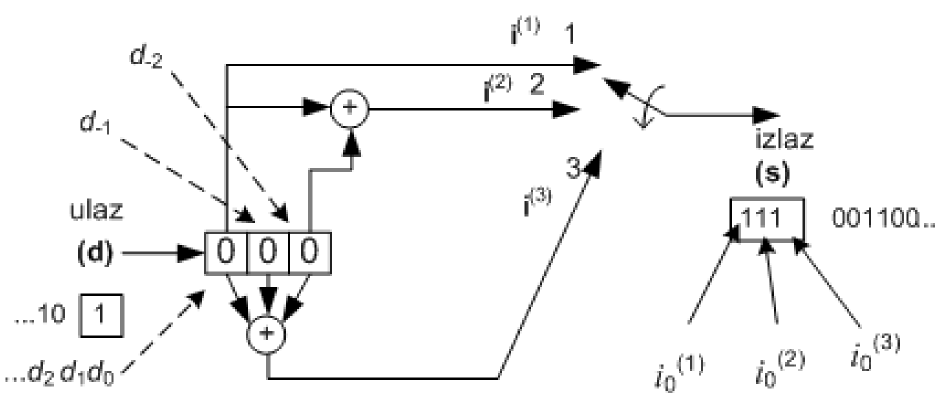
- Općenito, *r*-ti bit (*r* ≥ 0) *j*-tog kodiranog vektora određuje se izrazom:

$$\mathbf{i}_r^{(j)} = \sum_{i=1}^{k} \sum_{t=0}^{m} d_{r-t} h_{i,t}^{(j)} = d_r h_{1,0}^{(j)} + d_{r-1} h_{1,1}^{(j)} + \ldots + d_{r-m} h_{1,m}^{(j)} + \ldots$$

$$+ d_r h_{k,0}^{(j)} + d_{r-1} h_{k,1}^{(j)} + \ldots + d_{r-m} h_{k,m}^{(j)},$$

$$j = 1,\ldots,n.$$

- U početku sva su stanja u posmačnom registru postavljena u "0"



- $r = 0$

$$i_0^{(1)} = d_0 h_{1,0}^{(1)} \oplus d_{-1} h_{1,1}^{(1)} \oplus d_{-2} h_{1,2}^{(1)} = 1 \cdot 1 \oplus 0 \cdot 0 \oplus 0 \cdot 0 = 1$$

$$i_0^{(2)} = d_0 h_{1,0}^{(2)} \oplus d_{-1} h_{1,1}^{(2)} \oplus d_{-2} h_{1,2}^{(2)} = 1 \cdot 1 \oplus 0 \cdot 0 \oplus 0 \cdot 1 = 1$$

$$i_0^{(3)} = d_0 h_{1,0}^{(3)} \oplus d_{-1} h_{1,1}^{(3)} \oplus d_{-2} h_{1,2}^{(3)} = 1 \cdot 1 \oplus 0 \cdot 1 \oplus 0 \cdot 1 = 1$$

- $r = 1$

$$i_1^{(1)} = d_1 h_{1,0}^{(1)} \oplus d_0 h_{1,1}^{(1)} \oplus d_{-1} h_{1,2}^{(1)} = 0 \cdot 1 \oplus 1 \cdot 0 \oplus 0 \cdot 0 = 0 \ldots$$

- Izlaz: $\mathbf{s} = [i_0^{(1)}\ i_0^{(2)}\ i_0^{(3)}\ i_1^{(1)}\ i_1^{(2)}\ i_1^{(3)} \ldots]$

Generirajuća matrica konvolucijskih kodova (3/3)

- Generirajuća matrica **G** je u potpunosti određena s funkcijskim generatorima, dok njena dimenzija ovisi o promatranoj informacijskoj poruci **d**
- Općeniti oblik generirajuće matrice, **G**, je:

$$\mathbf{G}_\infty = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{G}_m & & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_{m-1} & \mathbf{G}_m & & \\ & & \mathbf{G}_0 & \cdots & \mathbf{G}_{m-2} & \mathbf{G}_{m-1} & \mathbf{G}_m & \\ & & & \ddots & & & & \ddots \end{bmatrix} \text{ s podmatricama } \mathbf{G}_l = \begin{bmatrix} h_{1,l}^{(1)} & h_{1,l}^{(2)} & \cdots & h_{1,l}^{(n)} \\ h_{2,l}^{(1)} & h_{2,l}^{(2)} & \cdots & h_{2,l}^{(n)} \\ \vdots & \vdots & & \vdots \\ h_{k,l}^{(1)} & h_{k,l}^{(2)} & \cdots & h_{k,l}^{(n)} \end{bmatrix}$$

$$l = 0,1,...,m$$

- <u>Primjer (3, 1, 3)/ generirajuća matrica, **G**:</u>

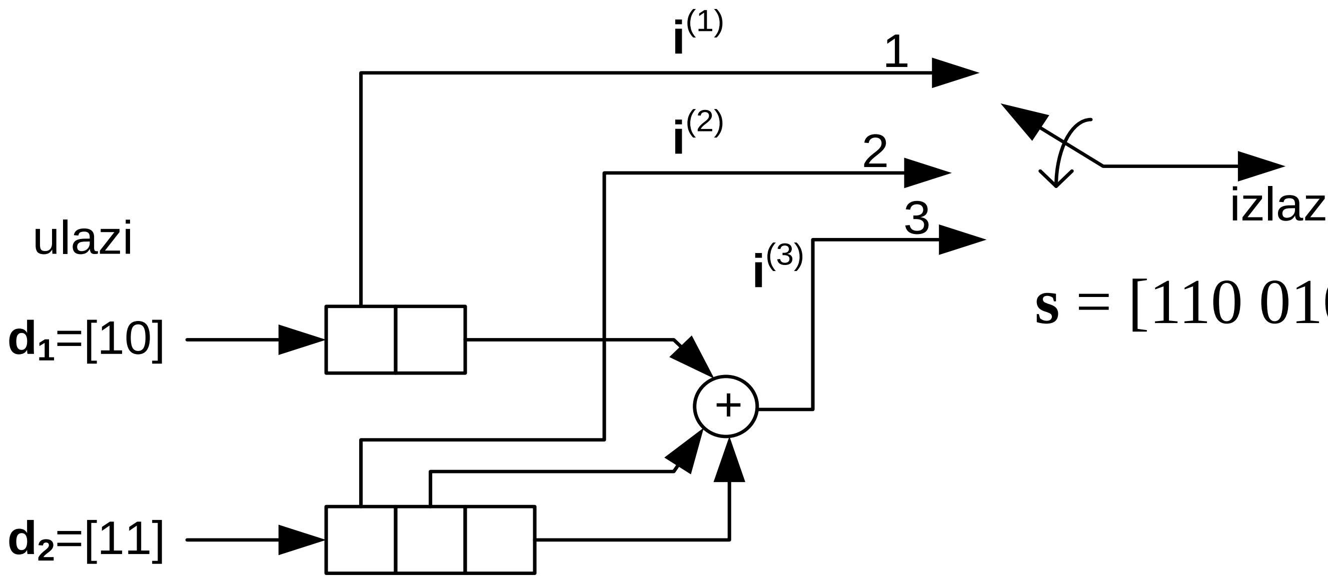
Na ulaz kodera dolazi poruka $\mathbf{d} = [101]$

$$\mathbf{G} = \begin{bmatrix} 111 & 001 & 011 & 0 & 0 \\ 0 & 111 & 001 & 011 & 0 \\ 0 & 0 & 111 & 001 & 011 \end{bmatrix}$$

$$\mathbf{s} = \mathbf{dG} = [101] \cdot \begin{bmatrix} 111 & 001 & 011 & 0 & 0 \\ 0 & 111 & 001 & 011 & 0 \\ 0 & 0 & 111 & 001 & 011 \end{bmatrix} = [111 \quad 001 \quad 100 \quad 001 \quad 011].$$

Primjer: konvolucijski koder kodne brzine 2/3

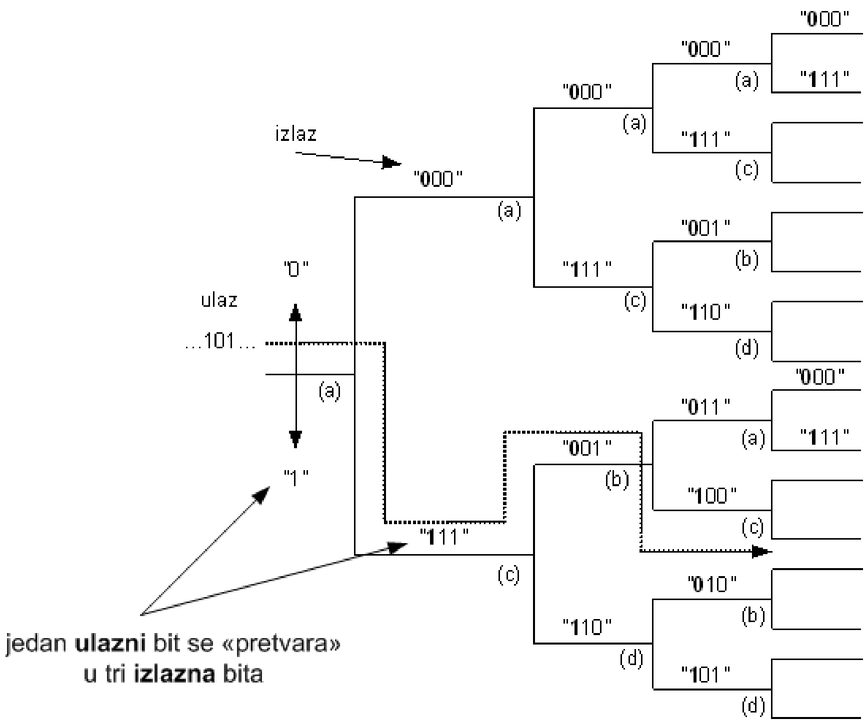
- Ulaz: poruka **d** = [11 01], gledano s lijeva na desno



$\mathbf{s}$ = [110 010 000 001]

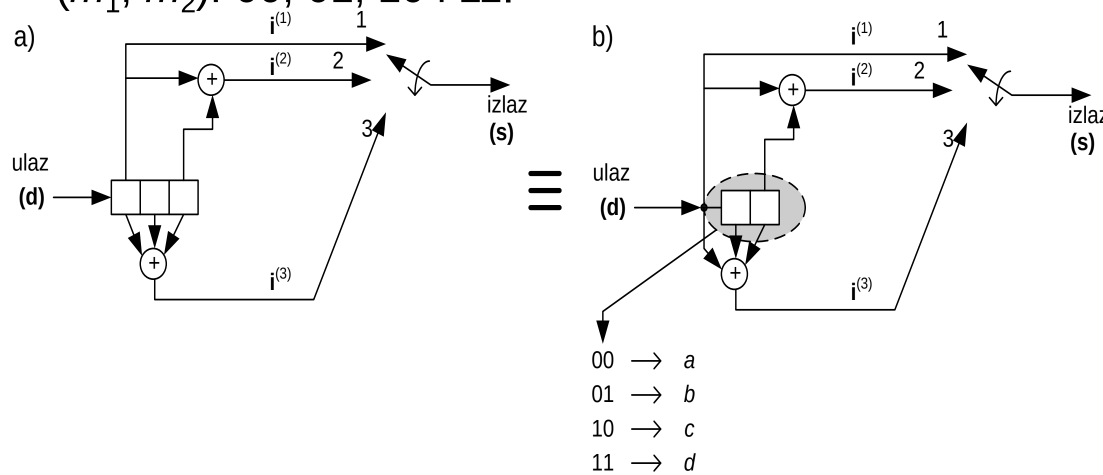
Grafički prikaz konvolucijskih kodova (1/2)

- Postoje tri metode za prikaz konvolucijkog kodiranja
  - stablasti dijagram (engl. *tree diagram*)
  - rešetkasti dijagram (engl. *trellis diagram*)
  - dijagram stanja (engl. *state diagram*)



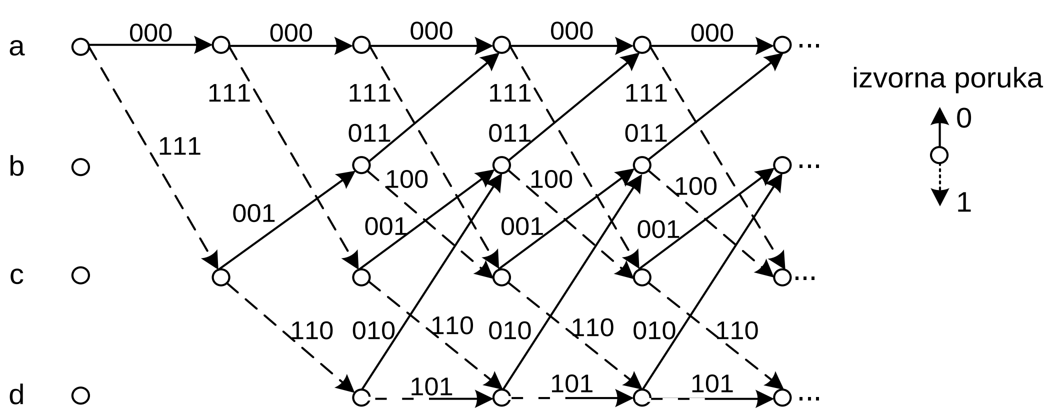
Stablasti dijagram za koder sa slajda 7

- Izlaz iz kodera određen je s ulaznim bitom (m_0) i jednim od četiri moguća stanja posmačnog registra (m_1, m_2): 00, 01, 10 i 11.

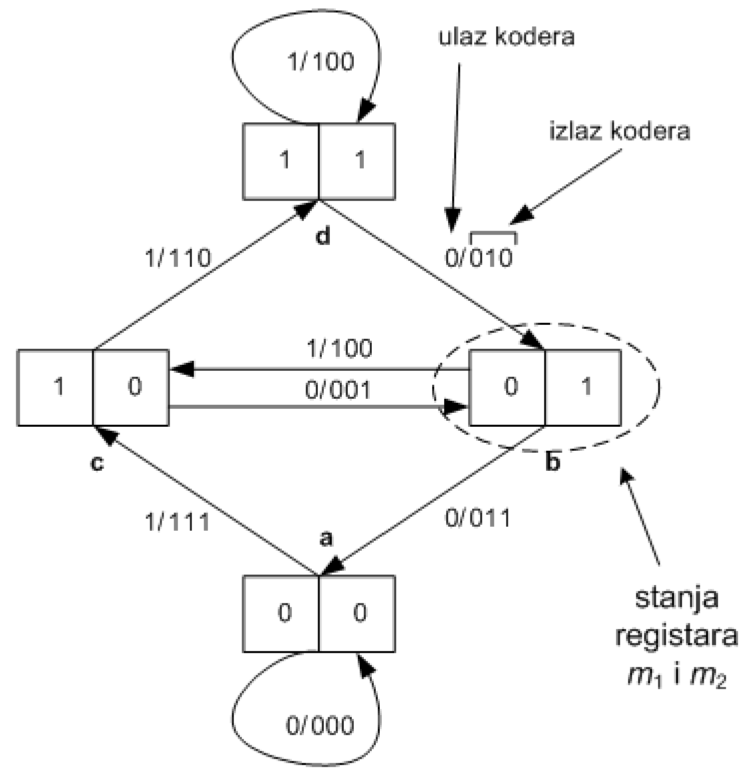


- Spajanjem čvorišta, u stablastom dijagramu, koja imaju isti izlazni slijed i istu oznaku nastaje rešetkasti dijagram.

Grafički prikaz konv. kodova (2/2)



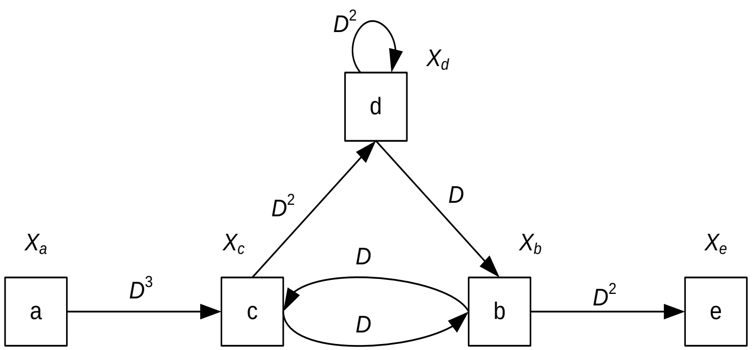
Rešetkasti dijagram za koder sa slajda 7



Dijagram stanja za koder sa slajda 7

Prijenosna funkcija konvolucijskog koda i udaljenost koda (1/2)

- Prijenosna funkcija (oznaka -> $T(D)$) određuje udaljenost i performanse koda koje se odnose na otkrivanje pogrešaka
- Prethodni dijagram stanja (slajd 10), uz izmjenu, iskoristit ćemo za objašnjenje i proračun udaljenosti konvolucijskog koda
- Potrebno je napraviti sljedeće:
  - Podijeliti stanje "a" na dva nova stanja, tj. "a" (ulaz) i "e" (izlaz) → potrebno je ukloniti vlastitu petlju u stanju "a"
  - Označiti svaku granu u grafu s D^i, gdje i označava <u>težinu kodne riječi</u> n (izlazni bitovi kodera)
  - Uvesti jednu varijablu (X_a, ..., X_e) za svako stanje (a, ...,e)

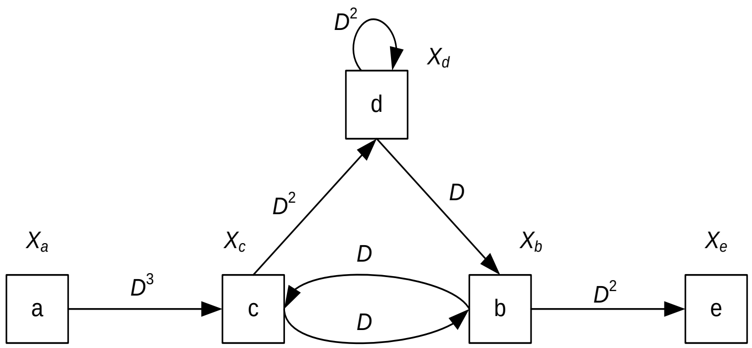


$$T(D) = \frac{X_e}{X_a}$$

- **Potrebno je pronaći sve putove od stanja "a" do stanja "e" i zbrojiti njihove težine**

Prijenosna funkcija konvolucijskog koda i udaljenost koda (2/2)

- Primjer (3, 1, 3)/ $T(D)$:



$$T(D) = \frac{X_e}{X_a}$$

$$X_c = D^3 X_a + D X_b$$

$$X_b = D X_c + D X_d$$

$$X_d = D^2 X_c + D^2 X_d$$

$$X_e = D^2 X_b$$

Rješavanjem sustava jednadžbi, dobivamo:

$$T(D) = \frac{D^6}{1-2D^2} = D^6\left(1+2D^2+4D^4+8D^6+\ldots\right) = 1D^6 + 2D^8 + 4D^{10} + 8D^{12} + \ldots$$

jedan put udaljenosti 6, d_{min}
a-c-b-e

dva puta udaljenosti 8

Napomena: Za dodatne mogućnosti prijenosne funkcije konv. koda vidjeti: PANDŽIĆ, I.S. BAŽANT, A. ILIĆ, Ž. VRDOLJAK, Z. KOS, M. SINKOVIĆ, V. *Uvod u teoriju informacije i kodiranje.* Element, 2 izdanje, 2010. (ISBN 978-953-197-605-3)

Dekodiranje konvolucijskih kodova: koderi

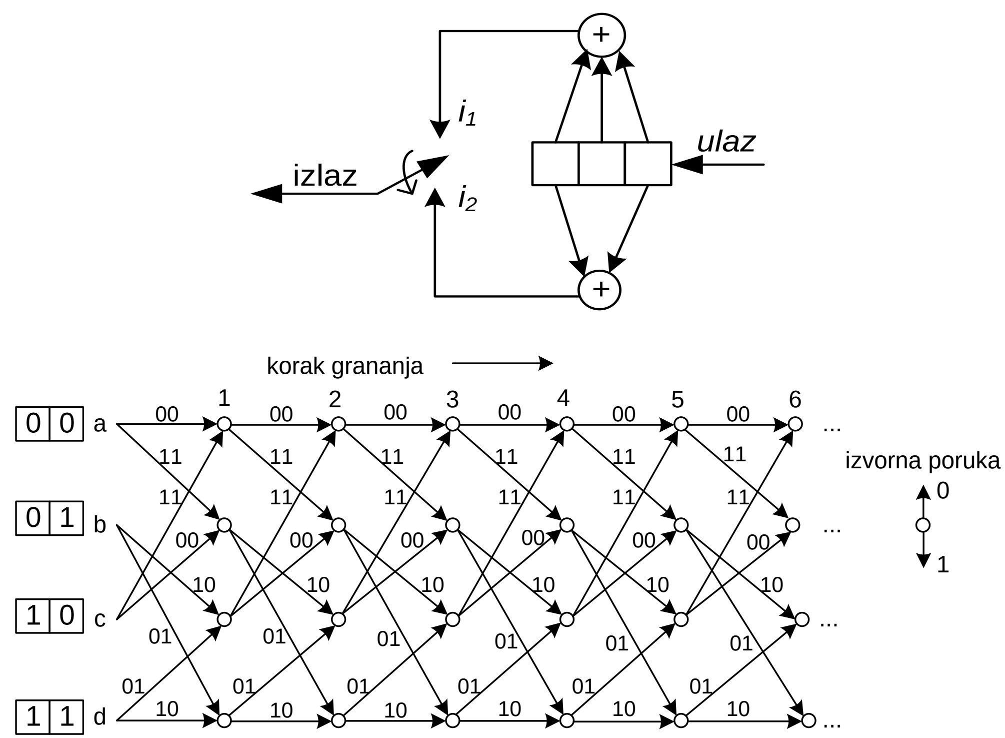
- **Udaljenost koda:** *mjera sličnosti* između dvaju kodnih riječi nekog koda *K*
  - Dvije mjere sličnosti: Hammingova i euklidska udaljenost koda
  - Primjena neke od navedenih mjera ovisi o odabranom kodnom sustavu, zahtijevanom BER-u, tipu prijenosnog kanala i vrsti demodulatora
- Hammingova udaljenost koda
  - Kod prijenosa binarnim simetričnim kanalom u kojem se svaki simbol promatra zasebno (bezmemorijski kanal)
  - U prijemu se koristi tzv. **većinska odluka** o tome koji je simbol primljen
  - Koristi se tzv. **dekoder s tvrdim odlučivanjem** (engl. *hard-decision decoder*)
- Euklidska udaljenost koda
  - Kod prijenosa kanalom s aditivnim bijelim Gaussovim šumom (skr. AWGN)
  - Odluka o primljenom simbolu provodi se **uzimajući u razmatranje cijelu kodnu riječ**
  - Koristi se tzv. **dekoder s mekim odlučivanjem** (engl. *soft-decision decoder*)

Dekodiranje konv. kodova: Viterbijev algoritam

- Optimalno dekodiranje konvolucijskih kodova svodi se na pronalaženje puta u rešetkastom dijagramu koji od primljene kodne riječi **c**’ ima minimalnu Hammingovu udaljenost, tj. traži se put u rešetkastom dijagramu od stanja *a* do stanja *e* s minimalnom Hammingovom udaljenosti

- <u>Problem dekodiranja ML</u> je što mora <u>odrediti sve putove</u> u rešetkastom dijagramu kako bi se dekodiranje provelo. To uključuje 2^L putova.
- **Viterbijev algoritam poboljšava proračun tako što uspoređuje dvije metrike za putove koji se spajaju u nekom stanju i <u>odbacuje</u> onaj put s manjom metrikom.** Navedeni postupak se ponavlja za sva stanja. **Na ovaj način na svakoj razini rešetke imamo 2^m “preživjelih” putova.**

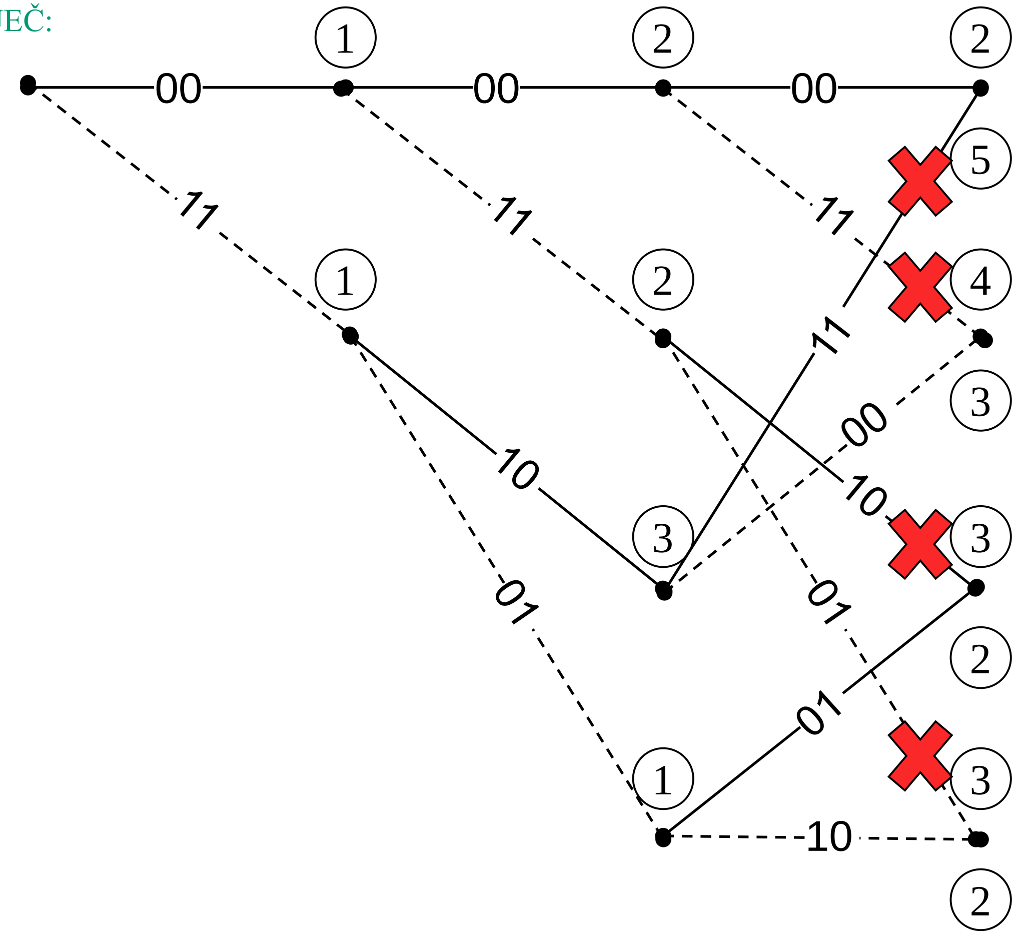
Primjer: koder L=3, k=1, n=3 (iz knjige)



Viterbijev algoritam: primjer (1/5)

KORAK: 1 2 3 4 5 6

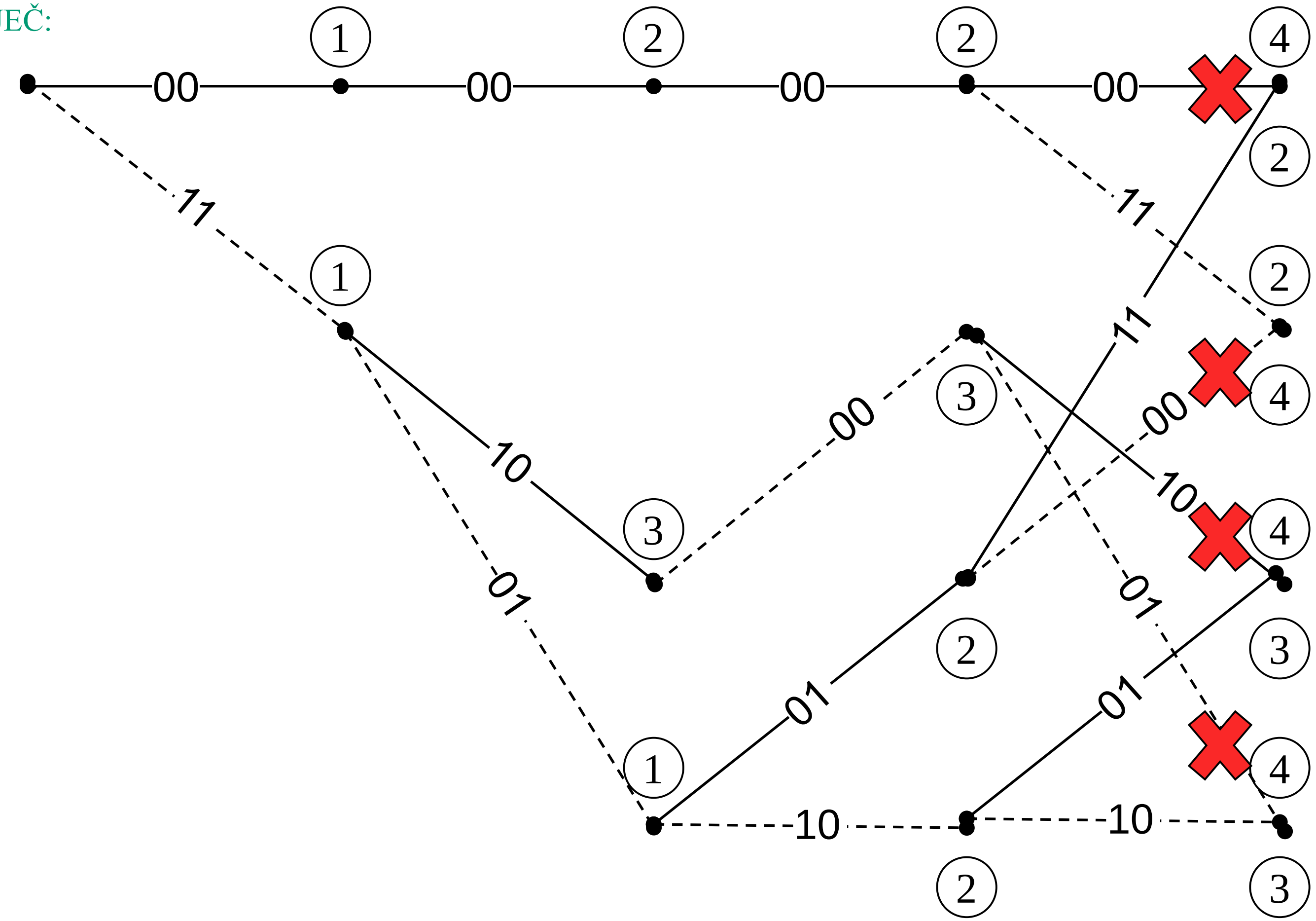
PRIMLJENA RIJEČ: 01 01 00 11 10 00



Viterbijev algoritam: primjer (2/5)

KORAK: 1 2 3 4 5 6

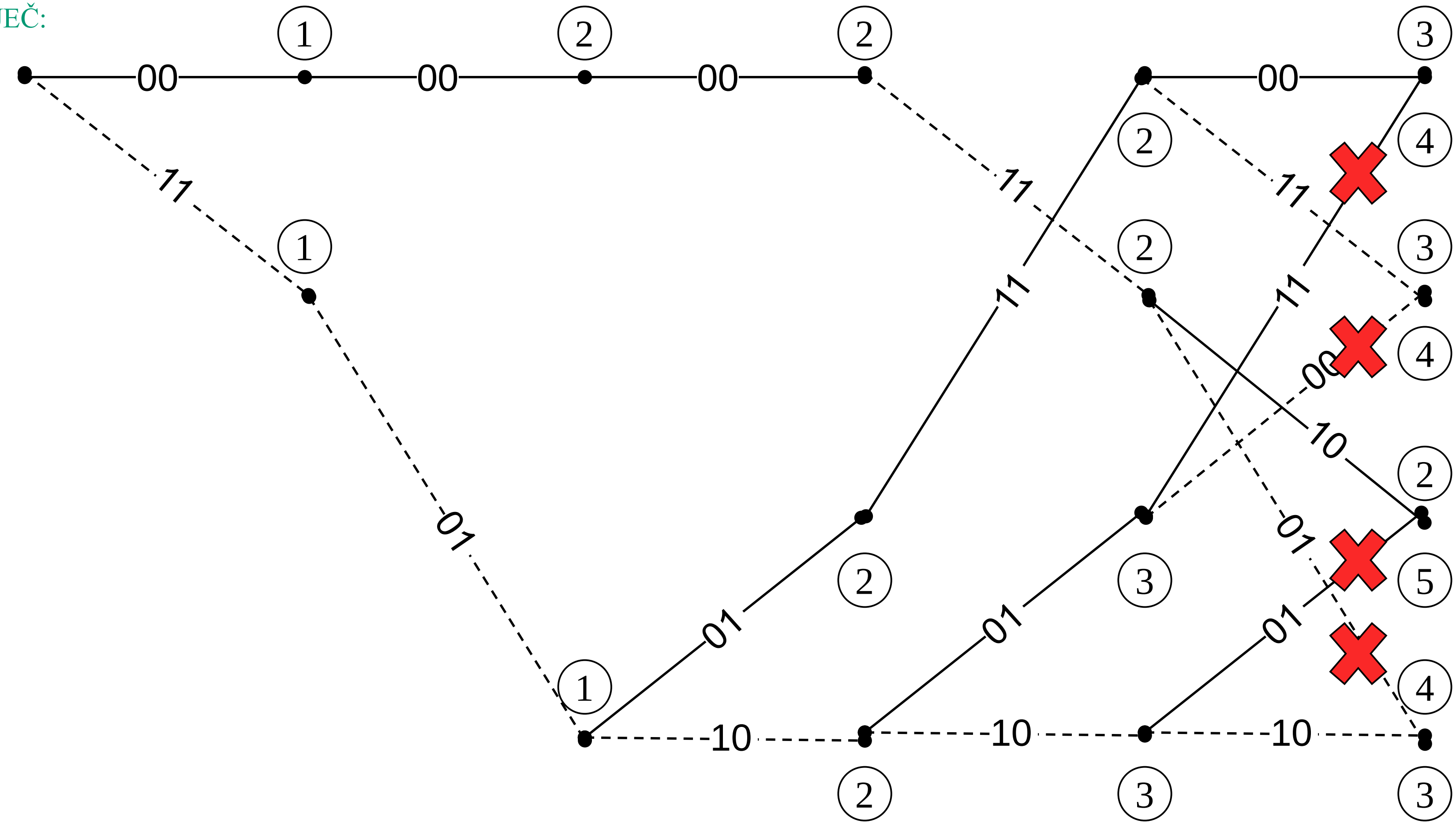
PRIMLJENA RIJEČ: 01 01 00 11 10 00



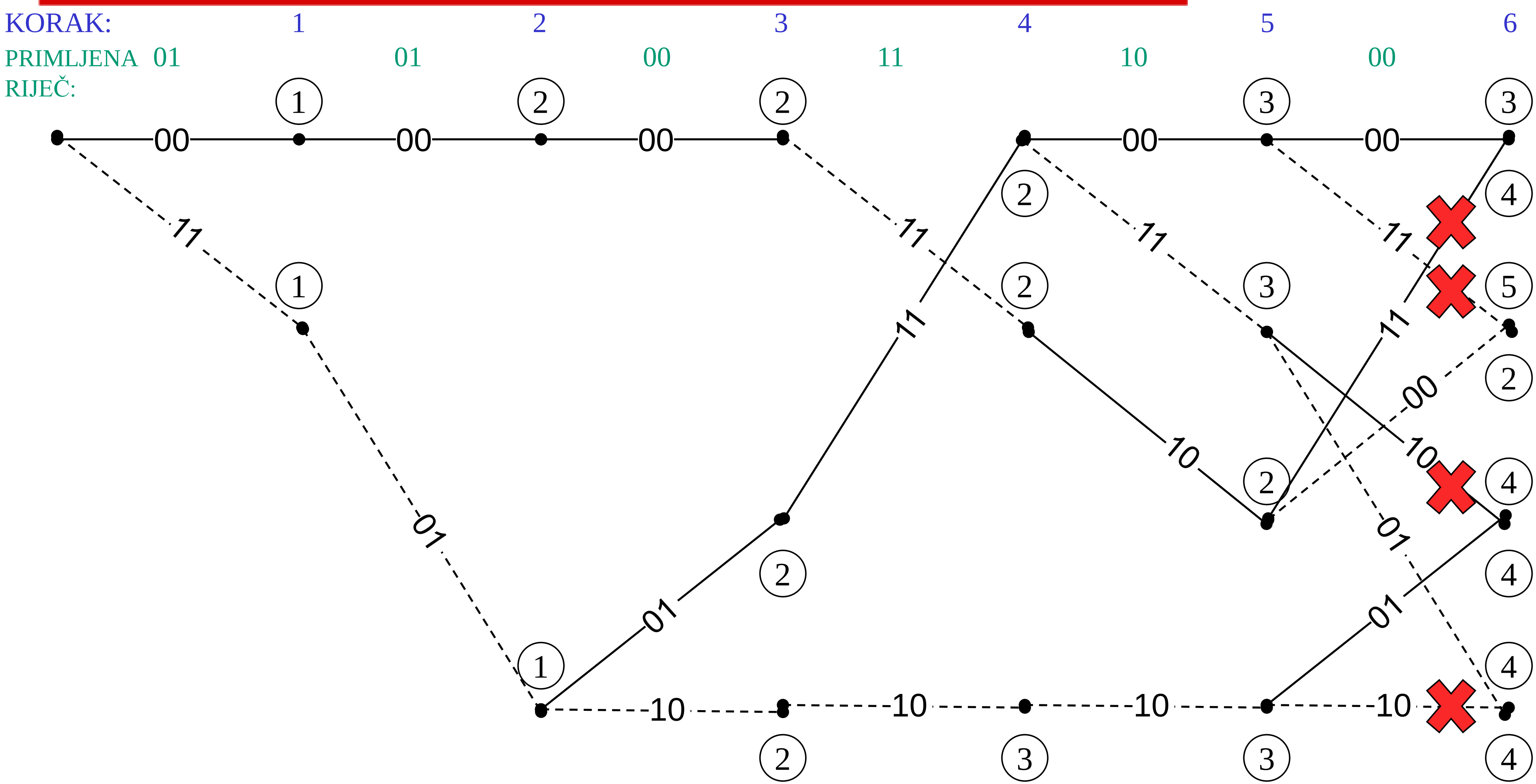
Viterbijev algoritam: primjer (3/5)

KORAK: 1 2 3 4 5 6

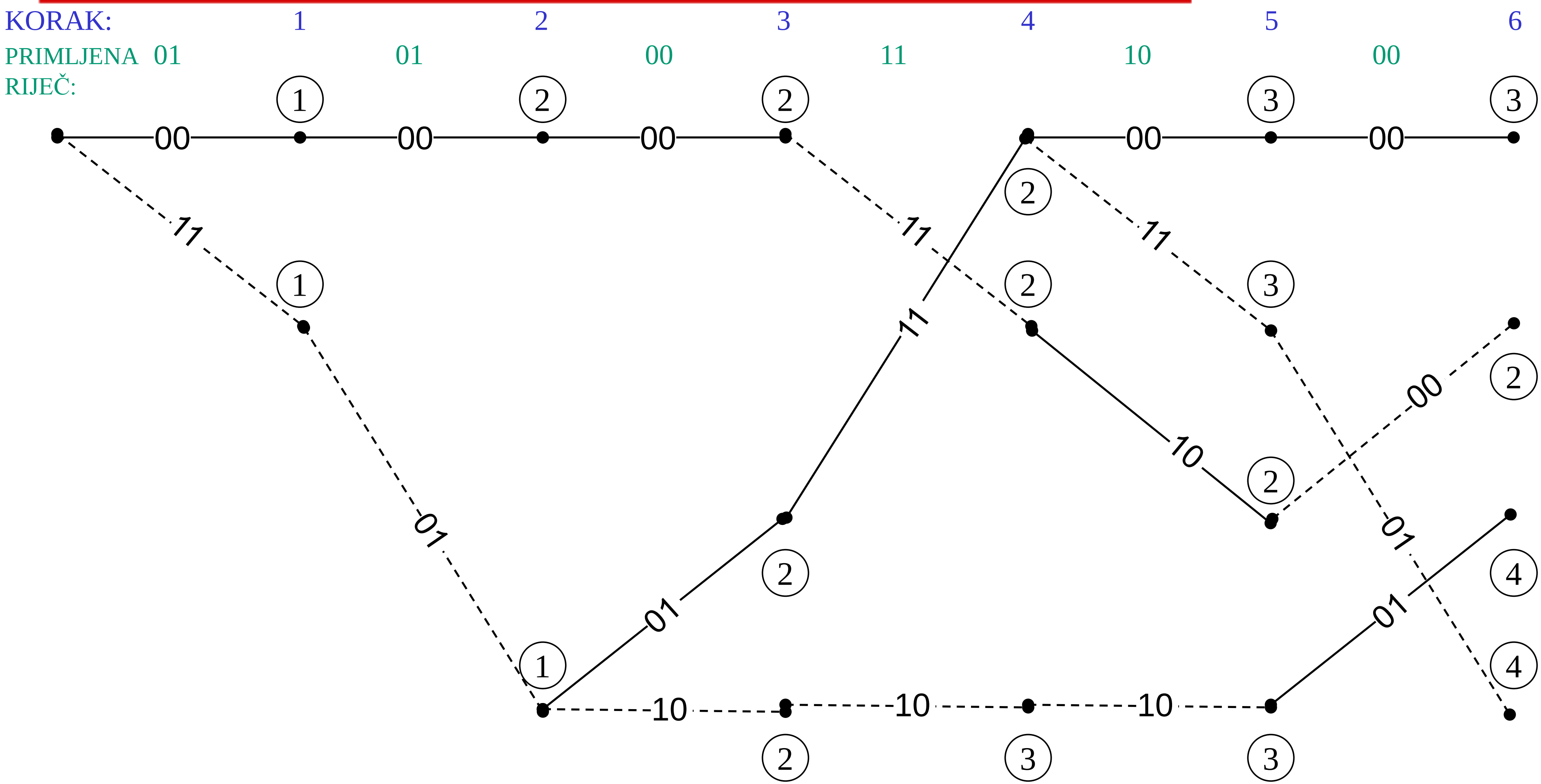
PRIMLJENA RIJEČ: 01 01 00 11 10 00



Viterbijev algoritam: primjer (4/5)



Viterbijev algoritam: primjer (5/5)



Viterbijev algoritam: neka praktična pitanja

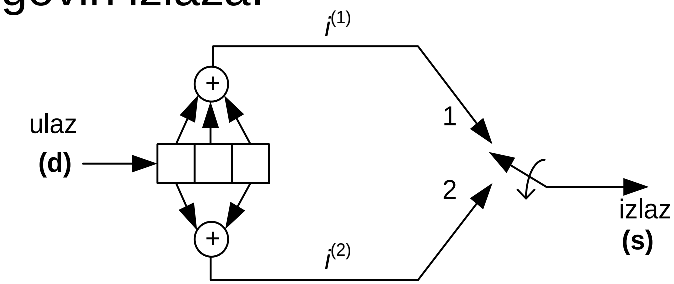
- Iz primjera je vidljivo da dekoder (koristi Viterbijev algoritam) u potpunosti može početi s radom (sva stanja su uključena) nakon trećeg koraka grananja.
- Pitanje: Koliko dugo (do kojeg koraka grananja) algoritam treba ponavljati, tj. kada treba donijeti odluku o primljenom slijedu bitova? Na ovaj način se određuje dio bitova koji pripadaju izvornoj poruci.
  - Odgovor na dano pitanje je jako bitan jer cijena dekodera ovisi o veličini memorije u koju se spremaju "preživjeli" putovi.
  - Pokazuje se da veličina memorije koja je 4 do 5 puta dulja od L daje performanse koda bliske optimumu.
    - Kao izlaz dekodera, tj. bitovima poruke proglašavaju se bitovi koji pripada najvjerojatnijem putu od svih "preživjelih".
    - Kad se donese odluka o izlazu dekodera svi "preživjeli" putovi u memoriji se brišu i u istu spremaju novi.
- Pitanje: Što dekoder radi ako u istom stanju ima dva puta koji imaju jednaku metriku?
  - U takvim prilikama dekoder odabire slučajno jedan od dva puta.
- Drugi algoritmi dekodiranja konvolucijskih kodova.
  - Sekvencijalni algoritam (dosta sličan Viterbijevom algoritmu);
  - Algoritam dekodiranja s povratnom vezom (engl. *feedback decoding*).

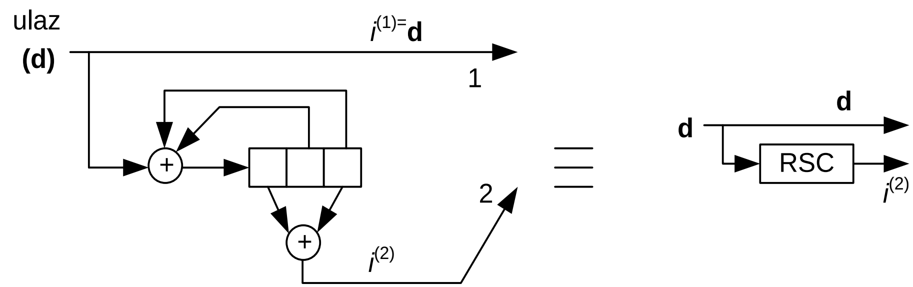
Druge primjene Viterbijevog algoritma

- Skriveni Markovljevi modeli (Hidden Markov Models, HMM)
  - Ne vidimo stanja nego *opažanja* uzrokovana stanjima
  - V.A. Omogućava da iz slijeda opažanja zaključimo najvjerojatniji slijed stanja
- Poznata primjena u raspoznavanju govora

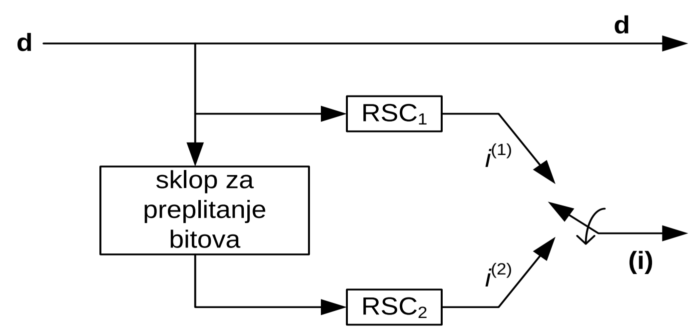
Turbo kodovi (1/2)

- Podgrupa konvolucijskih kodova.
- Kodiranje se temelji na paralelnom ulančavanju nekih klasa sistematskih konvolucijskih kodova, tzv. *rekurzivni sistematski konv. kodovi* (RSCC, engl. *recursive systematic convolutional codes*).
- RSCC → nesistematski koder u kojem se na njegov ulaz spaja jedan ili više njegovih izlaza.





- Turbo kodovi → paralelna veza najčešće dva jednaka RSC kodera odvojena sklopom za preplitanje bitova (engl. *interleaver*).



Turbo kodovi (2/2)

- Sklop za preplitanje bitova permutira bitove: jedan od glavnih razloga dobrih svojstava turbo kodova
- Uporaba: 3G i 4G pokretne mreže, DVB, satelitske komunikacije
- Berrou, Claude; Glavieux, Alain; Thitimajshima, Punya 1993
  - "U Francuskoj je poznata šala o "glupanu" koji nije znao da se nešto ne može, pa je to napravio."