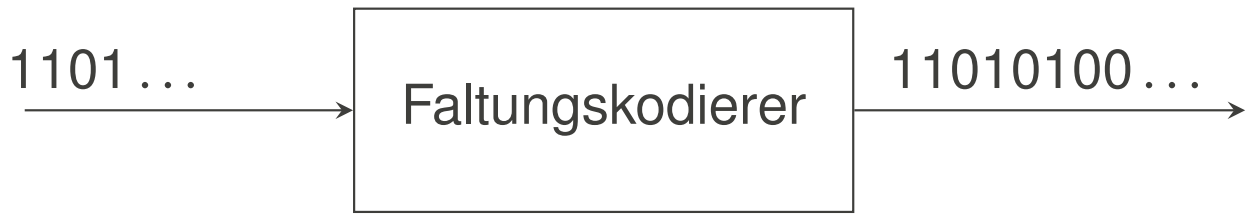
Martin Nocker

universität innsbruck

R-Paket für Kanalkodierung mit Faltungskodes

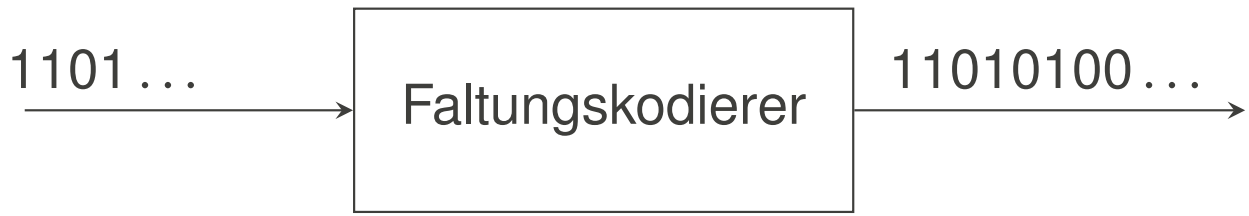
Faltungskodes

- Informationsstrom - keine Blöcke fester Länge
- ein einziges Kodewort als Resultat
- Redundanz wird kontinuierlich eingefügt



Faltungskodes

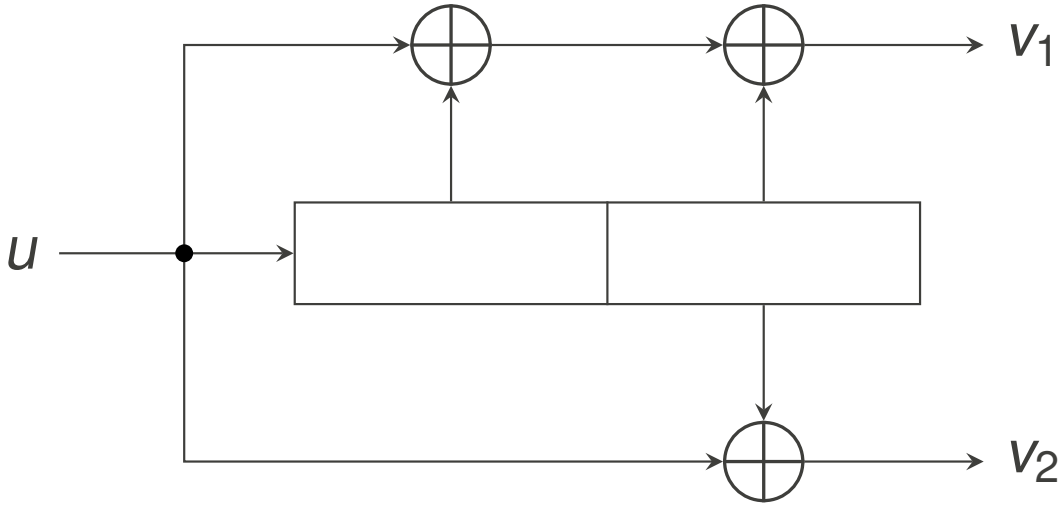
- Informationsstrom - keine Blöcke fester Länge
- ein einziges Kodewort als Resultat
- Redundanz wird kontinuierlich eingefügt



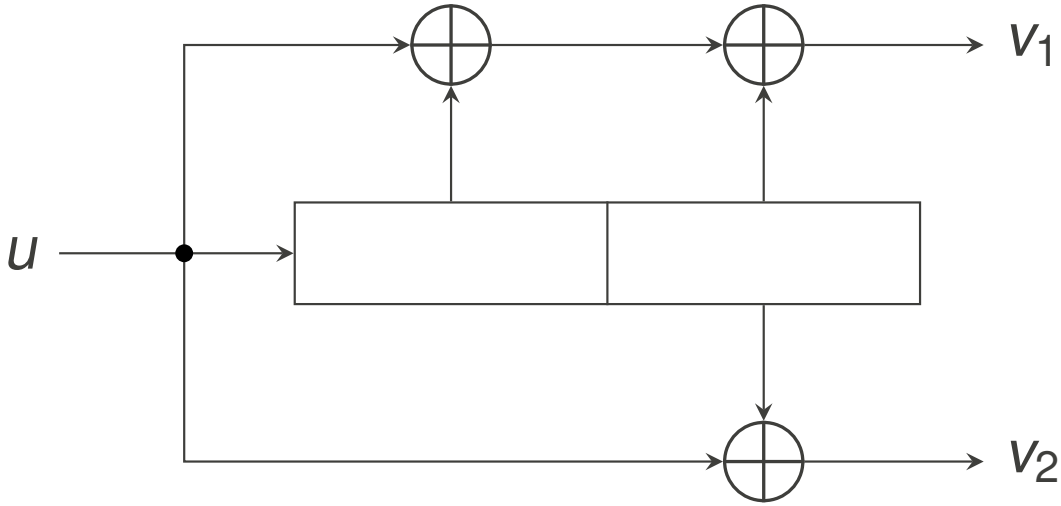
Verwendung

- Mobil- und Satellitenkommunikation
- Turbokodes

Faltungskodierer

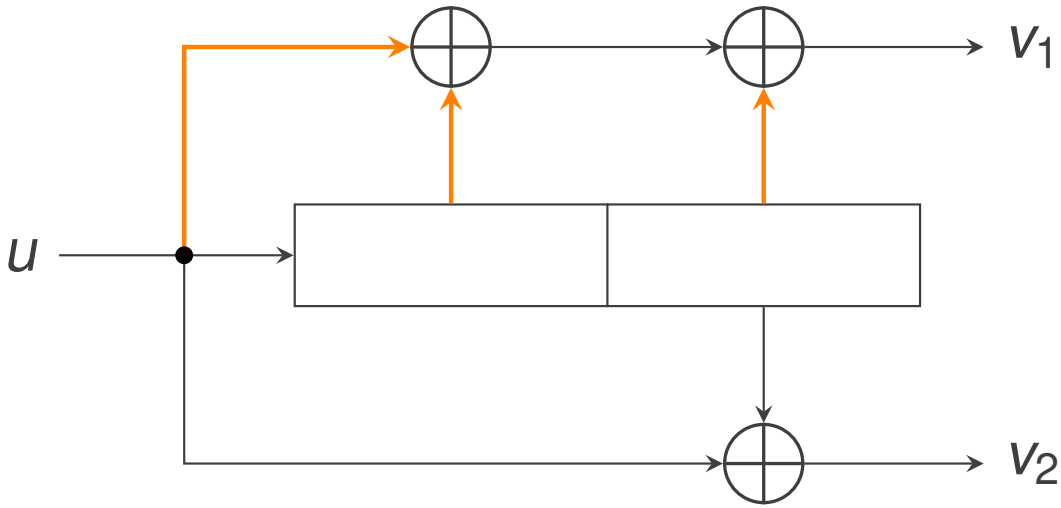


Faltungskodierer



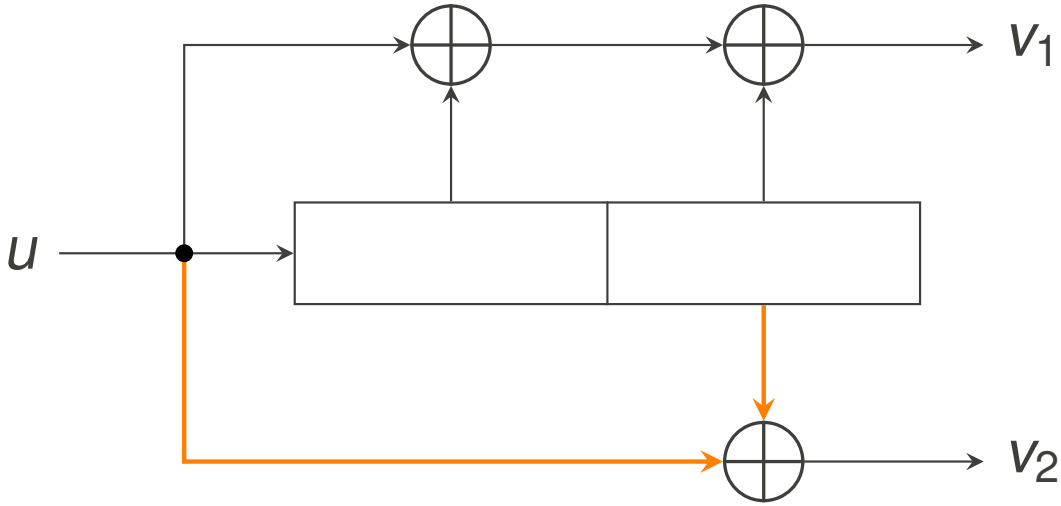
$$G = \left(\quad \right)$$

Faltungskodierer



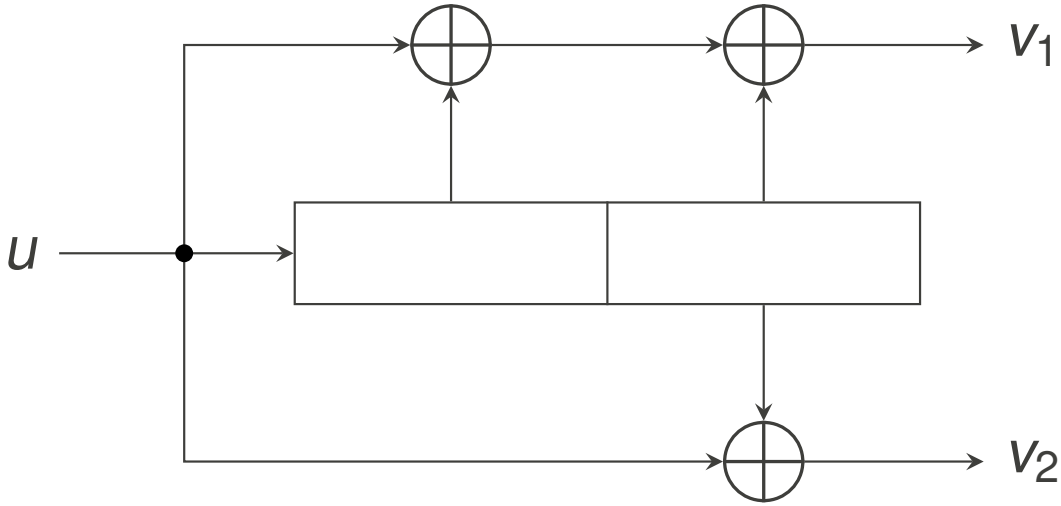
$$G = \begin{pmatrix} 111 \end{pmatrix}$$

Faltungskodierer



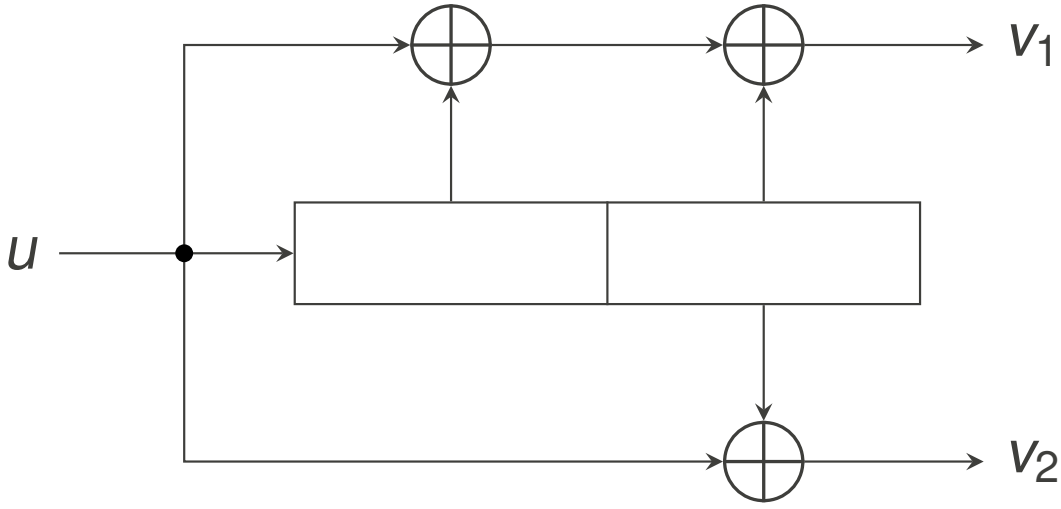
$$G = \begin{pmatrix} 111 \\ 101 \end{pmatrix}$$

Faltungskodierer

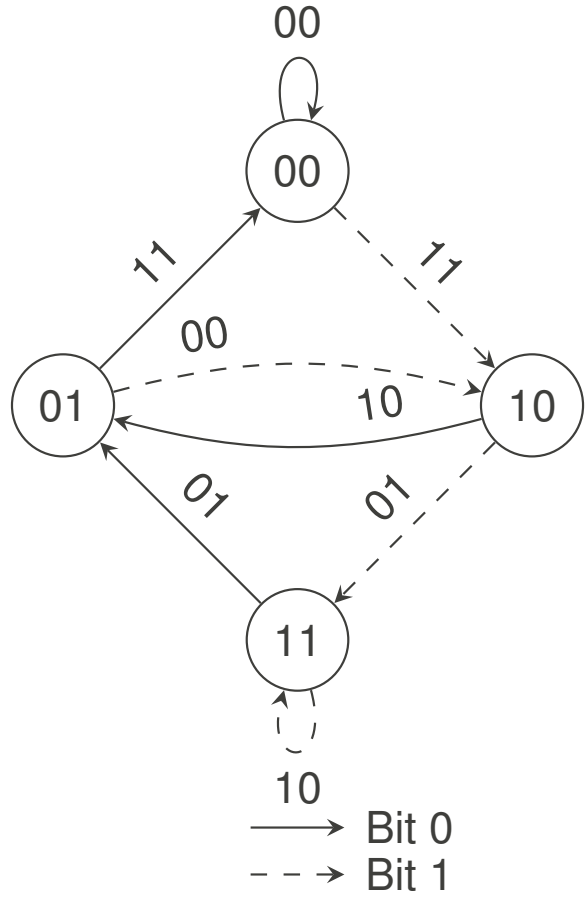


$$G = \begin{pmatrix} 111 \\ 101 \end{pmatrix} = (7_8, 5_8)$$

Faltungskodierer



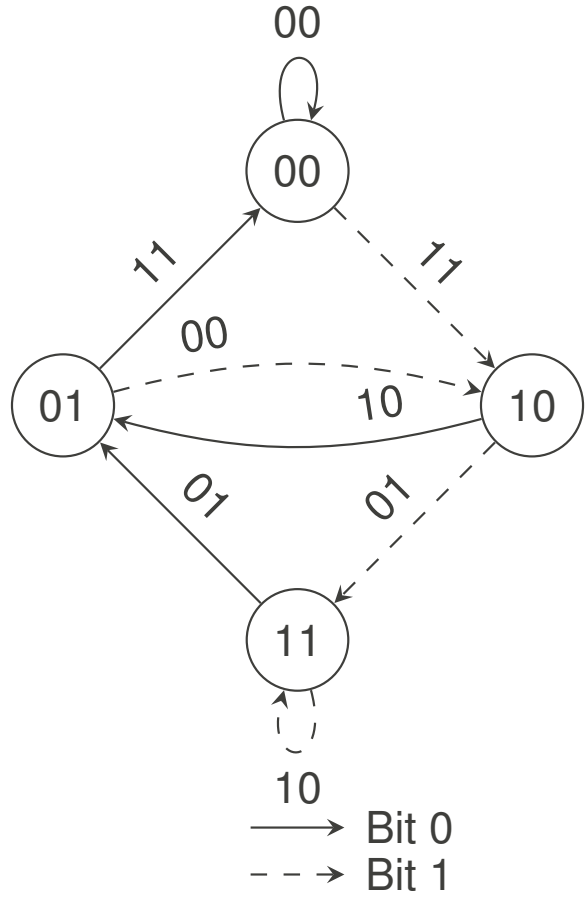
$$G = \begin{pmatrix} 111 \\ 101 \end{pmatrix} = (7_8, 5_8)$$



Faltungskodierung

Nachricht: (1, 1, 0, 1, 0, 0)

Input	Zustand	Folgezustand	Output

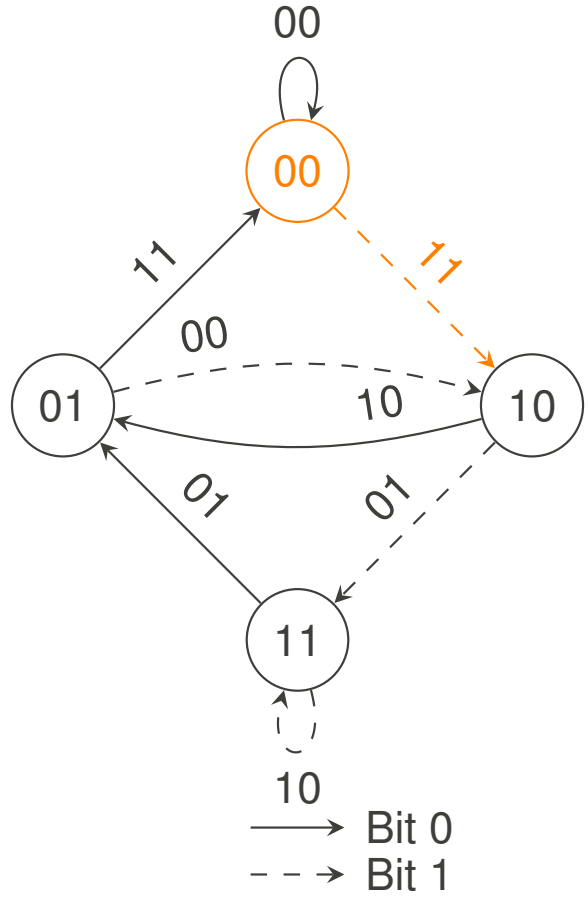


Kode :

Faltungskodierung

Nachricht: (1, 1, 0, 1, 0, 0)

Input	Zustand	Folgezustand	Output
1	00	10	11

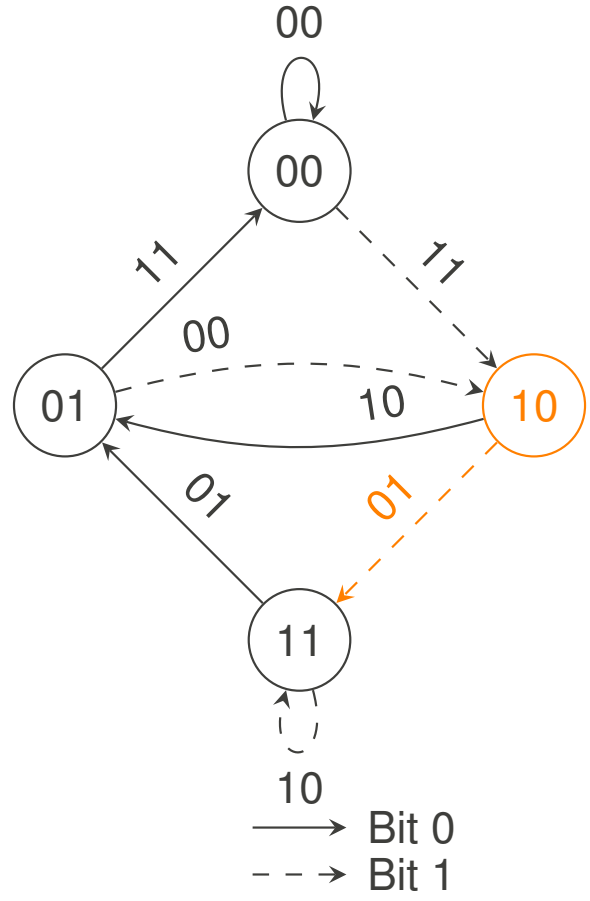


Kode : (1, 1

Faltungskodierung

Nachricht: (1, 1, 0, 1, 0, 0)

Input	Zustand	Folgezustand	Output
1	00	10	11
1	10	11	01

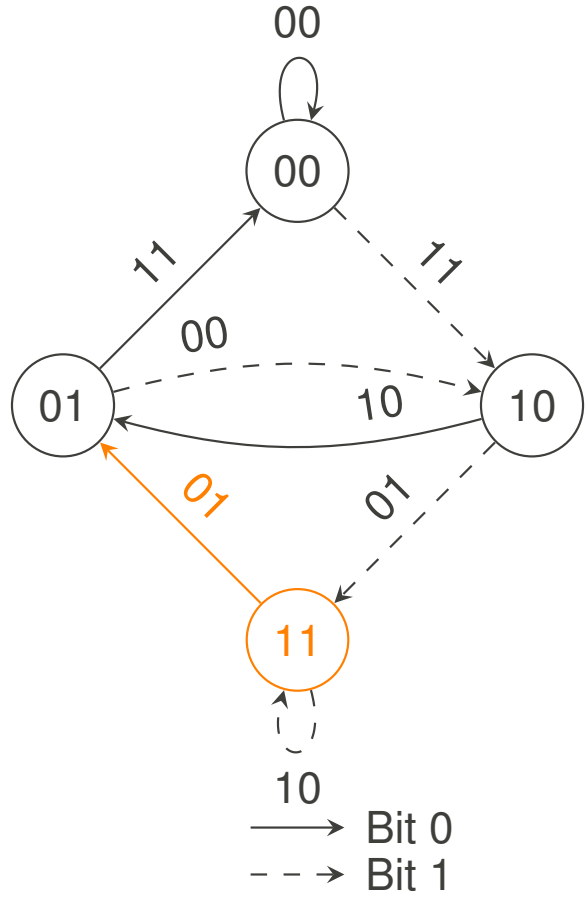


Kode : (1, 1, 0, 1

Faltungskodierung

Nachricht: (1, 1, 0, 1, 0, 0)

Input	Zustand	Folgezustand	Output
1	00	10	11
1	10	11	01
0	11	01	01

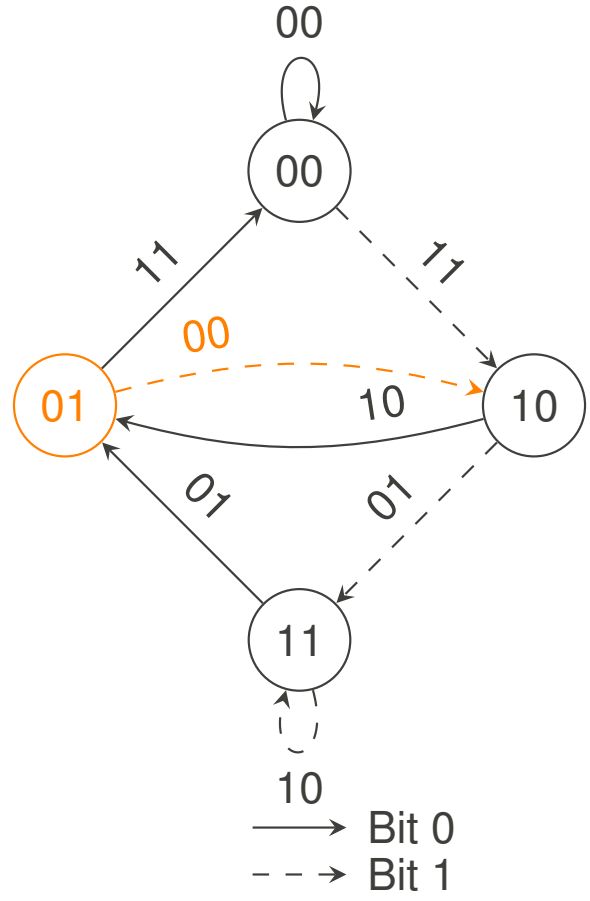


Kode : (1, 1, 0, 1, 0, 1

Faltungskodierung

Nachricht: (1, 1, 0, 1, 0, 0)

Input	Zustand	Folgezustand	Output
1	00	10	11
1	10	11	01
0	11	01	01
1	01	10	00

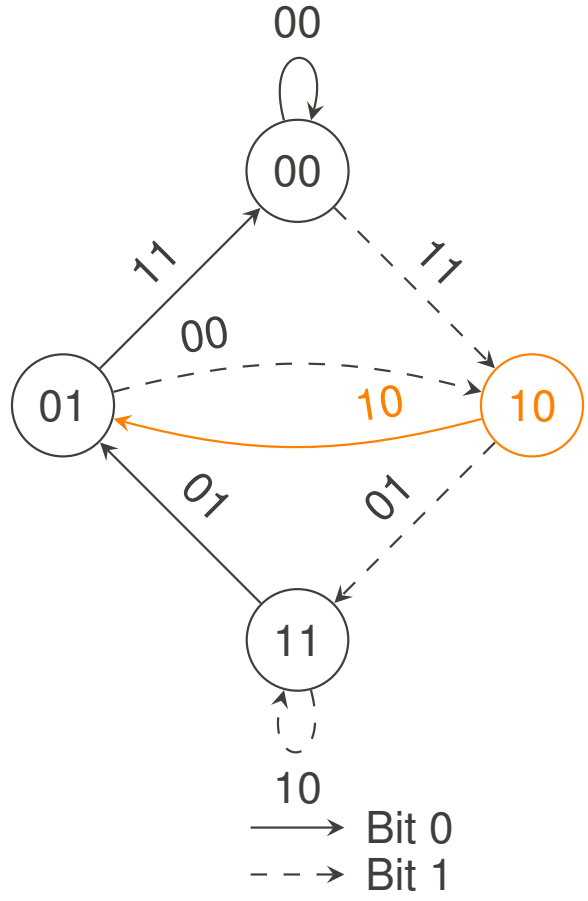


Kode : (1, 1, 0, 1, 0, 1, 0, 0

Faltungskodierung

Nachricht: (1, 1, 0, 1, 0, 0)

Input	Zustand	Folgezustand	Output
1	00	10	11
1	10	11	01
0	11	01	01
1	01	10	00
0	10	01	10

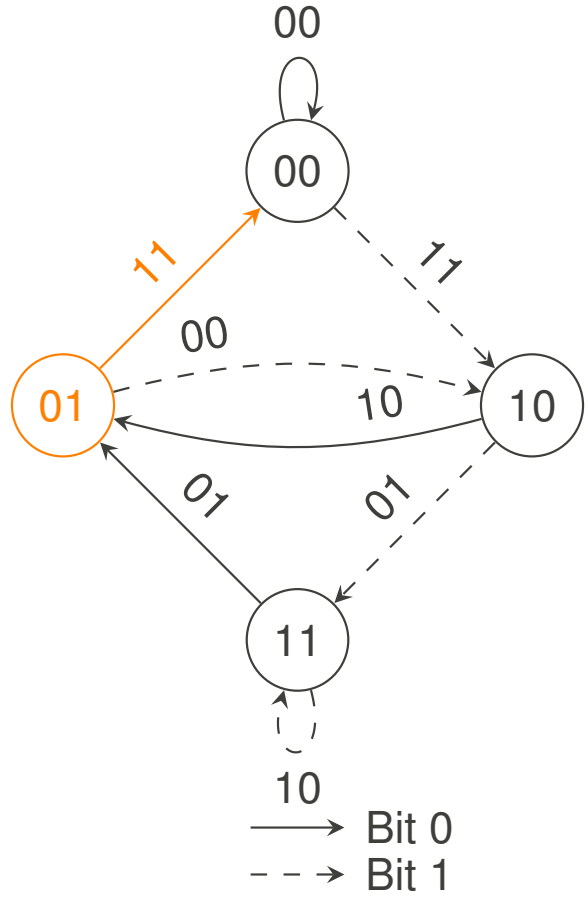


Kode : (1, 1, 0, 1, 0, 1, 0, 0, 1, 0

Faltungskodierung

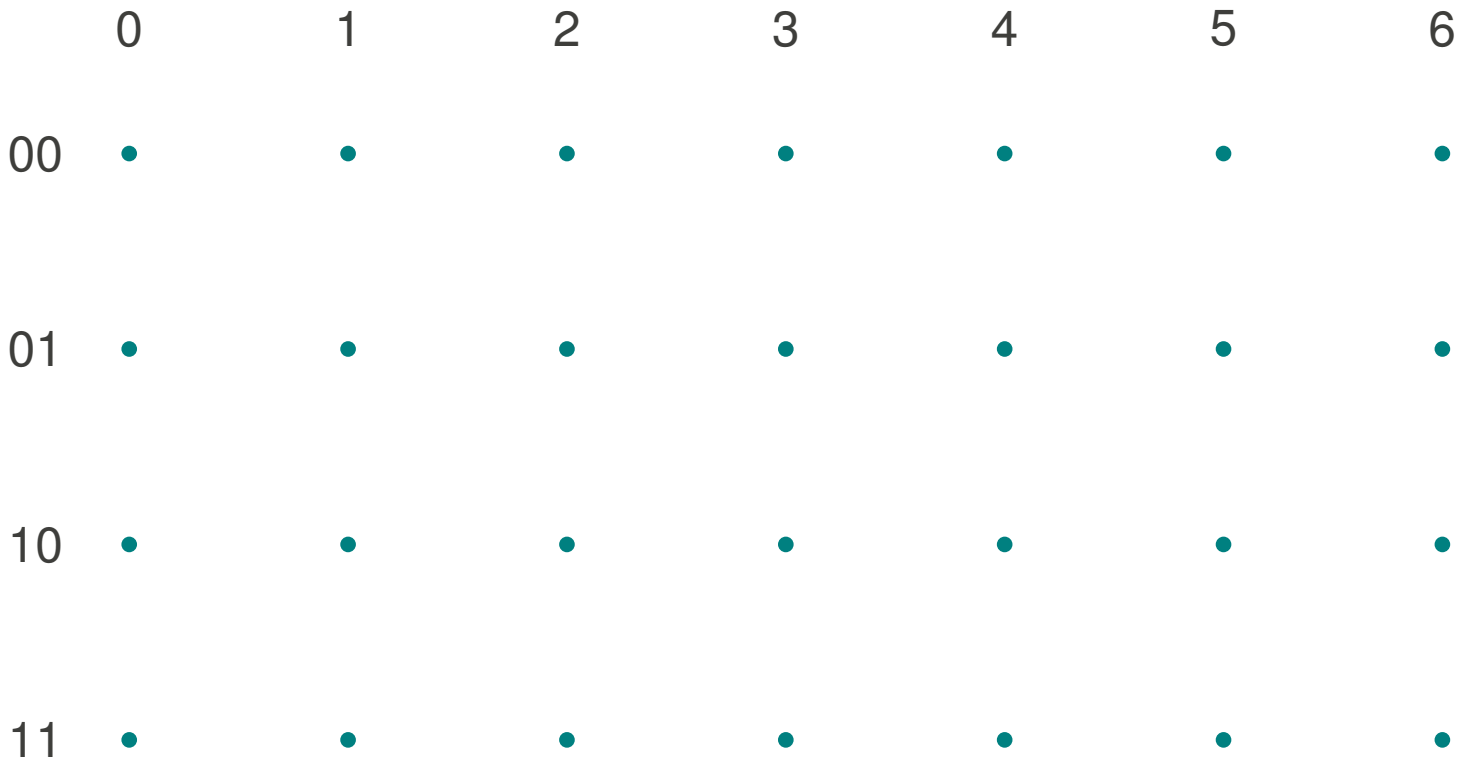
Nachricht: (1, 1, 0, 1, 0, 0)

Input	Zustand	Folgezustand	Output
1	00	10	11
1	10	11	01
0	11	01	01
1	01	10	00
0	10	01	10
0	01	00	11

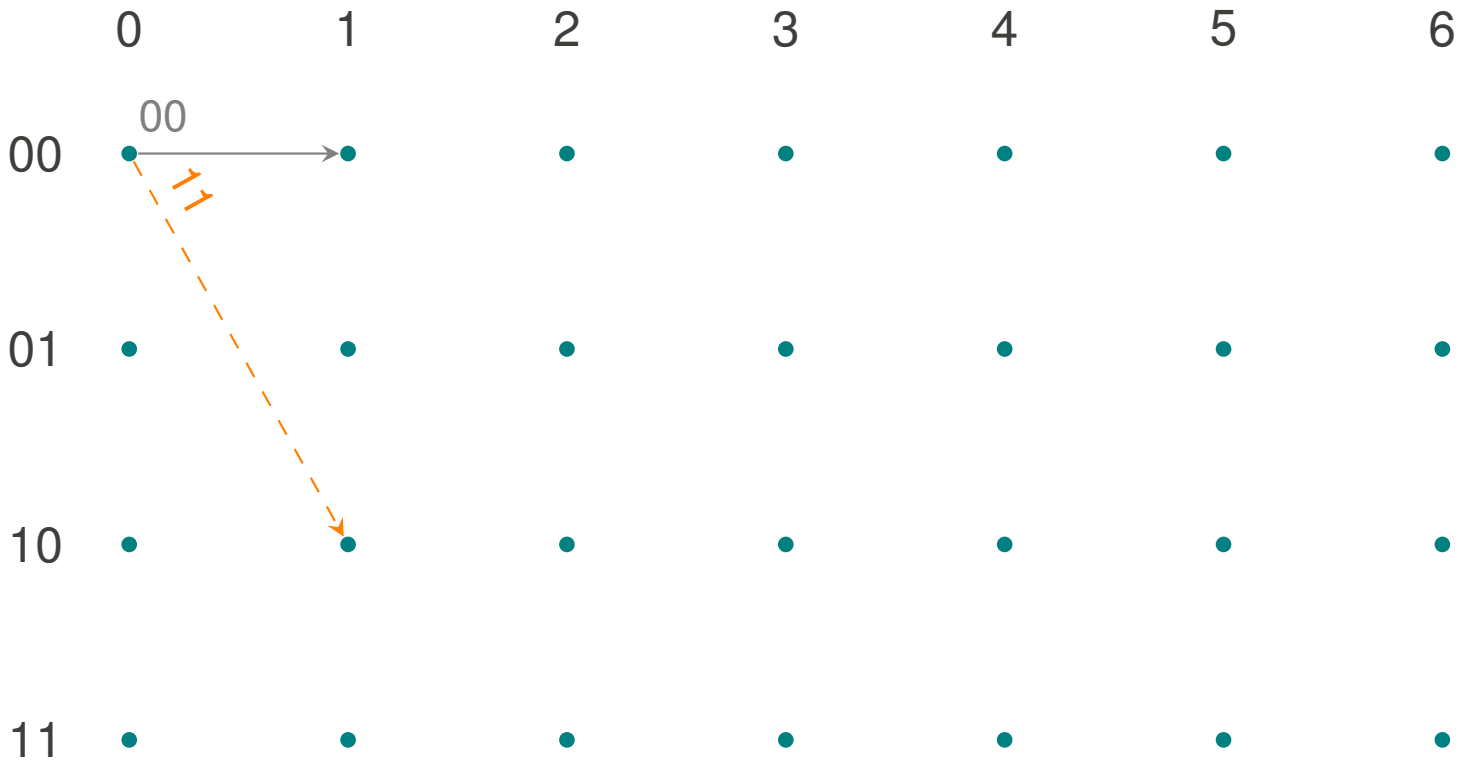


Kode : (1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1)

Trellis - Kodierung



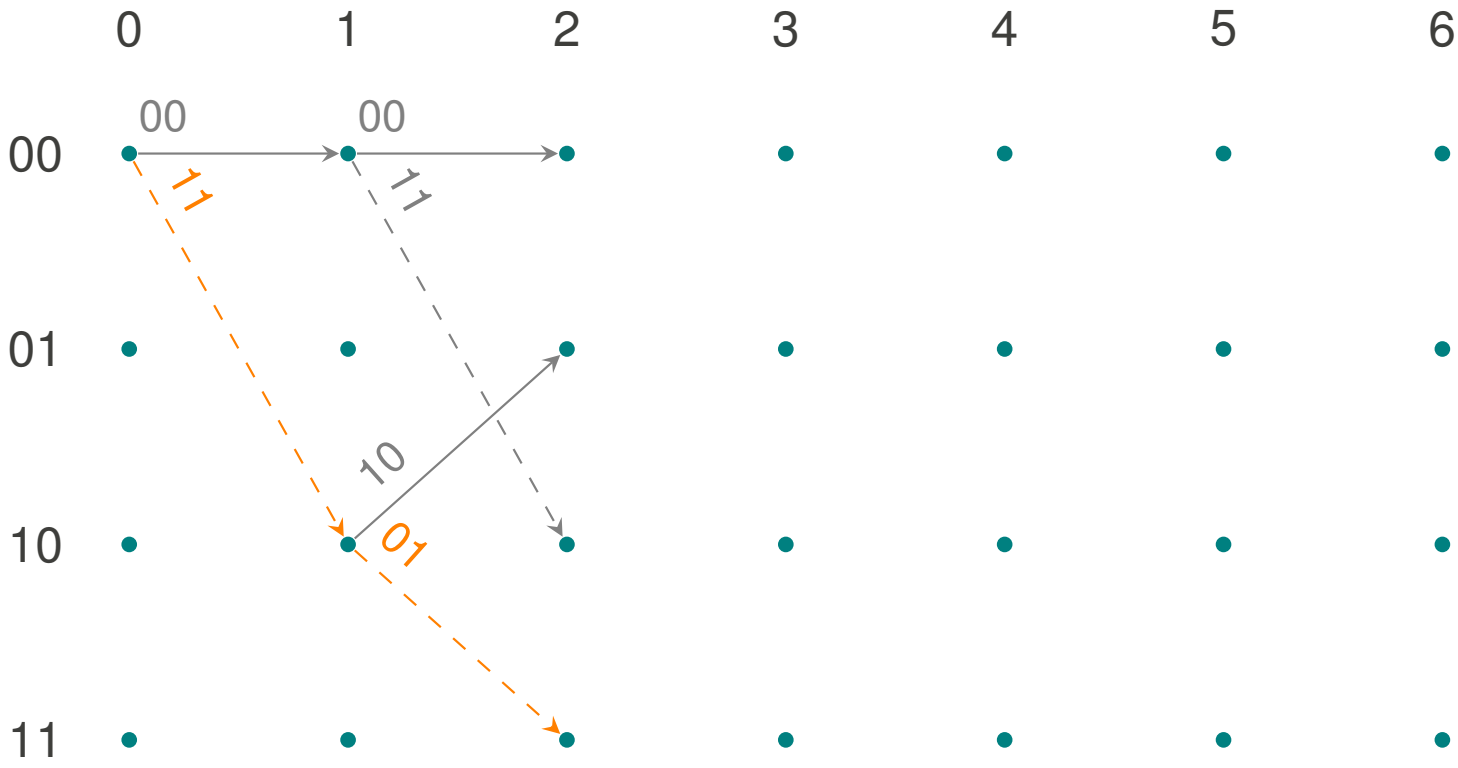
Trellis - Kodierung



Nachricht: 1 1 0 1 0 0

Kode: 11

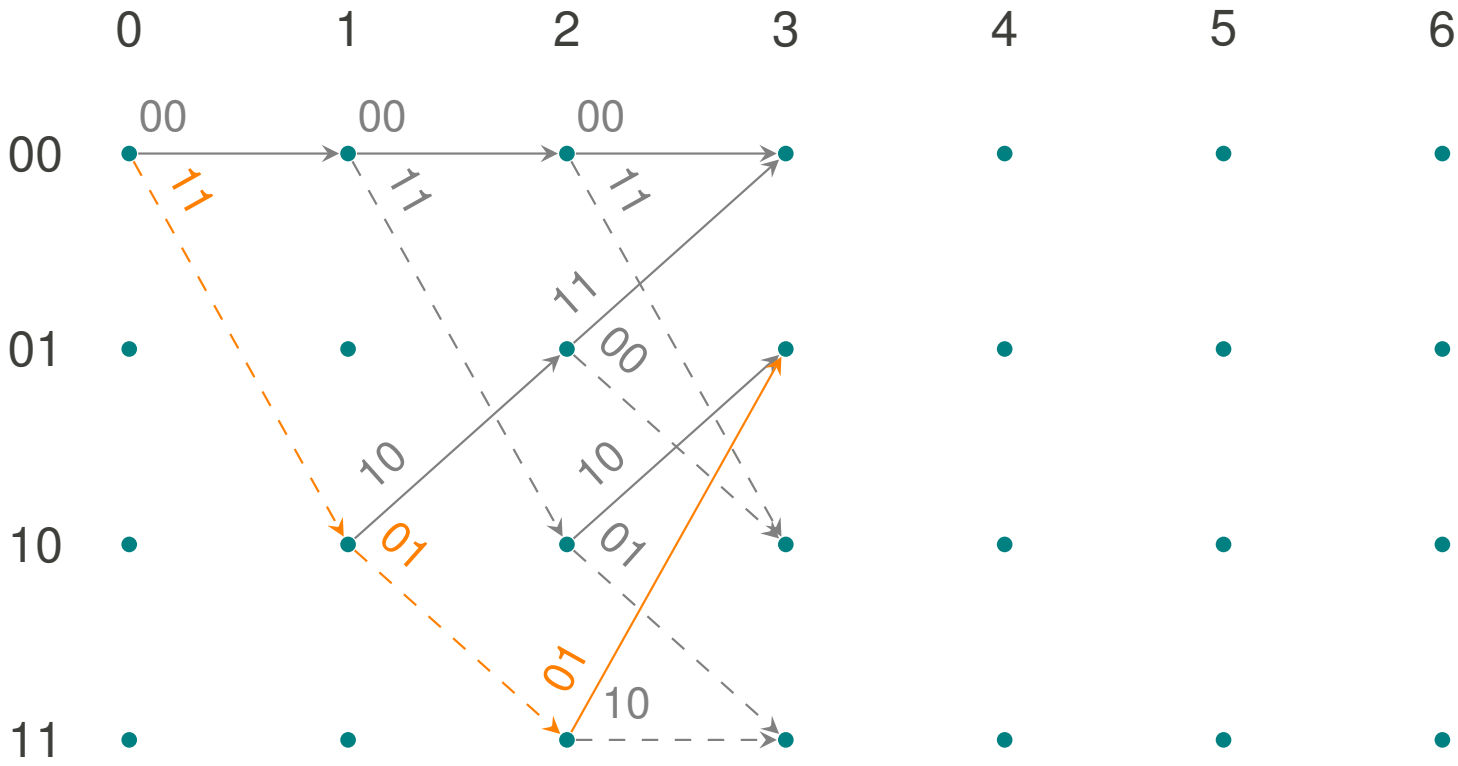
Trellis - Kodierung



Nachricht: 1 1 0 1 0 0

Kode: 11 01

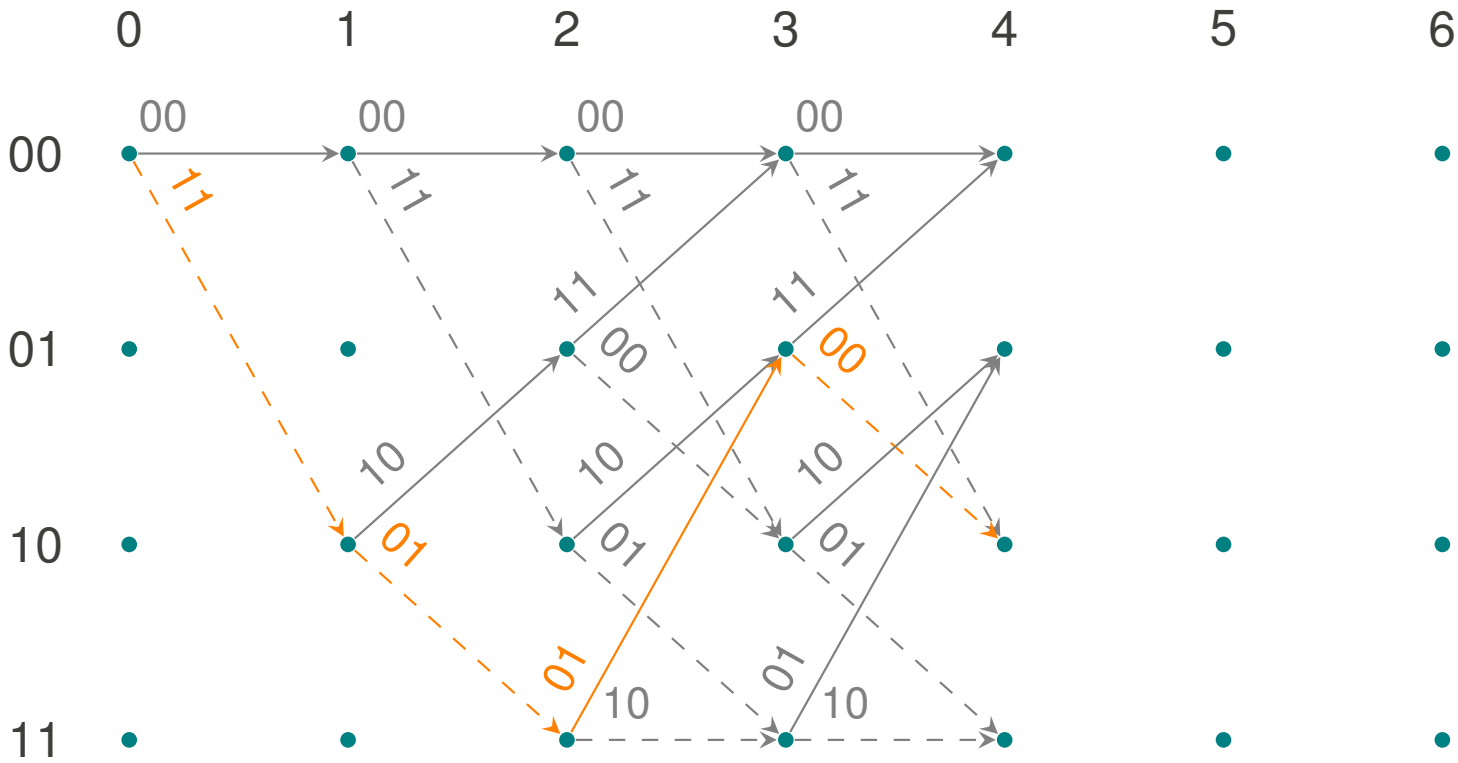
Trellis - Kodierung



Nachricht: 1 1 0 1 0 0

Kode: 11 01 01

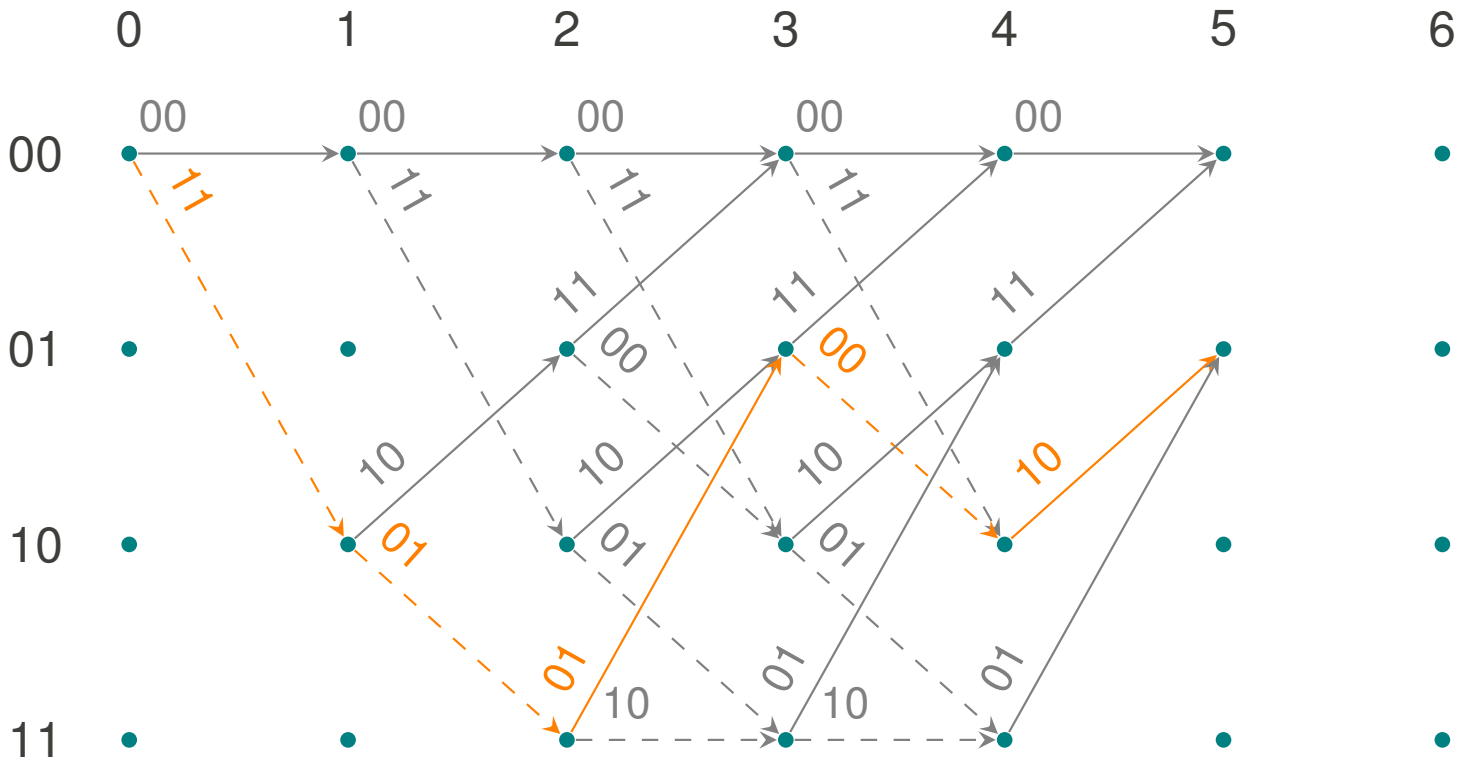
Trellis - Kodierung



Nachricht: 1 1 0 1 0 0

Kode: 11 01 01 00

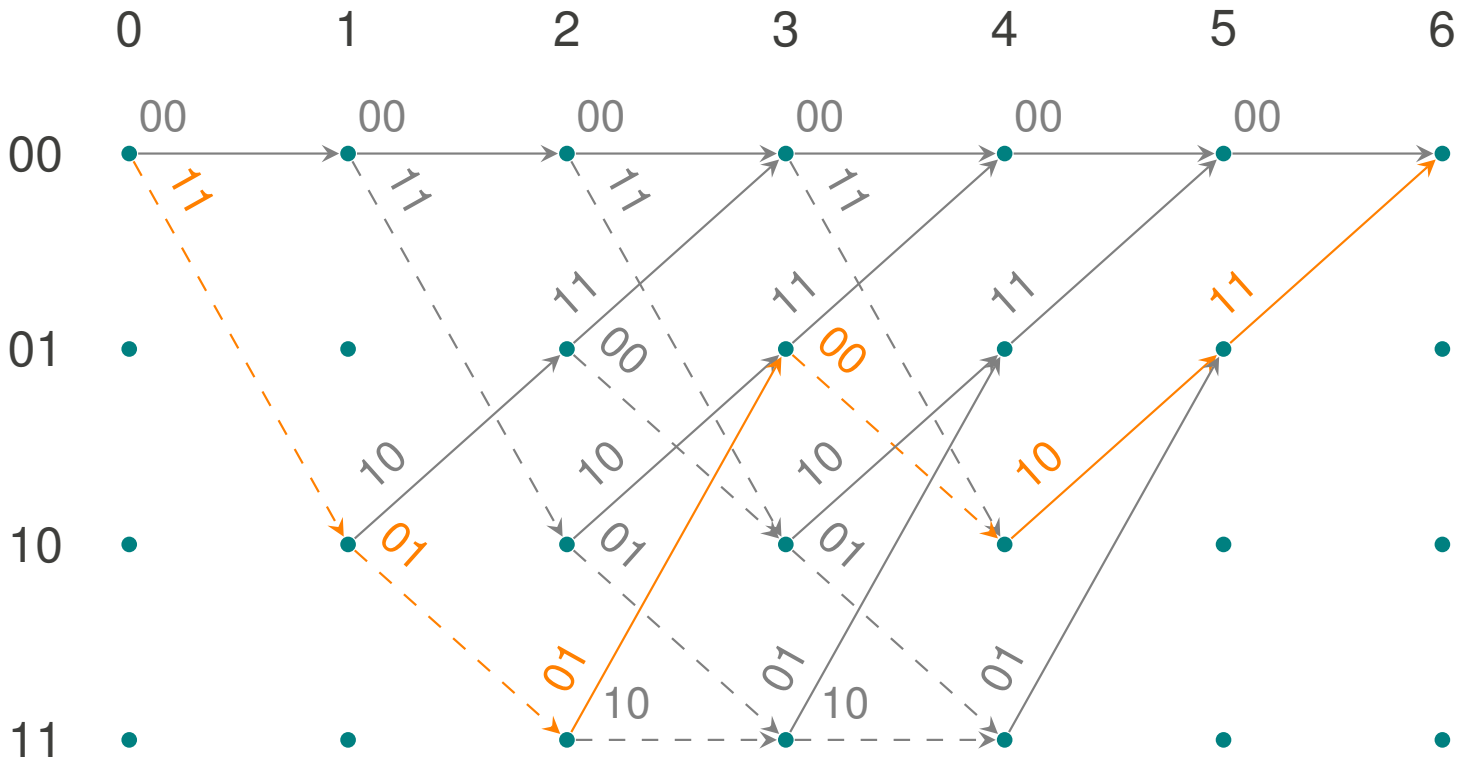
Trellis - Kodierung



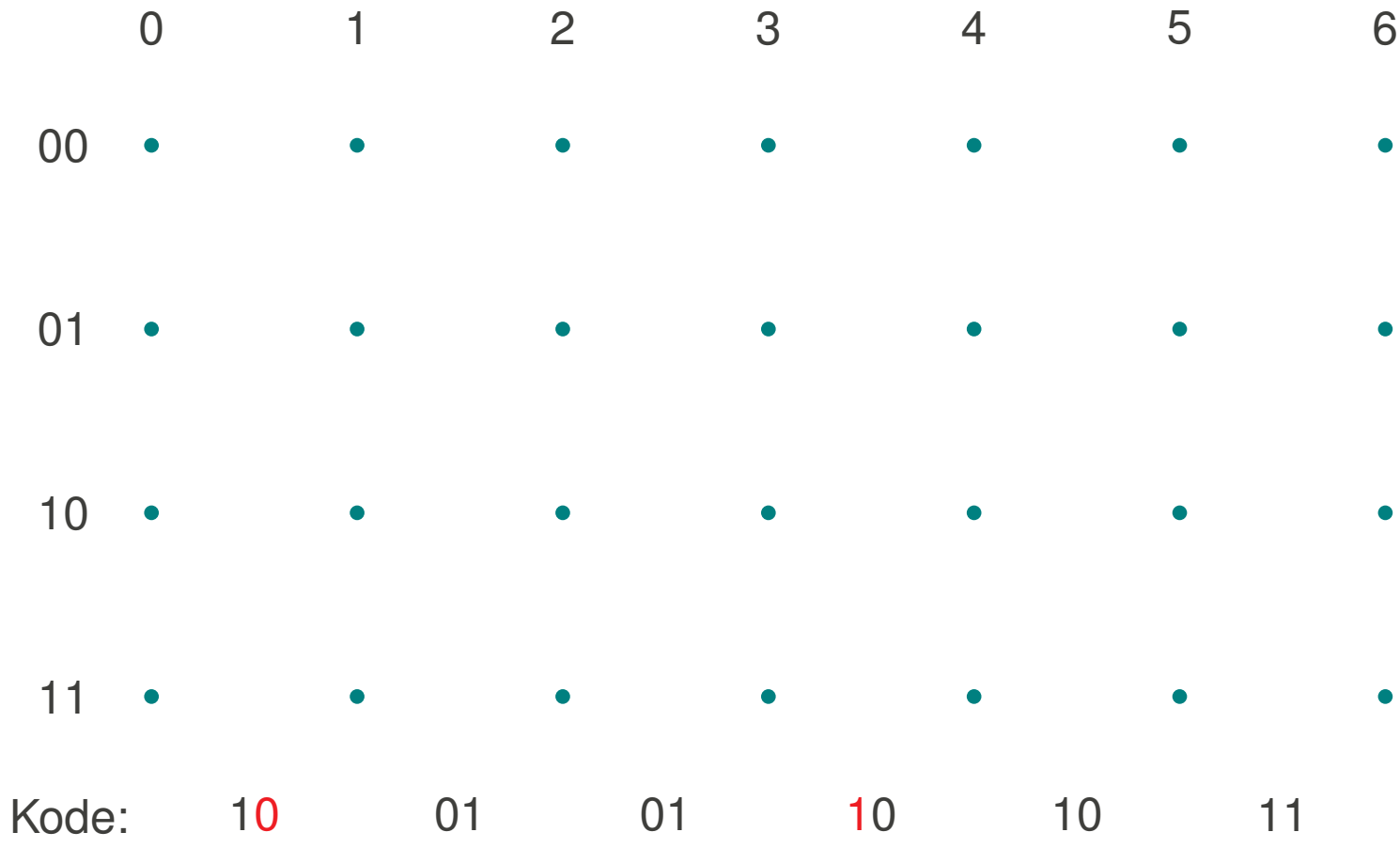
Nachricht: 1 1 0 1 0 0

Kode: 11 01 01 00 10

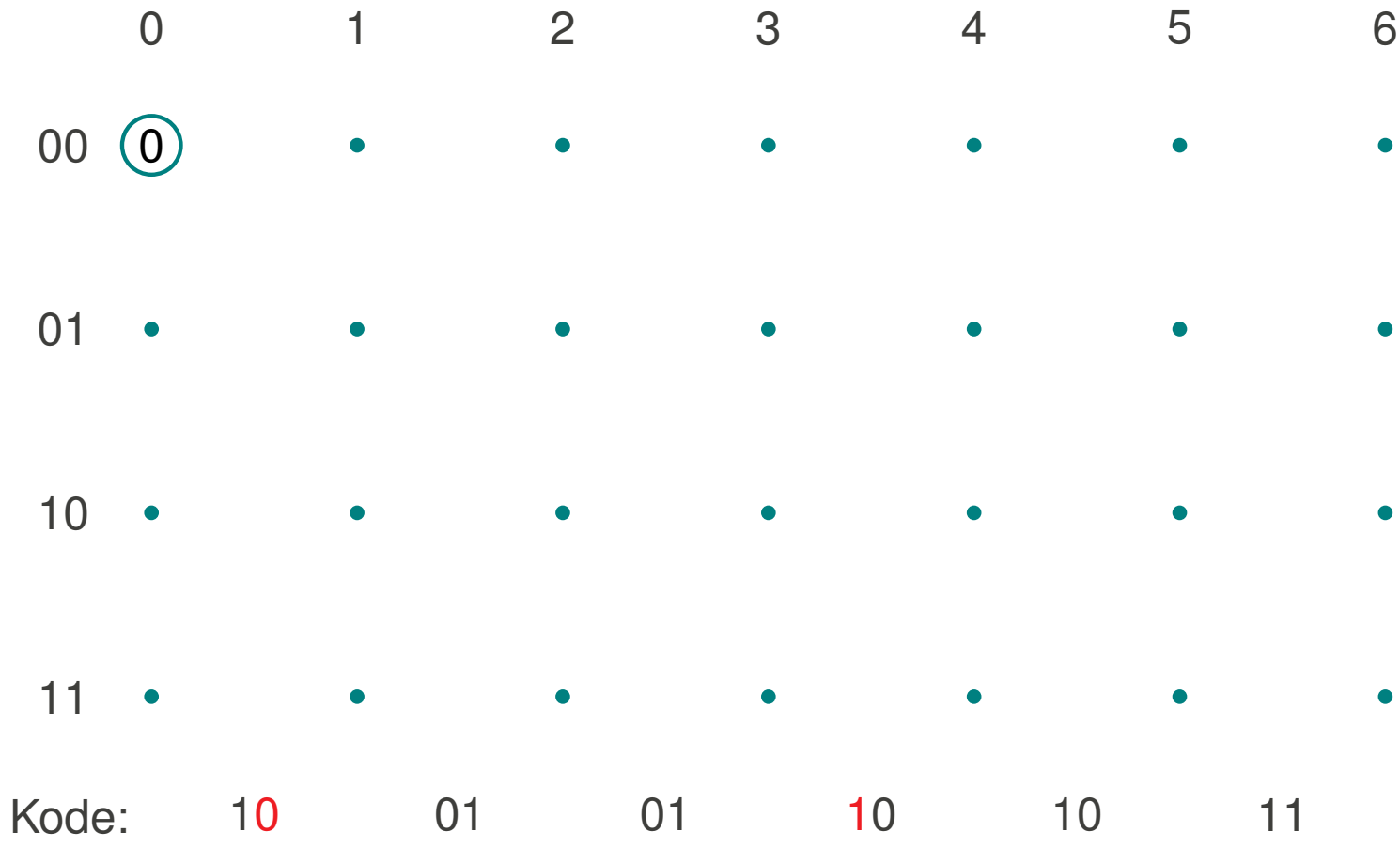
Trellis - Kodierung



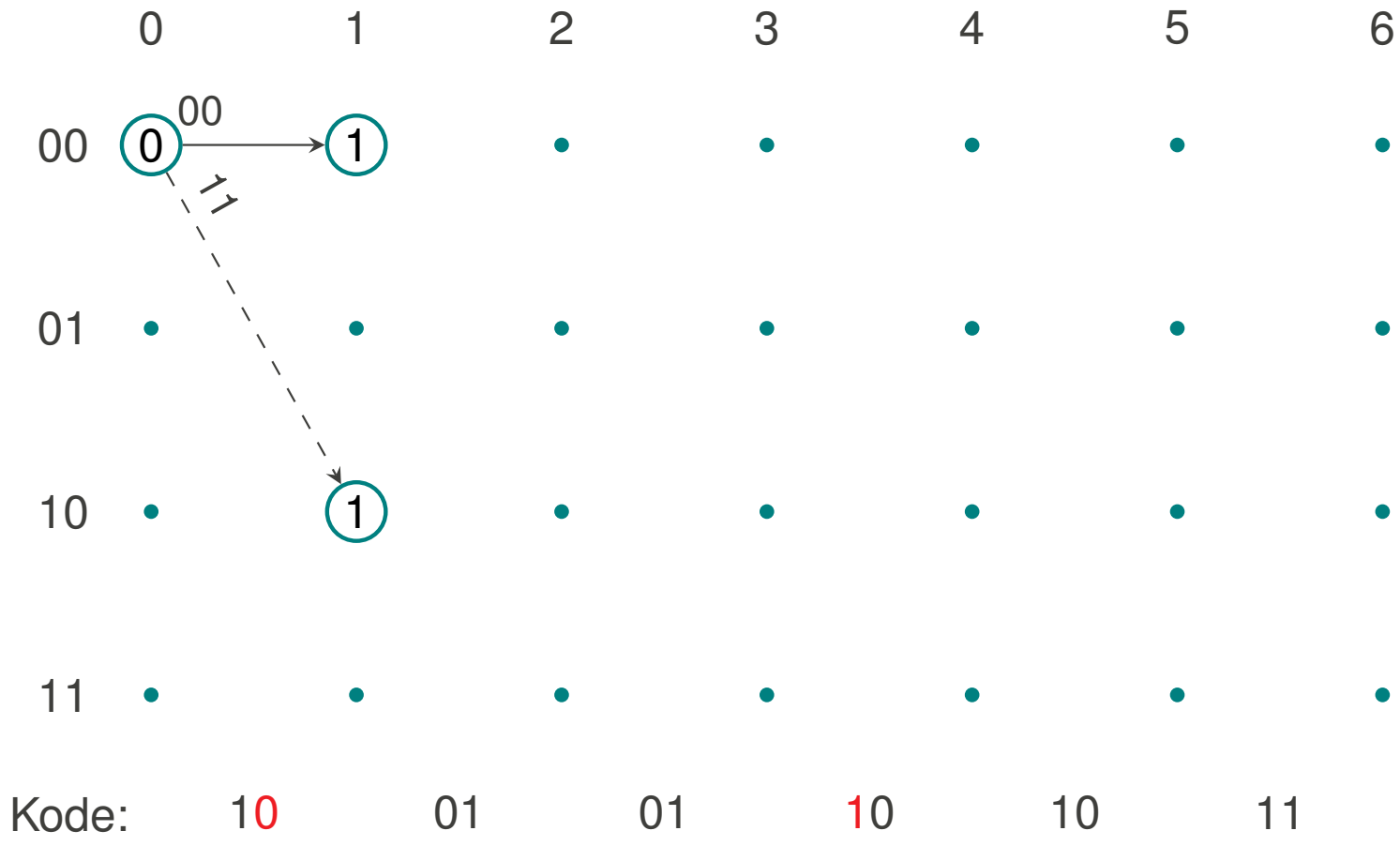
Faltungsdekodierung (Viterbi-Algorithmus)



Faltungsdekodierung (Viterbi-Algorithmus)

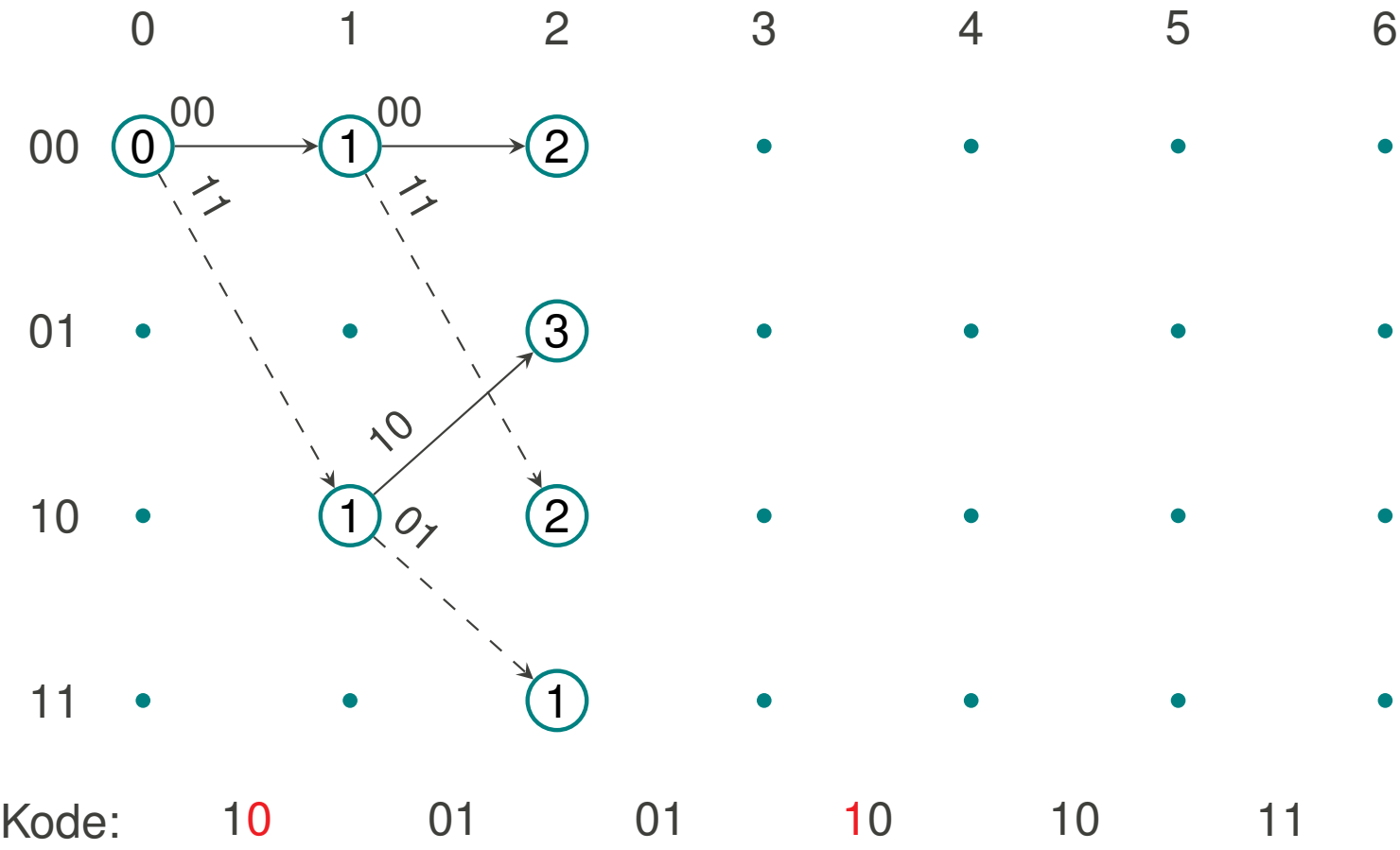


Faltungsdekodierung (Viterbi-Algorithmus)



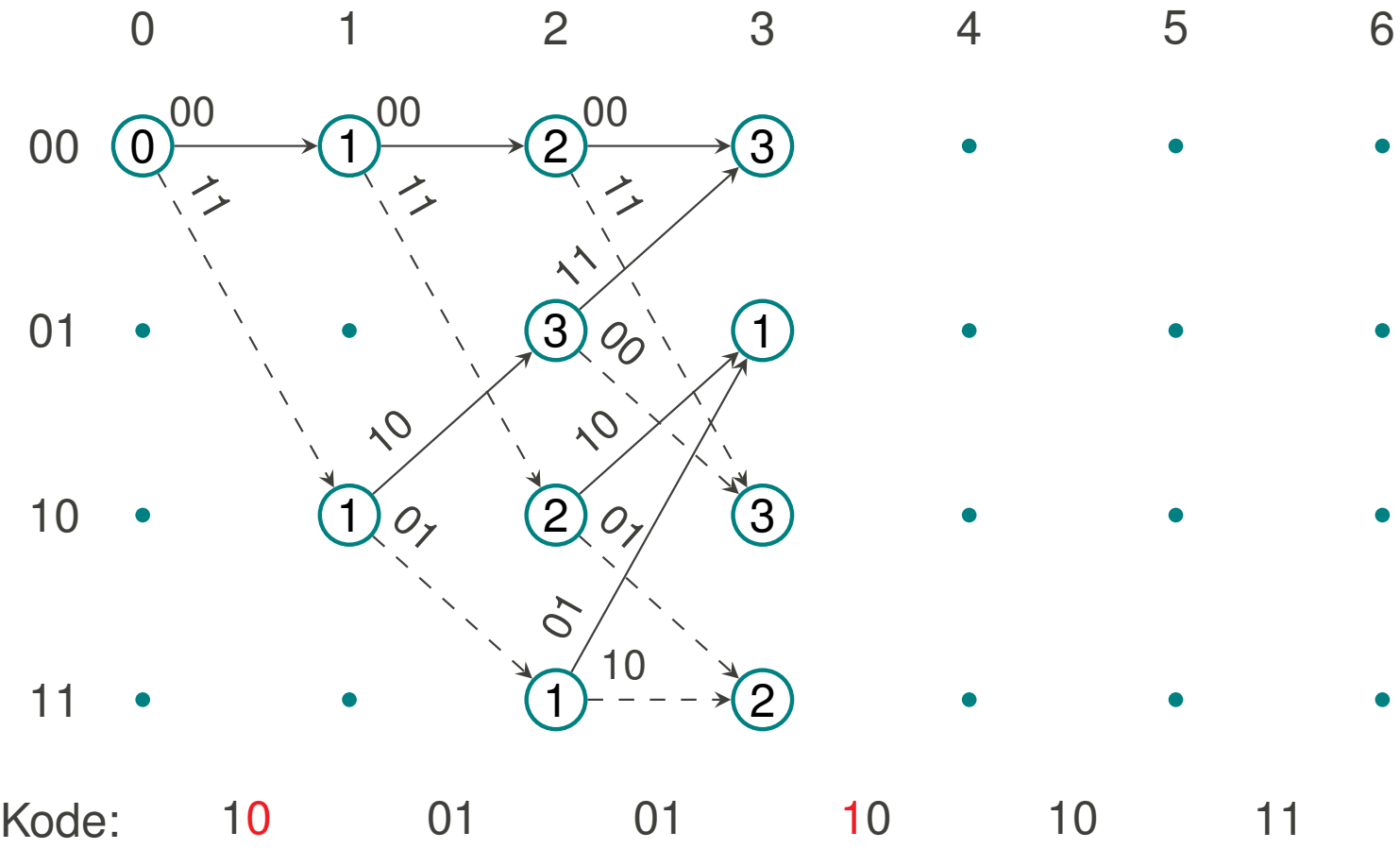
Nachricht:

Faltungsdekodierung (Viterbi-Algorithmus)



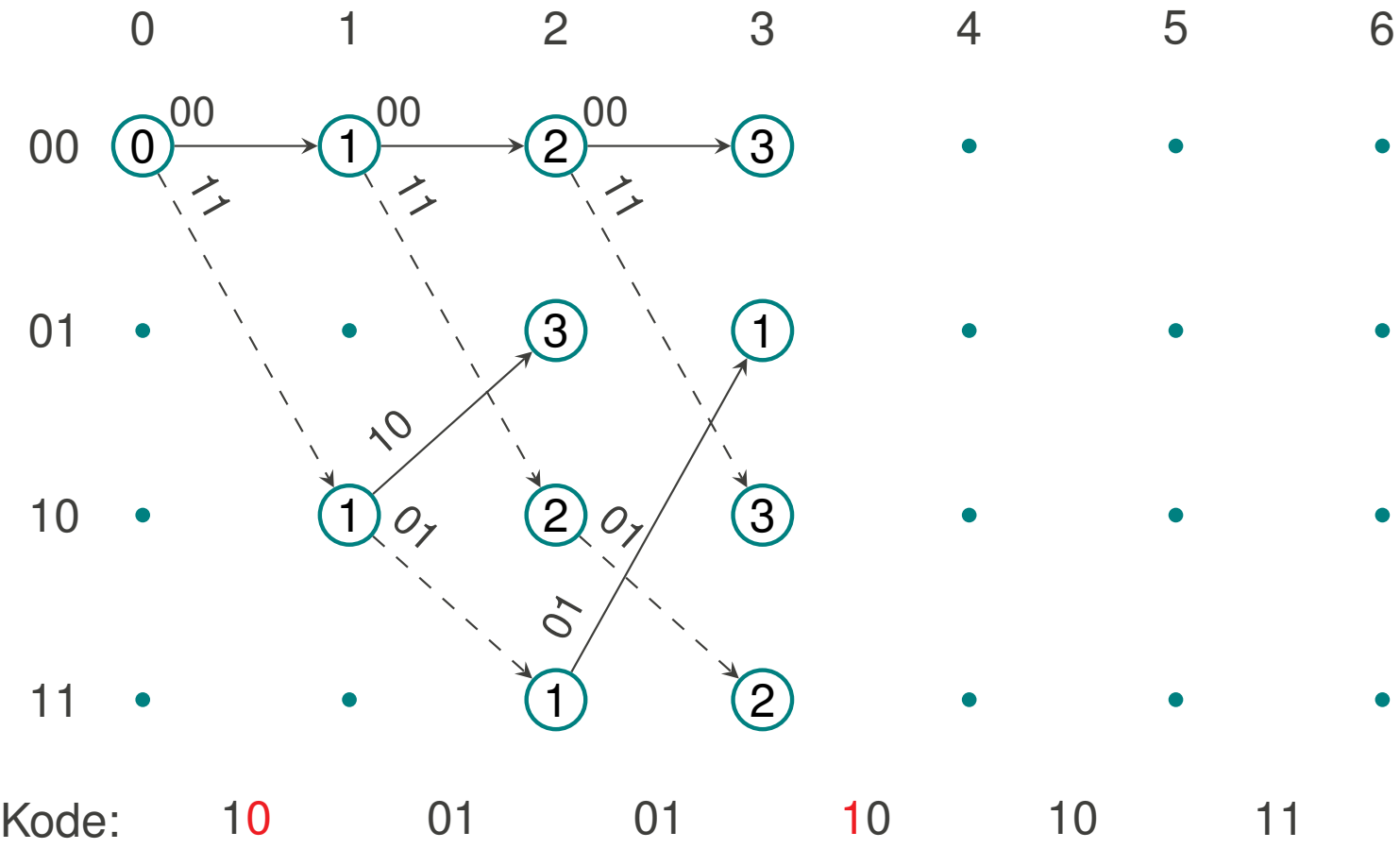
Nachricht:

Faltungsdekodierung (Viterbi-Algorithmus)



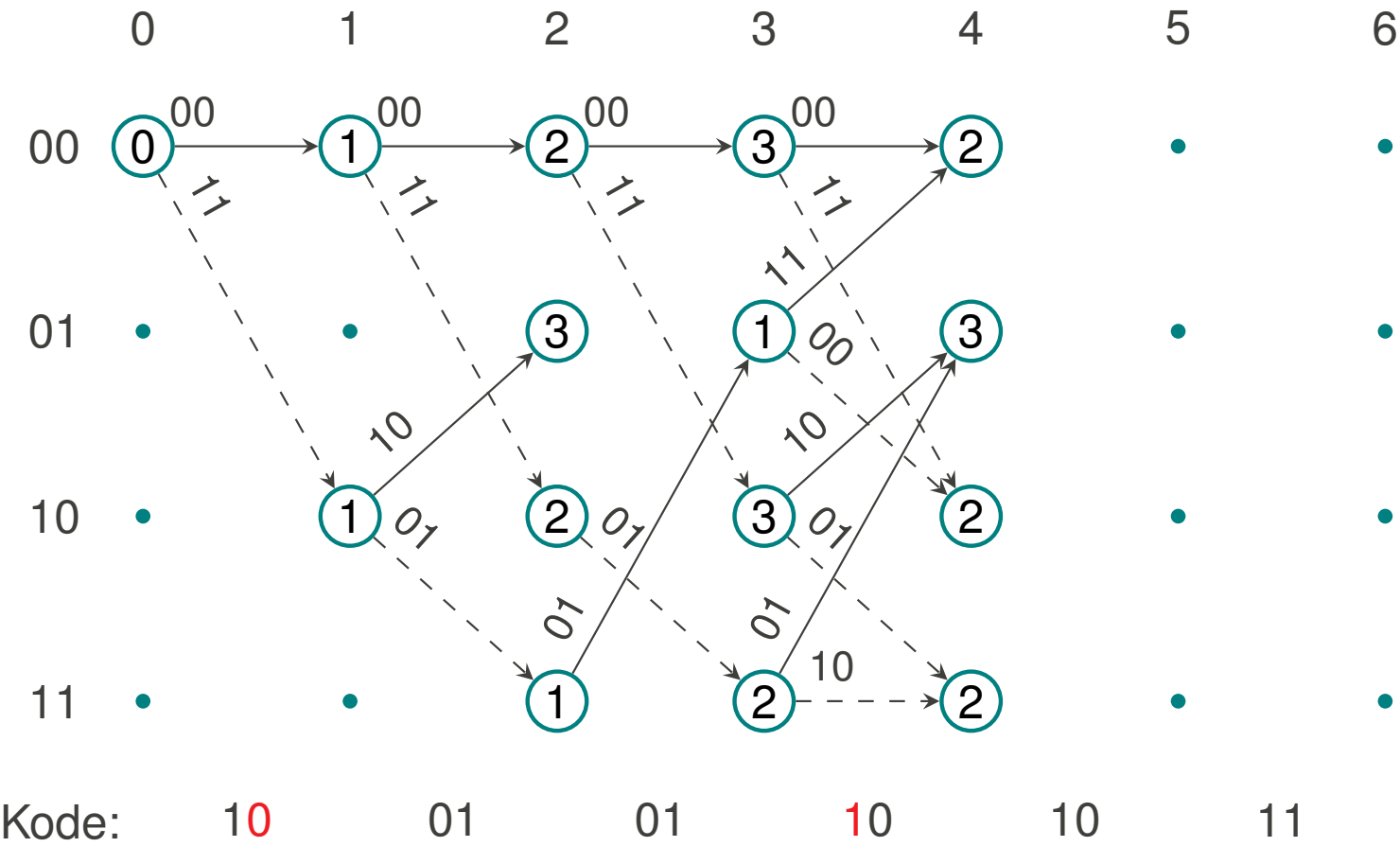
Nachricht:

Faltungsdekodierung (Viterbi-Algorithmus)

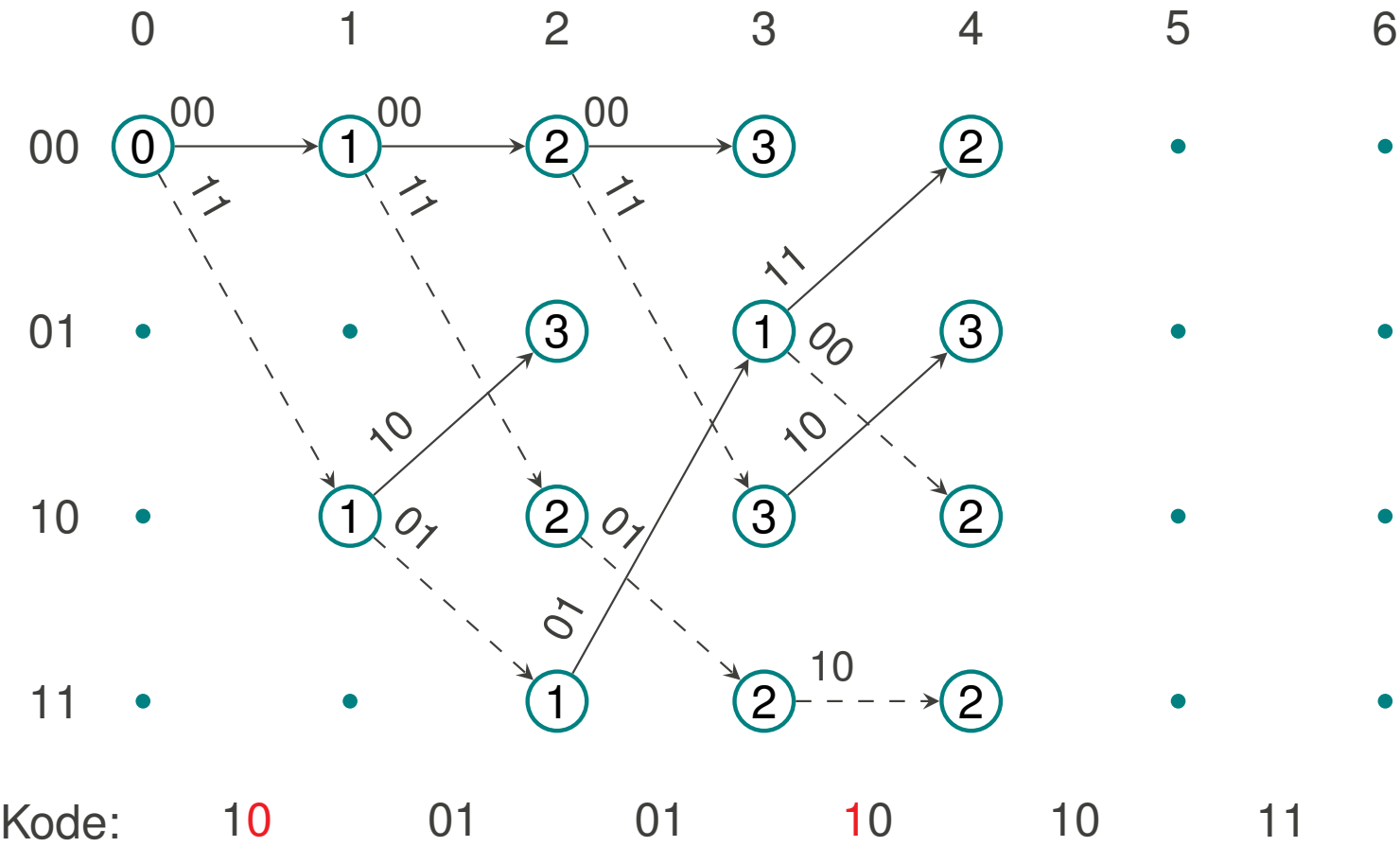


Nachricht:

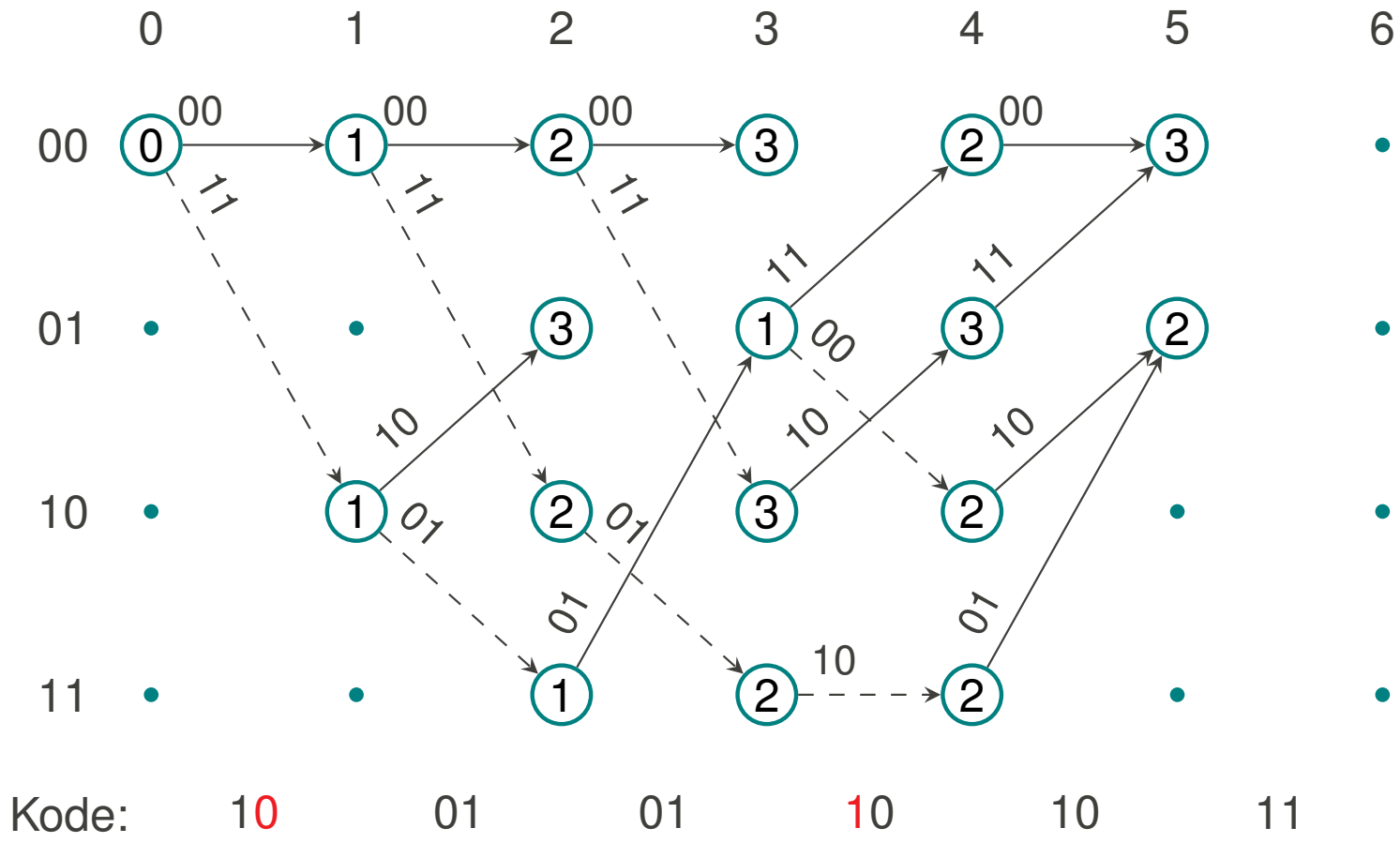
Faltungsdekodierung (Viterbi-Algorithmus)



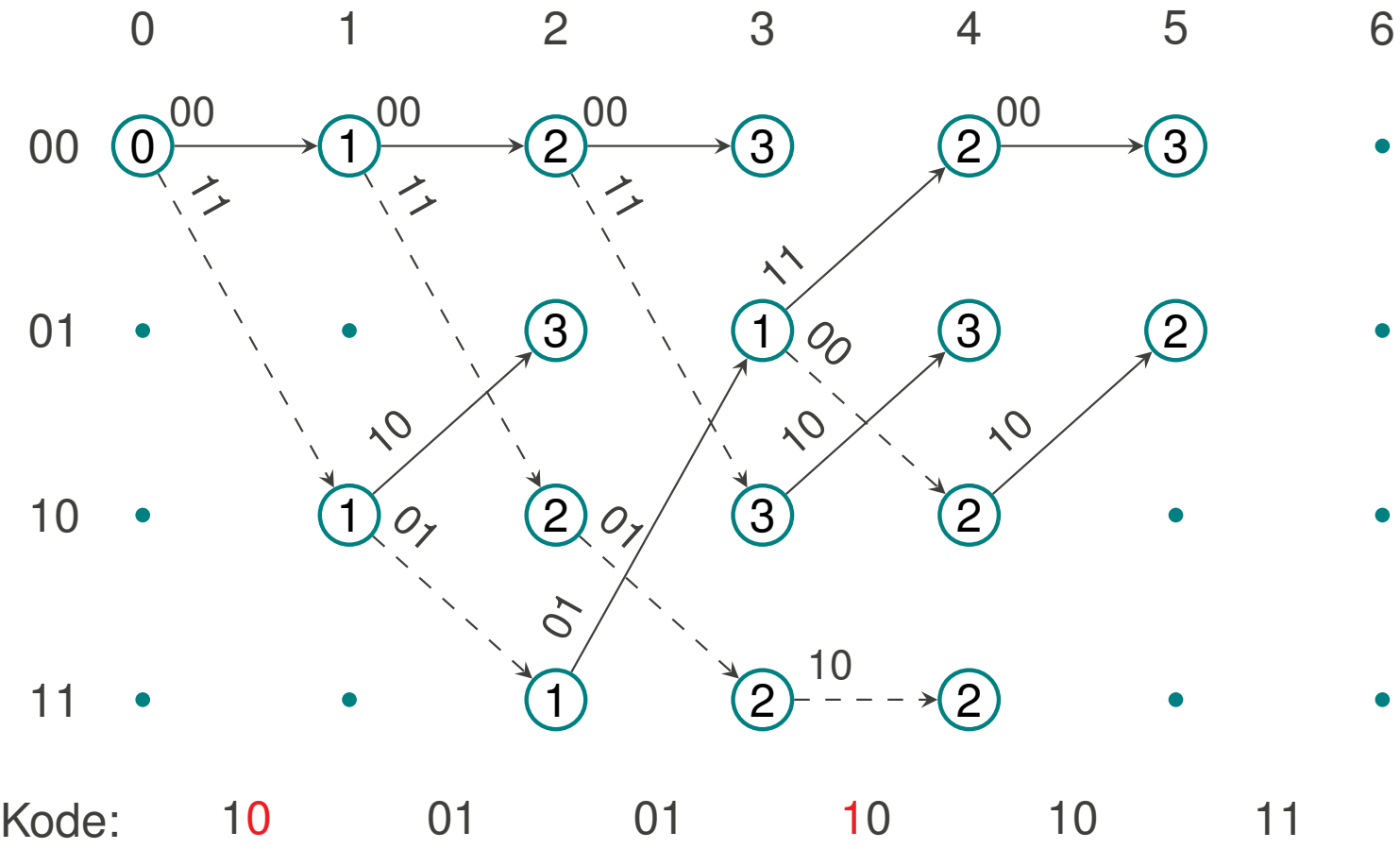
Faltungsdekodierung (Viterbi-Algorithmus)



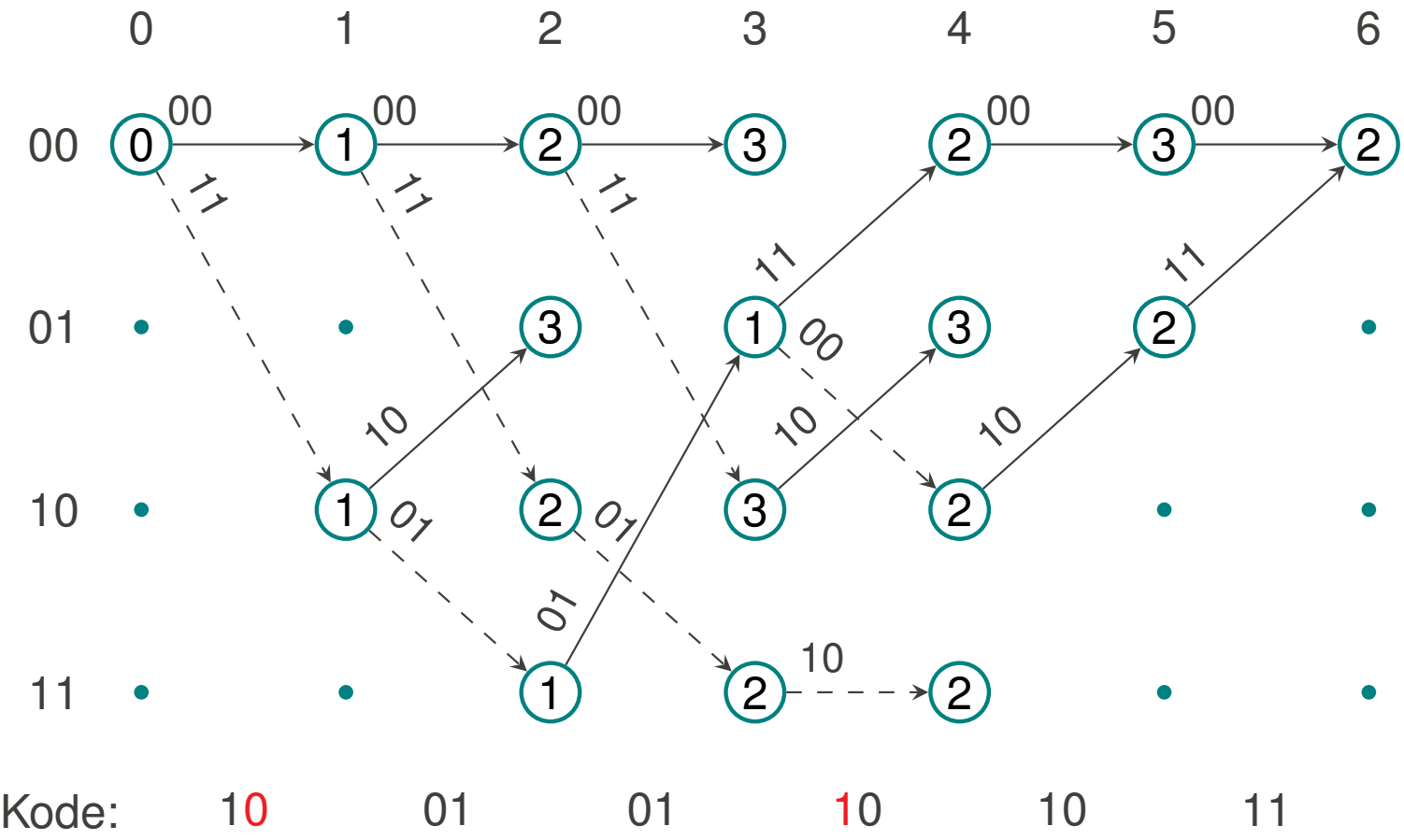
Faltungsdekodierung (Viterbi-Algorithmus)



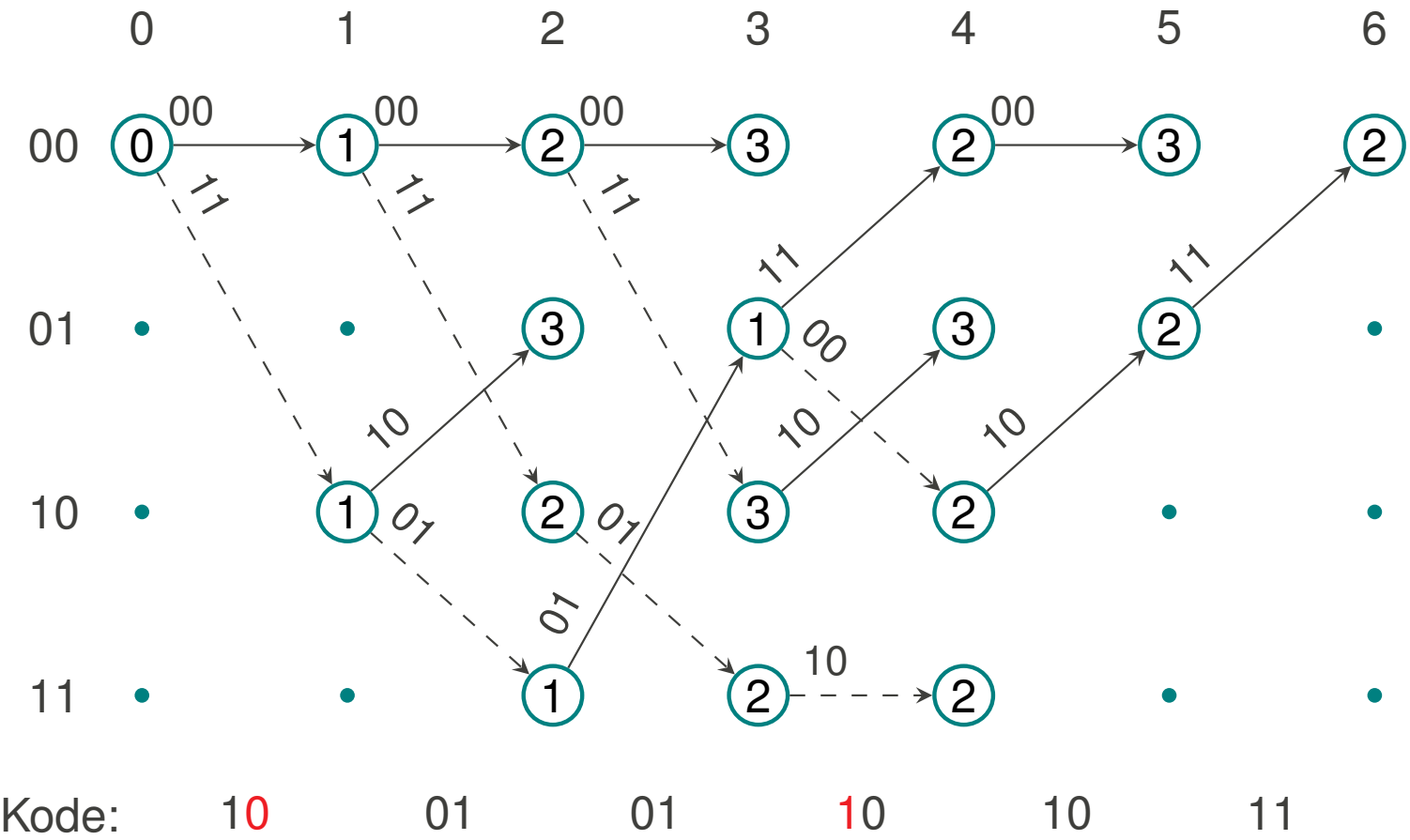
Faltungsdekodierung (Viterbi-Algorithmus)



Faltungsdekodierung (Viterbi-Algorithmus)

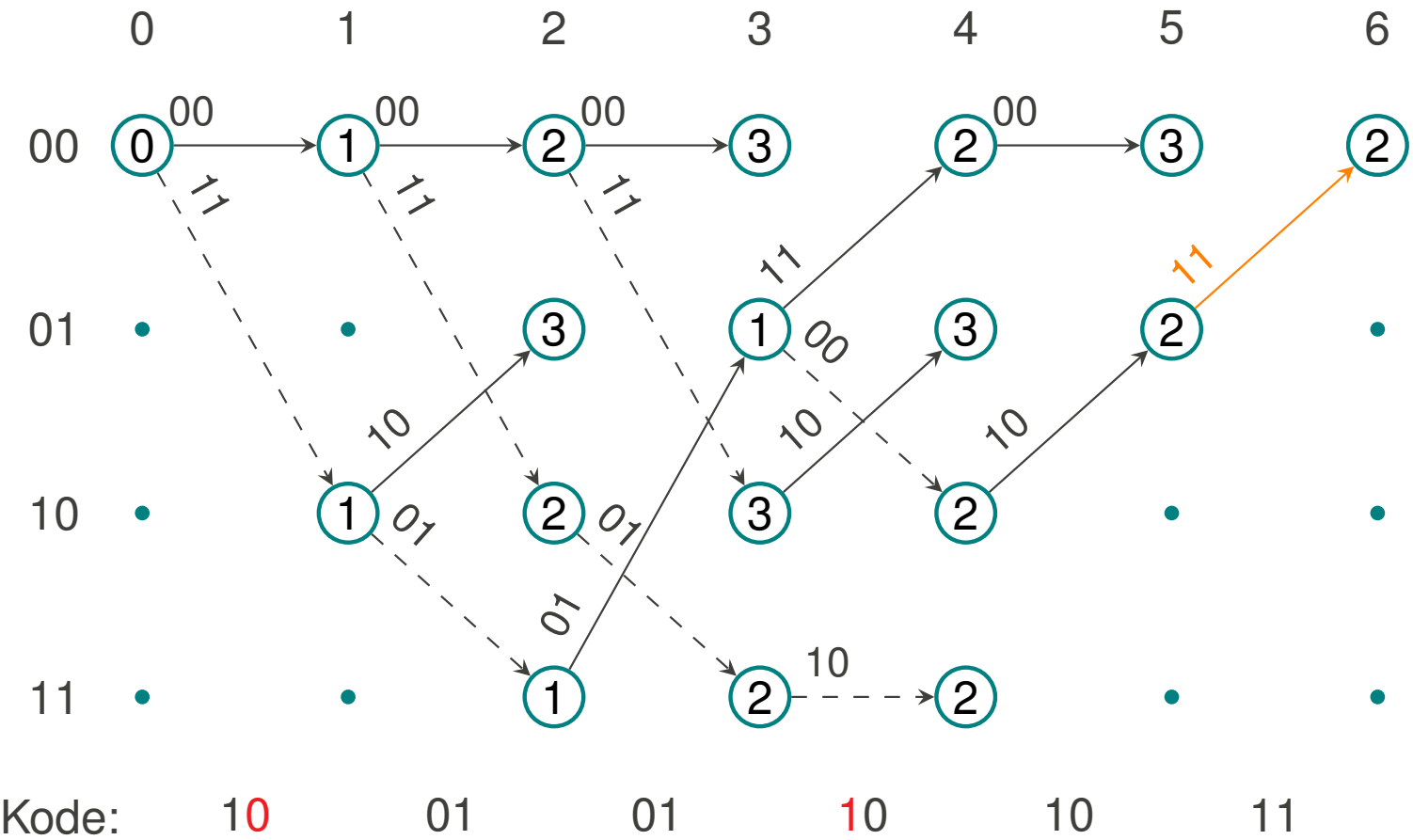


Faltungsdekodierung (Viterbi-Algorithmus)



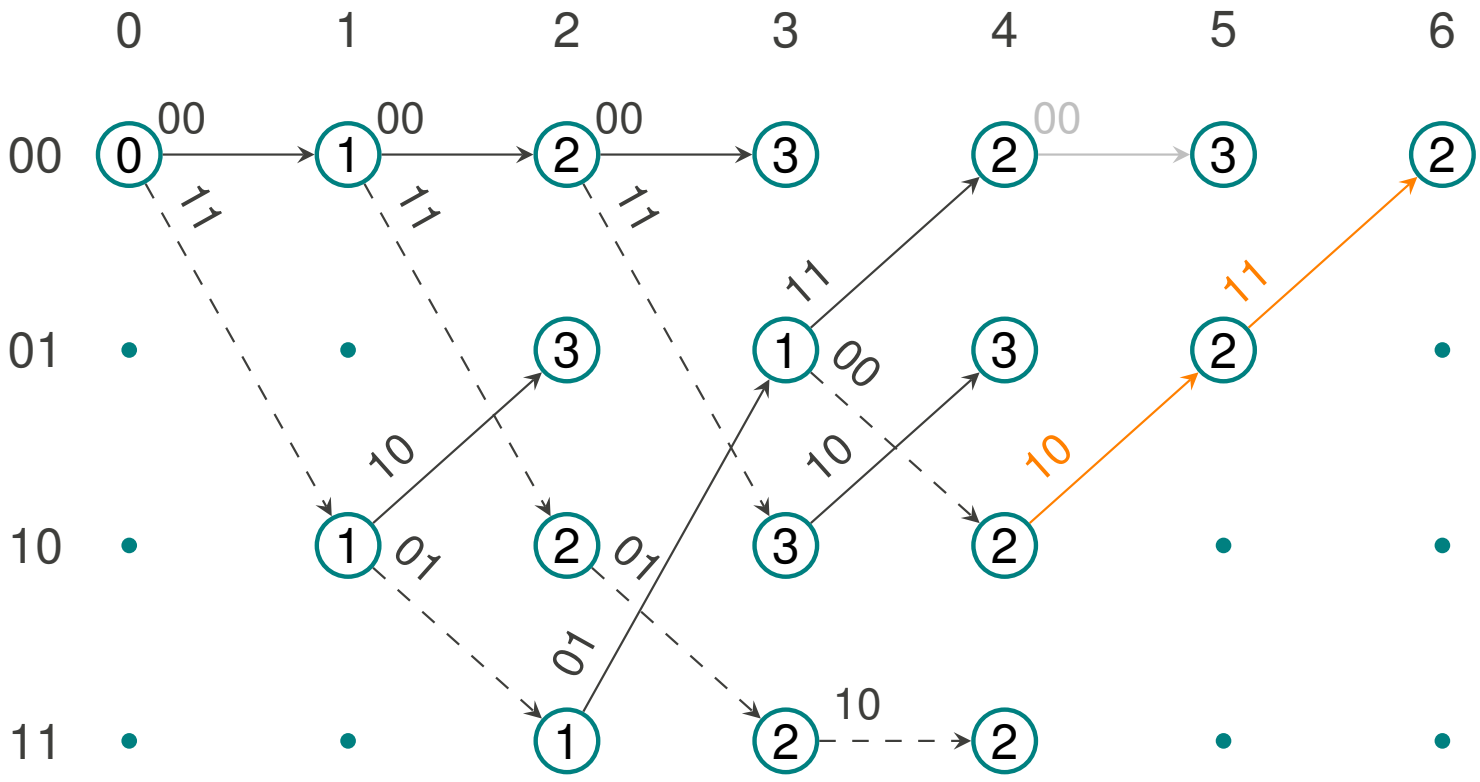
Nachricht:

Faltungsdekodierung (Viterbi-Algorithmus)

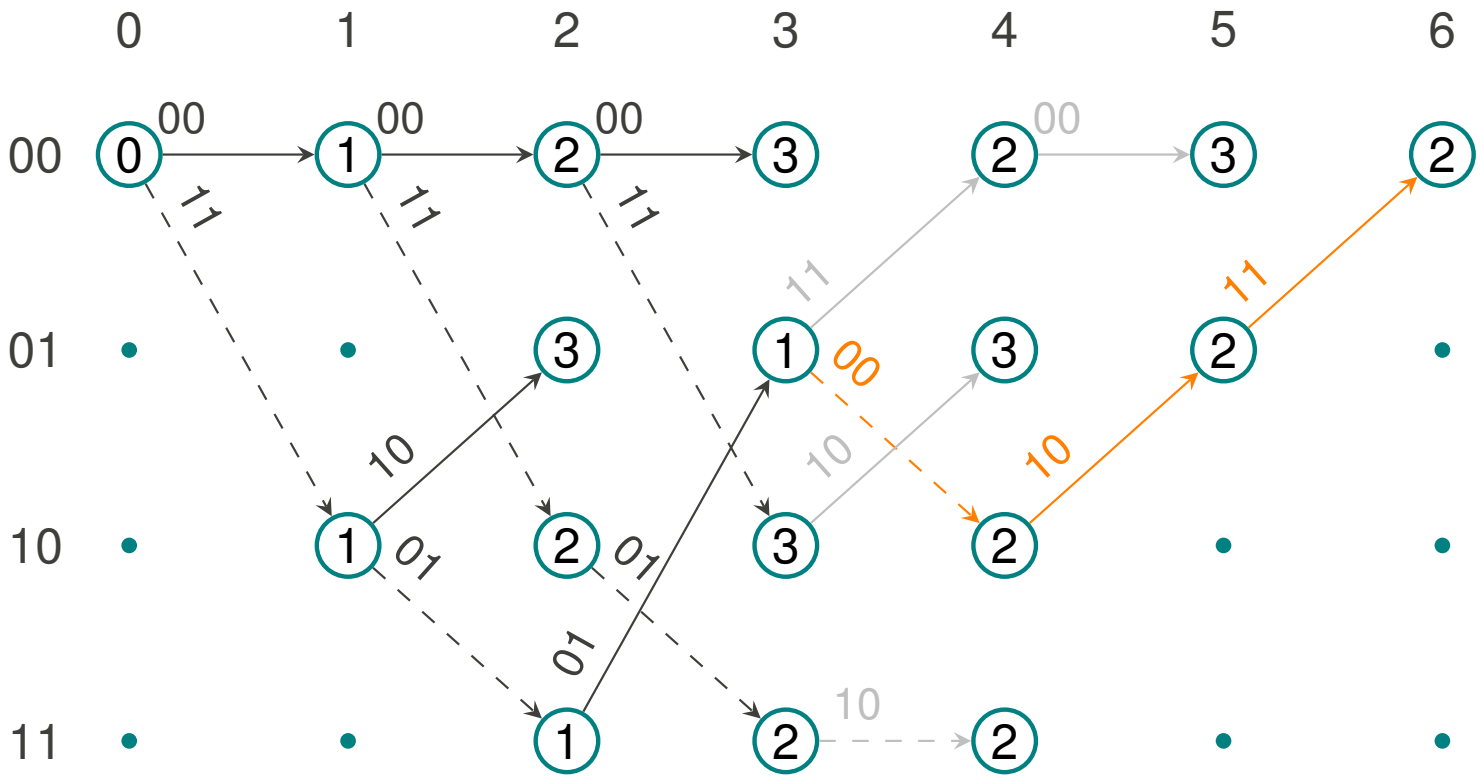


Nachricht: **0**

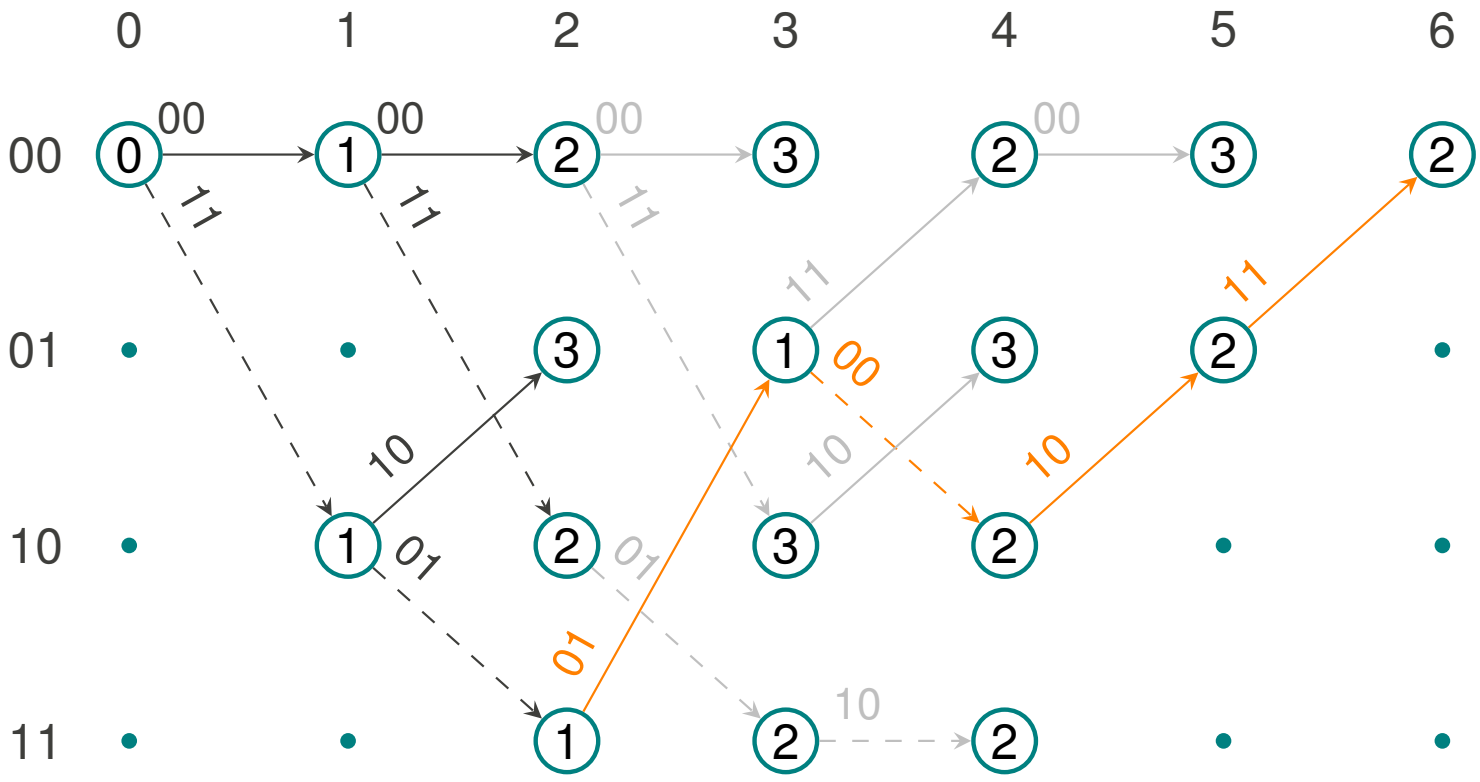
Faltungsdekodierung (Viterbi-Algorithmus)



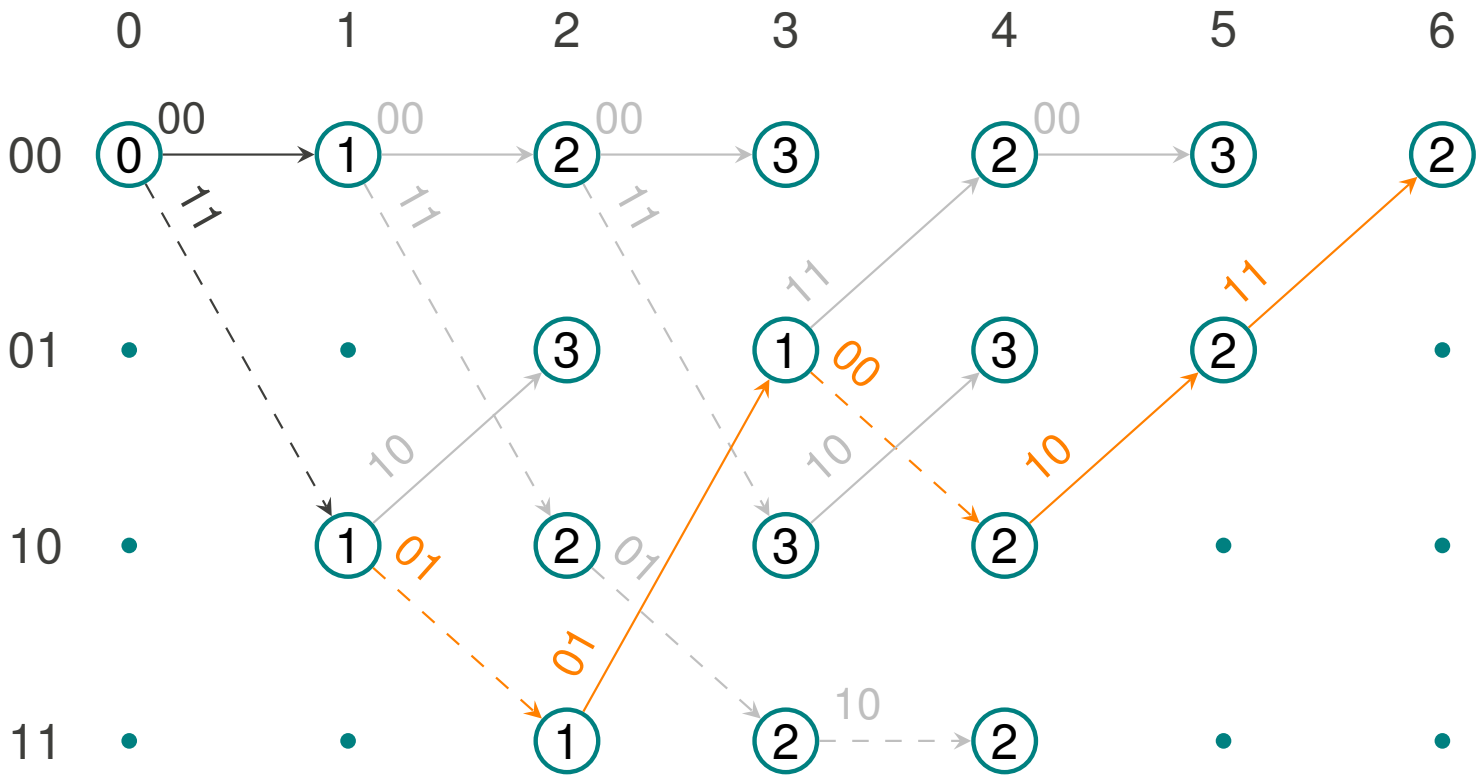
Faltungsdekodierung (Viterbi-Algorithmus)



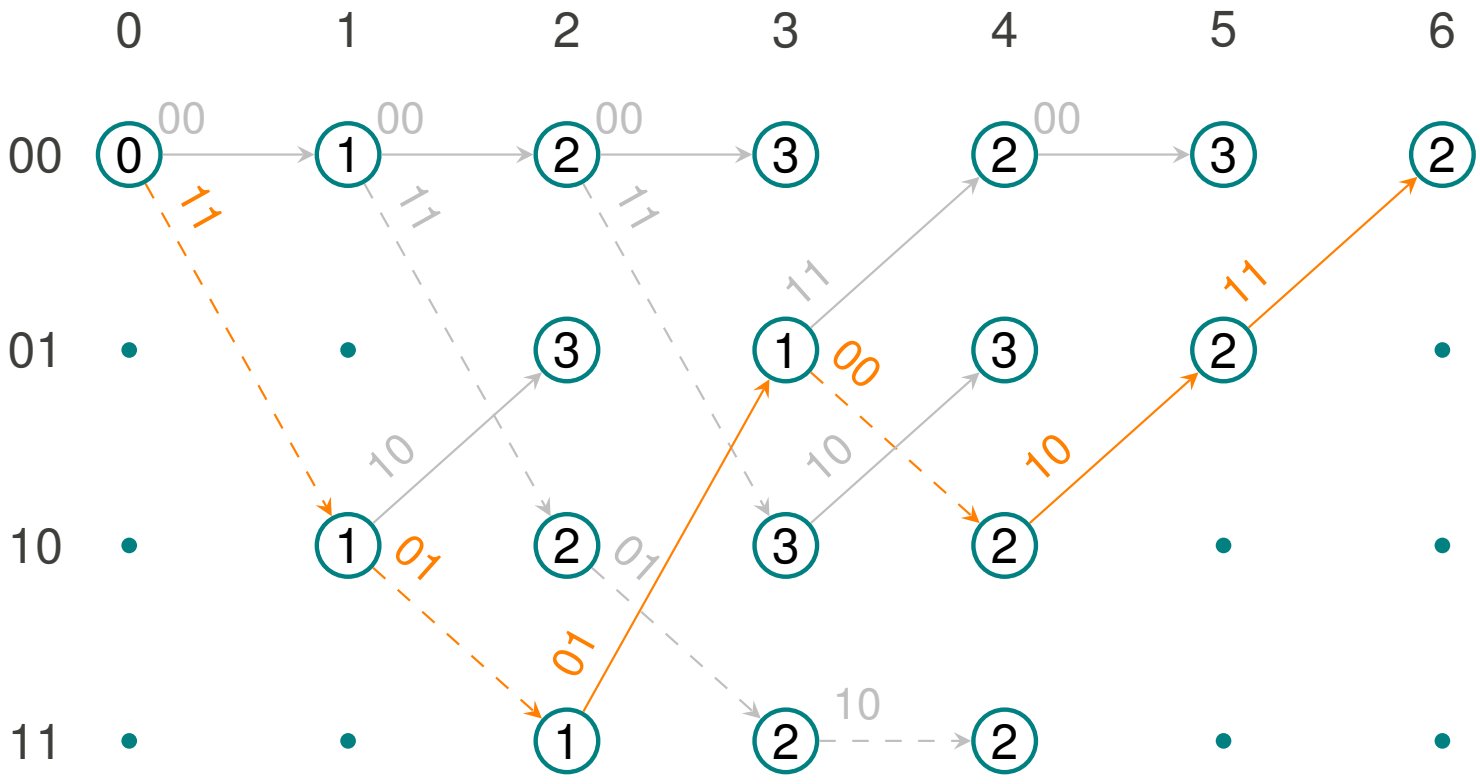
Faltungsdekodierung (Viterbi-Algorithmus)



Faltungsdekodierung (Viterbi-Algorithmus)



Faltungsdekodierung (Viterbi-Algorithmus)



Viterbi-Algorithmus

- Varianten
  - hard decision Dekodierung
  - soft decision Dekodierung
- Dekodierung aufwändig → Flaschenhals

Viterbi-Algorithmus

- Varianten
  - hard decision Dekodierung
  - soft decision Dekodierung
- Dekodierung aufwändig → Flaschenhals

R vs. C/C++

- R-Code wird interpretiert → langsam
- Flaschenhälse: Schleifen, rekursive Funktionen
- Performance mittels C/C++-Code verbessern (kompiliert)
- Rcpp-Paket

Rcpp

- R-Datentypen
- einfacher Funktionsaufruf

C++-Code

```
1 #include <Rcpp.h>
2 using namespace Rcpp;
3
4 // [[Rcpp::export]]
5 IntegerVector myFunction(int a, int b) {
6
7    /* my cool C++ algorithm */
8 }
```

R-Code

```
1 x <- myFunction(1, 2)
```

Quellen

- W. C. Huffman und V. Pless. *Fundamentals of Error-Correcting Codes*. 2010
- R. H. Morelos-Zaragoza. *The Art of Error Correcting Coding*. 2006
- D. Schönfeld, H. Klimant und R. Piotraschke. *Informations- und Kodierungstheorie*. 2012
- H. Wickham. *Advanced R*. 2015
- H. Wickham. *R Packages*. 2015