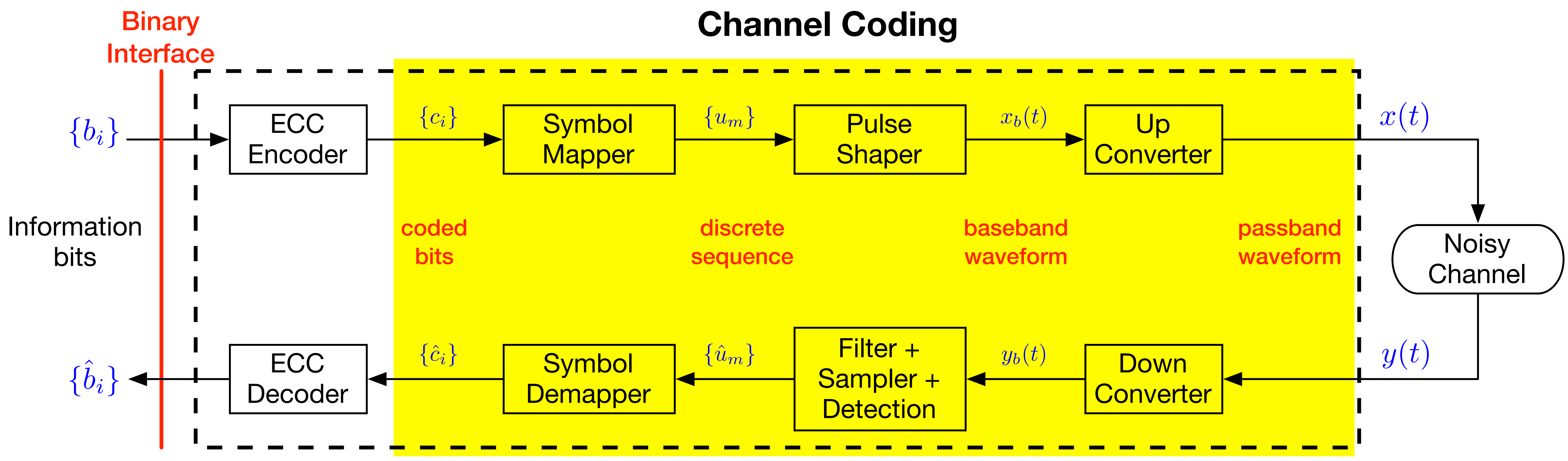
Communication Systems Lab, Spring 2018

Lecture 03
Reliable Communication

I-Hsiang Wang
ihwang@ntu.edu.tw
National Taiwan University

2018/04/18



Previous lectures:

Focusing on digital modulation, we can ensure that the coded bits $\{c_i\}$ can be reconstructed optimally (i.e., minimize avg. prob. of error) at the receiver

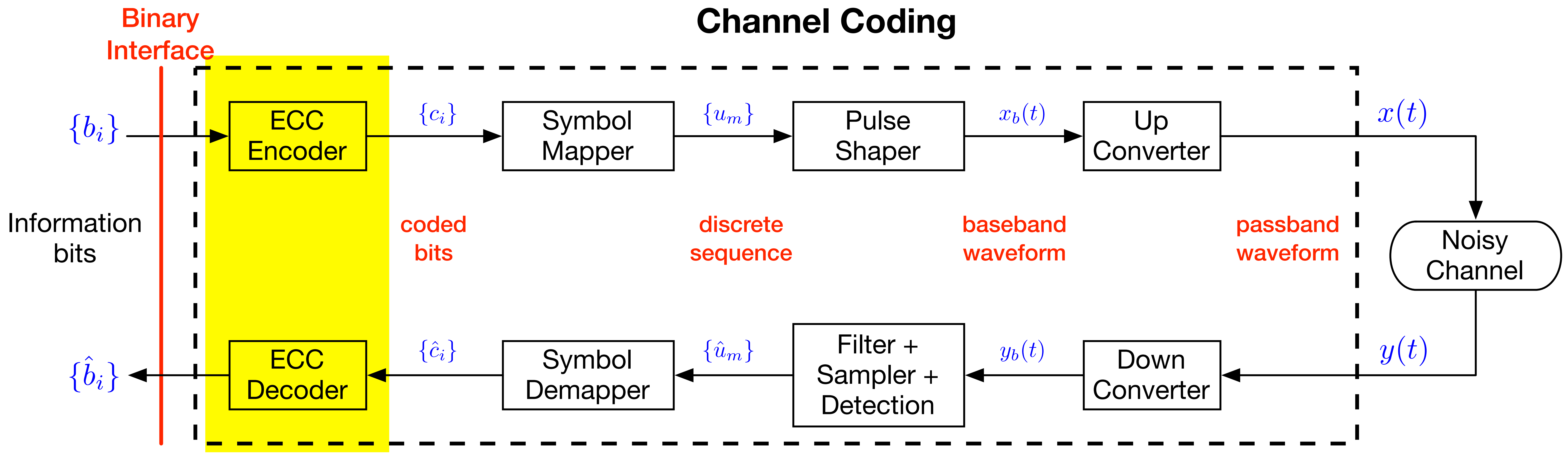
- Averaged symbol probability of error is exponentially decaying with SNR

$$\mathsf{P_e} \doteq \exp(-c\,\mathsf{SNR})$$

- For each symbol, $\mathsf{P_e} = 10^{-3}$ is already pretty good!

However, this is not good enough …

- Consider a file mapped and converted into $n = 250$ symbols
- The file cannot be reconstructed if one symbol is wrong
- The “file” probability of error is $1 - (1 - \mathsf{P_e})^n \approx n\mathsf{P_e} = 250/1000 = 0.25$
- Pretty bad … But we cannot do much because noise is inevitable, while modulation only focus on the symbol level, not the the file level



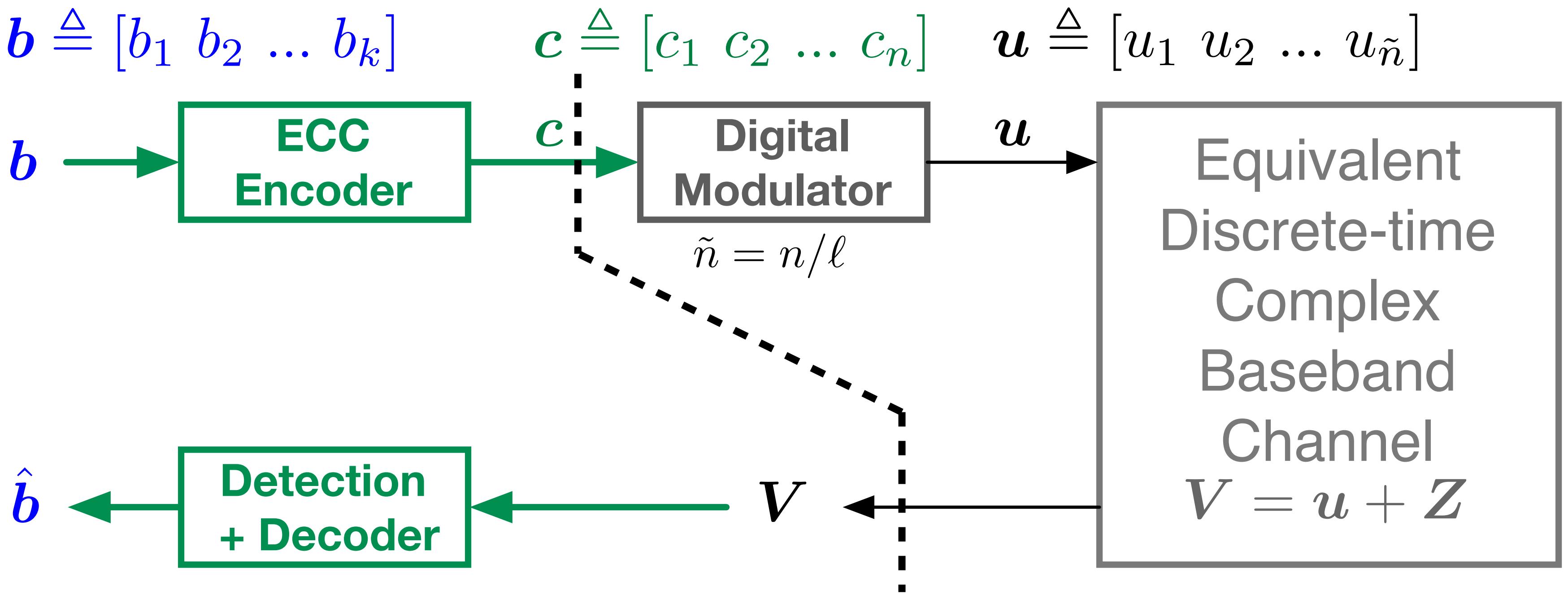
This lecture: **Reliable Communication!**

Introduce error correction coding, to add redundancy to the original file.

 We are able to make the overall “file” probability of error arbitrarily small!

Prices to pay: data rate and energy

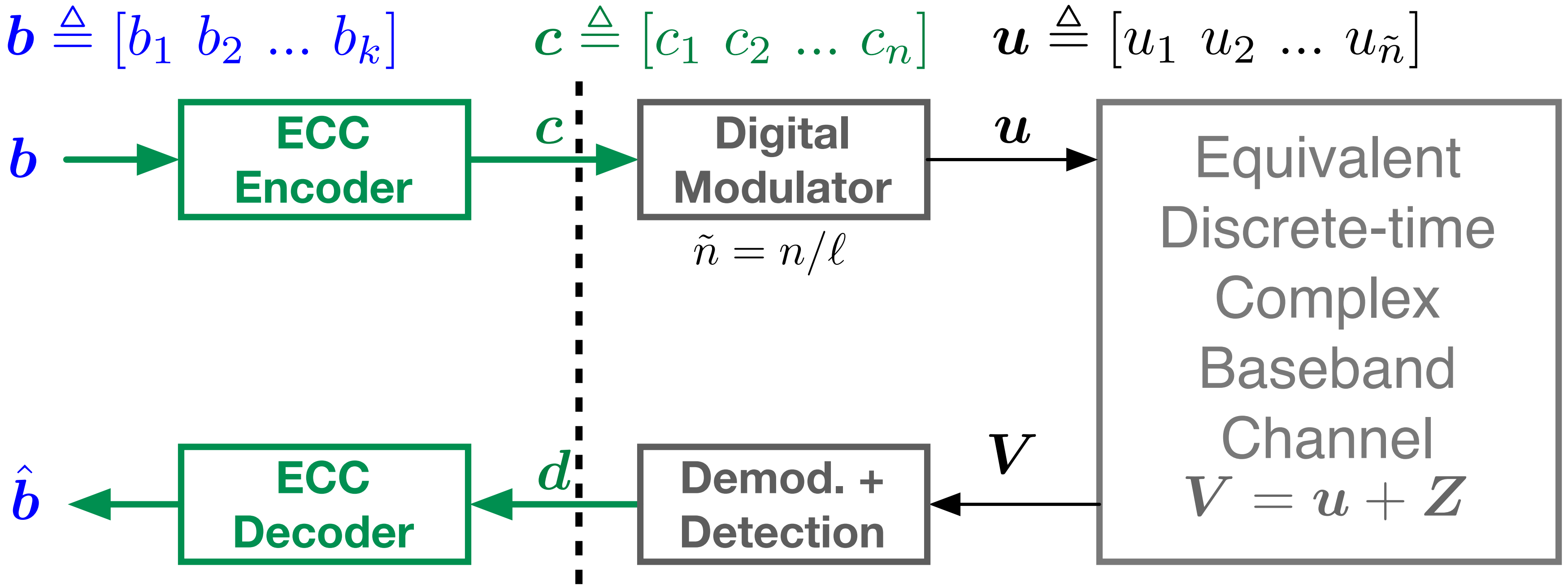
Soft decision: jointly consider detection and decoding; directly work on the demodulated symbols



Rate: $R = k/n$

We focus on soft decision first!

Hard decision: only consider decoding; directly work on the detected bit sequences



Outline

- Prelude: repetition coding
- Energy-efficient reliable communication: orthogonal code
- Rate-efficient reliable communication: linear block code
- Convolutional code

Part I. Prelude: Repetition Coding

Repetition code, Rate and Energy efficiency

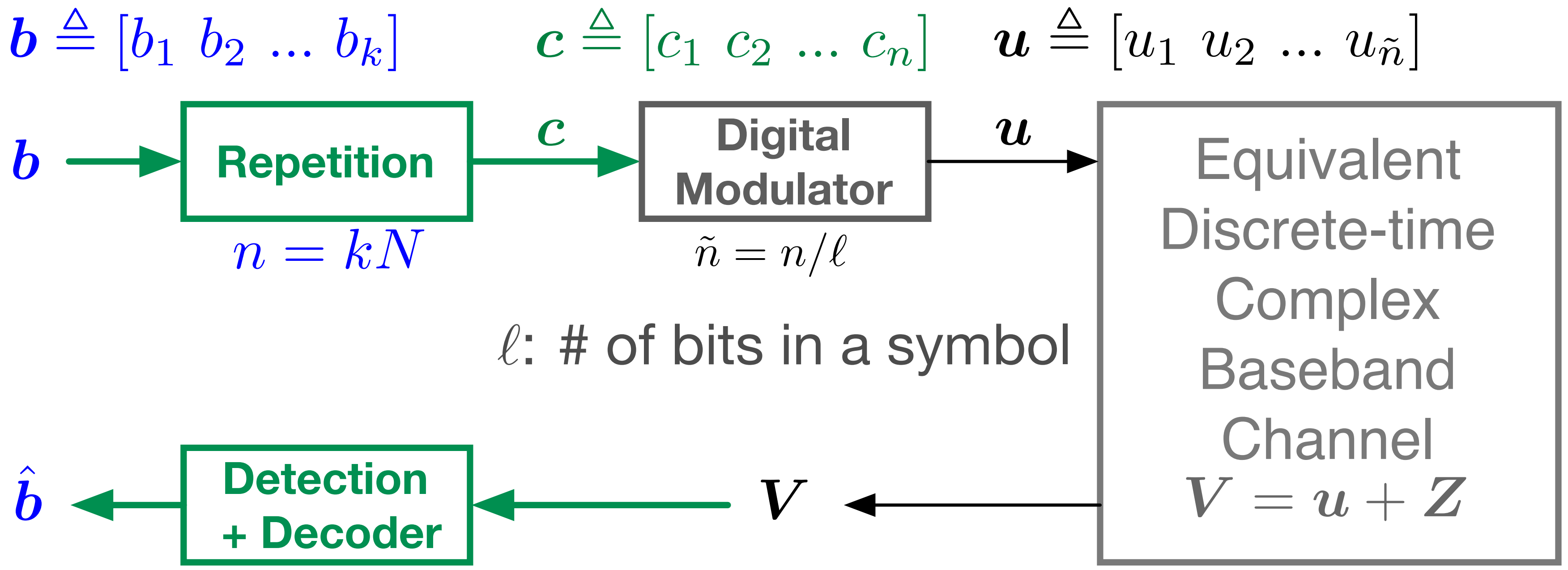
Repetition: a simple way to enhance reliability

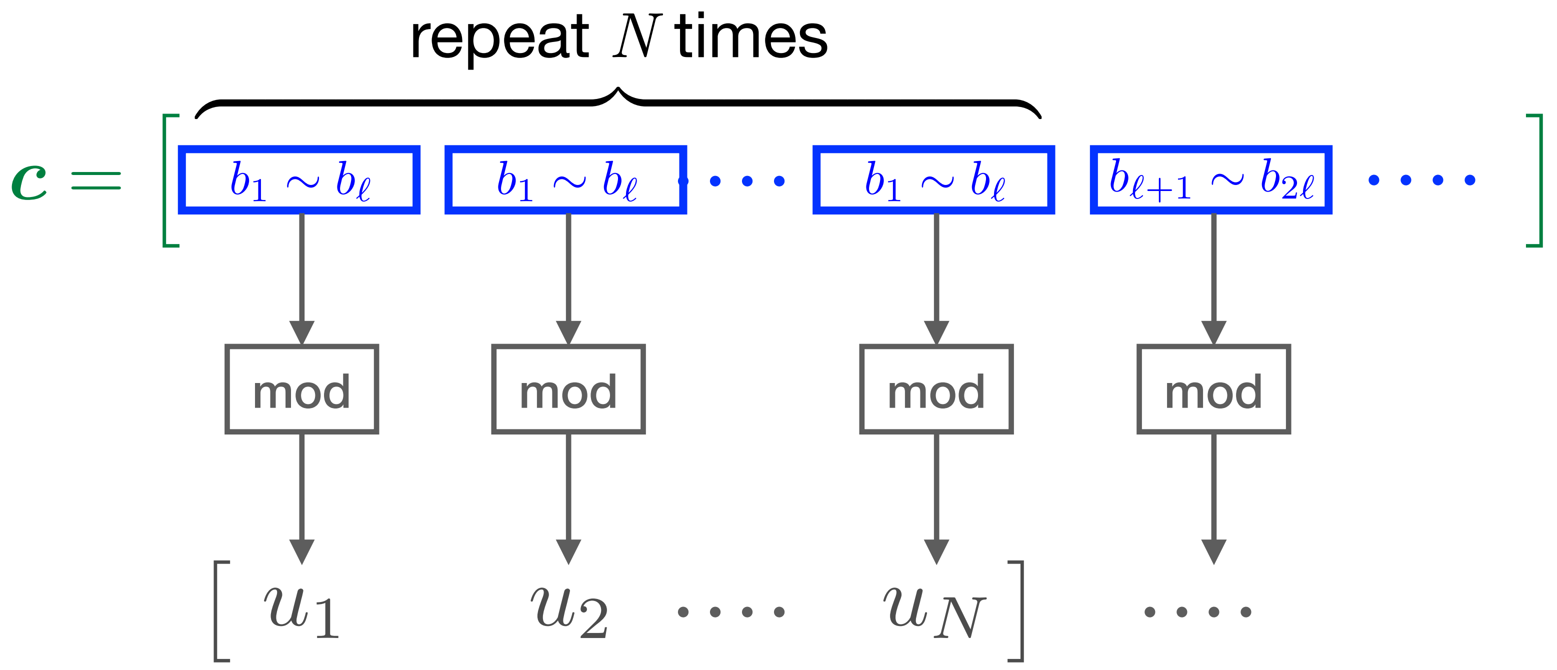
- Idea: repeat each bit N times □ data rate $R = 1/N$.

original bit seq.	b_1		b_2		b_3		b_4		b_5	
coded bit seq. 1	b_1	b_1	b_2	b_2	b_3	b_3	b_4	b_4	b_5	b_5
coded bit seq. 2	b_1	b_2	b_3	b_4	b_5	b_1	b_2	b_3	b_4	b_5

Many ways for repetition

- We focus on the architecture below: $\boldsymbol{c} = \left[\overbrace{\boxed{b_1 \sim b_\ell}\ \boxed{b_1 \sim b_\ell} \cdots \boxed{b_1 \sim b_\ell}}^{\text{repeat } N \text{ times}}\ \boxed{b_{\ell+1} \sim b_{2\ell}} \cdots\right]$





Equivalent vector symbol $\boldsymbol{u} \triangleq [u_1 \ u_2 \ \cdots \ u_N] \in \mathbb{C}^N$

- Since the noises are i.i.d., it suffices to use the N-dim. demodulated

$$\boldsymbol{V} = \boldsymbol{u} + \boldsymbol{Z}$$

to optimally decode $b_1 \sim b_\ell$

BPSK + repetition coding

- Equivalent channel model: $\boldsymbol{V} = \boldsymbol{u} + \boldsymbol{Z} \in \mathbb{C}^N \quad Z_1, \ldots Z_N \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0)$

- Equivalent constellation set: $\boldsymbol{u} \in \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$

$$\boldsymbol{a}_0 = -\begin{bmatrix} d & d & \cdots & d \end{bmatrix} \qquad \boldsymbol{a}_1 = +\begin{bmatrix} d & d & \cdots & d \end{bmatrix}$$

- Performance analysis:

$$\mathsf{P}_{\mathsf{e}}^{(N)} = \mathrm{Q}\left(\frac{\|\boldsymbol{a}_1 - \boldsymbol{a}_0\|}{2\sqrt{N_0/2}}\right) = \mathrm{Q}\left(\sqrt{\frac{N \cdot 4d^2}{2N_0}}\right) = \mathrm{Q}\left(\sqrt{N\frac{2d^2}{N_0}}\right)$$

Repetition effectively increase SNR by N-fold!

$$= \mathrm{Q}\left(\sqrt{N 2\mathsf{SNR}}\right) \doteq \exp(-N\mathsf{SNR})$$

$$\mathsf{SNR} \triangleq \frac{\text{average energy per } \textbf{uncoded} \text{ symbol}}{\text{total noise variance per symbol}} = \frac{d^2}{N_0}$$

Rate and energy efficiency

- Rate: $R = 1/N \to 0$ as $N \to \infty$

- Energy per bit: $E_{\mathrm{b}} = Nd^2 \to \infty$ as $N \to \infty$

- Achieving arbitrarily small prob. of error at the price of **zero rate** and **infinite energy per bit**

- Question: can we resolve the issue with more general constellation sets?

General modulation + repetition coding

- Equivalent channel model: $\boldsymbol{V} = \boldsymbol{u} + \boldsymbol{Z} \in \mathbb{C}^N \quad Z_1, \ldots Z_N \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0)$

- Equivalent constellation set: $\boldsymbol{u} \in \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M\} \quad M = 2^\ell$

- Rate: $R = \ell/N$

  $\to 0$ as $N \to \infty$

- Energy per bit: $\frac{E_\text{b}}{N_0} = \frac{N}{\ell}\text{SNR} = \frac{\text{SNR}}{R}$

  $\to \infty$ as $N \to \infty$

- Probability of error (take M-ary PAM as an example):

$$\mathsf{P}_\mathsf{e}^{(N)} = 2(1 - 2^{-\ell})\text{Q}\left(\sqrt{N\frac{6}{4^\ell - 1}\text{SNR}}\right) = 2(1 - 2^{-NR})\text{Q}\left(\sqrt{\frac{N}{4^{NR} - 1}6\text{SNR}}\right)$$

$$\lim_{N\to\infty} \mathsf{P}_\mathsf{e}^{(N)} = 0 \iff \lim_{N\to\infty} \frac{4^{NR} - 1}{N} = 0$$

it is necessary that $\lim_{N\to\infty} R = 0$

Why repetition coding is not very good

- Repetition coding: high reliability at the price of asymptotically zero rate and infinite energy per bit

- Repetition is too naive and does not utilize the available degrees of freedom in the N-dimensional space efficiently

- Is it possible to design better coding schemes with the following?
  - Vanishing probability of error
  - Positive rate
  - Finite energy per bit

YES!

Part II. Convolutional Code

Encoding Architecture, Trellis Representation, Maximum Likelihood Sequence Detection, Viterbi Algorithm

Convolutional Code

- Introduced by Peter Elias in 1955

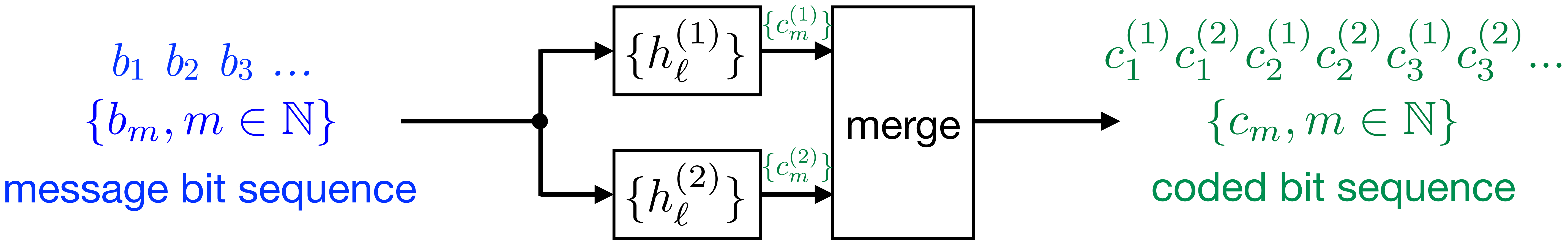
- Efficient ML decoding algorithm by Andrew Viterbi in 1967

- Used in NASA space exploration projects, from Voyager (1977) onwards
- Widely applied in digital video, radio, satellite communications, etc.

Encoding architecture

- Message bits passing through causal LTI filters to generate coded bits

- Multiple filters to introduce redundancy: (example: 2 filters)



Encoding $\equiv$ FIR filtering

- Each filter is causal and has finite impulse response (FIR)

$$\begin{bmatrix} \dots & 0 & 0 & h_0^{(j)} & h_1^{(j)} & \dots & h_{L-1}^{(j)} & 0 & 0 & \dots \end{bmatrix}$$

coefficients of filter taps are ±1

- The output bit sequence of one branch is the input convolve with the IR

j-th branch: $$c_m^{(j)} = (h * b)_m \triangleq \sum_{\ell=-\infty}^{\infty} h_\ell^{(j)} b_{m-\ell} = \sum_{\ell=0}^{L-1} h_\ell^{(j)} b_{m-\ell}$$

binary field arithmetic

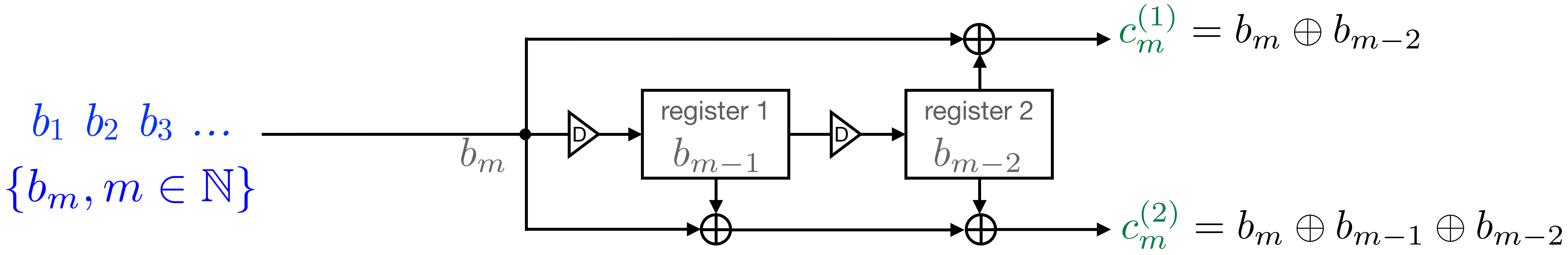
- More generally, there can be K input sequences and N output sequences, and the code rate is $R = K/N$

j-th branch: $$c_m^{(j)} \triangleq \sum_{i=1}^{K} \sum_{\ell=0}^{L-1} h_\ell^{(i,j)} b_{m-\ell}^{(i)}, \quad j = 1, \dots, N$$

Implementation with shift registers

- The encoder (FIR filtering) can be implemented with L–1 shift registers

$$L = 3 \quad K = 1, N = 2 \qquad \boldsymbol{h}^{(1)} = [1 \quad 0 \quad 1] \quad \boldsymbol{h}^{(2)} = [1 \quad 1 \quad 1]$$

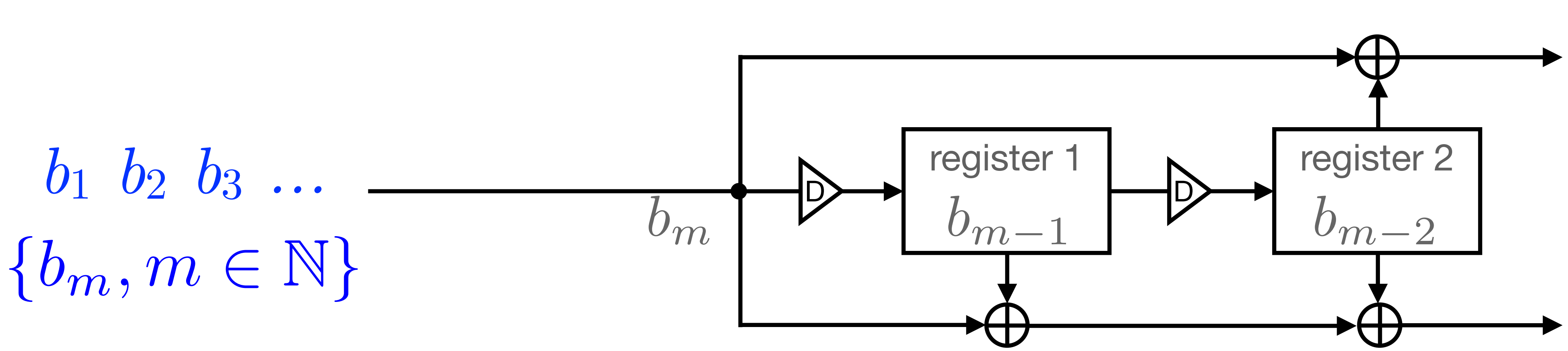


- Call the content of the two registers as the “state” of the encoder

input	State (before)	State (after)	Output $c_m^{(1)}$	Output $c_m^{(2)}$
1	00	10	1	1
0	10	01	0	1
0	01	00	1	1
1	00	10	1	1

finite state machine!

State transition diagram



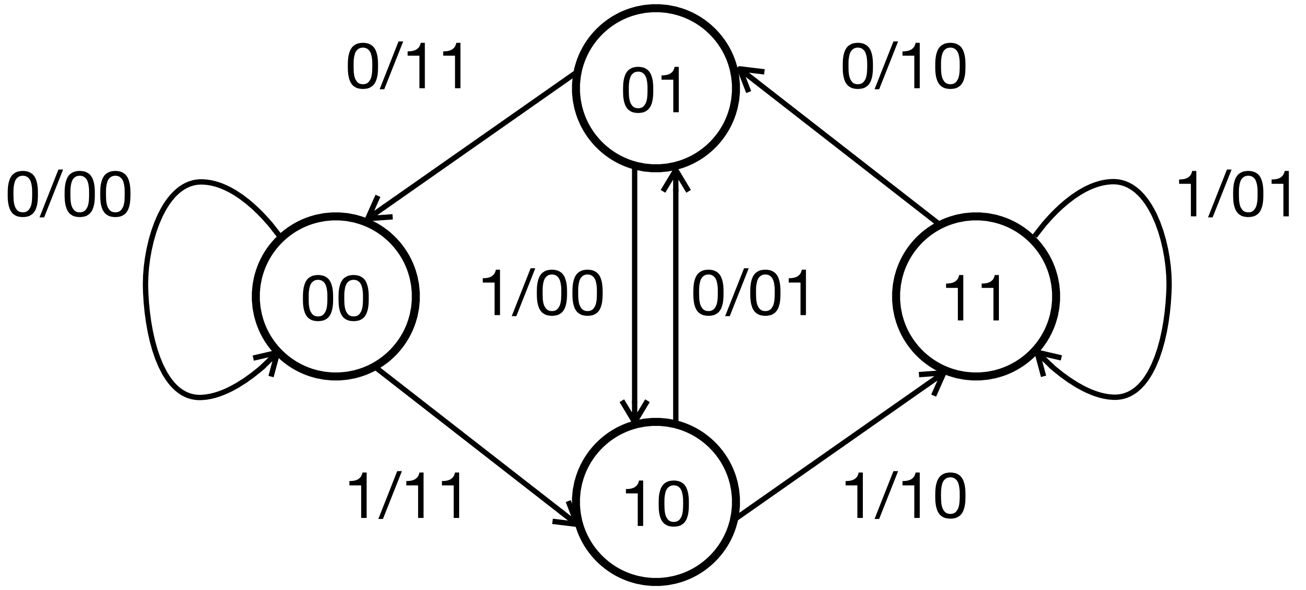
$c_m^{(1)} = b_m \oplus b_{m-2}$

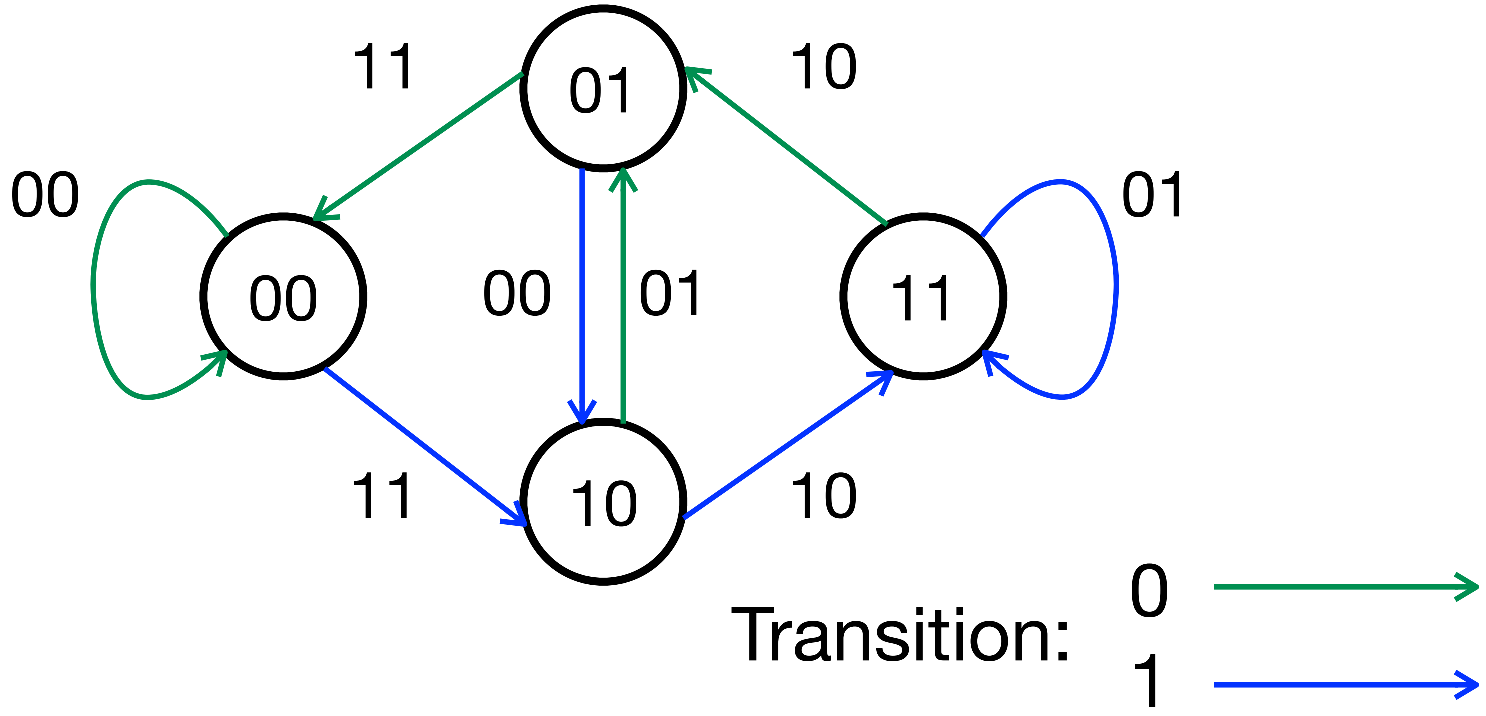
$\boldsymbol{h}^{(1)} = [1 \quad 0 \quad 1]$

$c_m^{(2)} = b_m \oplus b_{m-1} \oplus b_{m-2}$

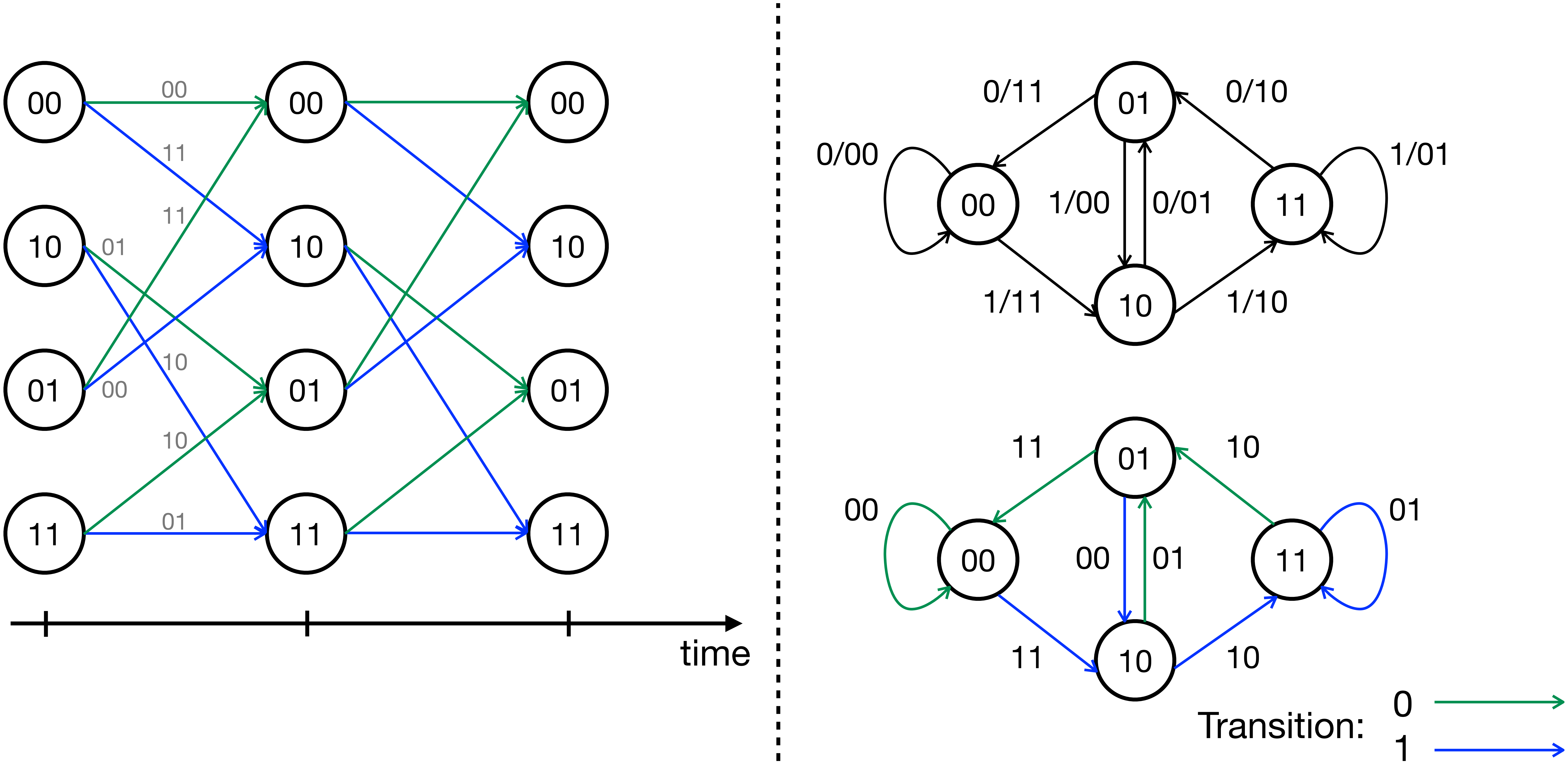
$\boldsymbol{h}^{(2)} = [1 \quad 1 \quad 1]$

- For the finite state machine, its state transition diagram can be drawn



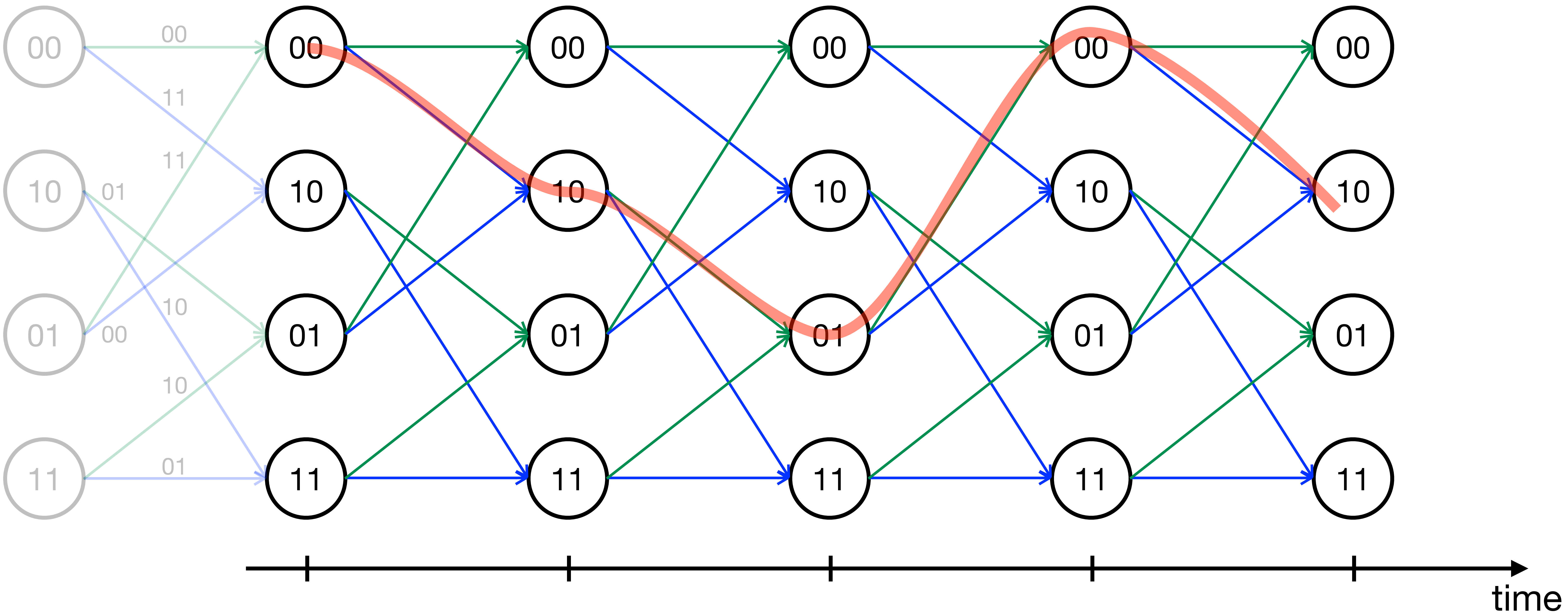


Trellis representation of codewords

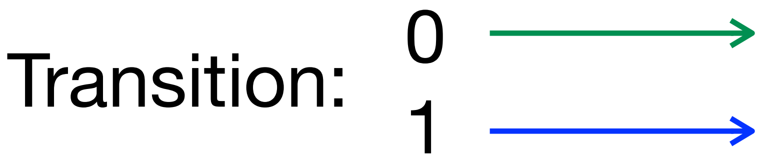


message: 1 0 0 1

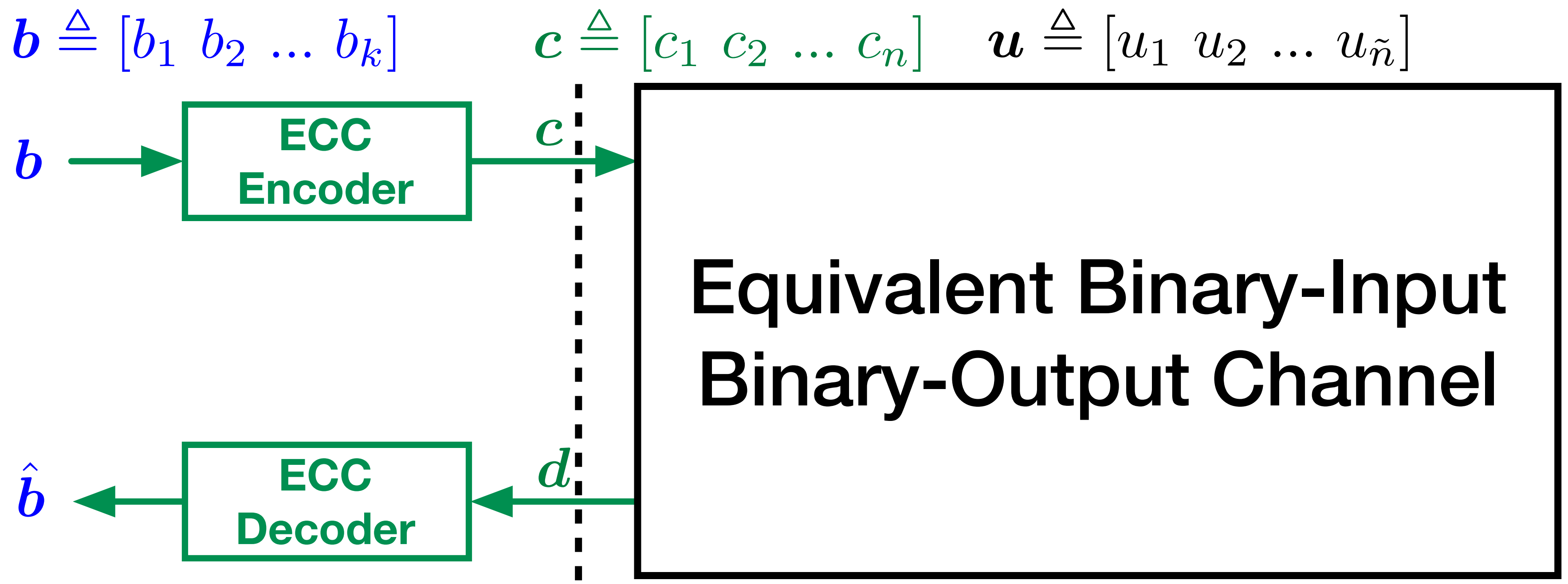
codeword: 11 01 11 11



Each path represents a message and its corresponding codeword!



Equivalent channel model under hard decision



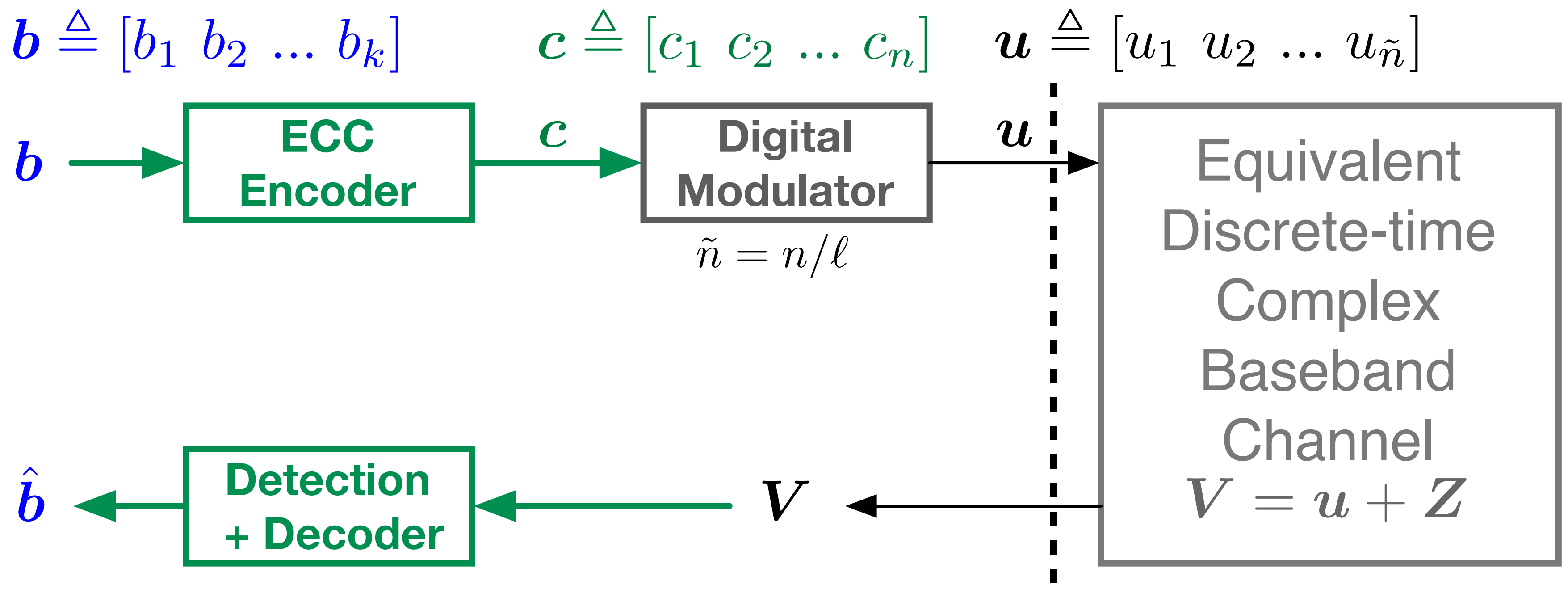
- Binary-input, binary-output: for the each integer i, $c_i, d_i \in \{0, 1\}, \ i = 1, \dots, n$
- Each input bit is flipped with certain probability p :

$$D_i = c_i \oplus E_i, \quad E_i \sim \mathrm{Ber}(p)$$

- For hard decision (bit-level detection), assume that the flips are i.i.d.

p : <u>bit error probability</u> of the modulation scheme

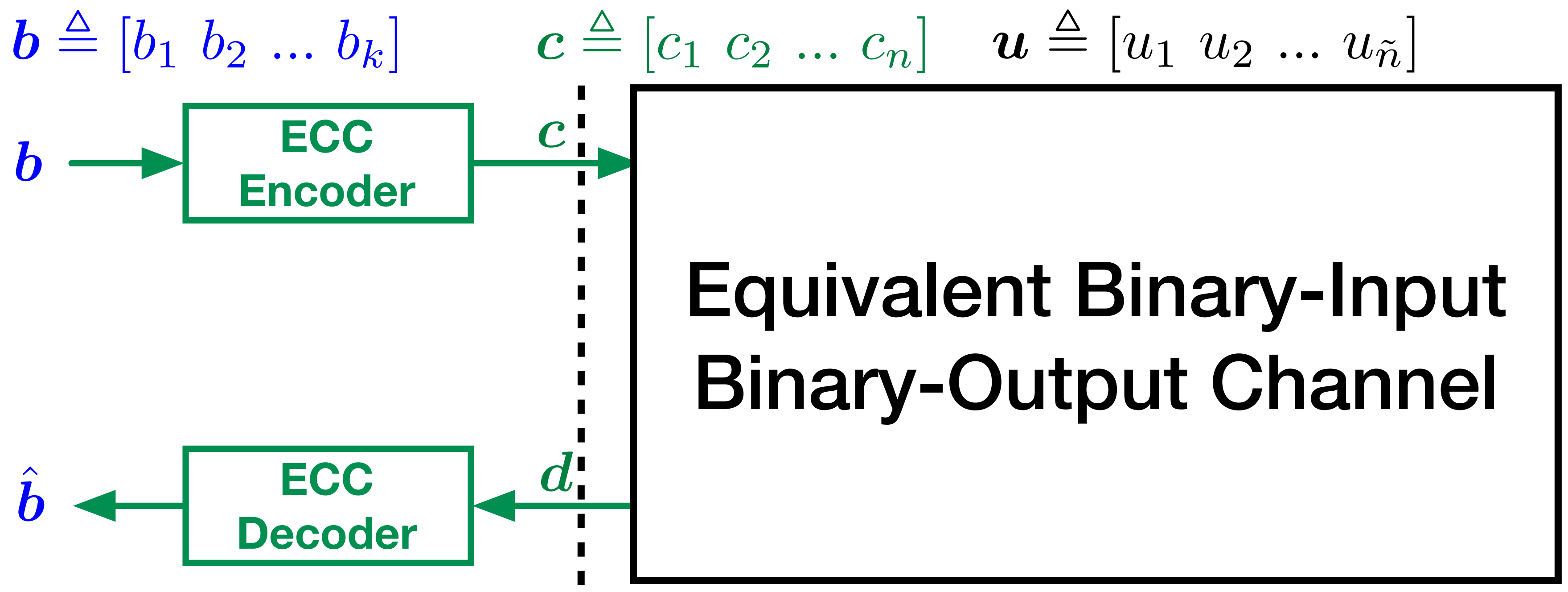
Hard decision vs. soft decision



Soft decision:

Decode $\boldsymbol{b}$ from $\boldsymbol{V}$

$$V_i = u_i + Z_i, \;\; Z_i \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, N_0)$$



Hard decision:

Decode $\boldsymbol{b}$ from $\boldsymbol{D}$

$$D_i = c_i \oplus E_i, \;\; E_i \overset{\text{i.i.d.}}{\sim} \mathrm{Ber}(p)$$

Focus on hard decision next

Maximum likeligood sequence detection

$$\boldsymbol{D} = \boldsymbol{c} \oplus \boldsymbol{E} \longrightarrow \boxed{\text{ML Decoder}} \longrightarrow \hat{\boldsymbol{B}} = \phi_{\mathrm{ML}}(\boldsymbol{D})$$

- Equivalently, finding a length-k path on the trellis diagram such that the likelihood is maximized

- Likelihood (conditional pmf of $\boldsymbol{D}$ given $\boldsymbol{c}$):

WLOG $p < 1/2$

$$P_{\boldsymbol{D}|\boldsymbol{C}}(\boldsymbol{d}|\boldsymbol{c}) = (1-p)^{n-w(\boldsymbol{d}\oplus\boldsymbol{c})} p^{w(\boldsymbol{d}\oplus\boldsymbol{c})} = (1-p)^n (\tfrac{p}{1-p})^{w(\boldsymbol{d}\oplus\boldsymbol{c})}$$

- Maximum likelihood is equivalent to minimum **Hamming distance**!

$$\phi_{\mathrm{ML}}(\boldsymbol{d}) = \operatorname*{argmax}_{\boldsymbol{c}\in\mathcal{C}} P_{\boldsymbol{D}|\boldsymbol{C}}(\boldsymbol{d}|\boldsymbol{c}) = \operatorname*{argmin}_{\boldsymbol{c}\in\mathcal{C}} \underbrace{w(\boldsymbol{d}\oplus\boldsymbol{c})}_{\mathrm{d_H}(\boldsymbol{d},\boldsymbol{c})}$$

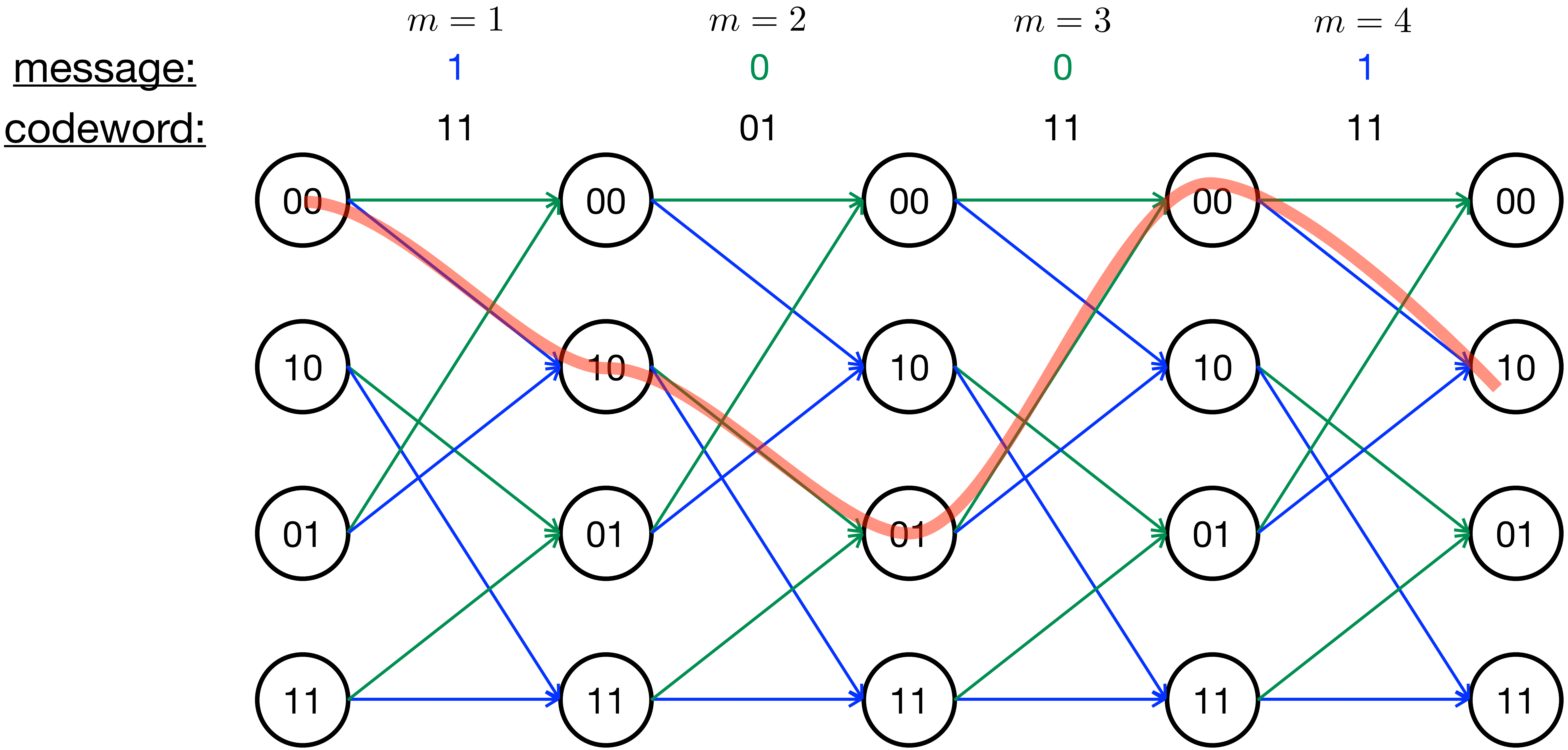
# of locations where $\boldsymbol{d}$ and $\boldsymbol{c}$ disagree

Again, ML ≡ MD!

Decompose the target function (distance)

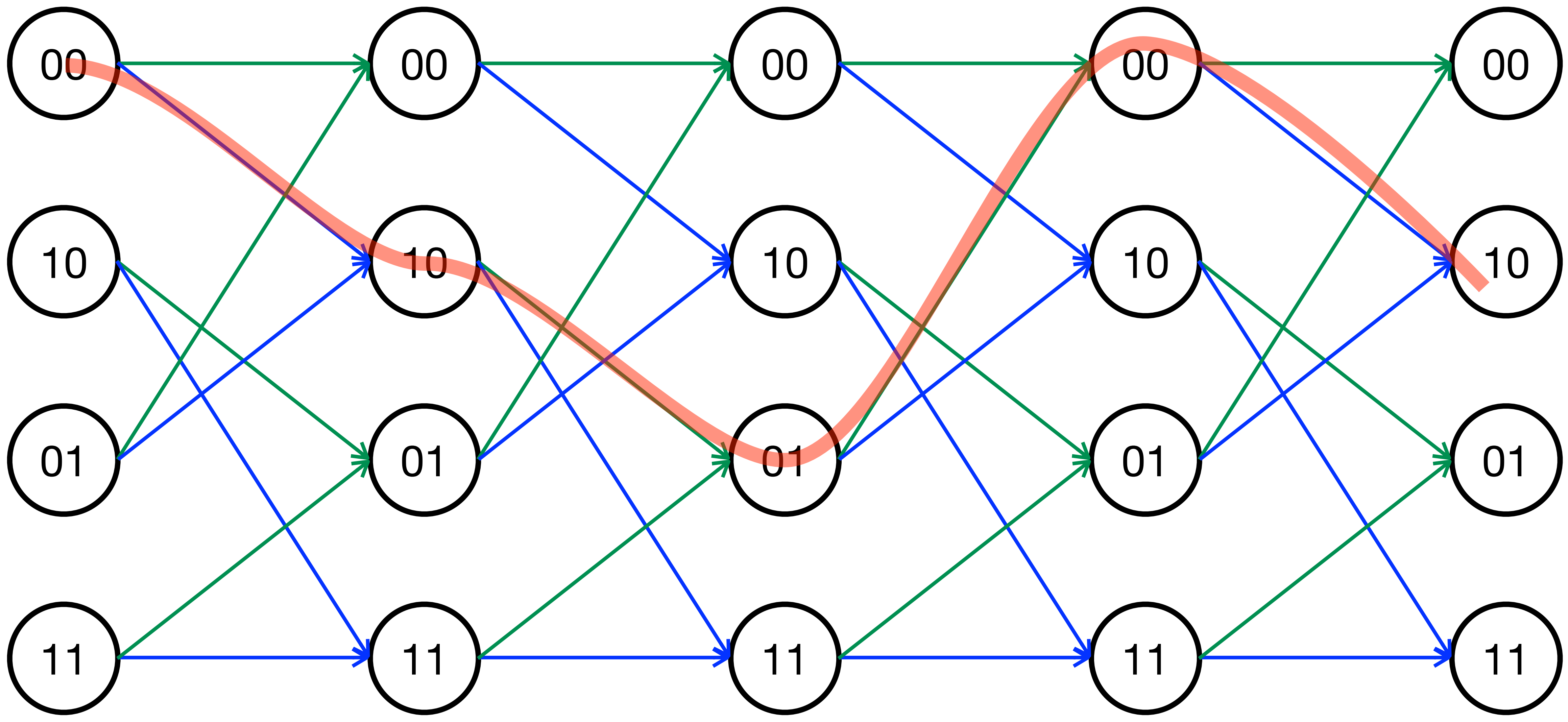
$$\mathrm{d_H}(\boldsymbol{d}, \boldsymbol{c}) = \sum_{i=1}^{n} \mathrm{d_H}(d_i, c_i) = \sum_{m=1}^{k} \mathrm{d_H}(\boldsymbol{d}_m, \boldsymbol{c}_m)$$

decompose into stages of the encoding finite state machine



Decoding: finding the minimum-cost path

	$m = 1$	$m = 2$	$m = 3$	$m = 4$	
received:	10	00	11	00	
codeword:	11	01	11	11	
cost: (distance)	1	1	0	2	$\mathrm{d_H}(\boldsymbol{d}, \boldsymbol{c}) = 4$

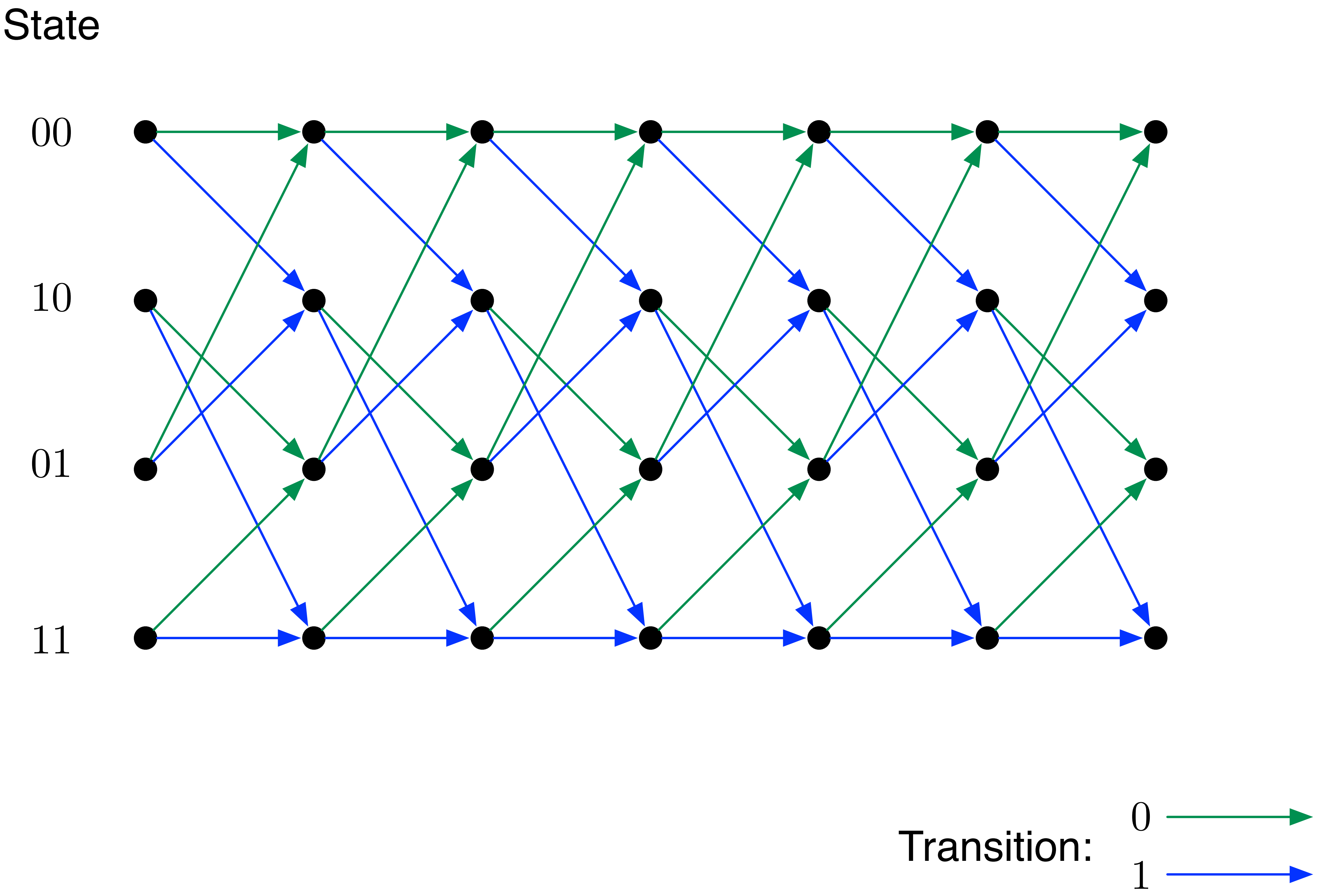


Viterbi algorithm

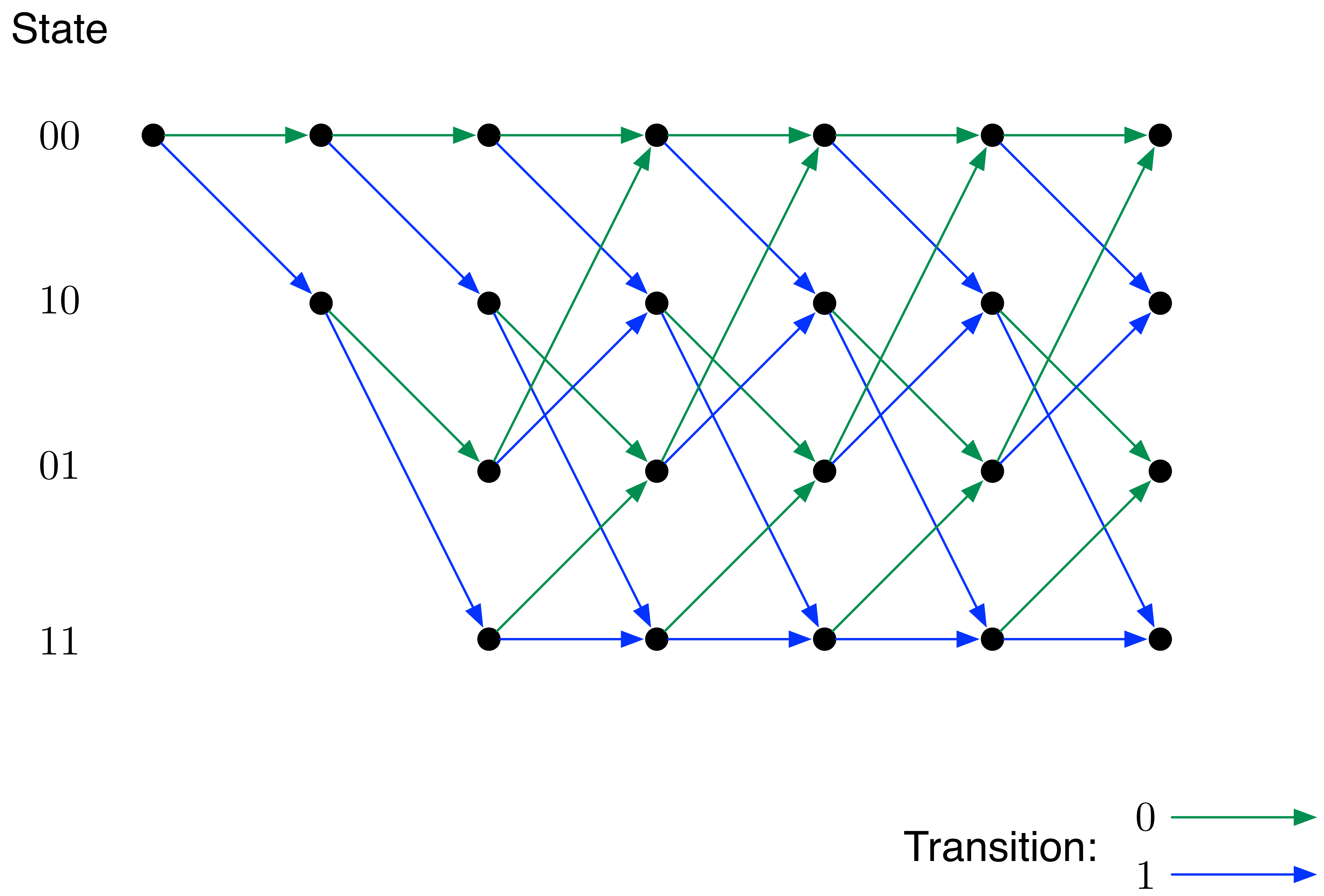
- How to efficiently find a minimum cost path on a trellis?

- For a directed acyclic graph is acyclic, one can use dynamic programming to find the min-cost path, with computational complexity polynomial in the size of the graph

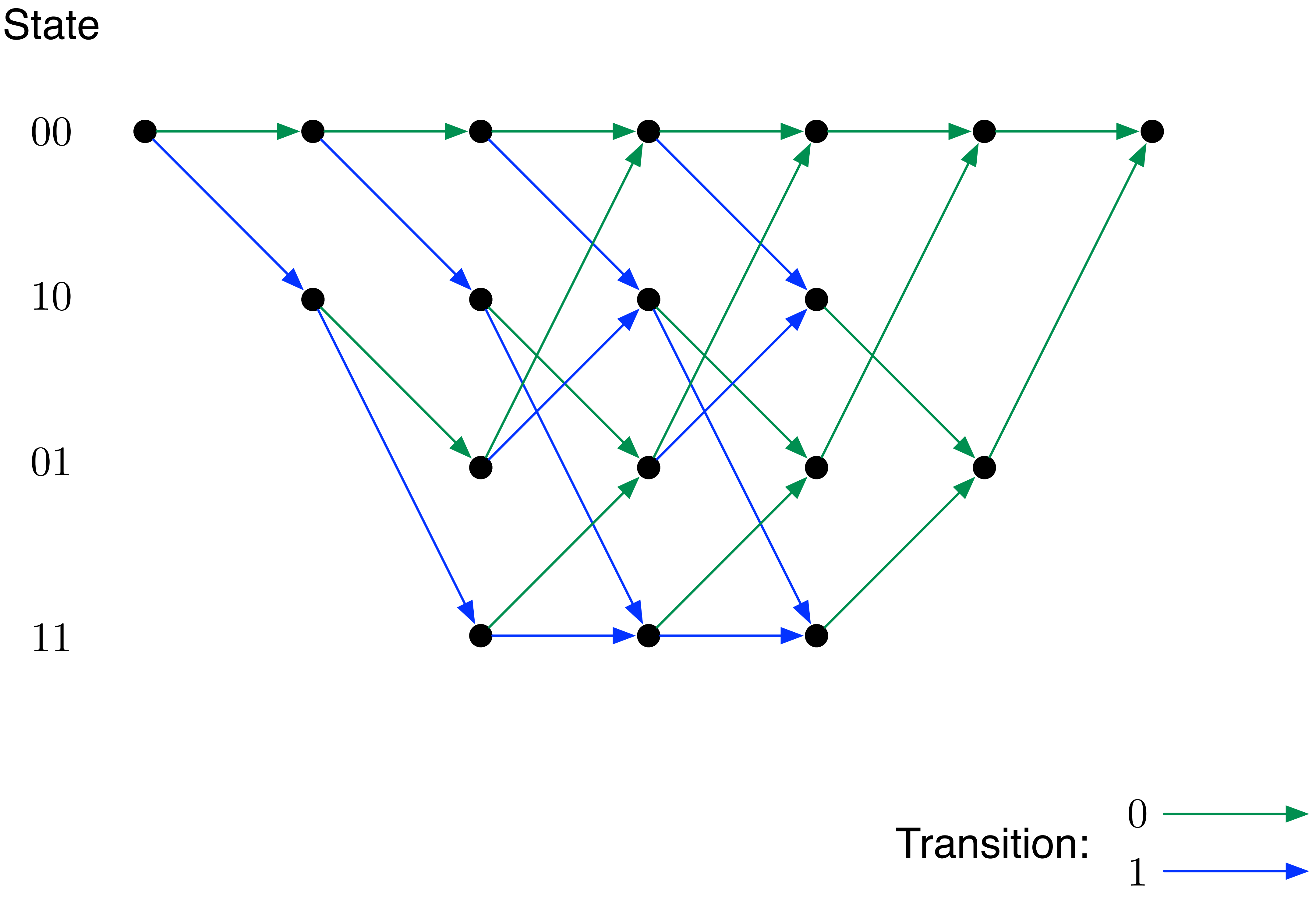
- Viterbi algorithm is a special case of finding the shortest path on a trellis

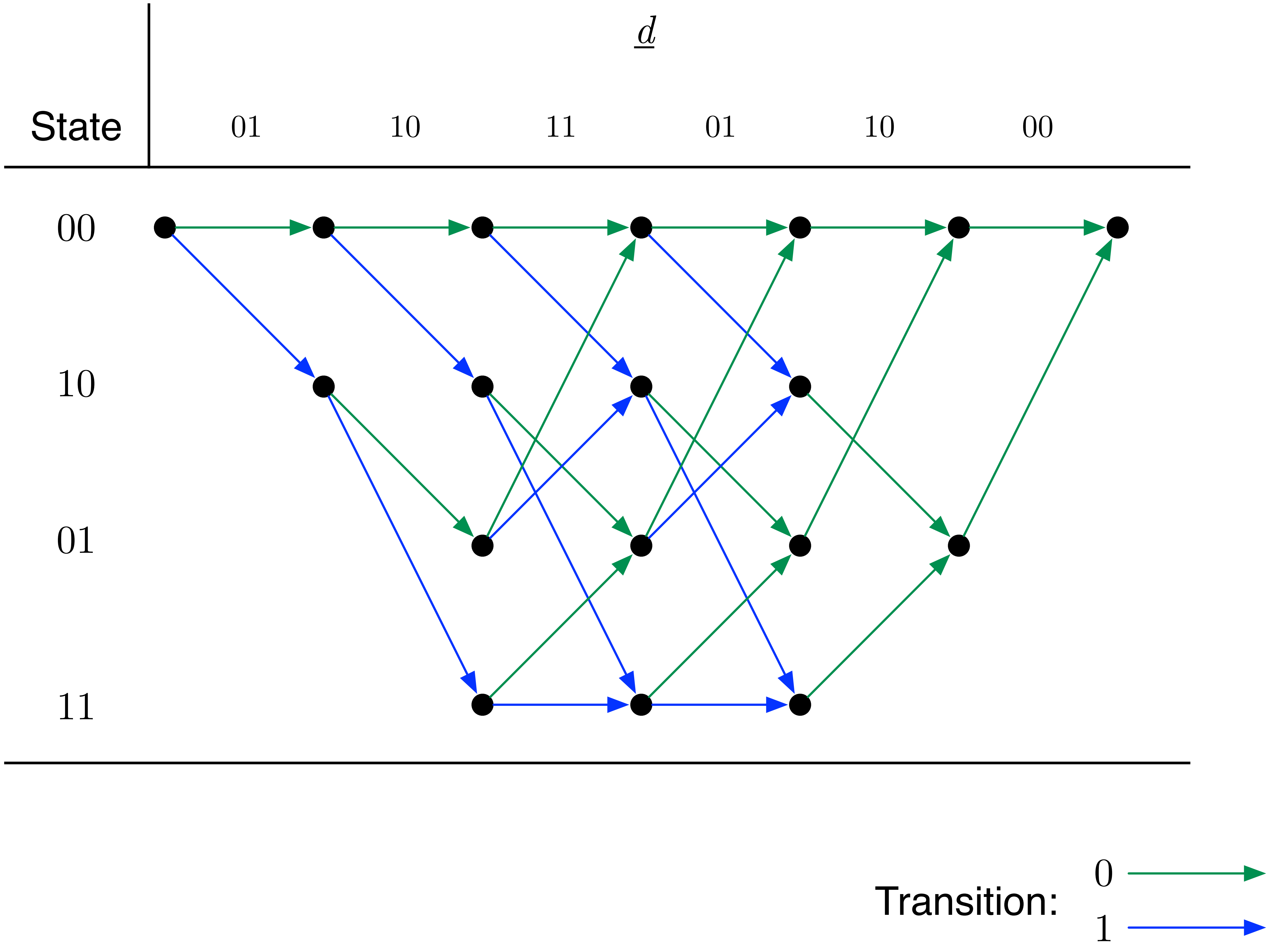

State
00
10
01
11
Transition:
0
1

- Initialization: start with State 00

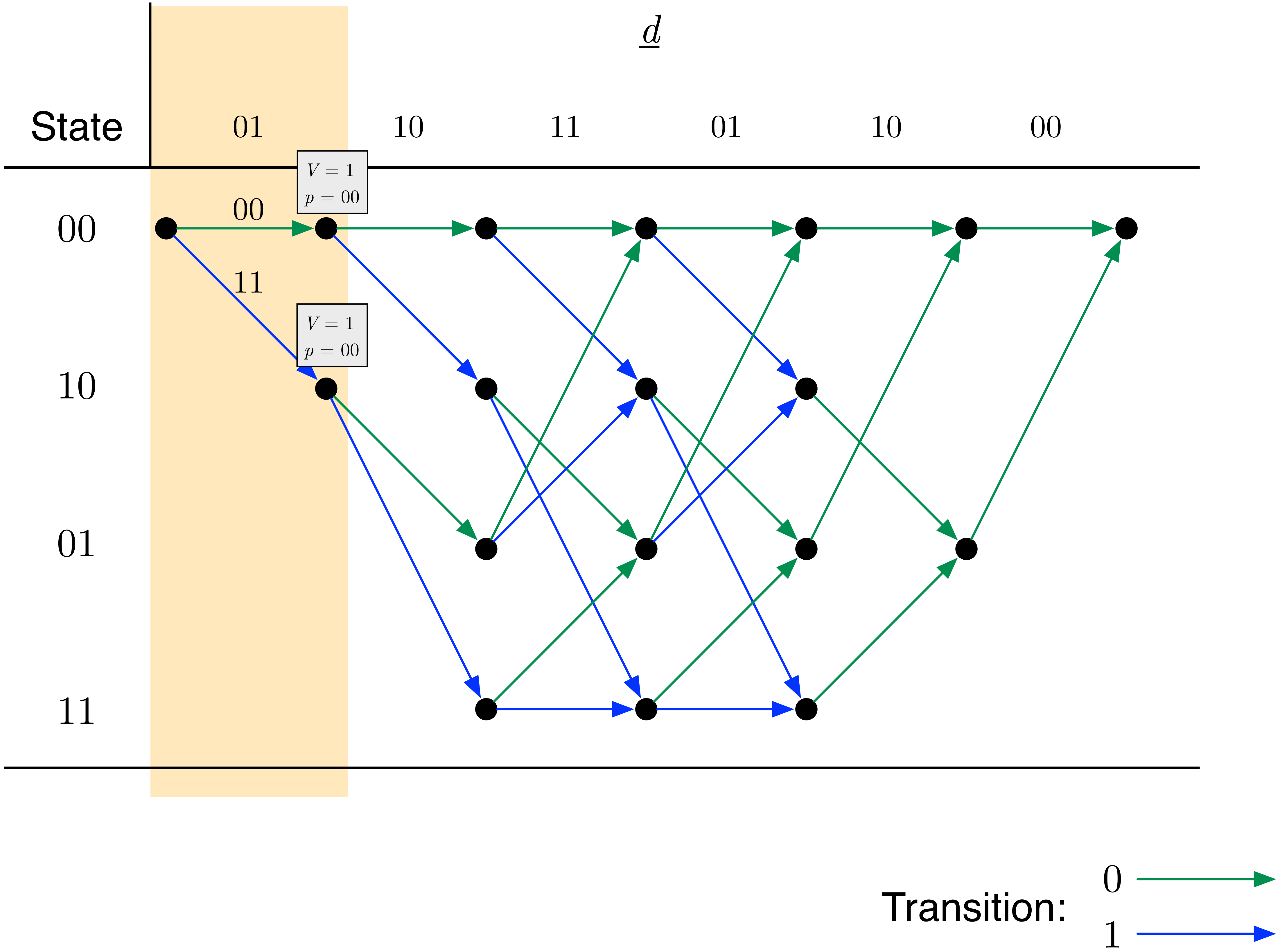


- Termination: end with State 00 by inserting 0,0 in the last two input bits

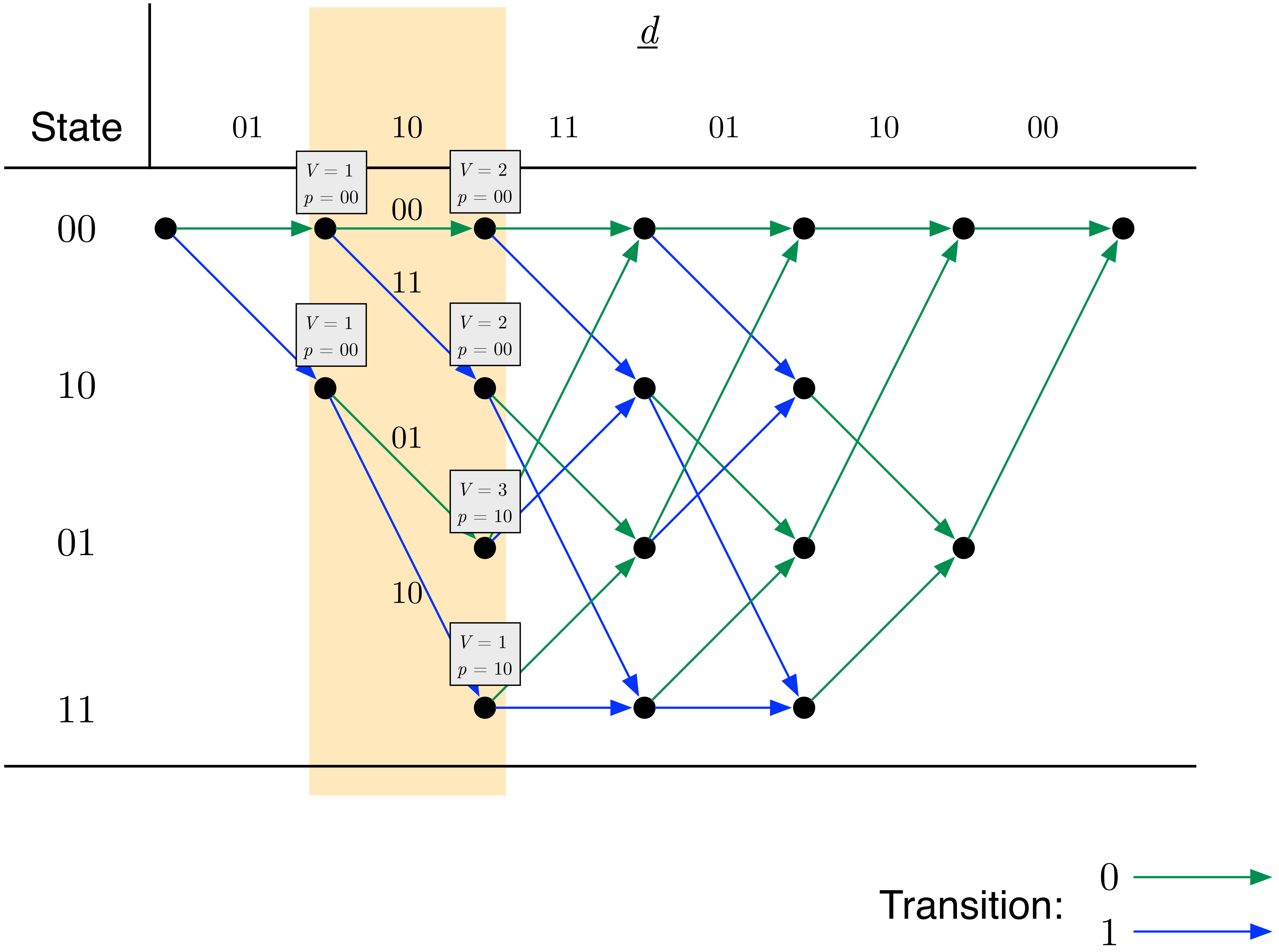




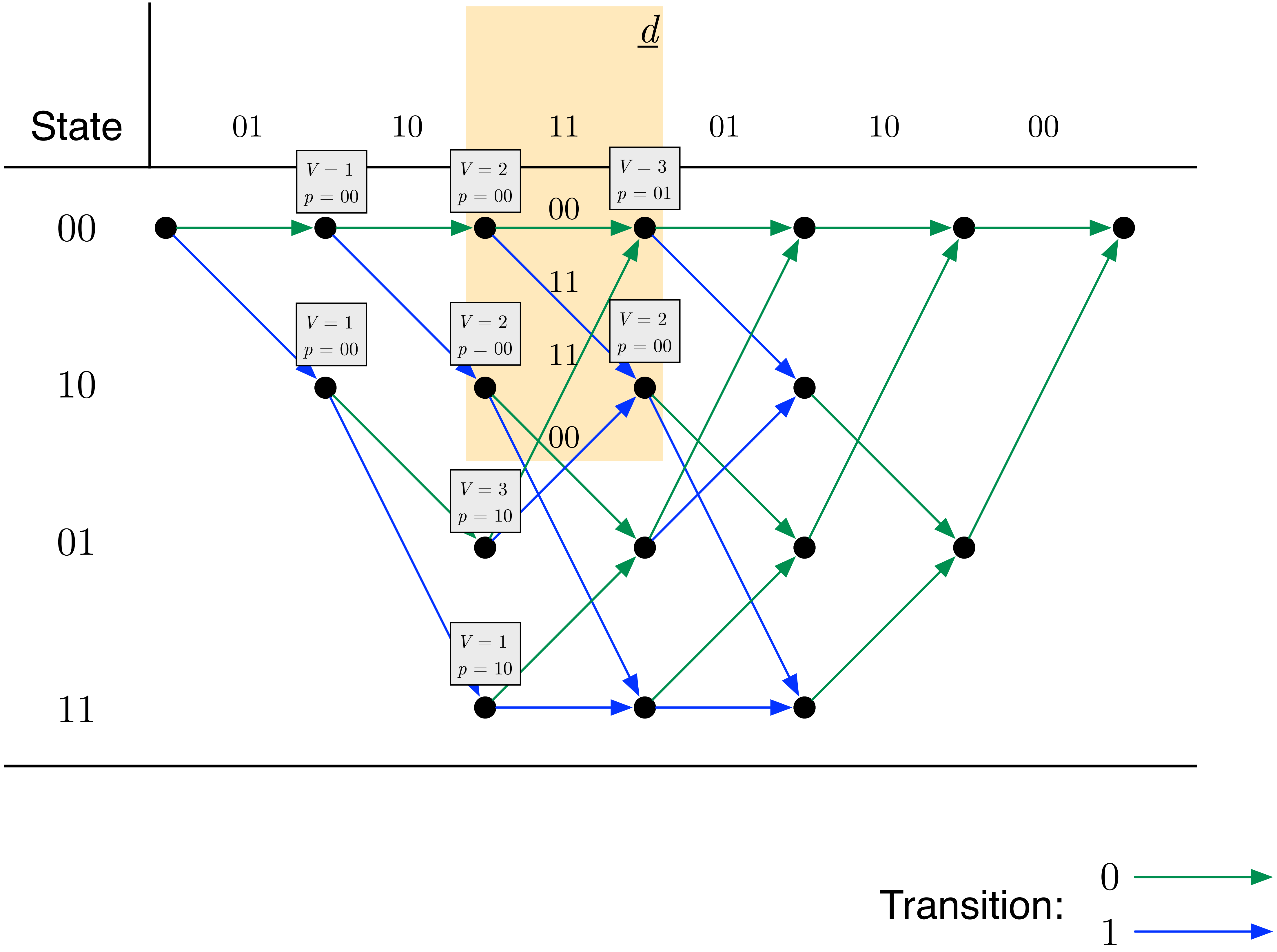
$\underline{d}$
State
01 10 11 01 10 00
00
10
01
11
Transition:
0
1

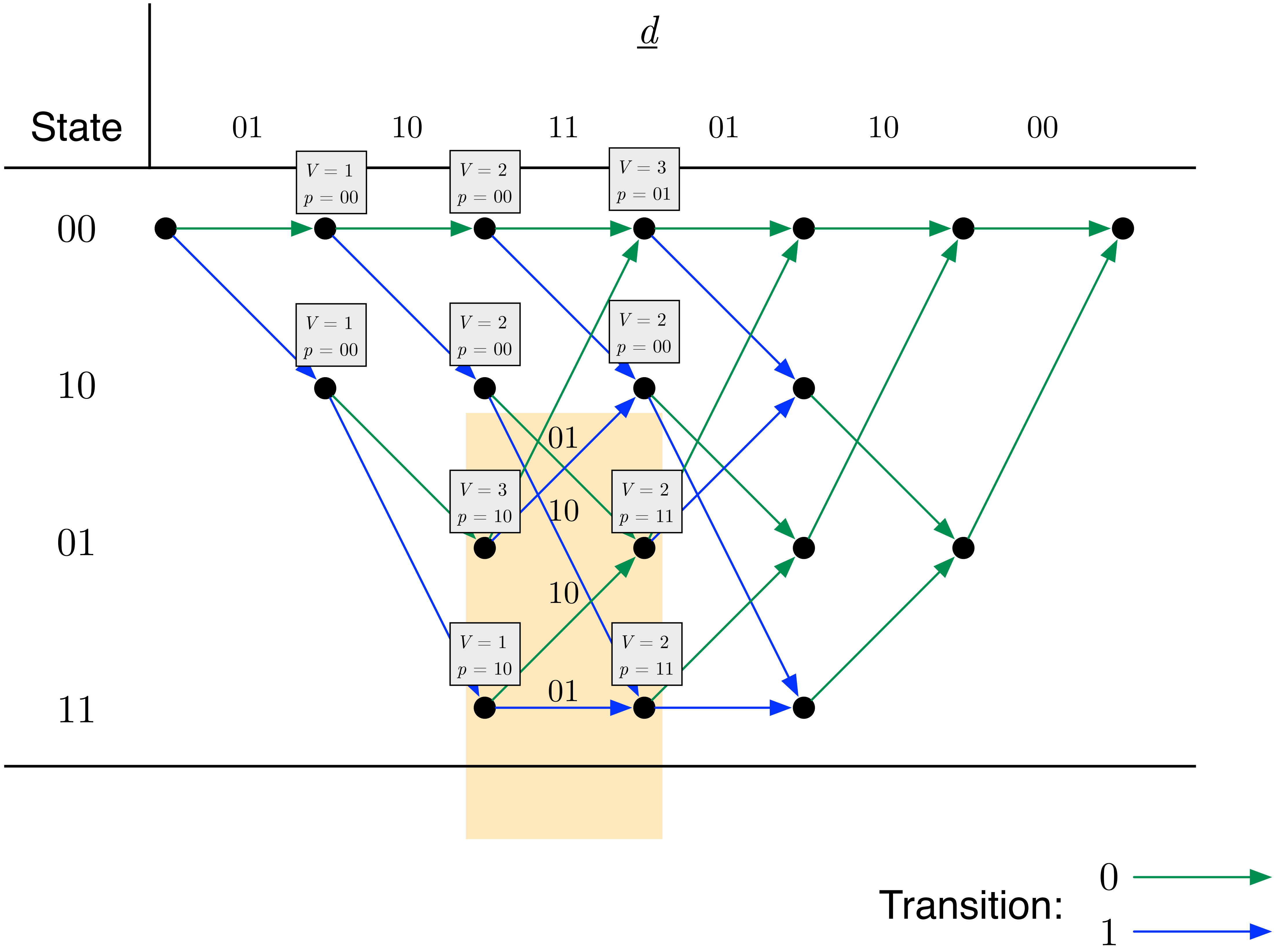


$\underline{d}$
State
01
10
11
01
10
00
00
10
01
11
00
11
$V = 1$
$p = 00$
$V = 1$
$p = 00$
Transition:
0
1

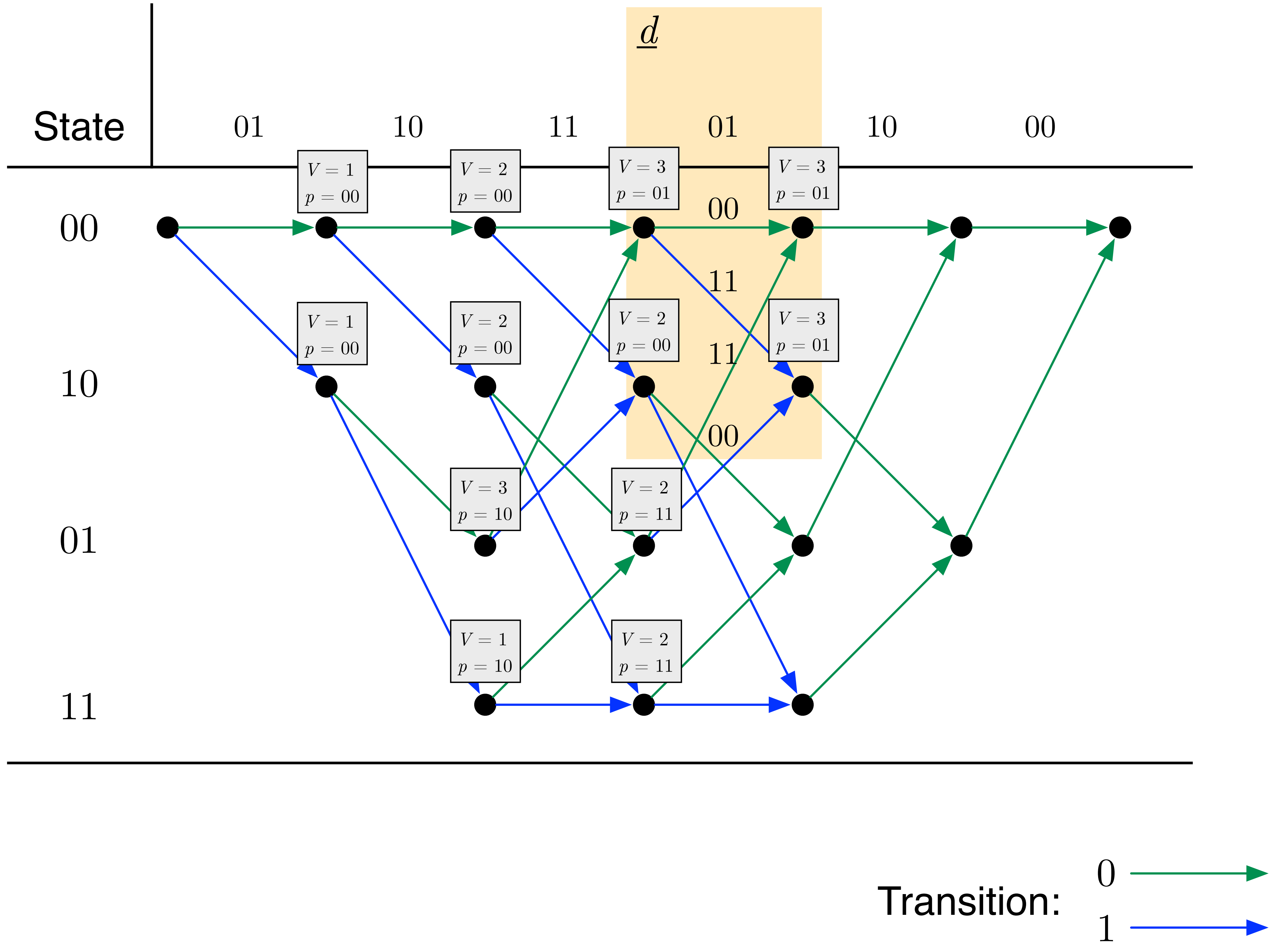


$\underline{d}$
State
01
10
11
01
10
00
00
10
01
11
$V = 1$ $p = 00$
$V = 2$ $p = 00$
$V = 1$ $p = 00$
$V = 2$ $p = 00$
$V = 3$ $p = 10$
$V = 1$ $p = 10$
00
11
01
10
Transition:
0
1

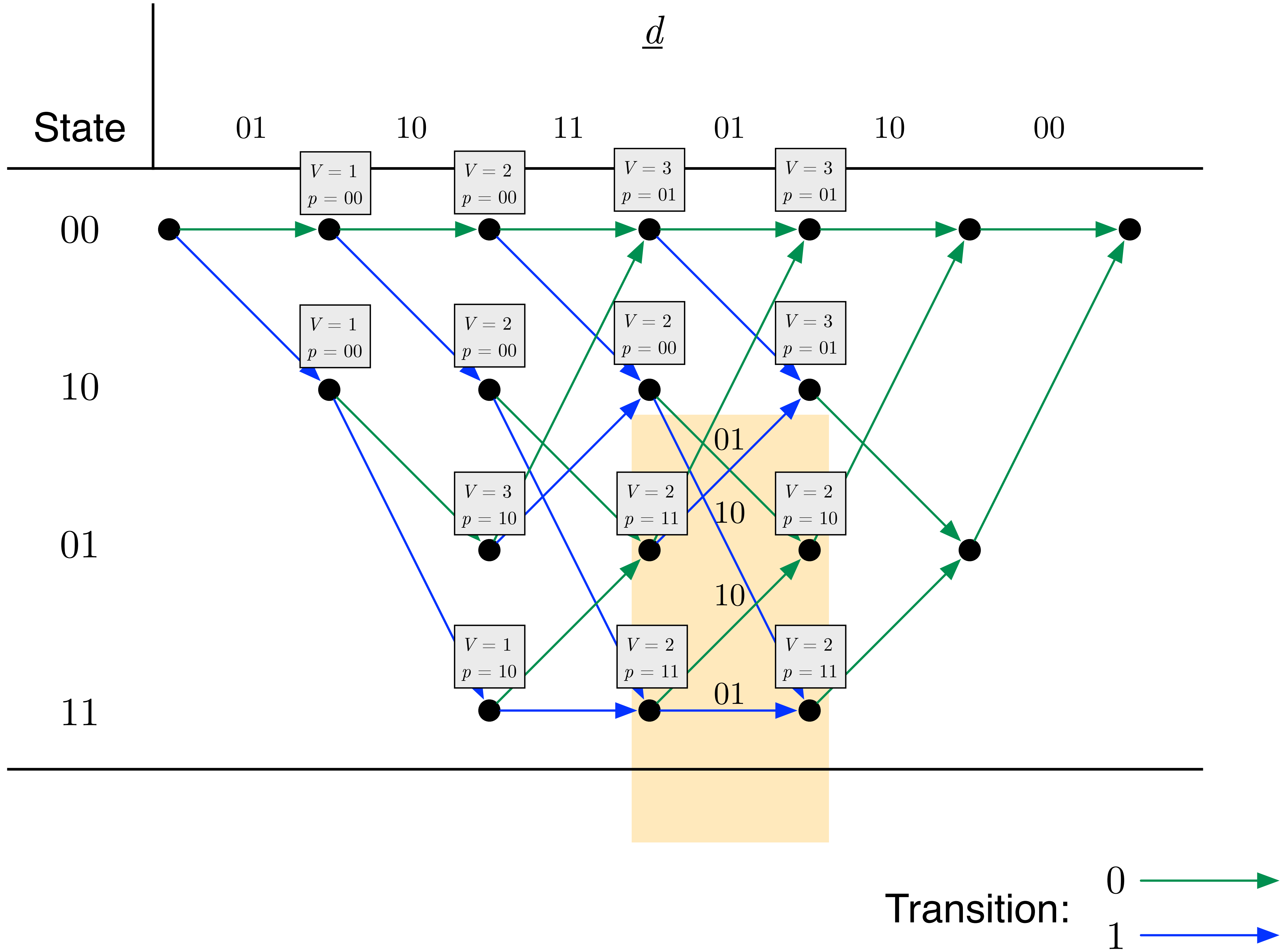

State
01
10
11
01
10
00
$\underline{d}$
00
10
01
11
$V=1$ $p=00$
$V=2$ $p=00$
$V=3$ $p=01$
$V=1$ $p=00$
$V=2$ $p=00$
$V=2$ $p=00$
$V=3$ $p=10$
$V=1$ $p=10$
00
11
11
00
Transition:
0
1

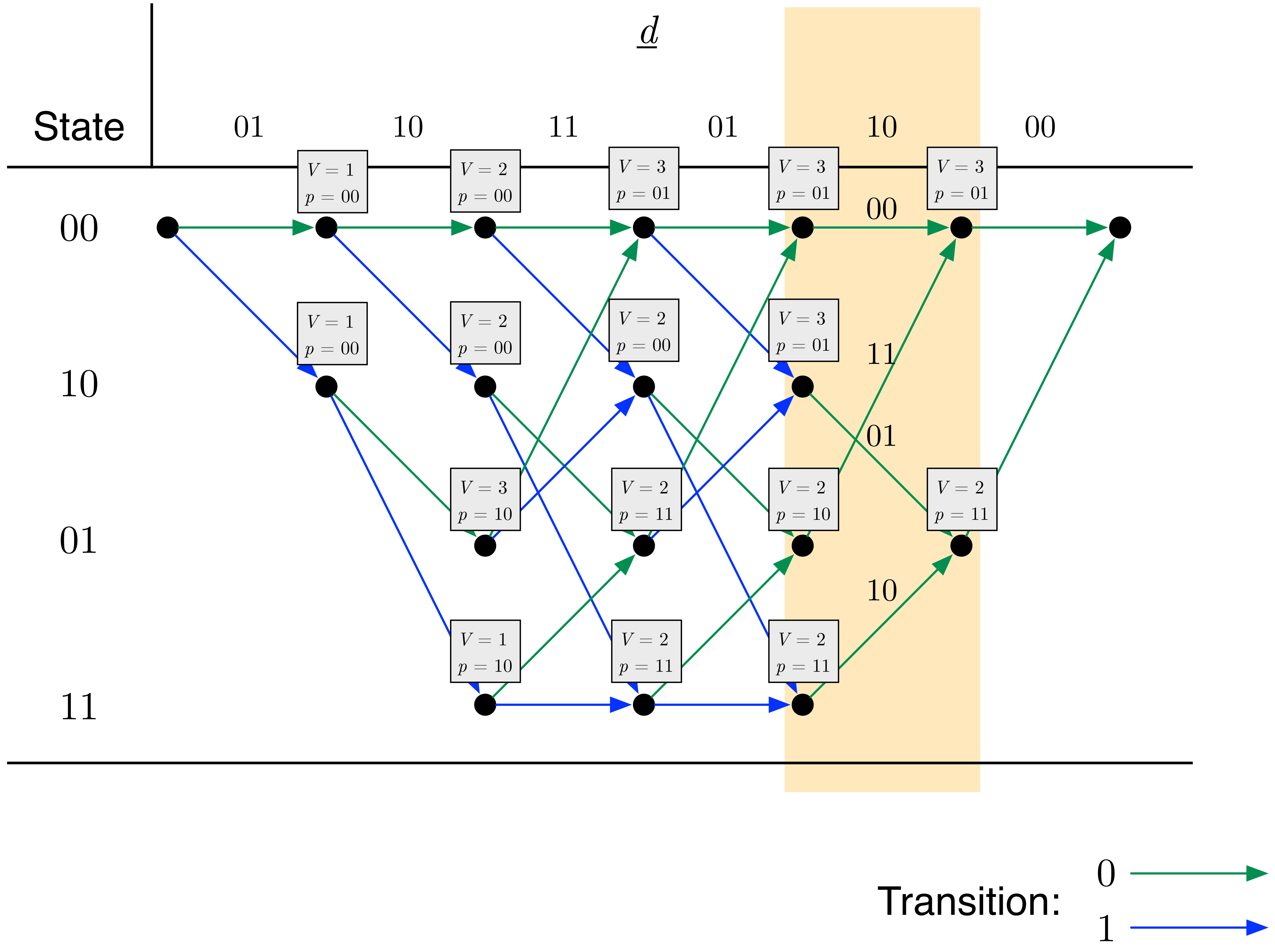


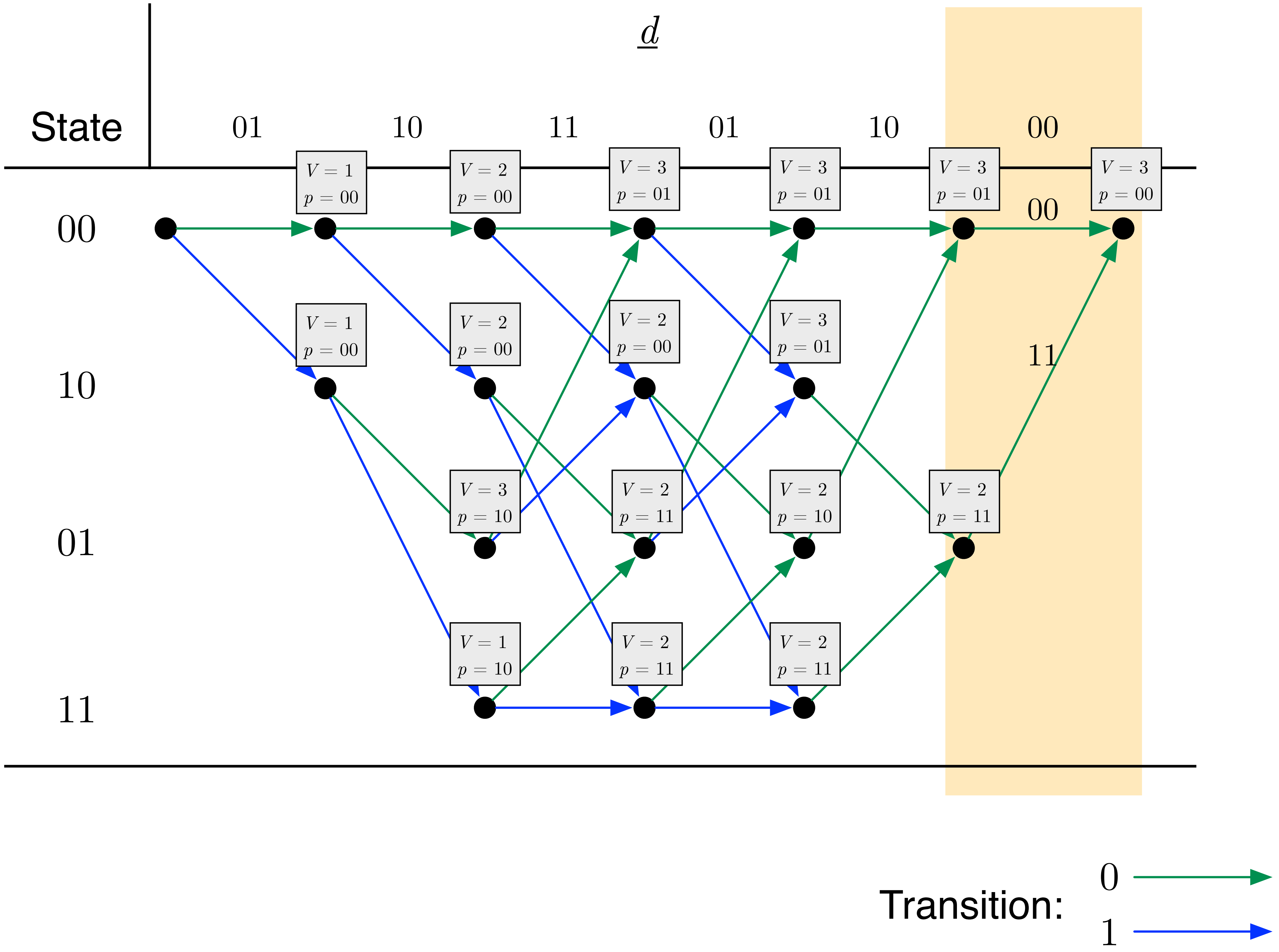
State
d
01
10
11
01
10
00
00
10
01
11
V = 1
p = 00
V = 2
p = 00
V = 3
p = 01
V = 1
p = 00
V = 2
p = 00
V = 2
p = 00
V = 3
p = 10
V = 2
p = 11
V = 1
p = 10
V = 2
p = 11
01
10
10
01
Transition:
0
1



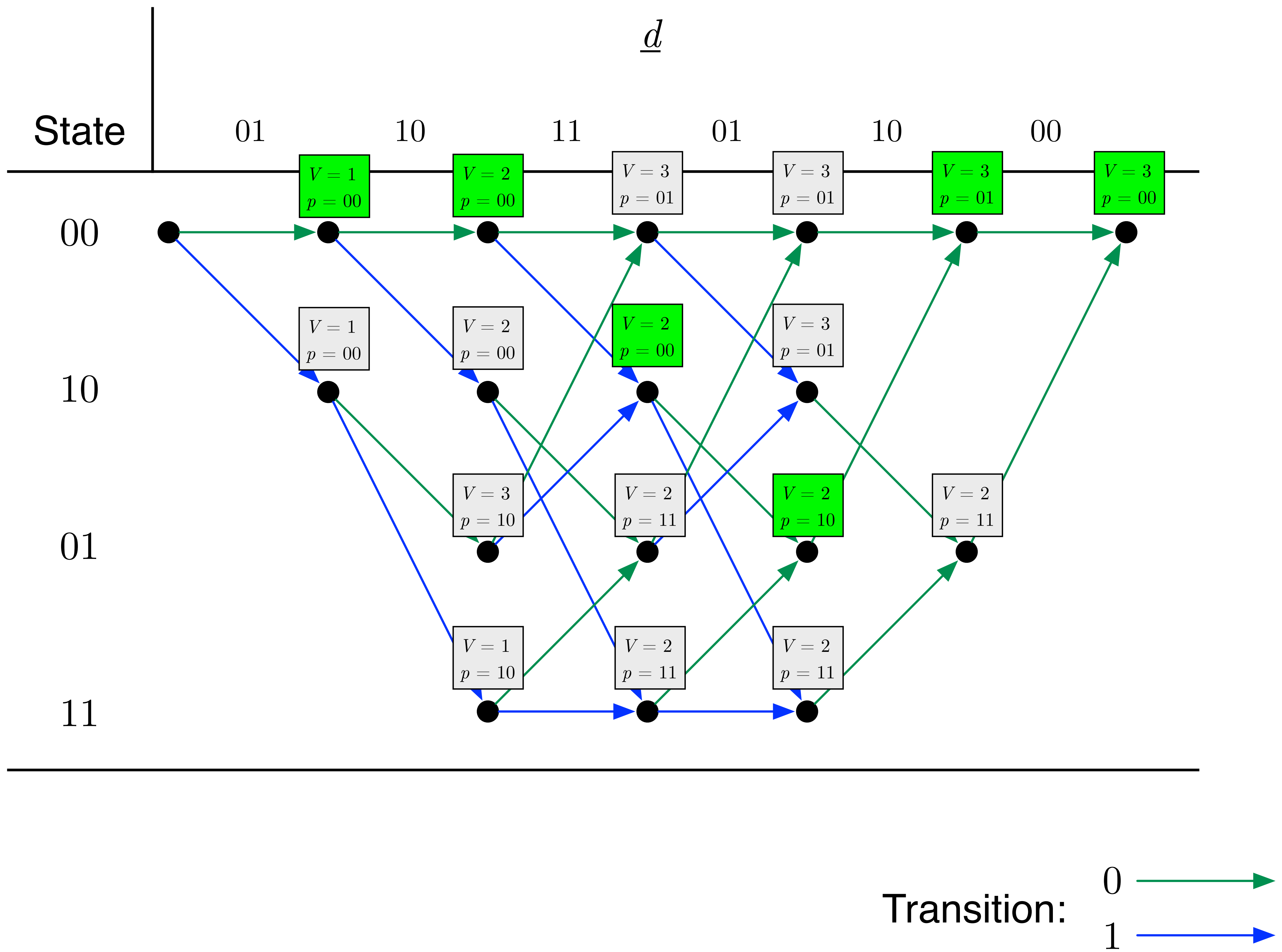
$\underline{d}$
State
01
10
11
01
10
00
00
10
01
11
$V = 1$ $p = 00$
$V = 2$ $p = 00$
$V = 3$ $p = 01$
$V = 3$ $p = 01$
$V = 1$ $p = 00$
$V = 2$ $p = 00$
$V = 2$ $p = 00$
$V = 3$ $p = 01$
$V = 3$ $p = 10$
$V = 2$ $p = 11$
$V = 1$ $p = 10$
$V = 2$ $p = 11$
00
11
11
00
Transition:
0
1


$\underline{d}$
State
01 10 11 01 10 00
00
10
01
11
V = 1 p = 00
V = 2 p = 00
V = 3 p = 01
V = 3 p = 01
V = 1 p = 00
V = 2 p = 00
V = 2 p = 00
V = 3 p = 01
V = 3 p = 10
V = 2 p = 11
V = 2 p = 10
V = 1 p = 10
V = 2 p = 11
V = 2 p = 11
01
10
10
01
Transition:
0
1

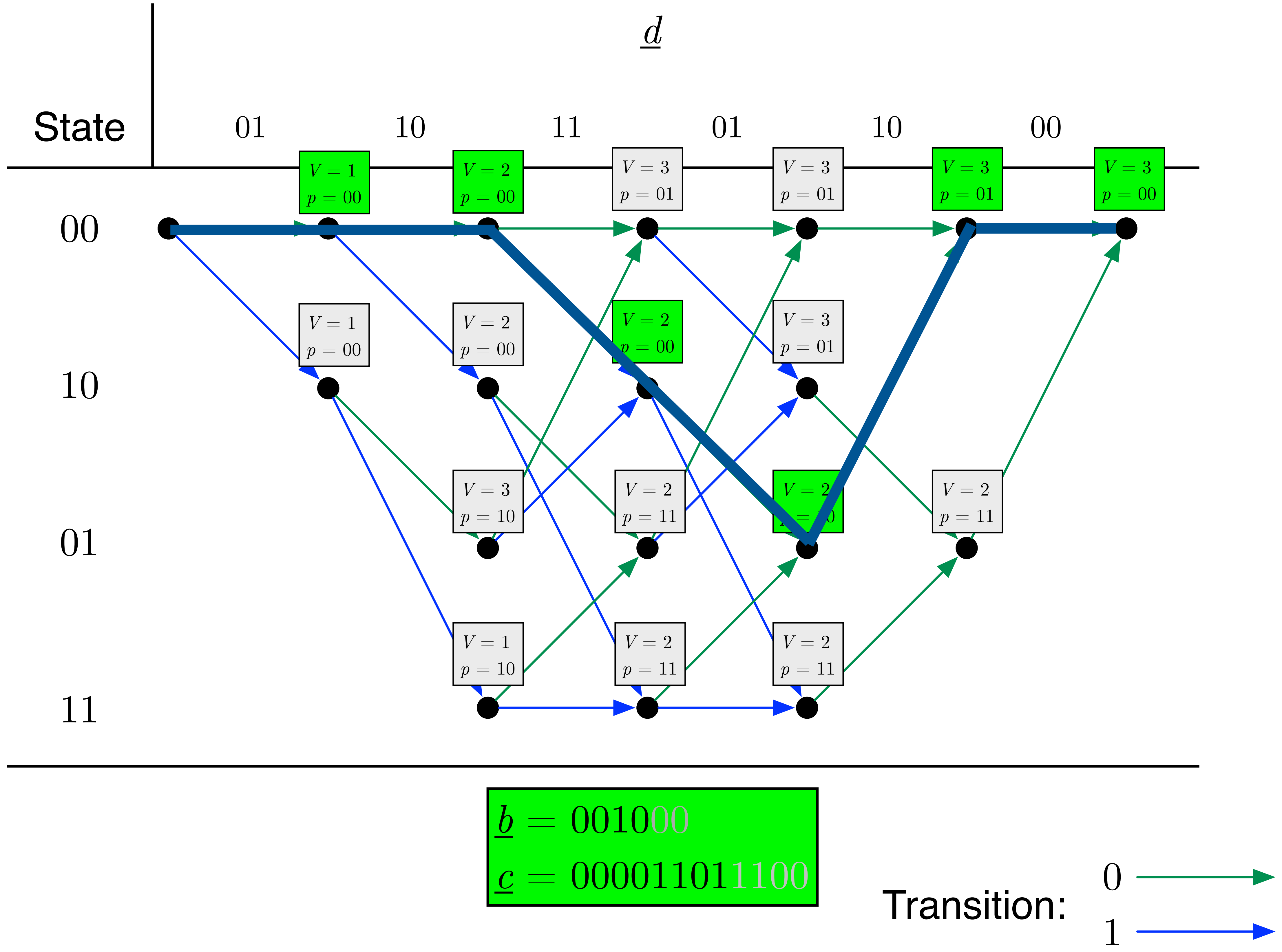

State
$\underline{d}$
01
10
11
01
10
00
00
10
01
11
$V = 1$ $p = 00$
$V = 2$ $p = 00$
$V = 3$ $p = 01$
$V = 3$ $p = 01$
$V = 3$ $p = 01$
$V = 1$ $p = 00$
$V = 2$ $p = 00$
$V = 2$ $p = 00$
$V = 3$ $p = 01$
$V = 3$ $p = 10$
$V = 2$ $p = 11$
$V = 2$ $p = 10$
$V = 2$ $p = 11$
$V = 1$ $p = 10$
$V = 2$ $p = 11$
$V = 2$ $p = 11$
00
11
01
10
Transition:
0
1

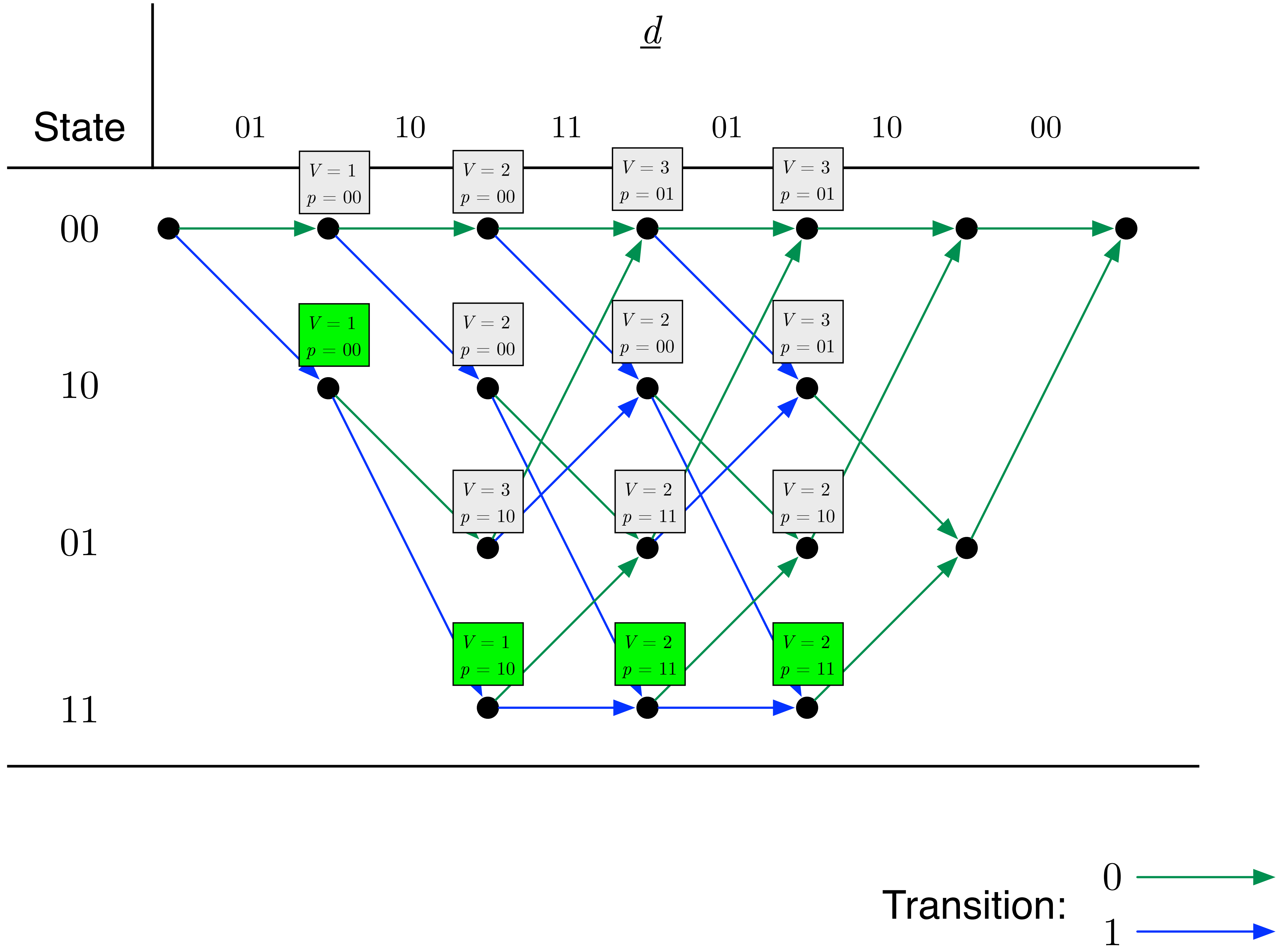


d
State
01
10
11
01
10
00
00
10
01
11
V = 1 p = 00
V = 2 p = 00
V = 3 p = 01
V = 3 p = 01
V = 3 p = 01
V = 3 p = 00
V = 1 p = 00
V = 2 p = 00
V = 2 p = 00
V = 3 p = 01
V = 3 p = 10
V = 2 p = 11
V = 2 p = 10
V = 2 p = 11
V = 1 p = 10
V = 2 p = 11
V = 2 p = 11
00
11
Transition:
0
1

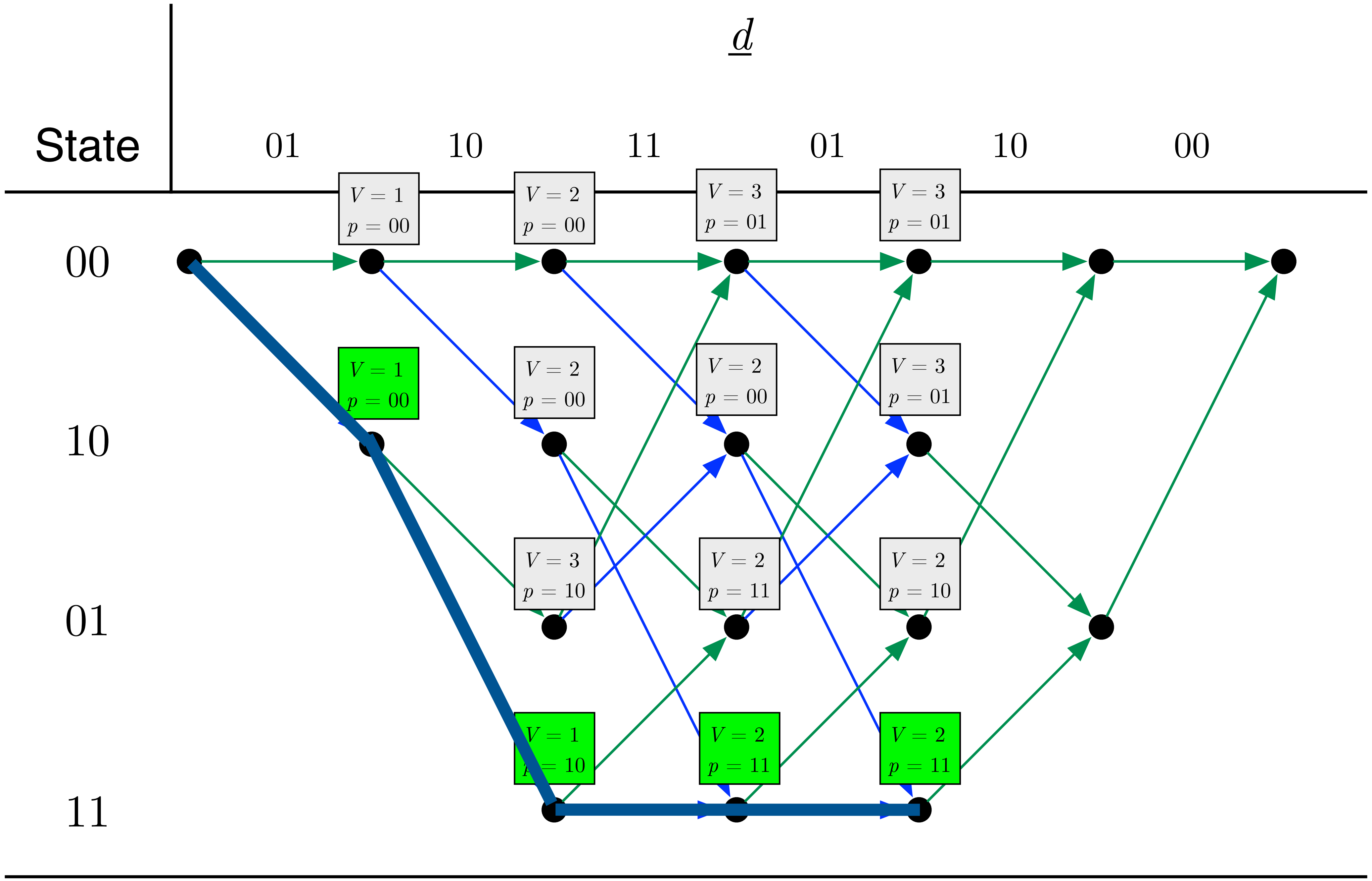


$\underline{d}$
State
01
10
11
01
10
00
00
10
01
11
V = 1 p = 00
V = 2 p = 00
V = 3 p = 01
V = 3 p = 01
V = 3 p = 01
V = 3 p = 00
V = 1 p = 00
V = 2 p = 00
V = 2 p = 00
V = 3 p = 01
V = 3 p = 10
V = 2 p = 11
V = 2 p = 10
V = 2 p = 11
V = 1 p = 10
V = 2 p = 11
V = 2 p = 11
Transition:
0
1


d
State
01
10
11
01
10
00
00
10
01
11
V = 1 p = 00
V = 2 p = 00
V = 3 p = 01
V = 3 p = 01
V = 3 p = 01
V = 3 p = 00
V = 1 p = 00
V = 2 p = 00
V = 2 p = 00
V = 3 p = 01
V = 3 p = 10
V = 2 p = 11
V = 2 p = 11
V = 1 p = 10
V = 2 p = 11
V = 2 p = 11
b = 001000
c = 000011011100
Transition:
0
1



$\underline{d}$
State
01
10
11
01
10
00
00
10
01
11
$V = 1$ $p = 00$
$V = 2$ $p = 00$
$V = 3$ $p = 01$
$V = 3$ $p = 01$
$V = 1$ $p = 00$
$V = 2$ $p = 00$
$V = 2$ $p = 00$
$V = 3$ $p = 01$
$V = 3$ $p = 10$
$V = 2$ $p = 11$
$V = 2$ $p = 10$
$V = 1$ $p = 10$
$V = 2$ $p = 11$
$V = 2$ $p = 11$
Transition:
0
1



$\underline{b} = 111100$

$\underline{c} = 111001011011$

Transition: 0 → 1 →

Other channel models

- Soft decision: the additive cost function becomes the square of Euclidean distance from the estimated signal to the received signal
  - Remark: things can be a bit trickier when the modulation size (# of bits in one symbol) is larger than the # of output streams.
  - Think about how to draw the state transition diagram and the trellis!
- Erasure channel: each bit is either obtained without any error, or it is erased
  - This can be realized by a detector which report the decoded bits if the likelihood function of the decoded symbol is significantly larger than the threshold of other candidates
  - For an erasure channel, decoding is simple: find the codeword that match the received sequence at the non-erased locations
  - Aside: derive the pairwise error probability!
  - Can you derive the Viterbi decoder for a convolutional code in the erasure channel?