2.1 Find a generator polynomial of a single-error-correcting RS code of length 15 over $\mathbb{F}_{16}$. The polynomial $x^4 + x + 1$ is primitive over $\mathbb{F}_2$.

Solution:

To be able to correct one error, we need the minimum distance 3. Thus, the generator matrix should have degree 2.

$$g(y) = (y - x)(y - x^2) = y^2 + (x^2 + x)y + x^3.$$

2.2 Find a generator polynomial of a double-error-correcting RS code of length 31 over $\mathbb{F}_{32}$. The polynomial $x^5 + x^2 + 1$ is primitive over $\mathbb{F}_2$.

Solution:

To be able to correct two errors, we need the minimum distance 5. Thus, the generator matrix should have degree 4.

$$\begin{aligned} g(y) &= (y-x)(y-x^2)(y-x^3)(y-x^4) = \ldots = \\ &= y^4 + (x^4 + x^3 + x^2 + x)y^3 + (x^2 + x)y^2 + (x^3 + 1)y + x^4 + 1. \end{aligned}$$

2.3 A convolutional code is defined by the generator matrix

$$G(D) = \big(1 + D + D^3, 1 + D + D^2 + D^3\big)$$

a. Determine the rate of the code.

b. Draw the corresponding encoder.

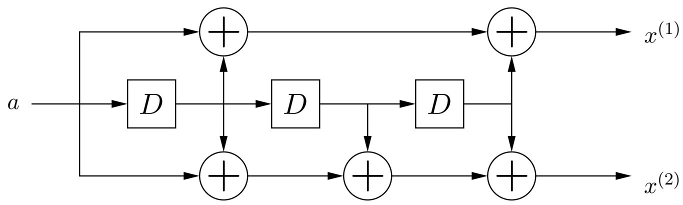
c. Determine the free distance of the code.

Solution:
We are given the generator matrix

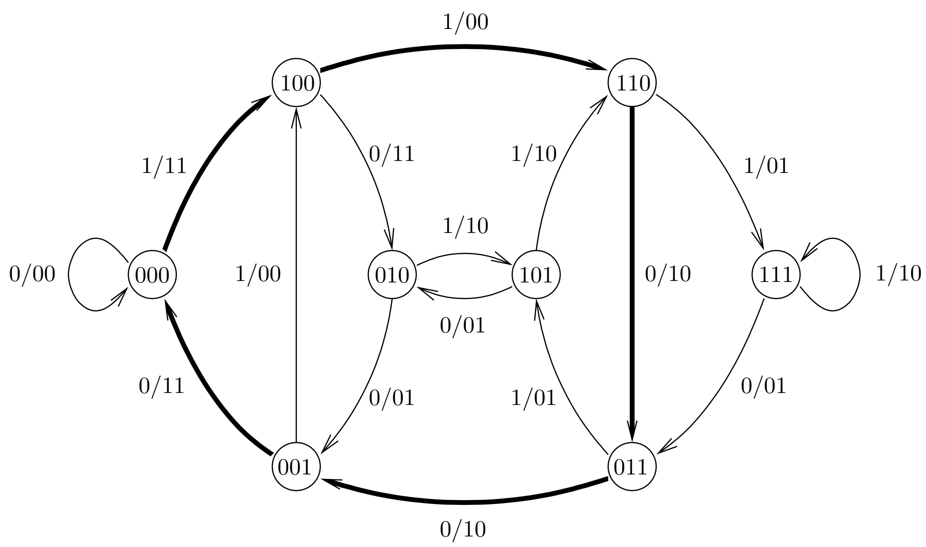
$$G(D) = \big(1 + D + D^3, 1 + D + D^2 + D^3\big)$$

a. $G(D)$ is a 1×2 matrix. The rate of the code is therefore 1/2.

b. The generator matrix gives us the following encoder.



c. To determine the free distance, we start by drawing the state diagram of the encoder, where we label the edges by $a_i/x_i^{(1)}x_i^{(2)}$.



The path corresponding to a min-weight codeword is emphasized with bold edges. Convolutional codes are linear. Thus, the free distance is equal to the smallest non-zero weight. We count to six ones along the bold path. Therefore the code has free distance $d_{\text{free}} = 6$.

2.4 A convolutional code maps a binary sequence

$$a = a_0, a_1, a_2, \ldots$$

on a pair $\{x^{(1)}, x^{(2)}\}$ of sequences according to

$$x_i^{(1)} = a_i + a_{i-1}, \qquad x_i^{(2)} = a_i.$$

As usual, we assume $a_{-1} = 0$. The channel distorts the sequences, and we receive the sequences

$$r_i^{(1)} = x_i^{(1)} + e_i^{(1)}, \qquad r_i^{(2)} = x_i^{(2)} + e_i^{(2)}.$$

The first received pairs are

$$\begin{array}{lccccc} r^{(1)}: & 1 & 1 & 0 & 1 & 1 \\ r^{(2)}: & 0 & 0 & 0 & 1 & 0 \end{array}$$

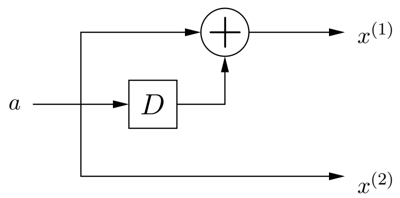
Estimate the sequence a as far as the Viterbi algorithm gives a unique estimate.

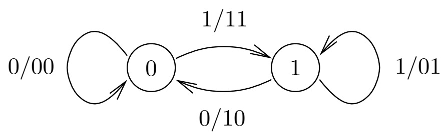
Solution:

The encoding rule gives us the generator matrix

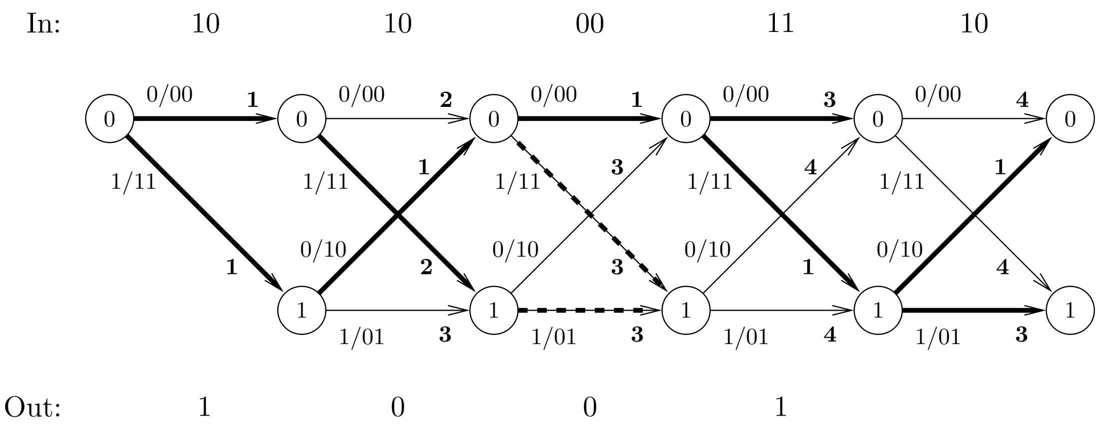
$$G(D) = \big(1 + D, 1\big).$$

The encoder and state diagram are given below, where we label the edges by $a_i/x_i^{(1)}x_i^{(2)}$.





We draw the trellis of the code, and run the Viterbi algorithm based on the given input.



The bold edges are the survivors up to that point. Note that there is a draw between two alternatives in the third interval. At that point we can choose any of the two alternatives. Both are equally good. As can be seen in the figure, we can uniquely decode the first four information bits, and those are 1001.

2.5 A convolutional code maps a binary sequence

$$a = a_0, a_1, a_2, \ldots$$

on a triplet $\{x^{(1)}, x^{(2)}, x^{(3)}\}$ of sequences according to

$$x_i^{(1)} = a_i, \qquad x_i^{(2)} = a_i + a_{i-1}, \qquad x_i^{(2)} = a_i + a_{i-1} + a_{i-2}.$$

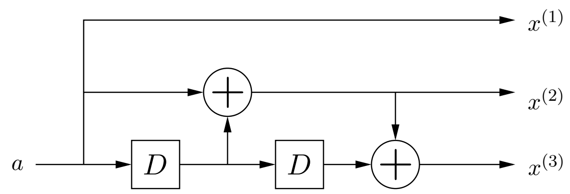
As usual, we assume $a_{-1} = a_{-2} = 0$. Determine the free distance of the code.

Solution:

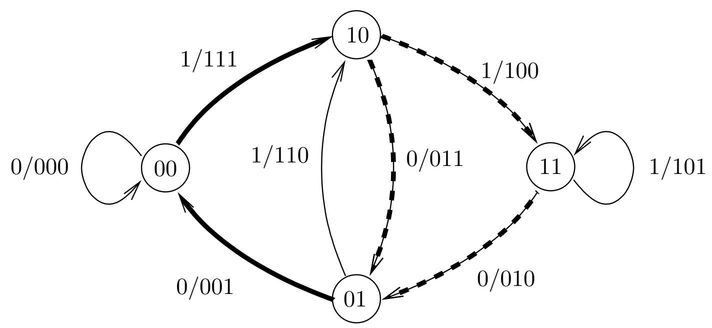
The generator matrix corresponding to the given rules is

$$G(D) = (1, 1 + D, 1 + D + D^2),$$

and the corresponding encoder is given below.

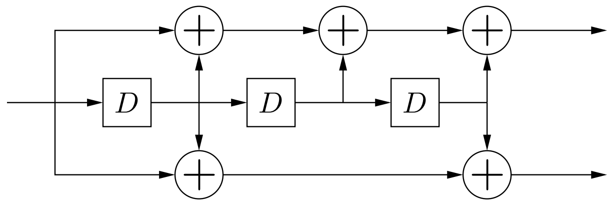


The state diagram of the encoder is given below.



The bold edges correspond to min-weight paths. The dashed bold edges correspond to two different min-weight paths. We immediately see that the free distance is 6.

2.6 Consider the following convolutional encoder.



a. Determine the rate of the code.

b. Draw the state diagram of the encoder. Indicate both input and output on all edges.

c. Determine the free distance of the code.

Solution:
This problem is problem number 2.3, slightly reformulated. See the solution to that problem.

2.7 A convolutional code is defined by the generator matrix

$$G(D) = \begin{pmatrix} 1+D & D & 1 \\ D^2 & 1 & 1+D+D^2 \end{pmatrix}.$$

a. Determine the rate of the code.

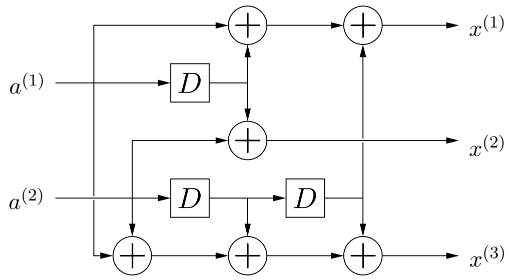
b. Draw the corresponding encoder.

c. Draw one section of the trellis of the encoder. Indicate both input and output on all edges.

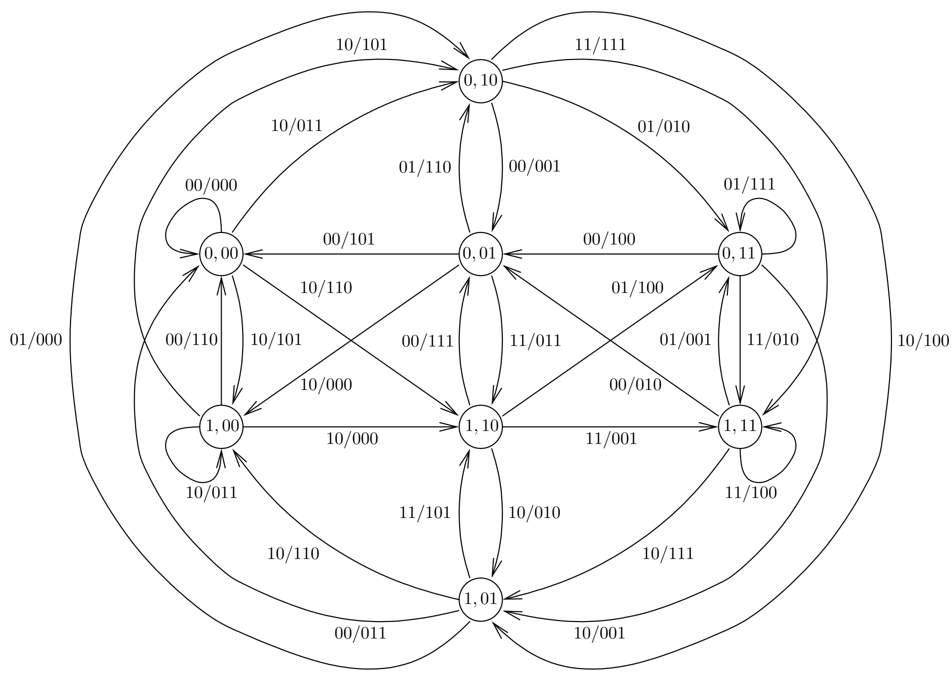
Solution:

a. The generator matrix is a 2×3 matrix. Thus, the rate is $2/3$.

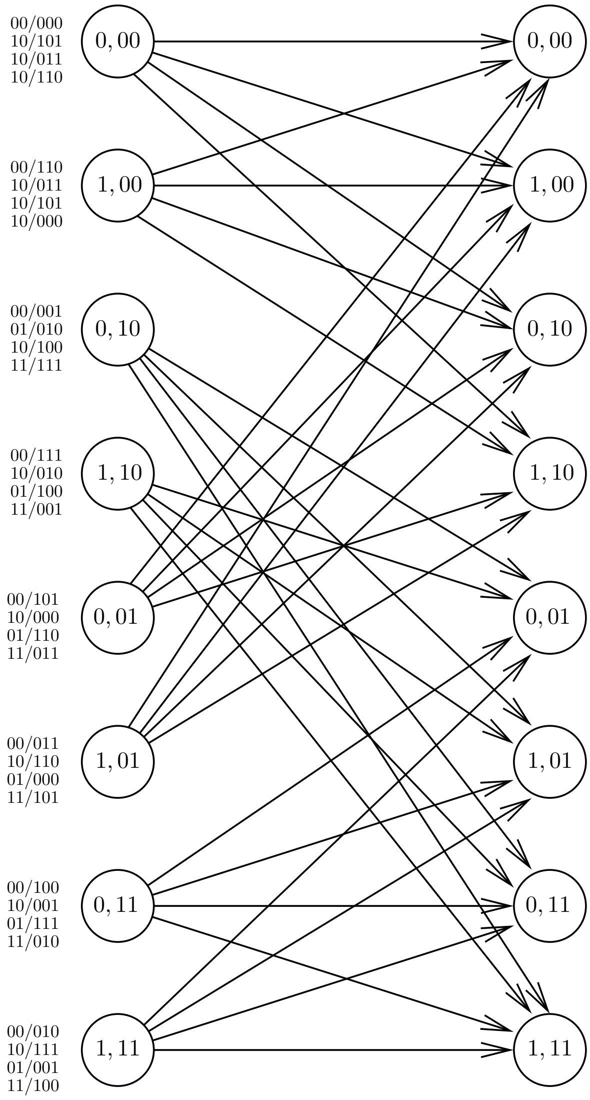
b. The corresponding encoder is given below.



c. First we draw the state diagram of the encoder.

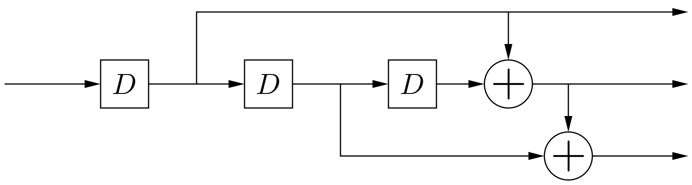


And a section of the trellis:



The labels of the edges are written to the left of its origin state, since there is not enough space near the edges themselves.

2.8 Consider the following convolutional encoder.

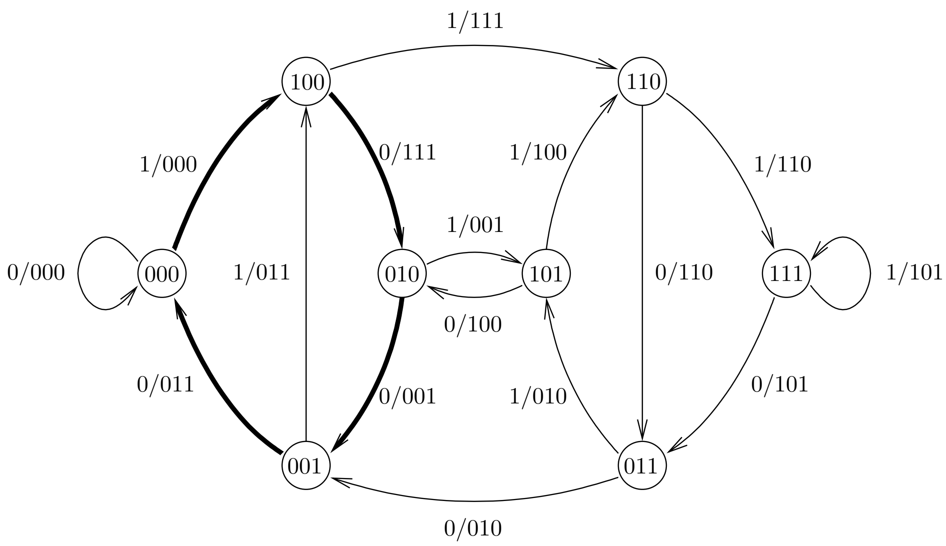


a. Draw one section of the trellis of this encoder. Indicate both input and output on all edges.

b. The sequence $a = a_0, a_1, a_2, a_3, 0, 0, \ldots$ has been encoded by this encoder. The received sequence is $(111, 111, 111, 111, 111, 111)$. Use the Viterbi algorithm to determine the most likely sent sequence.

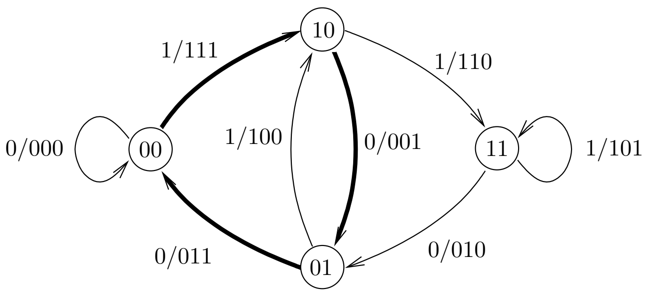
Solution:

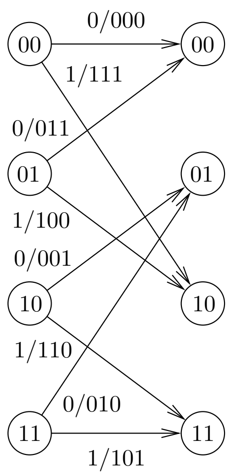
a. A complete state diagram is given below. The edges are labelled $a_i/x_i^{(1)}x_i^{(2)}x_i^{(3)}$.



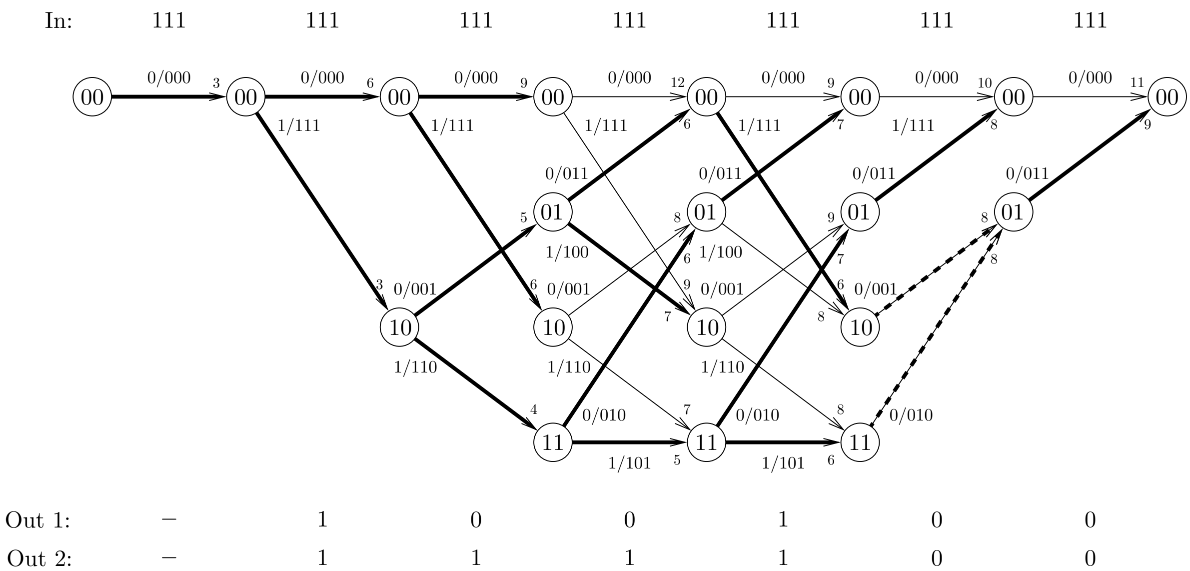
However, we can notice that the input at time instance i does not affect the output at that time instance. Instead the first delay element in the encoder only delays the signal, and the input at time instance i starts affecting the output at time instance $i + 1$.

Below is a state diagram only including the last two delay elements of the encoder, and a section of the corresponding trellis. The edges are here labelled $a_{i-1}/x_i^{(1)}x_i^{(2)}x_i^{(3)}$.





b. The complete trellis of the given situation is given below. We run the Viterbi algorithm:



The bold edges are the survivors up to that point. Note that there is a draw between two alternatives in the second last interval. At that point we can choose any of the two alternatives. Both are equally good. Those two alternatives are in the end the two best alternatives. The decoder can output any of those two, which correspond to the two sequences indicated above, i.e. (a_0, a_1, a_2, a_3) is either $(1, 0, 0, 1)$ or $(1, 1, 1, 1)$.