ELG 5372 Error Control Coding

Lecture 21: Polynomial, Rational and Systematic Encoders and Introduction to Decoding of Convolutional Codes.

Université d'Ottawa | University of Ottawa

uOttawa

L'Université canadienne
Canada's university

uOttawa.ca

Polynomial and Rational Encoders

- Every rational encoder has an equivalent basic encoder.
  - This implies that it is sufficient to use only feedforward encoders to represent every code
  - However, there may not be an equivalent basic systematic code.
  - If a systematic code is desired (for example, Turbo codes), it may be necessary to use a rational encoder.

Invariant Factor Decomposition

- Let **G**(*D*) be a *k*×*n* polynomial matrix.
- **G**(*D*) can be written as **A**(*D*)Γ(*D*)**B**(*D*), where **A**(*D*) is a *k*×*k* polynomial matrix and **B**(*D*) is an *n*×*n* polynomial matrix where det(**A**(*D*)) = det(**B**(*D*)) = 1 and Γ(*D*) is the *k*×*n* diagonal matrix given below:

$$\Gamma(D) = \begin{bmatrix} \gamma_1(D) & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \gamma_2(D) & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \gamma_3(D) & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \gamma_k(D) & \cdots & 0 \end{bmatrix}$$

Invariant Factor Decomposition 2

- The nonzero elements of $\Gamma(D)$ are polynomials called invariant factors of $\mathbf{G}(D)$.
- Then invariant factors satisfy the property that $\gamma_i(D)$ divides $\gamma_{i+1}(D)$.
- If $\mathbf{G}(D)$ is rational, $\mathbf{G}(D) = \mathbf{A}(D)\Gamma(D)\mathbf{B}(D)$ is still true, only $\Gamma(D)$ is now rational and takes the form

$$\Gamma(D) = \begin{bmatrix} \frac{\alpha_1(D)}{\beta_1(D)} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{\alpha_2(D)}{\beta_2(D)} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \frac{\alpha_3(D)}{\beta_3(D)} & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \cdots & \frac{\alpha_k(D)}{\beta_k(D)} & \cdots & 0 \end{bmatrix}$$

Invariant Factor Decomposition 3

- Let us express $\mathbf{B}(D)$ as

$$\mathbf{B}(D) = \begin{bmatrix} \mathbf{G}'(D) \\ \mathbf{B}_2(D) \end{bmatrix}$$

- Where $\mathbf{G}'(D)$ is a $k \times n$ polynomial matrix and $\mathbf{B}_2(D)$ is a $(n-k) \times n$ polynomial matrix.
- Since the last $(n-k)$ columns of $\Gamma(D)$ are zero, $\Gamma(D)\mathbf{B}(D) = \Gamma'(D)\mathbf{G}'(D)$

Invariant Factor Decomposition 4

- Where $\Gamma'(D)$ is given by

$$\Gamma'(D) = \begin{bmatrix} \frac{\alpha_1(D)}{\beta_1(D)} & 0 & 0 & \cdots & 0 \\ 0 & \frac{\alpha_2(D)}{\beta_2(D)} & 0 & \cdots & 0 \\ 0 & 0 & \frac{\alpha_3(D)}{\beta_3(D)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{\alpha_k(D)}{\beta_k(D)} \end{bmatrix}$$

Invariant Factor Decomposition 5

- Therefore, invariant factor decomposition states that for rational **G**(*D*), it can be expressed as **G**(*D*)=**A**(*D*)Γ'(*D*)**G**'(*D*).
- Since **A**(*D*)Γ'(D) is non singular, **G**(*D*) and **G**'(*D*) are equivalent encoders. Since **B**(*D*) is polynomial, so is **G**'(*D*).
- Also, since det(**B**(*D*)) = 1, the right inverse of **B**(*D*) is polynomial. Since **G**'(*D*) is part of **B**(*D*), it must also have a polynomial inverse. Thus **G**'(*D*) is a basic encoder.
- **Every rational encoder has an equivalent basic transfer function matrix**

Constraint length and minimal encoders

- Let $\mathbf{G}(D)$ be a basic encoder.
- Let $v_i = \max_j \deg(g_{ij}(D))$ denote the maximum degree of the polynomials in row i of $\mathbf{G}(D)$.
- The constraint length $v = v_1+v_2+\ldots+v_k$. This represents the number of memory elements required by the encoder.
- A minimal basic encoder is a basic encoder that has the smallest constraint length among all equivalent basic encoders.
- We are interested in equivalent basic encoders as they require the least amount of hardware and have the smallest number of states.

Encoder matrix decomposition

- In general a basic encoder matrix $\mathbf{G}(D)$ can be written as:

$$\mathbf{G}(D) = \begin{bmatrix} D^{\nu_1} & 0 & \cdots & 0 \\ 0 & D^{\nu_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D^{\nu_k} \end{bmatrix} \mathbf{G}_h + \tilde{\mathbf{G}}(D) = \mathbf{\Lambda}(D)\mathbf{G}_h + \tilde{\mathbf{G}}(D)$$

- Where $\mathbf{G}_h$ is a binary matrix which contains a 1 indicating the position in each row where the highest degree term D^{vi} occurs.

Example

$$\mathbf{G} = \begin{bmatrix} 1 & D^2 & D \\ D & 1 & 0 \end{bmatrix} = \begin{bmatrix} D^2 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & D \\ 0 & 1 & 0 \end{bmatrix}$$

Basic Encoder Theorem 1 (BET1)

- Let $\mathbf{G}(D)$ be a $k \times n$ basic encoding matrix, then G(D) is a minimal basic encoding matrix if
  - The maximum degree of the $k \times k$ subdeterminants of $\mathbf{G}(D)$ is equal to v. (1)
  - $\mathbf{G}_h$ is full rank. (2)
- Statements (1) and (2) are equivalent.
- See proof on pages 466-467 in text.

Examples

$$\mathbf{G}_1 = \begin{bmatrix} 1 & D^2 & D \\ D & 1 & 0 \end{bmatrix} = \begin{bmatrix} D^2 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & D \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 1+D^2+D^3 & D+D^2 \\ 0 & D+D^3 & D^2 \end{bmatrix} = \begin{bmatrix} D^3 & 0 \\ 0 & D^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1+D^2 & D+D^2 \\ 0 & D & D^2 \end{bmatrix}$$

Producing equivalent basic encoder of reduced constraint length

- Let **G** be a basic encoder.
- If $\mathbf{G}_h(D)$ is rank deficient, then G(D) is not a minimal basic code.
- Let $\mathbf{g}_i$ be the row of greatest degree.
  - Then $\mathbf{g}_i = \mathbf{g}_i + \sum_{j \neq i}^{k} D^{v_i - v_d} \mathbf{g}_\mathbf{j}$
  - Determine row of maximum degree. If it is still $\mathbf{g}_i$, stop. Otherwise repeat above.
- See page 466-467 for proof.

Example cont'd

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 1+D^2+D^3 & D+D^2 \\ 0 & D+D^3 & D^2 \end{bmatrix}$$

$$\mathbf{g}_1 = \begin{bmatrix} 1 & 1+D^2+D^3 & D+D^2 \end{bmatrix}, \mathbf{g}_2 = \begin{bmatrix} 0 & D+D^3 & D^2 \end{bmatrix}$$

Both have degree 3. Let $\mathbf{g}_1$ = [1 1+D^2+D^3 D+D^2]+[0 D+D^3 D^2] = [1 1+D+D^2 D], which now has degree 2.

Let $\mathbf{g}_2$ = [0 D+D^3 D^2]+D[1 1+D+D^2 D] = [D D^2 0]

$$\mathbf{G}_3 = \begin{bmatrix} 1 & 1+D+D^2 & D \\ D & D^2 & 0 \end{bmatrix}$$

$$\mathbf{g}_1 = \begin{bmatrix} 1 & 1+D+D^2 & D \end{bmatrix}, \mathbf{g}_2 = \begin{bmatrix} D & D^2 & 0 \end{bmatrix}$$

$$\mathbf{G}_4 = \begin{bmatrix} 1+D & 1+D & D \\ D^2 & D & D^2 \end{bmatrix} = \begin{bmatrix} 1+D & D \\ D^2 & 1+D+D^2 \end{bmatrix} \mathbf{G}_2$$

Decoding convolutional codes

- Several algorithms exist for the decoding of convolutional codes.
- Most common is Viterbi algorithm.
- Variation is the soft output Viterbi Algorithm (SOVA) which not only provides the decoded output but a reliability measure of each decoded symbol.
- Suboptimal decoding algorithms exist. These are used to reduce complexity, especially when the constraint length is large. Stack and Fano algorithms are of particular interest.

Viterbi Algorithm

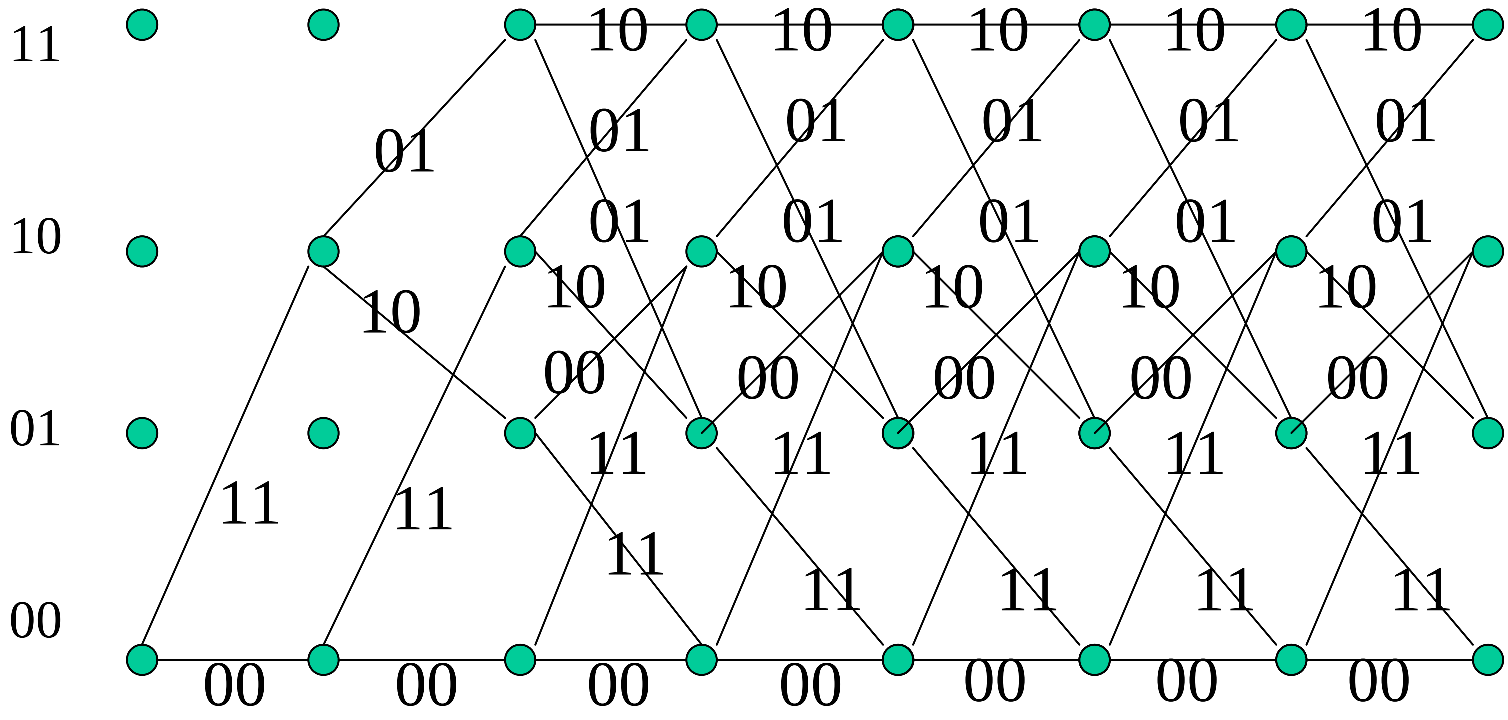
- Originally proposed by Andrew Viterbi.
- Only later was it shown to provide the maximum likely code sequence given the received data.
- It is essentially a shortest path algorithm.

Viterbi algorithm for hard decision decoding

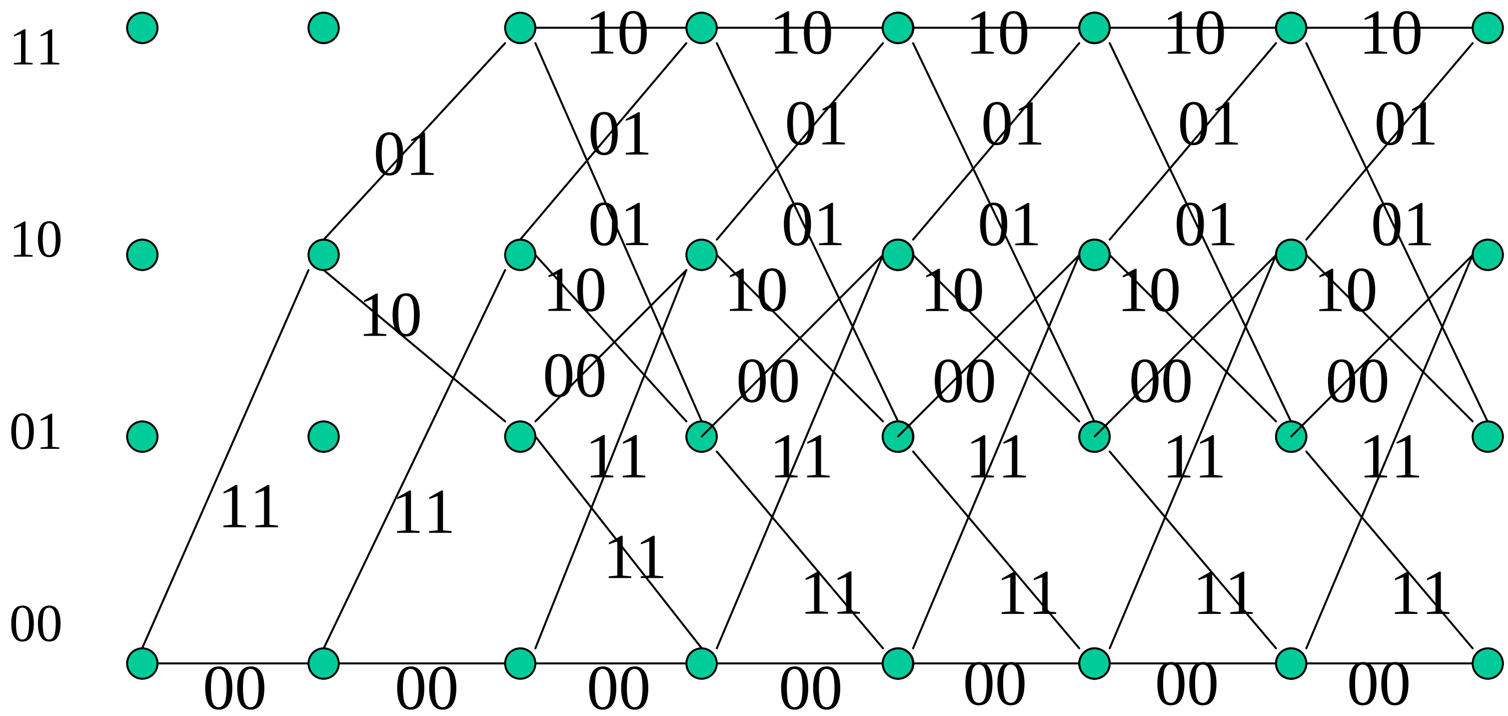
- Received data is “hard” (decisions rather than likelihoods are given to the decoder).
- The algorithm attempts to find the path that produces the code sequences that is closest in terms of Hamming distance.
- The algorithm uses the trellis diagram introduced in a previous lecture.

Example

- Consider the rate ½ code **G**(D) = [$1+D+D^2$ $1+D^2$].

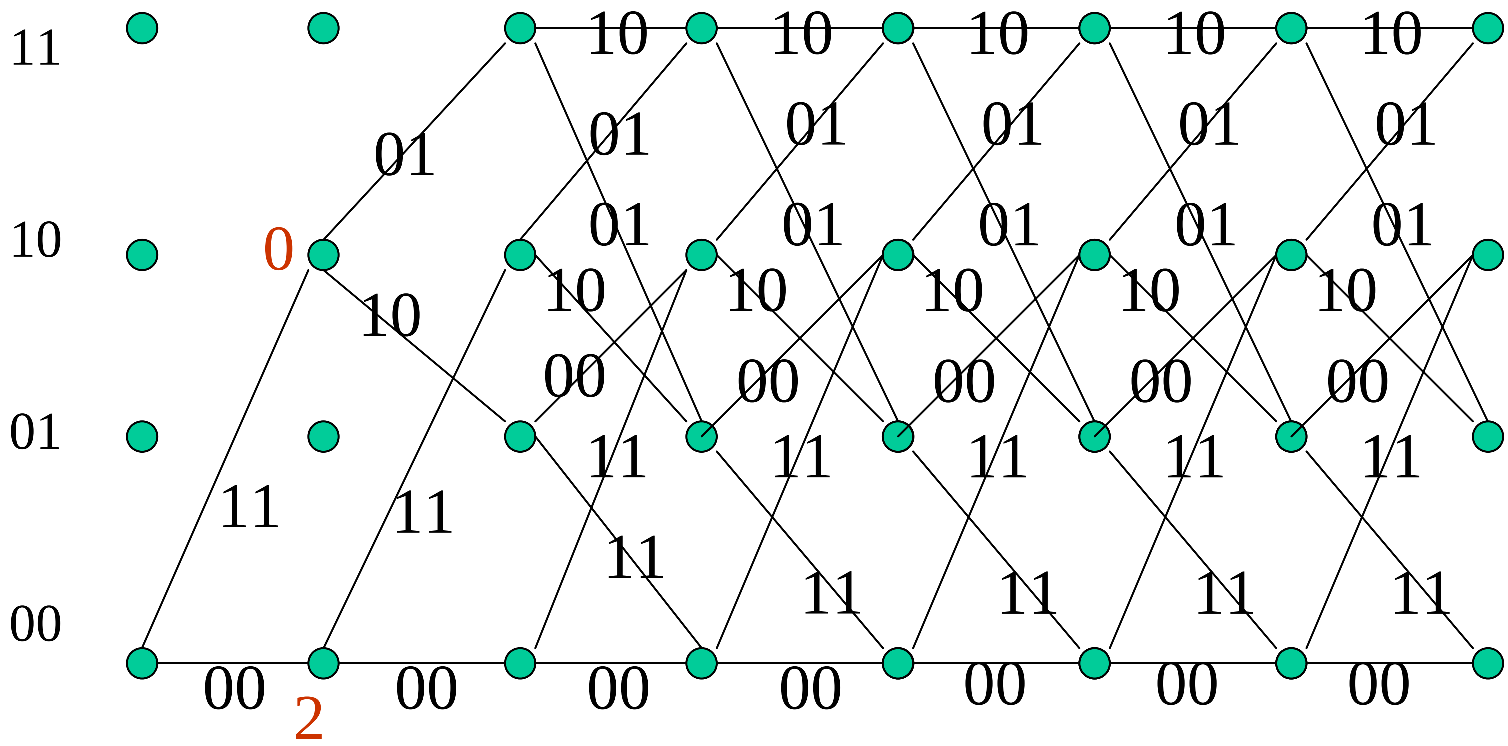


Example



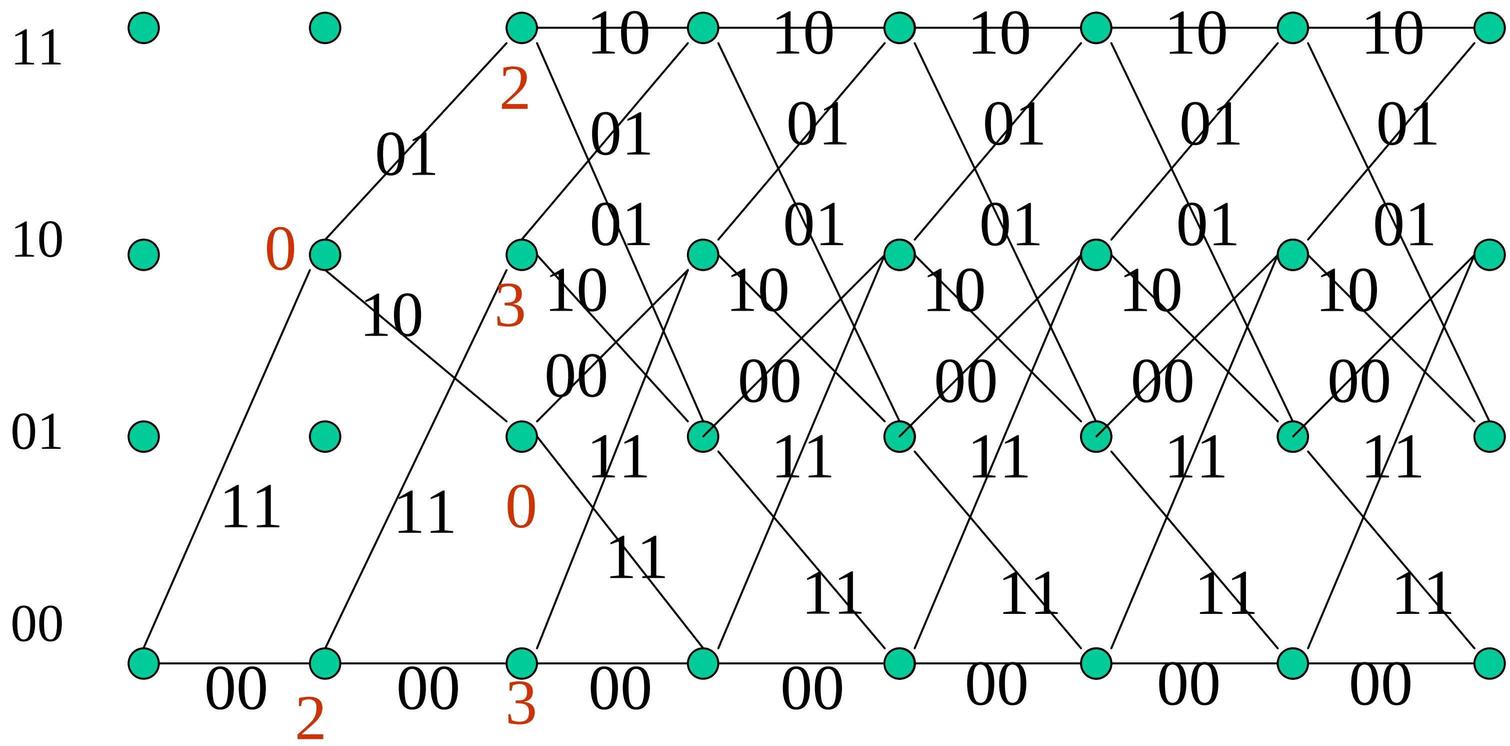
r = 11, 10, 00, 10, 01, 11, 00

Example



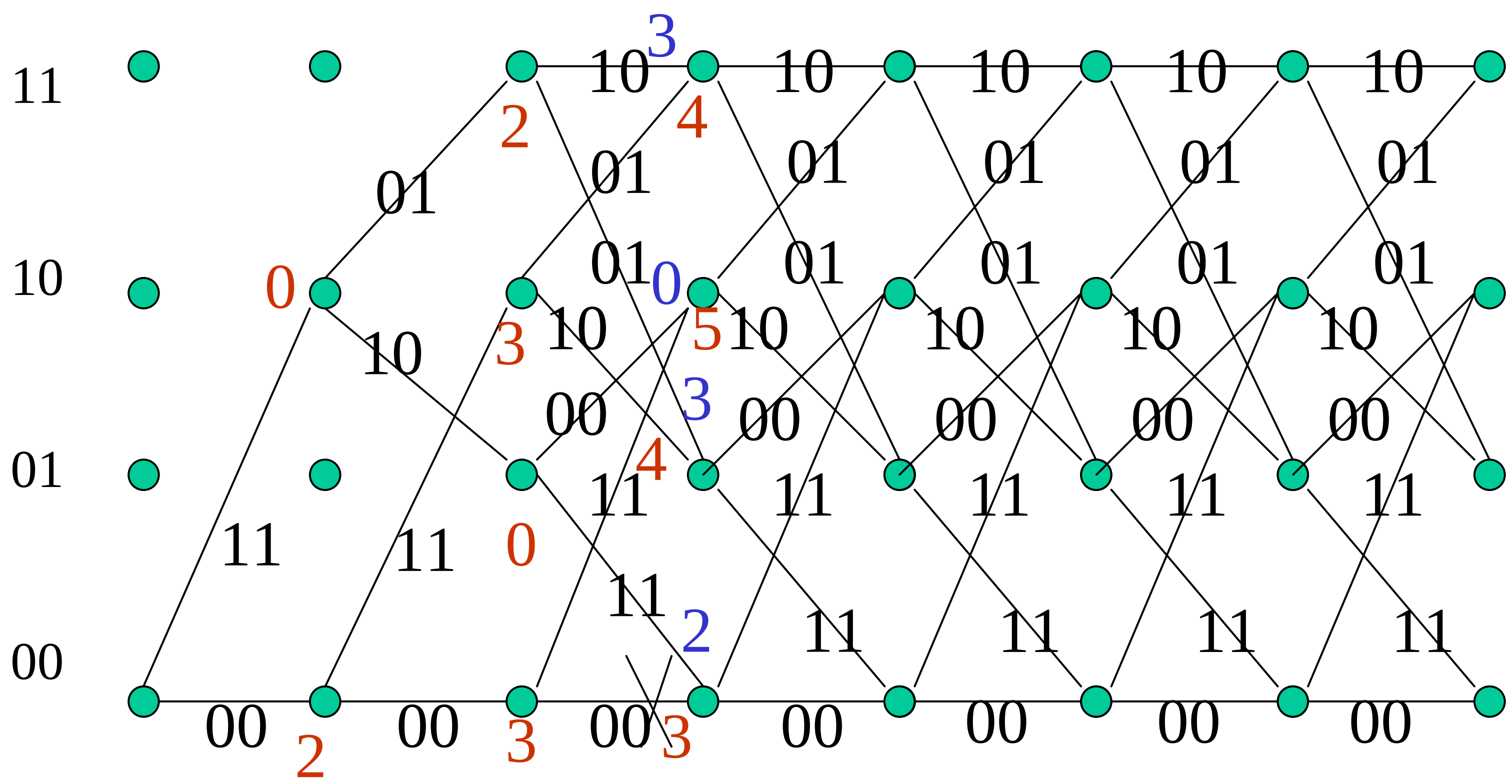
r = 11, 10, 00, 10, 01, 11, 00

Example



r = 11, 10, 00, 10, 01, 11, 00

Example

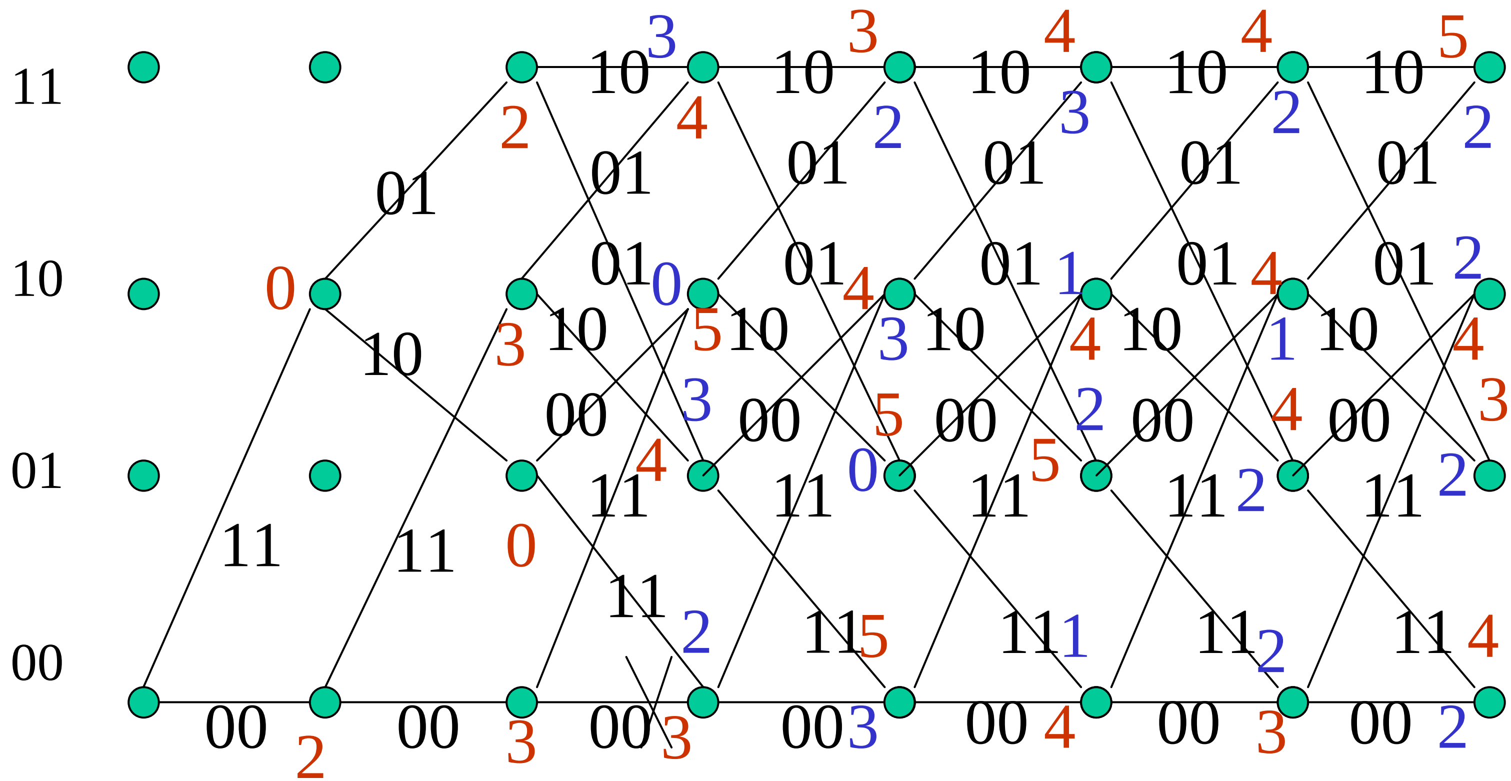


r = 11, 10, 00, 10, 01, 11, 00

Example

If terminated **c** = 11,10, 00,01, 01,11,00

M = 1,0,1,1,0,0,0



r = 11, 10, 00, 10, 01, 11, 00