Nicolas Sendrier

Majeure d'informatique

Introduction à la théorie de l'information

Cours n°8

Codes convolutifs

Codeur convolutif

Définition Un *codeur convolutif* binaire (n, k, m) est un circuit linéaire invariant dans le temps, sans feedback, d'ordre m, à k entrées et n sorties sur $\{0, 1\}$.

Définition Un *code convolutif* est l'ensemble des séquences produites par un codeur convolutif.

Pour toute séquence binaire infinie $\mathbf{u}$ la séquence codée (infinie) correspondante est $\mathbf{v} = \mathbf{u}G$, où G est une matrice binaire infinie

$$G = \begin{pmatrix} G_0 & G_1 & \cdots & G_m & & & \\ & G_0 & G_1 & \cdots & G_m & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & G_0 & G_1 & \cdots & G_m \\ & & & & \ddots & \ddots & \ddots \end{pmatrix}$$

et les G_i sont des matrices binaires $k \times n$.

Codeur convolutif – Description algébrique

Matrice de transfert :

$$G(D) = \sum_{i=1}^{m} G_i D^i = \begin{pmatrix} g_{1,1}(D) & \cdots & g_{1,n}(D) \\ \vdots & \ddots & \vdots \\ g_{k,1}(D) & \cdots & g_{k,n}(D) \end{pmatrix}$$

l'ordre du codeur est défini par $m = \max_{i,j} \deg g_{i,j}(D)$.

Entrée : k séries formelles $\left(u^{(1)}(D), \ldots, u^{(k)}(D)\right)$.

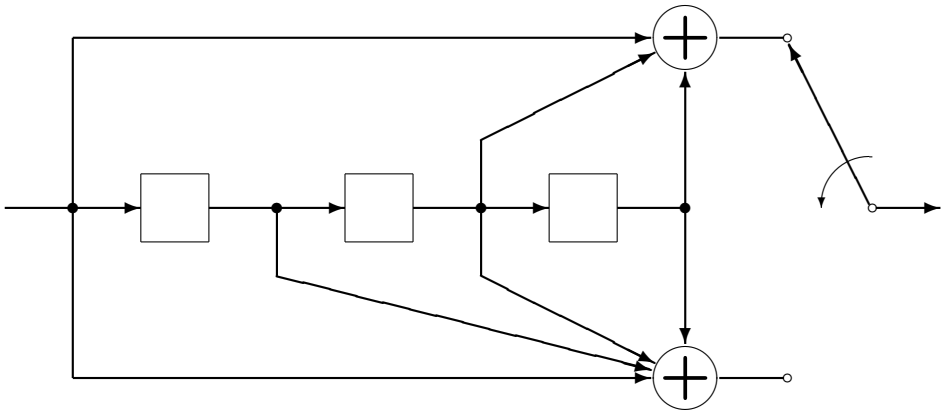
Sortie : n séries formelles $\left(v^{(1)}(D), \ldots, v^{(n)}(D)\right)$.

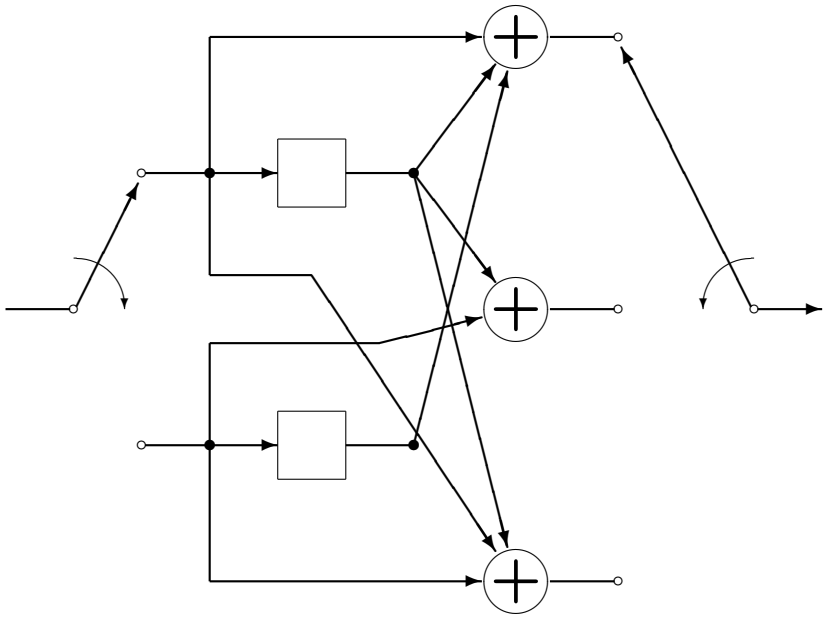
$$\forall j \in \{1, \ldots, n\}, \quad v^{(j)}(D) = \sum_{i=1}^{k} u^{(i)}(D) g_{i,j}(D)$$

Exemples

Codeur convolutif binaire $(2, 1, 3)$

$$G(D) = \left(\begin{array}{cc} 1 + D^2 + D^3 & 1 + D + D^2 + D^3 \end{array} \right)$$





Codeur convolutif binaire $(3, 2, 1)$

$$G(D) = \left(\begin{array}{ccc} 1 + D & D & 1 + D \\ D & 1 & 1 \end{array} \right)$$

Codage

Pour $t < 0$, les $u_t^{(i)}$ sont tous nuls (autrement dit l'état initial du codeur est 0).

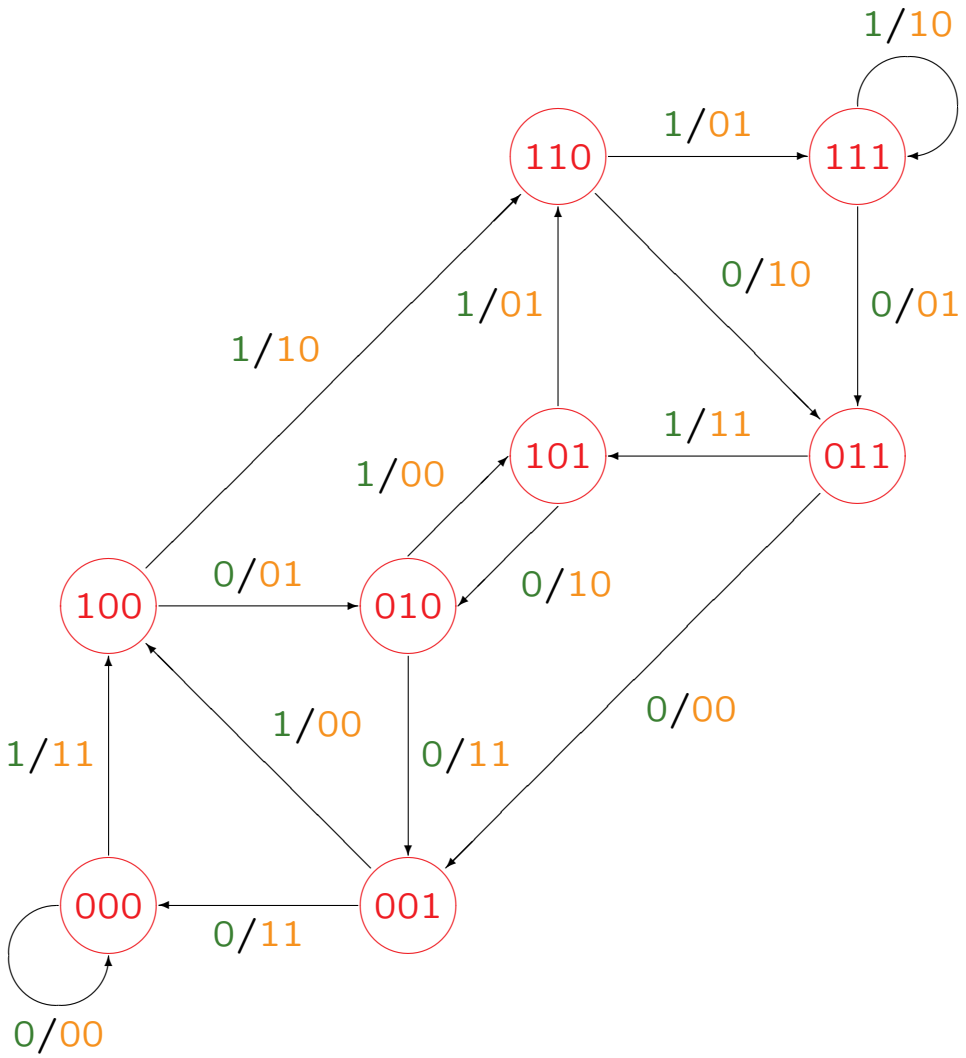
entrées ancien état sorties nouvel état

$$\begin{bmatrix} u_t^{(1)} \\ \vdots \\ u_t^{(k)} \end{bmatrix} \quad \begin{bmatrix} u_{t-1}^{(1)} & \cdots & u_{t-m}^{(1)} \\ \vdots & & \vdots \\ u_{t-1}^{(k)} & \cdots & u_{t-m}^{(k)} \end{bmatrix} \rightarrow \begin{bmatrix} v_t^{(1)} \\ \vdots \\ v_t^{(n)} \end{bmatrix} \quad \begin{bmatrix} u_t^{(1)} & \cdots & u_{t-m+1}^{(1)} \\ \vdots & & \vdots \\ u_t^{(k)} & \cdots & u_{t-m+1}^{(k)} \end{bmatrix}$$

$$\left(v_t^{(1)}, \ldots, v_t^{(n)}\right) = \sum_{l=0}^{m} \left(u_{t-l}^{(1)}, \ldots, u_{t-l}^{(k)}\right) G_l$$

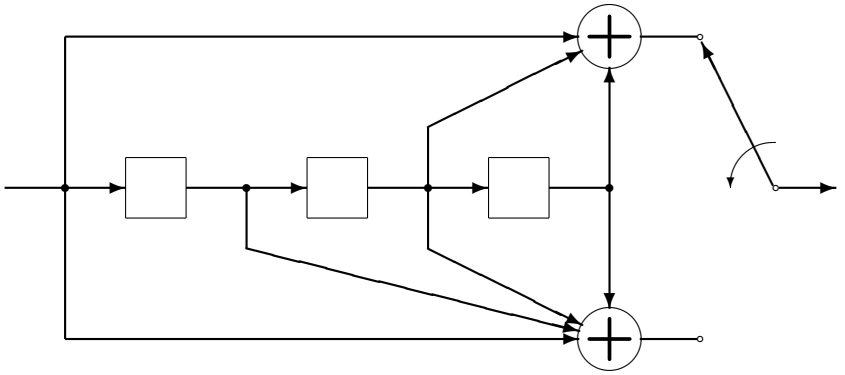
$$\left(v_t^{(1)}, \ldots, v_t^{(n)}\right) = \left(u_t^{(1)}, \ldots, u_t^{(k)}, \ldots, u_{t-m}^{(1)}, \ldots, u_{t-m}^{(k)}\right) \begin{pmatrix} G_0 \\ \vdots \\ G_m \end{pmatrix}$$

Diagramme d'état – Exemple pour un codeur $(2, 1, 3)$



À chaque état du codeur correspond un sommet du graphe.

Chaque arête du graphe représente une transition possible d'un état à un autre avec comme étiquette l'entrée/sortie correspondante.

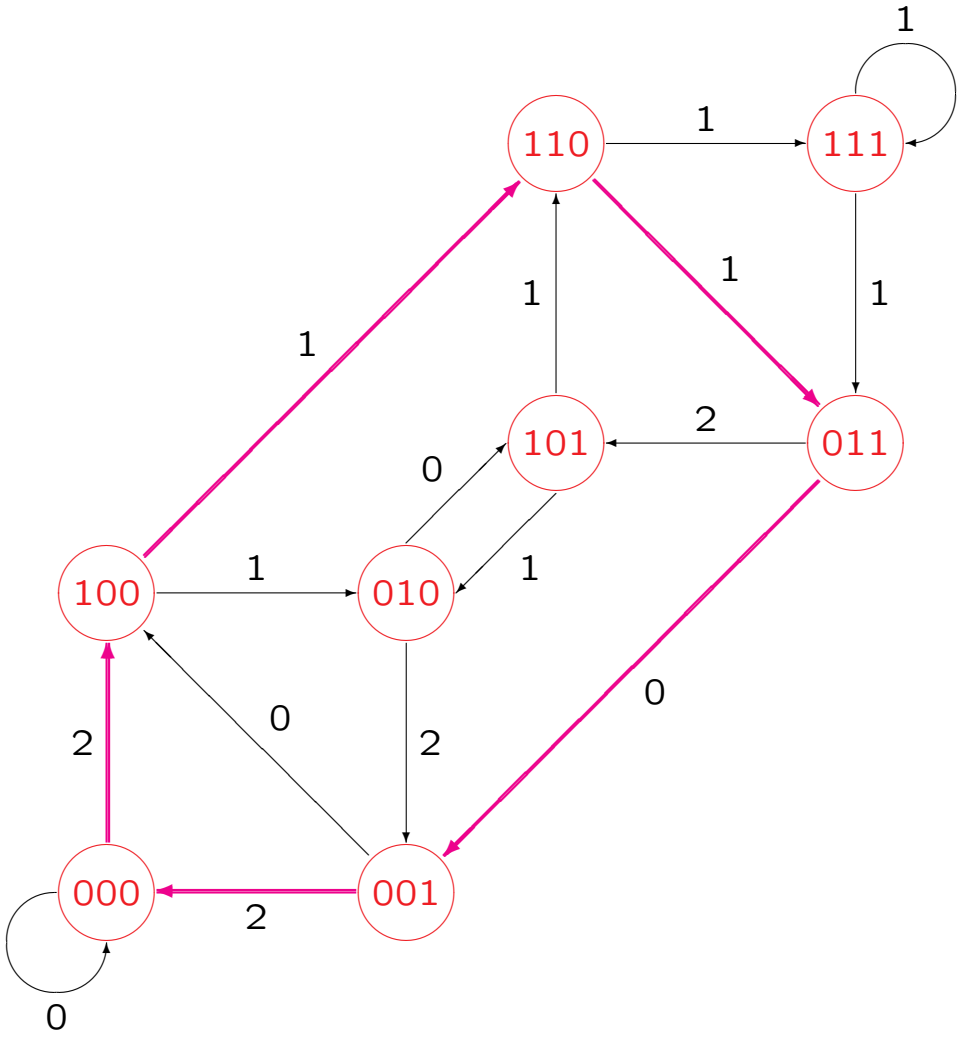


Distance libre

Définition La *distance libre* d'un codeur convolutif est la plus petite distance de Hamming entre deux mots de code distincts.

Comme les codeurs convolutifs sont linéaires, la distance libre est aussi égale au poids de Hamming minimal d'une séquence codée.

Distance libre – Exemple



La distance libre est le poids minimal d'un chemin de 000 à 000 .

Ici la distance libre vaut 6. On notera que le chemin de poids minimal n'est pas le plus court.

Les chemins minimaux déterminent les plus petits motifs d'erreurs conduisant à un mauvais décodage.

Codeur (non) catastrophique

Certains codeurs ont la propriété suivante :

> *Une séquence d'information de poids infini est codée en une séquence de poids fini.*

Par exemple $\mathbf{u} = (1, 1, \dots)$ et

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

La séquence codée sera $\mathbf{v} = (1, 1, 0, 1, 0, 0, \dots)$. Si au cours de la transmission les 2 premiers bits sont modifiés, un décodeur à vraisemblance maximale décodera la séquence en $\tilde{\mathbf{u}} = (0, 0, \dots)$, donc commettra un nombre infini d'erreurs (dans l'information).

On parle alors de codeur catastrophique. Il existe des conditions nécessaires et suffisantes pour qu'un codeur ne soit pas catastrophique.

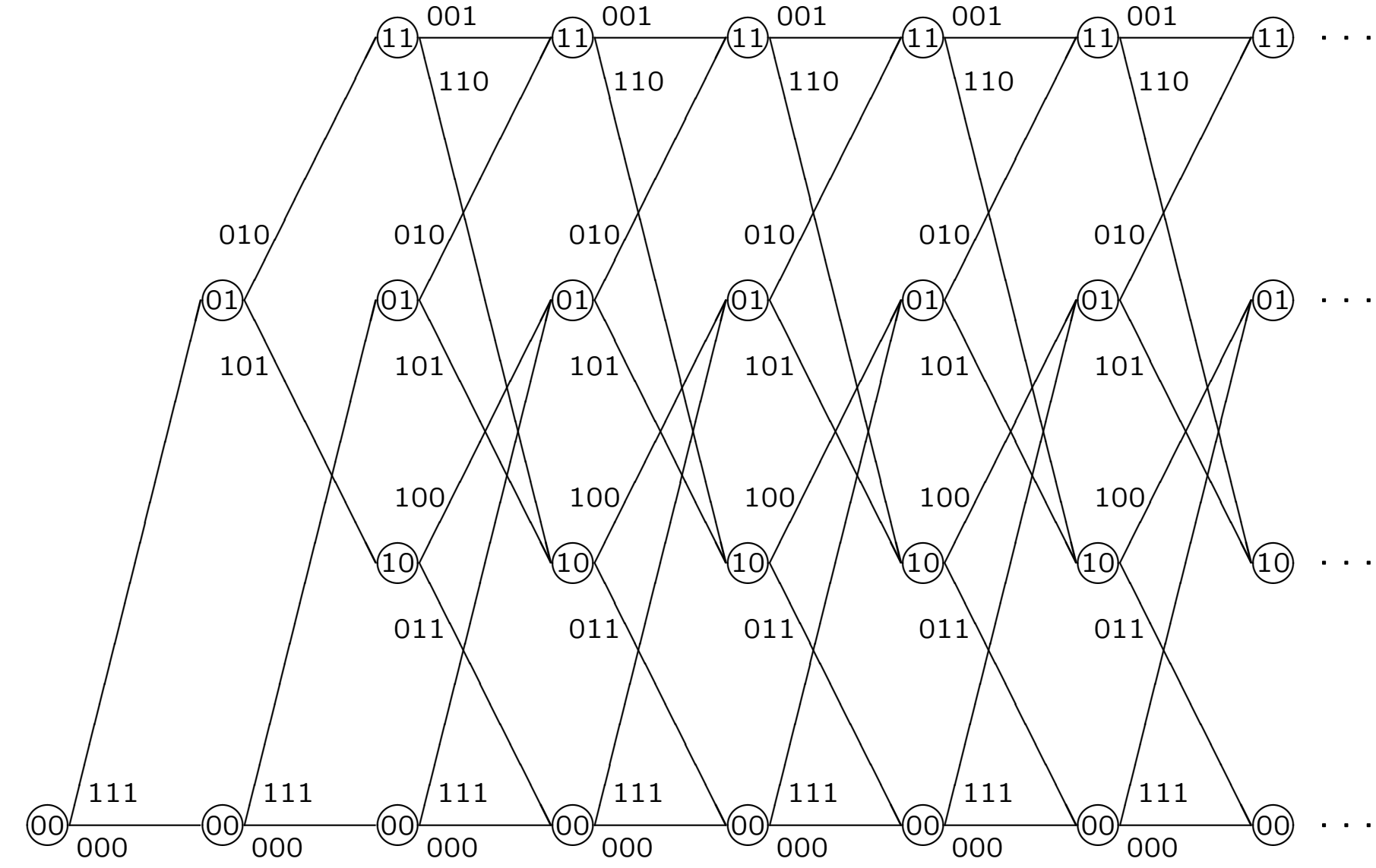
Treillis de codage

Le treillis de codage est obtenu en étendant le diagramme d'état dans le temps.

Il possède les propriétés suivantes :

- Chaque sommet correspond à un état du codeur.
- Chaque arête correspond à une transition et a pour étiquette la sortie correspondante du codeur.
- Tout chemin correspond à une séquence codée, obtenue en concaténant les étiquettes.

Treillis de codage



Algorithme de Viterbi

Soit la séquence binaire reçue

$$\mathbf{y} = \left(\underbrace{y_0^{(1)}, \ldots, y_0^{(n)}}_{y_0}, \underbrace{y_1^{(1)}, \ldots, y_1^{(n)}}_{y_1}, \cdots \right)$$

Nous noterons $e \to f$ l'arète du graphe allant de l'état e à l'état f et $s_{e \to f}$ la sortie du codeur correspondante. On définit pour $t \geq 0$

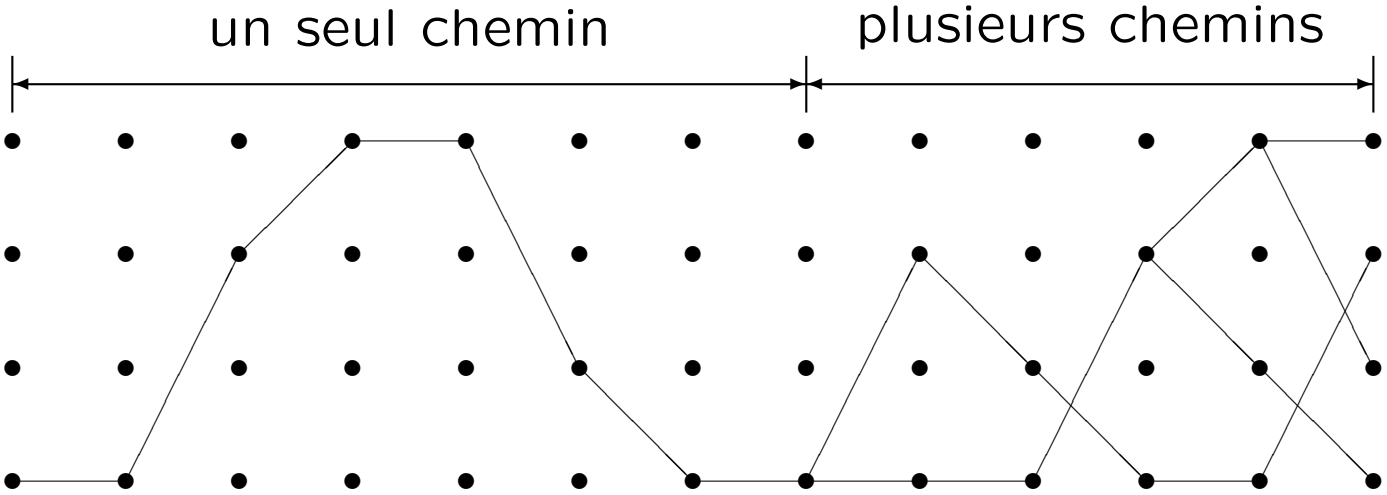
$$\begin{aligned} w_{t+1}(f) &= \min_{e \to f} \left(w_t(e) + d_H(y_t, s_{e \to f}) \right) \\ pred_{t+1}(f) &= e \quad \text{tel que} \quad w_t(e) + d_H(y_t, s_{e \to f}) = w_{t+1}(f) \end{aligned}$$

avec $w_0(0) = 0$ et pour $e \neq 0$, $w_0(e) = +\infty$.

L'entier $w_t(e)$ est la plus petite distance de Hamming entre la séquence reçue et une séquence codée à l'état e au temps t. Les prédécesseurs $pred_t(e)$ permettent de reconstituer pour tout e un chemin réalisant ce minimum, appelé chemin survivant.

Implémentation avec une mémoire bornée – Reconvergence

À chaque étape de l'algorithme de Viterbi, il existe un chemin survivant par état du codeur. Ces chemins finissent par reconverger. La *longueur de reconvergence* est le nombre d'unités de temps pour que tous les chemins survivants soit confondus (5 dans l'exemple ci-dessous).



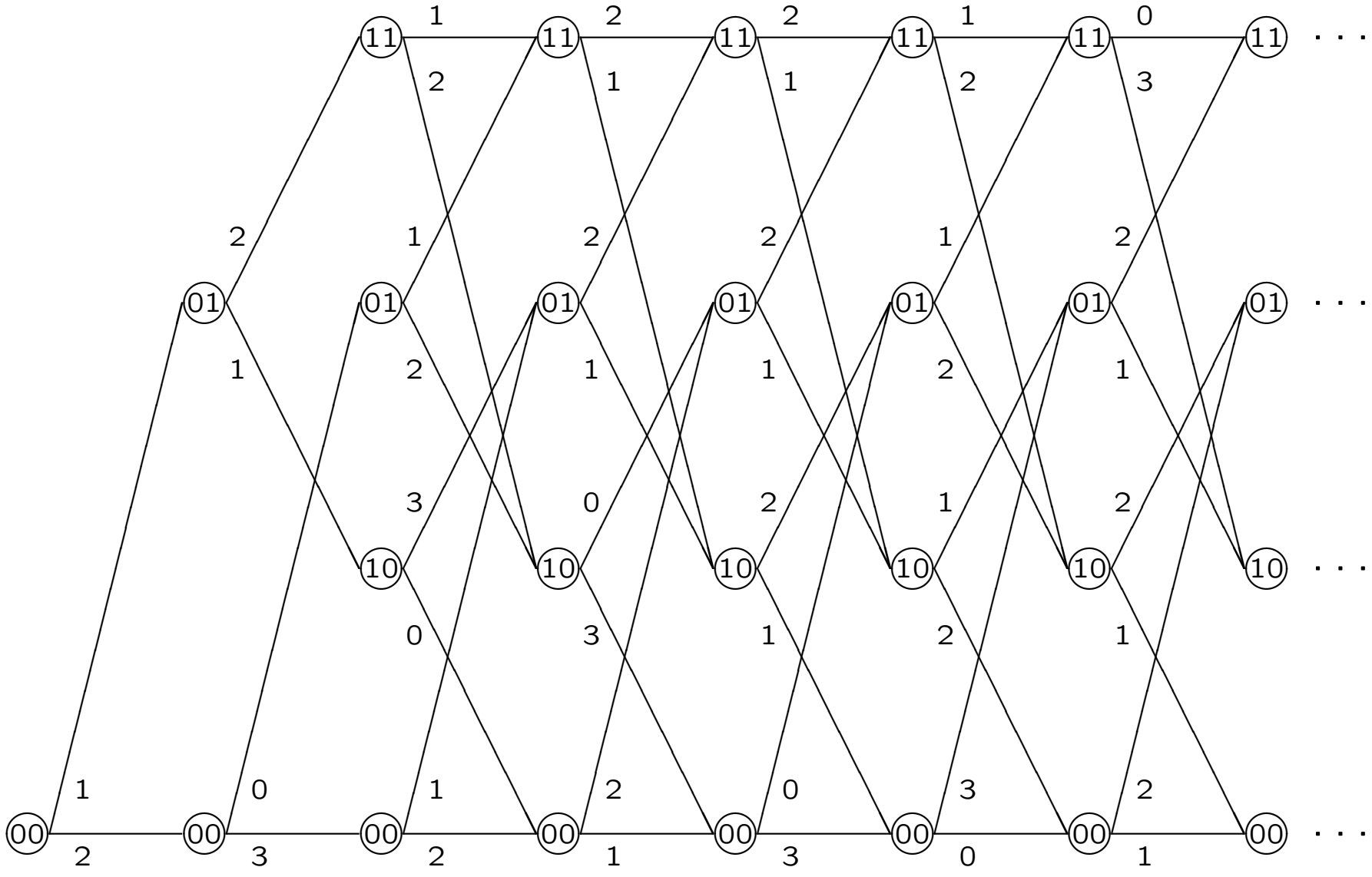
Dans un canal binaire symétrique de probabilité d'erreur > 0 donnée, la longueur de reconvergence n'est pas bornée. Dans les implémentations (en particulier en hardware), on borne la longueur de reconvergence. S'il existe encore plusieurs chemins survivants, on choisit le meilleur.

Décodage séquentiel

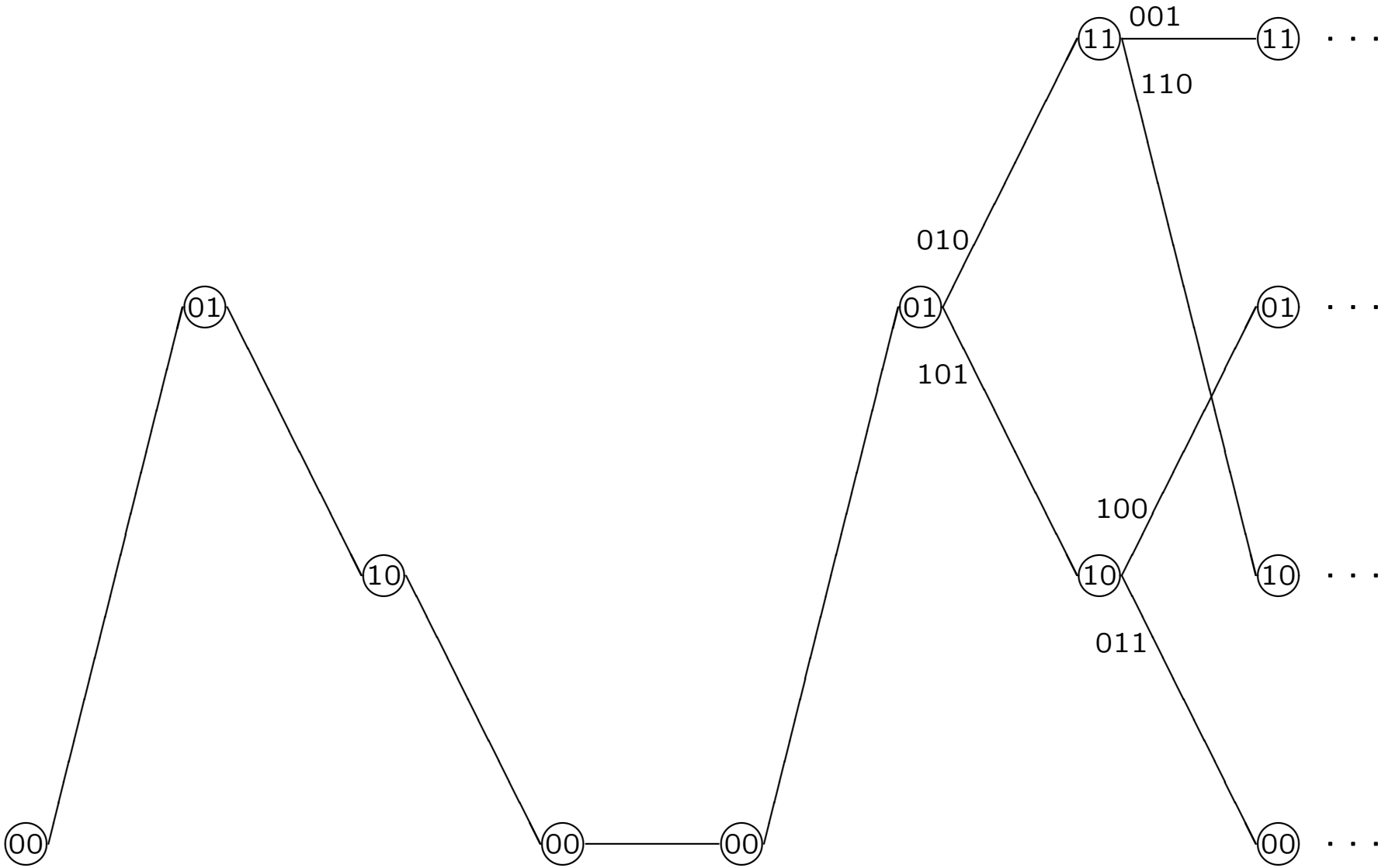
L'autre grande technique de décodage des codes convolutifs est le décodage séquentiel. Comme pour l'agorithme de Viterbi, on cherche un parcours optimal dans un graphe orienté sans cycles.

La principale différence est que l'on parcourt un unique chemin tant qu'il parait bon (*depth first*) plutôt qu'évaluer la qualité de tous les chemins à une profondeur donnée (*breadth first*).

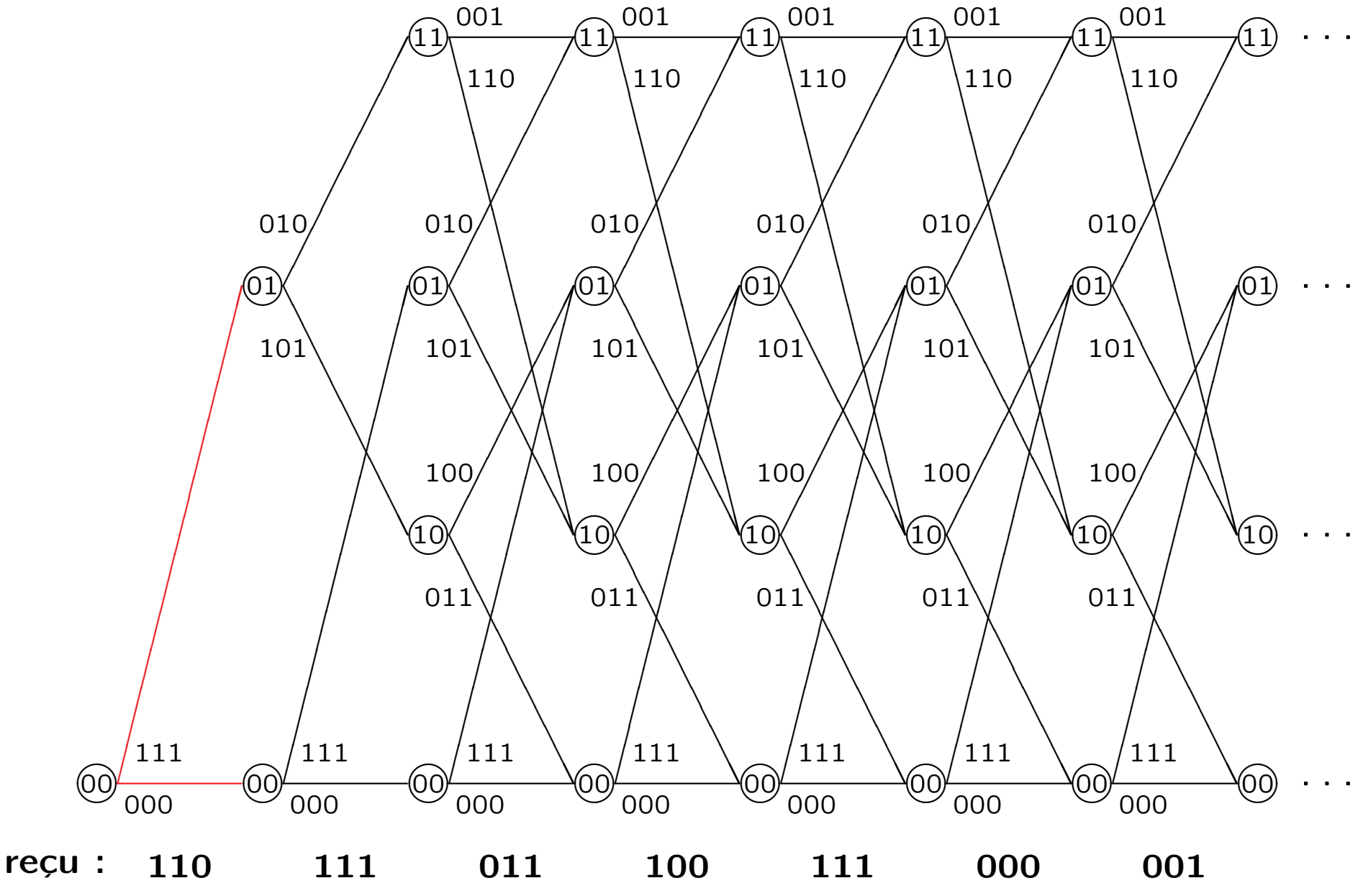
Viterbi exemple



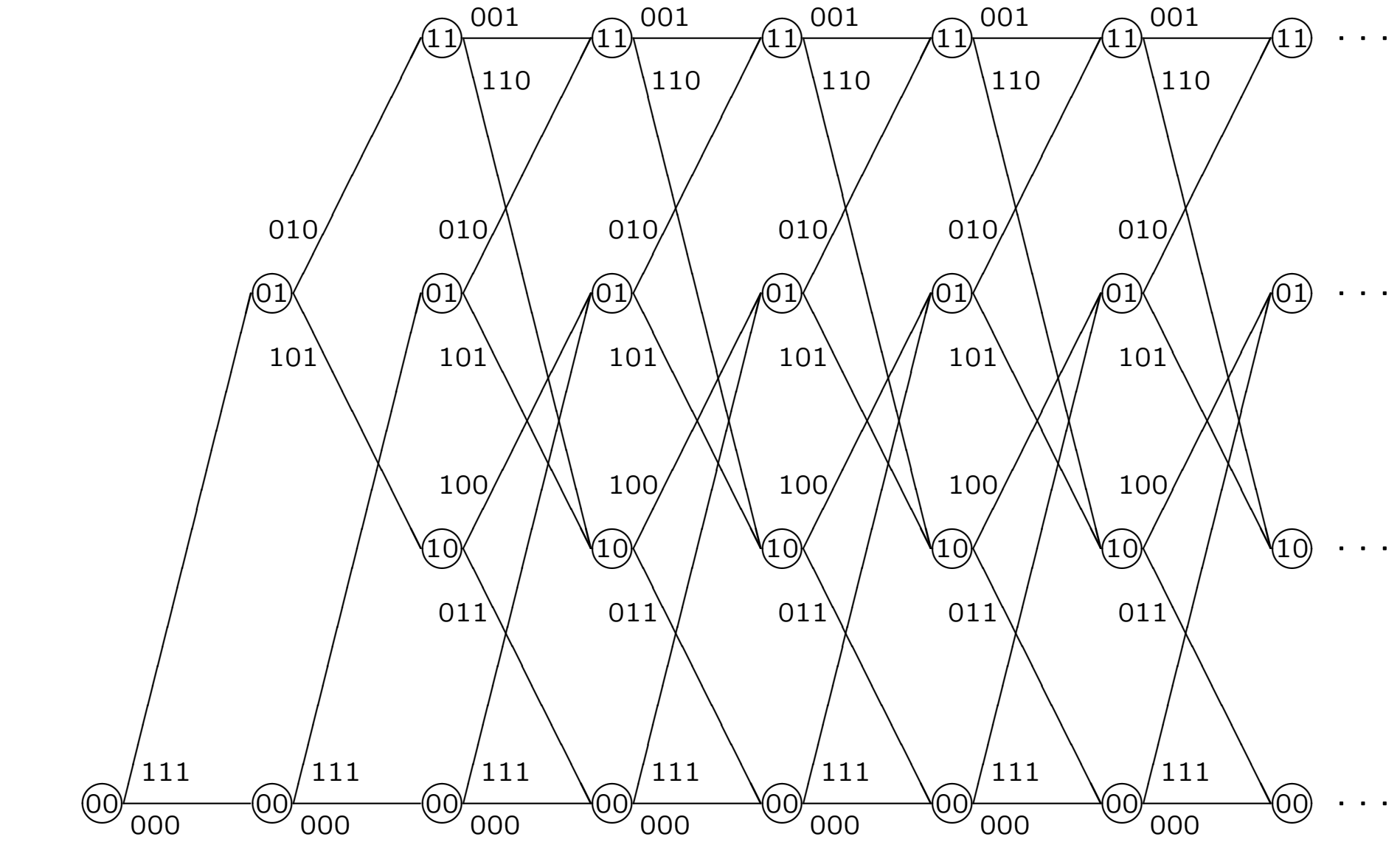
Viterbi exemple – Survivants



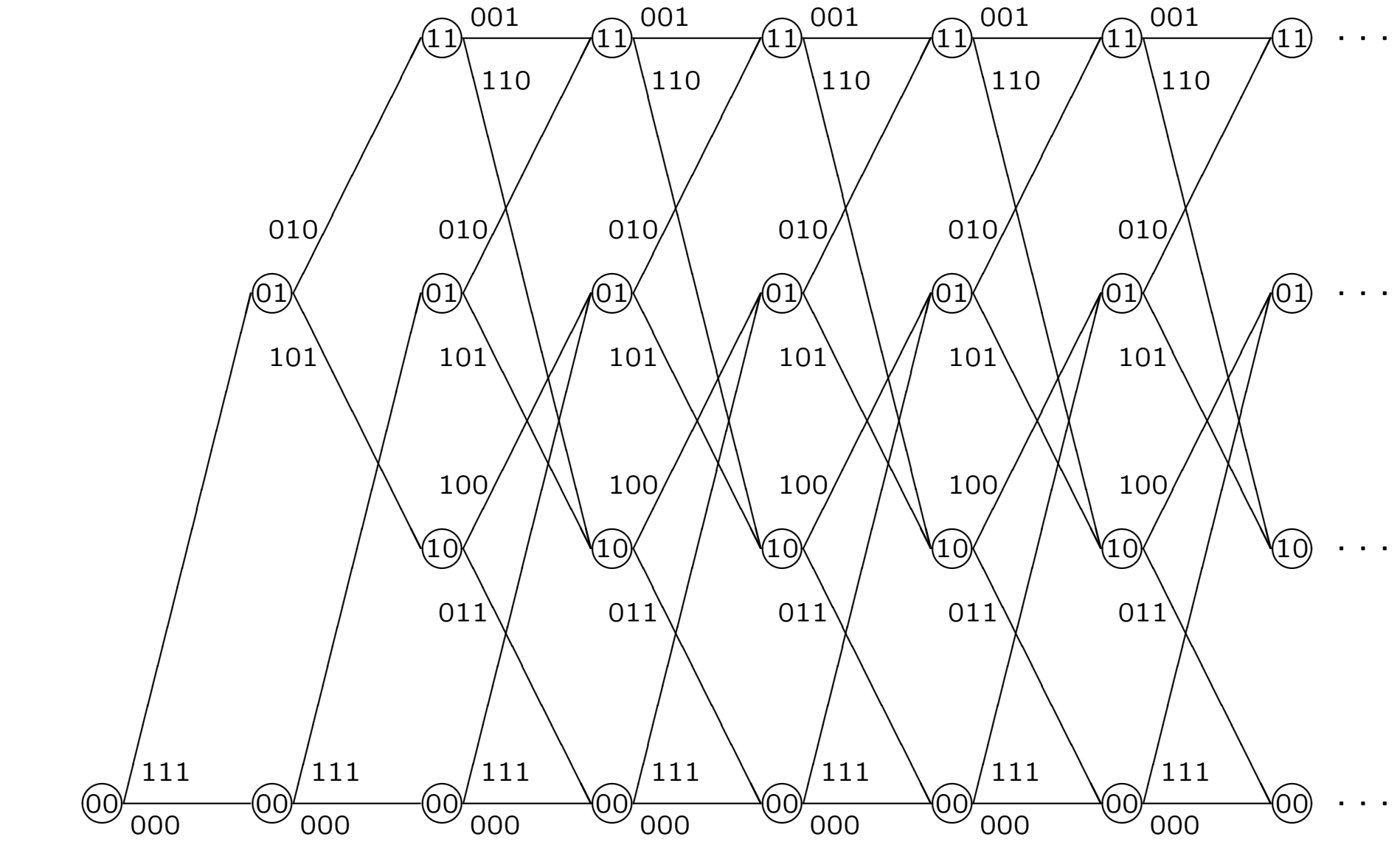
reçu :	**110**	**111**	**011**	**100**	**111**	**000**	**001**
décodé :	**111**	**101**	**011**	**000**	**011**	**???**	**???**



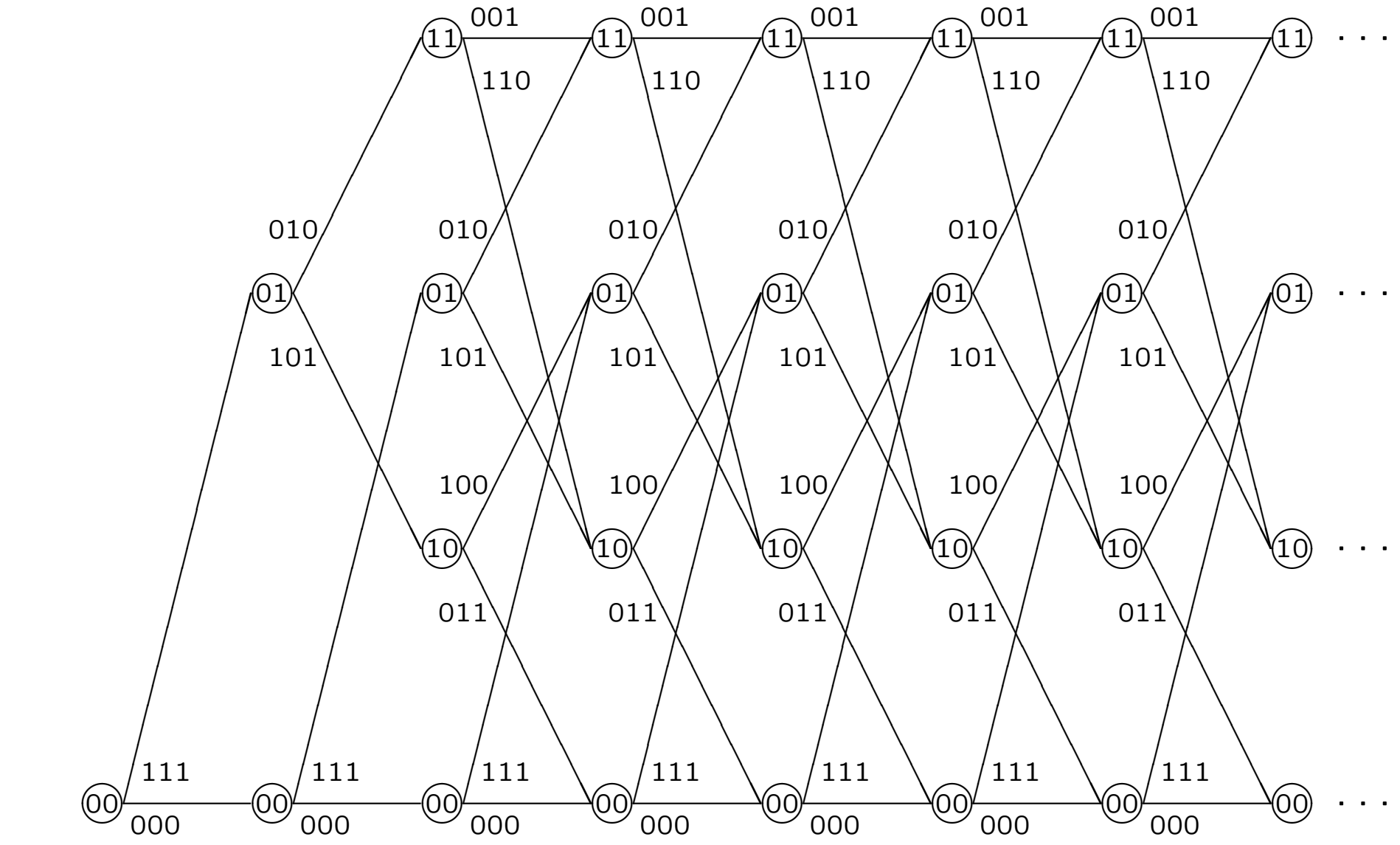
reçu : 110 111 011 100 111 000 001



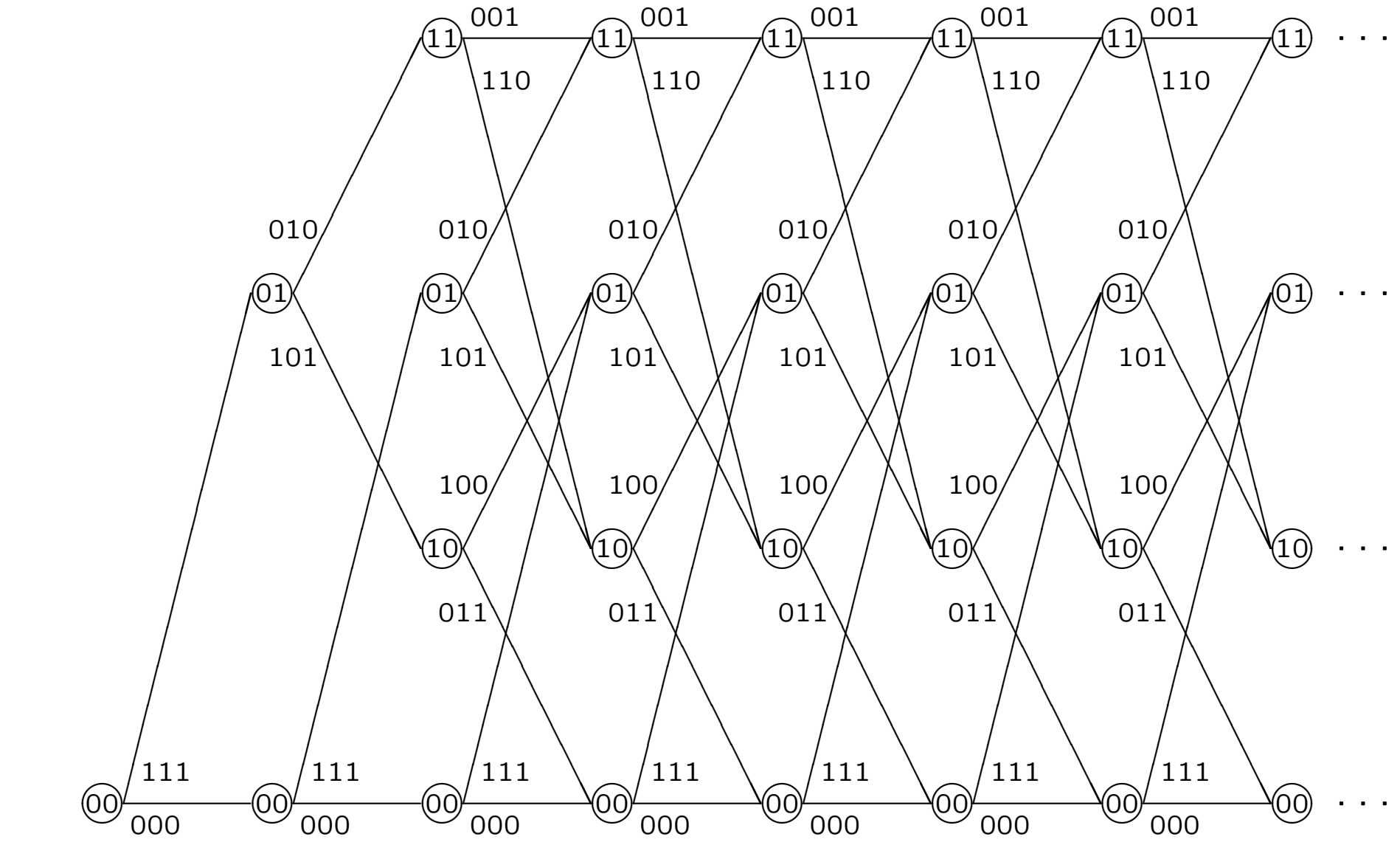
reçu : 110 111 011 100 111 000 001



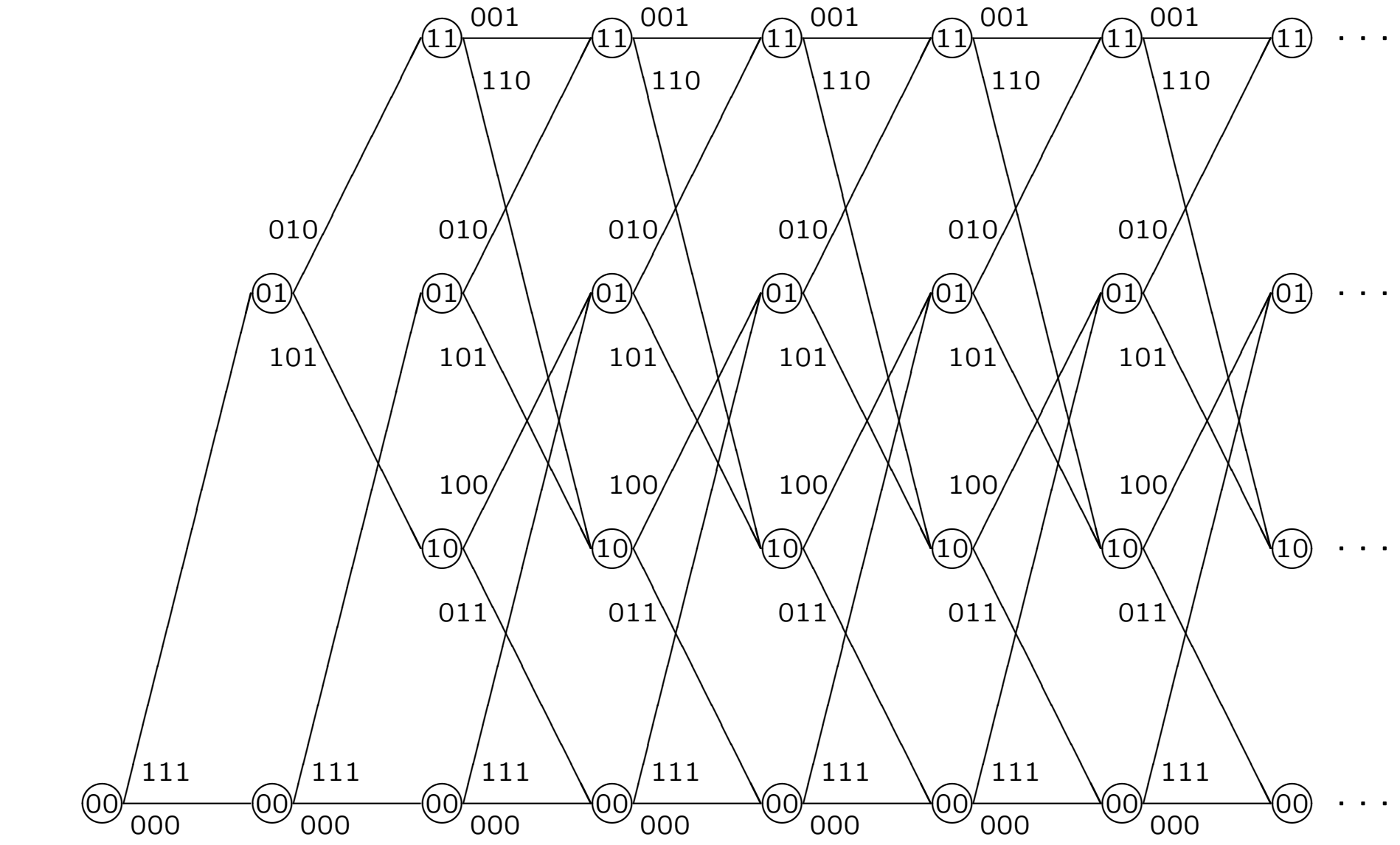
reçu : 110 111 011 100 111 000 001



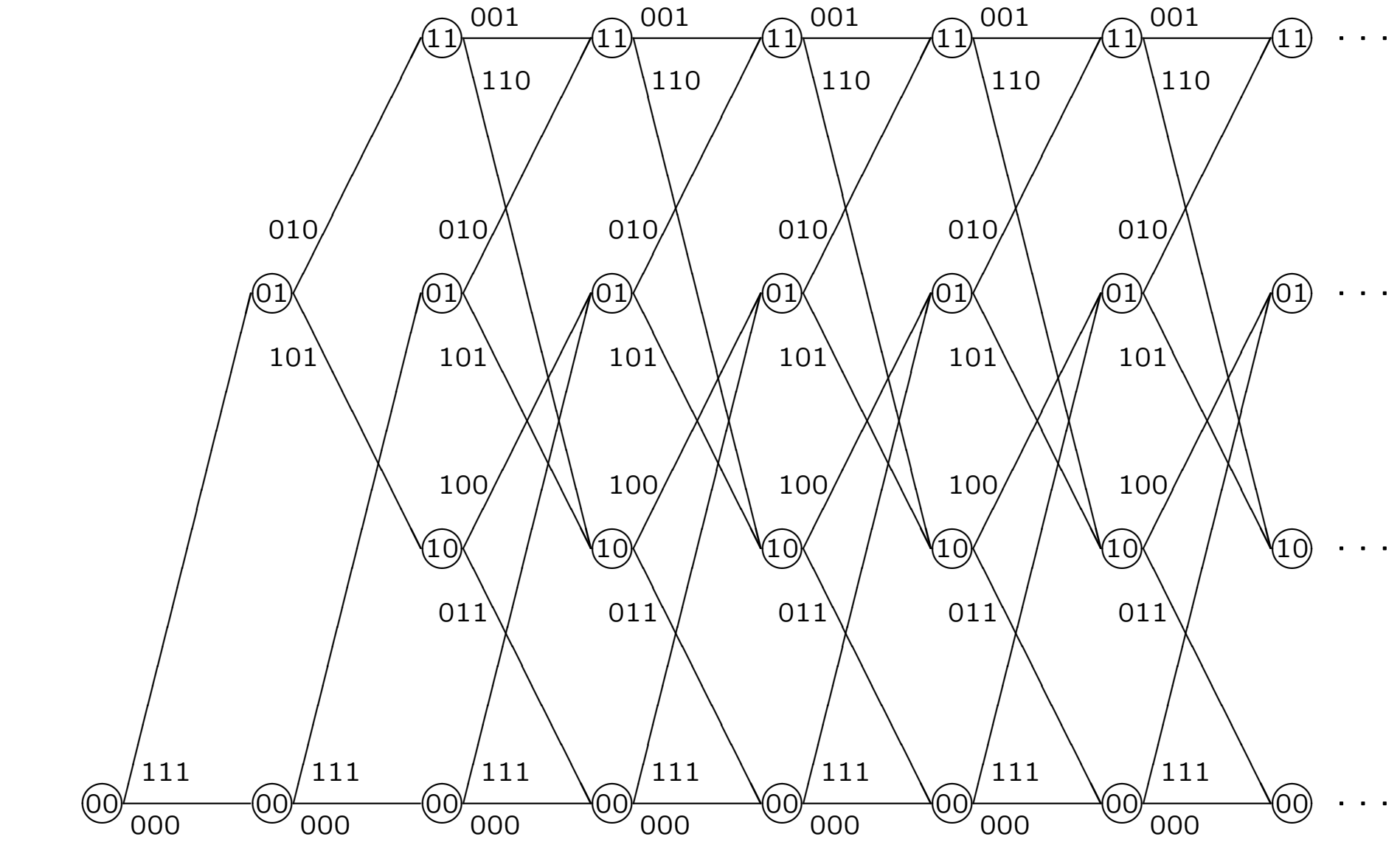
reçu : 110 111 011 100 111 000 001



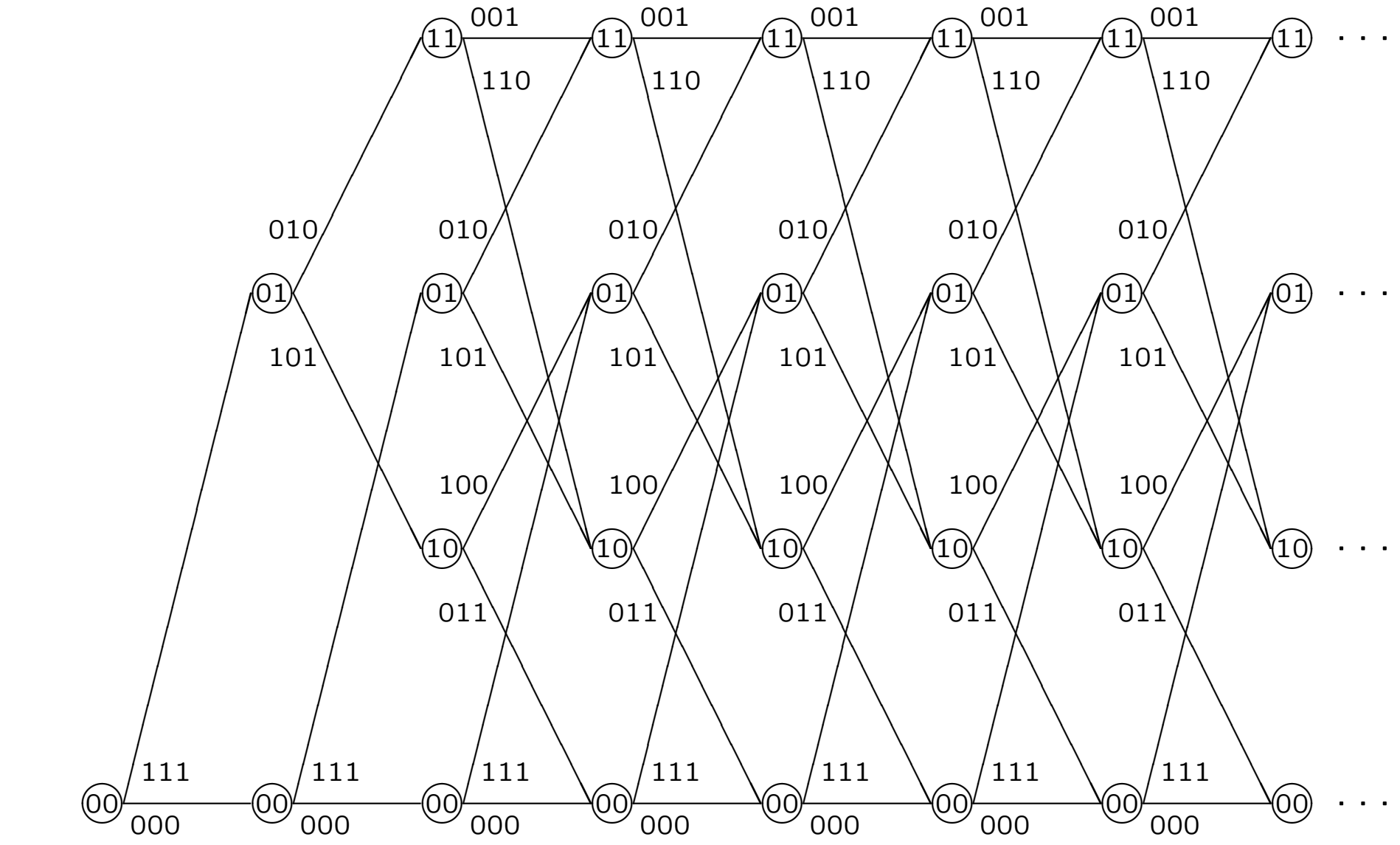
reçu : 110 111 011 100 111 000 001



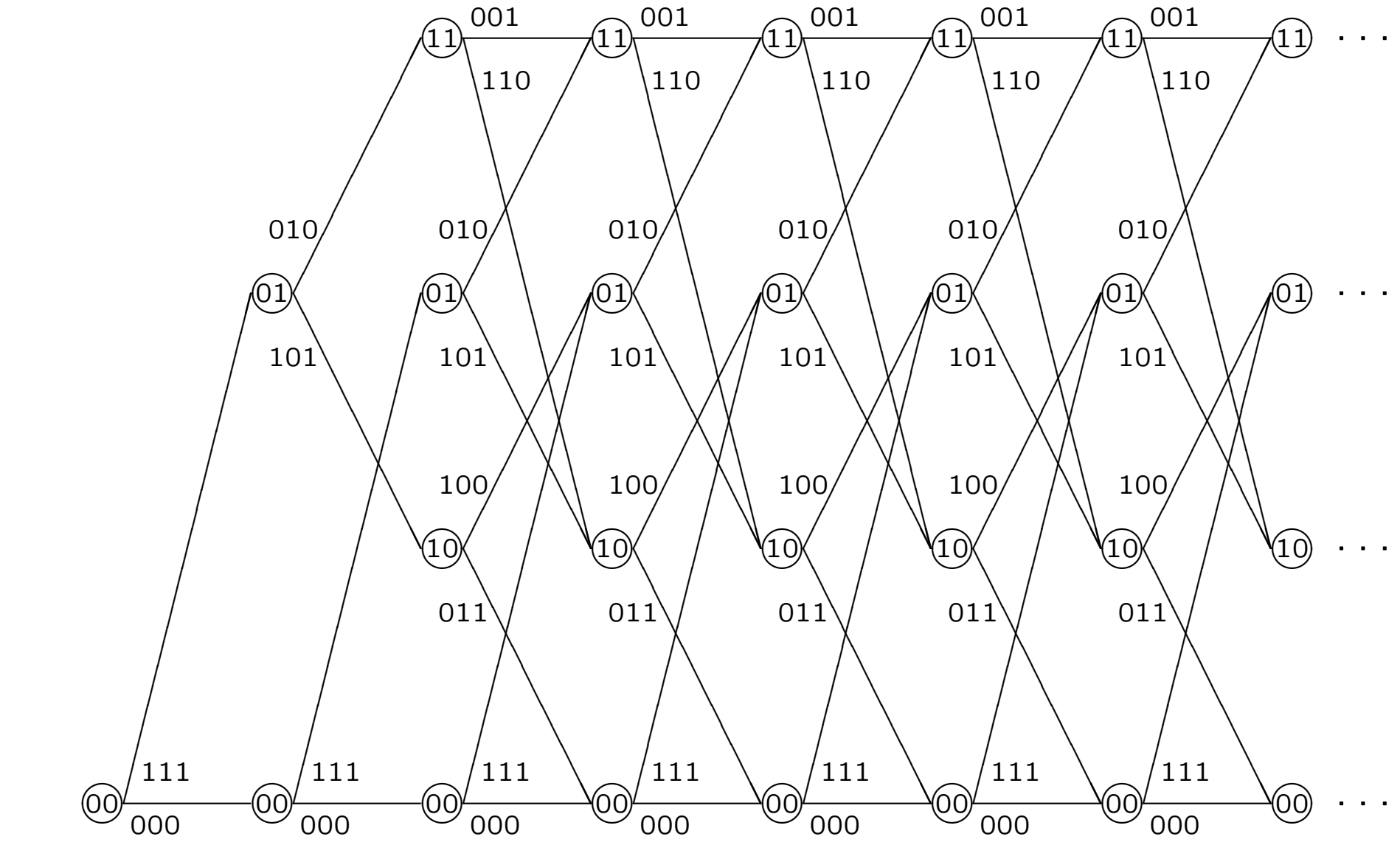
reçu : 110 111 011 100 111 000 001



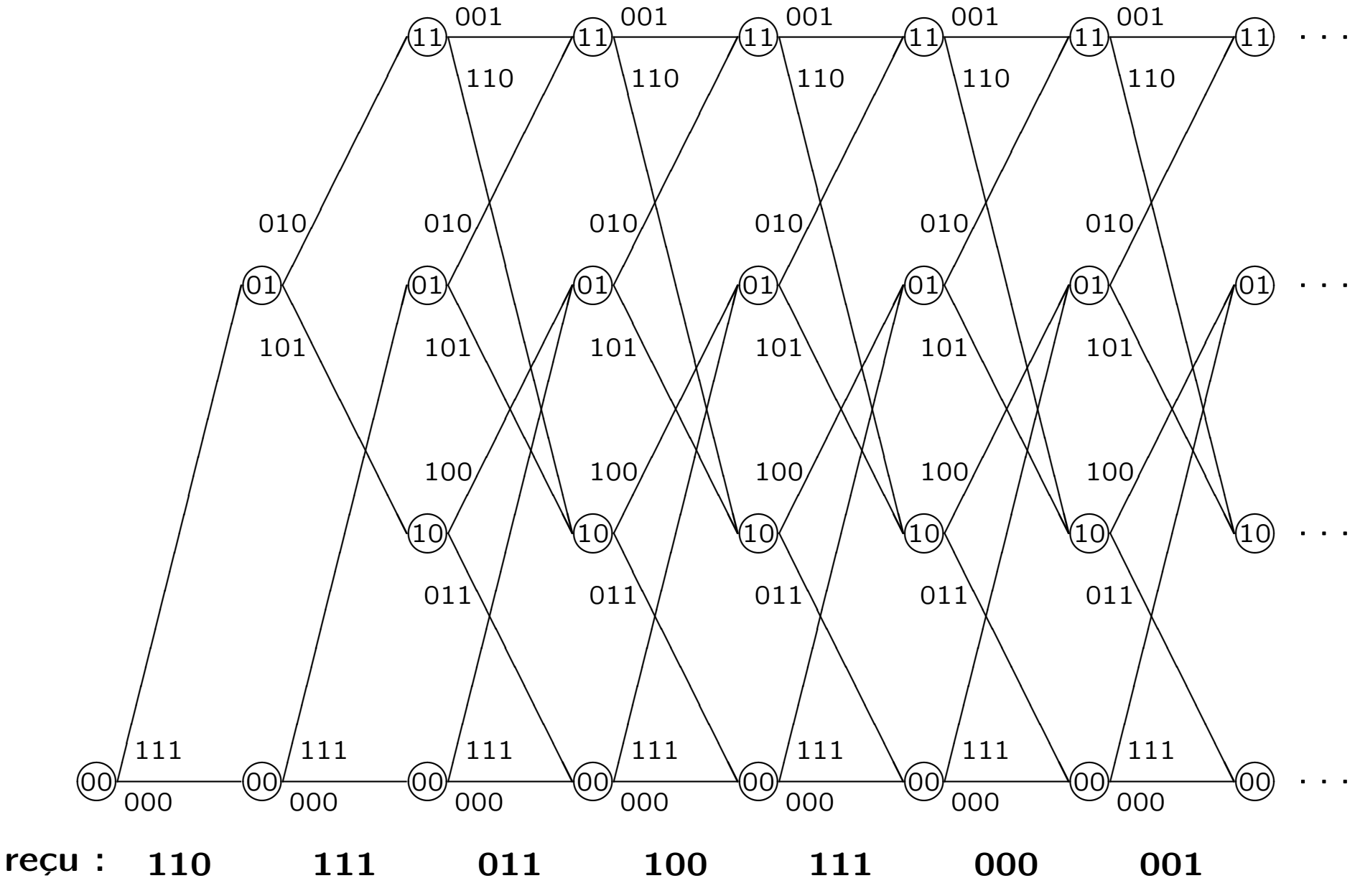
reçu : 110 111 011 100 111 000 001



reçu : 110 111 011 100 111 000 001



reçu : 110 111 011 100 111 000 001



reçu : 110 111 011 100 111 000 001