Département d'informatique et de génie logiciel

Compression de données
IFT-4003/IFT-7023

Notes de cours
Codage arithmétique

Édition Hiver 2012

Mohamed Haj Taieb

Local: PLT 2113

Courriel: mohamed.haj-taieb.1@ulaval.ca

Faculté des sciences et de génie
Département de génie électrique et de génie informatique

UNIVERSITÉ LAVAL

Plan de la présentation

- Codage arithmétique:
  - Idée principale
  - Génération d'étiquette
  - Déchiffrage de l'étiquette
  - Génération d'un code binaire
  - Unicité et efficacité du code
  - Algorithme d'implémentation
  - Implémentation entière
  - Comparaison avec le codage de Huffman

Rappel sur les codes de Huffman

- Longueur du code de Huffman
  - Le code de Huffman peut garantir un taux de codage (nombre moyen de bits par symbole) proche de l'entropie de 1 bit.

$$H(S) \leq l_{moy} < H(S) + 1$$

  - Cette borne supérieure peut être plus stricte.
  - Si les probabilités des symboles sont bien réparties avec la probabilité du symbole le plus fréquent pmax < 0.5 on obtient:

$$H(S) \leq l_{moy} < H(S) + p_{\max} + 0.086$$

  - On retrouve cette borne supérieure généralement lorsque le nombre de symboles est élevés.
  - Mais que faire si on a peu de symboles de probabilités distancées.

Codes de Huffman étendu (1)

- Regroupement de symboles
  - Le regroupement de symboles pour la génération d'un code de Huffman étendu peut améliorer les performances. Mais ce ne fonctionne pas toujours!
- Exemple
  - Considérons une source i.i.d suivante et le code de Huffman associé:

Lettres	a	b	C
Probabilité	0.95	0.02	0.03
Code	0	11	10

  - L'entropie de la source: 0.335 bits/symbole.
  - La longueur moyenne de ce code: 1.05 bits/symbole.
  - Redondance: 0.715 bits/symbole.
  - Efficacité: ξ=31.905%

Code de Huffman étendu (2)

- Regroupement de symboles
  - Le regroupement de symboles deux par deux donne le code suivant:

aa	ab	ac	ba	bb	Bc	ca	Cb	cc
0.902	0.019	0.0285	0.019	0.0004	0.0006	0.0285	0.0006	0.0009
0	111	100	1101	110011	110001	101	110010	110000

- Entropie=0.335 bit/symb
- lmoy=1.222/2=0.611 bit/symb
- Redondance=0.276
- Efficacité: ξ=54.828%

- Pour une efficacité acceptable il faut regrouper les symboles 8 par 8 → Alphabet de taille 3^8=6561 → Capacité de stockage du code élevée.

Code de Huffman étendu (3)

- Inefficacité des codes de Huffman étendu
  - La moindre perturbation des statistiques entrainant un changement des probabilité affecte grandement les performances.
  - Pour déterminer un mot-code pour une séquence de longueur m, on doit générer des mot-codes pour toutes autres séquences de même longueur possibles.
  - → Croissance exponentielle de la taille de l'alphabet.
  - Il faut un moyen pour générer un mot-code à une séquence donnée sans avoir à générer des mot-codes pour les autres séquences de même longueur.
  - → **Codes arithmétiques**

Idée principale des codes arithmétiques

- Fonctionnement
  - Dans le codage arithmétique un identificateur ou une étiquette unique est générée pour la séquence à encoder.
  - Cette étiquette correspond à une fraction binaire à partir de la quelle on obtient le code binaire de la séquence.
  - En pratique la génération de l'étiquette et du code binaire suivent la même procédure.
  - Mais pour mieux comprendre les codes arithmétiques on présente ces deux phases distinctement.
    - Phase 1: Génération d'étiquette (tag) pour la séquence
    - Phase 2: assignation d'un code binaire a ce tag

Codage d'une séquence

- Distinction entre les séquences de symboles
  - A chaque séquence de symboles un tag unique doit être fourni.
  - L'intervalle unité [0, 1) est un ensemble possible pour tout les tags.
  - En effet le nombre de réel dans cet intervalle sont infini on peut générer autant de tags que l'on veut.
  - Il faut alors une fonction f: séquence→ x ∈ [0, 1).
  - Une parmi ces fonctions on cite la fonction de distribution cumulative (*cdf*) de variable aléatoire.
  - La fonction cdf est utilisé dans le développement des code arithmétique.

Les fonctions de distribution de probabilités

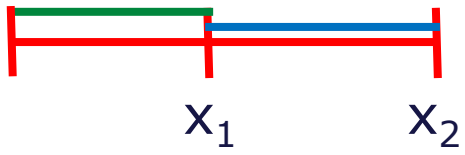
- Définition de la fonction cumulative
  - Soit X une variable aléatoire résultant d'une expérience donnée.
  - La fonction de distribution cumulative (cdf) dénotée par $F_X(x)$ est définie par: $F_X(x) = Pr(X \leq x)$.
- Quelques Propriétés
  - $0 \leq F_X(x) \leq 1$: une probabilité est toujours entre 0 et 1.
  - La cdf est une fonction croissante: $x_1 \leq x_2 \rightarrow F_X(x_1) \leq F_X(x_2)$

En effet $F_X(x_2) = Pr(X \leq x_2) = Pr(X \leq x_1 \text{ ou } x_1 < X \leq x_2)$

$= Pr(X \leq x_1) + Pr(x_1 < X \leq x_2)$

$= F_X(x_1) + Pr(x_1 < X \leq x_2)$

$\geq F_X(x_1)$



Quelques notations

- Fonction cumulative
  - Soit l'alphabet d'éléments discrets: A={ a_1, a_2, …, a_m}.
  - Soit une expérience qui résulte en un des éléments de A.
  - Soit la variable aléatoire X prenant des valeurs dans l'ensemble {1,…,m}.
  - La fonction de densité de probabilité pour cette variable aléatoire: P(X=i)=P(a_i).
  - La fonction de densité cumulative:

$$F_X\ i\ = \Pr(X \le i) = \Pr(X = 1\ ou\ \ldots\ ou\ X = i) = \Pr(X = 1) + \ldots + \Pr(X = i) = \sum_{k=1}^{i} P(X = k)$$

  - Pour chaque symbole a_i de probabilité non nulle on associe une valeur distincte de $F_X(i)$→ on peut alors utiliser $F_X(i)$ comme un tag dans les codes arithmétiques.

Génération de tag

- Procédure
  - Réduction de la taille de l'intervalle contenant le tag à chaque lecture d'un symbole de la séquence.
  - On commence par subdiviser l'intervalle [0, 1] en sous-intervalles de la forme $[F_X(i-1), F_X(i))$ avec i=1, …, m.
  - On associe à chaque sous-intervalles $[F_X(i-1), F_X(i))$ le symbole a_i.
  - La lecture du premier symbole, soit a_k, permet de restreindre l'intervalle du tag au sous-intervalle associé à a_k: $[F_X(k-1), F_X(k))$.
  - Ce sous-intervalle est subdiviser à son tour en sous-intervalles suivant la même partition de que l'intervalle initiale.

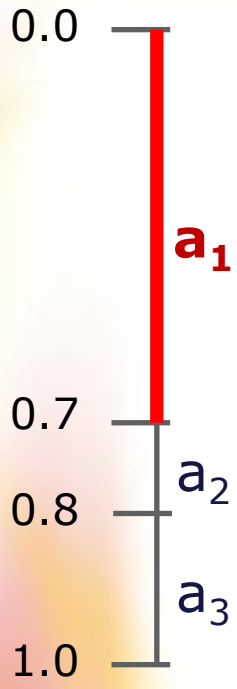
Partitionnement des sous-intervalles

- Procédure
  - Chaque symbole lu partitionne le sous-intervalle résultant de la lecture des symboles précédents et la position du tag devient de plus en plus restreinte.
  - Premier symbole a_k → $[F_X(k-1), F_X(k))$.
  - Deuxième symbole a_j →

$$[F_X(k-1)+F_X(j-1)\times\ F_X(k)-F_X(k-1)\ ,\ F_X(k-1)+F_X(j)\times(F_X(k)-F_X(k-1)))$$

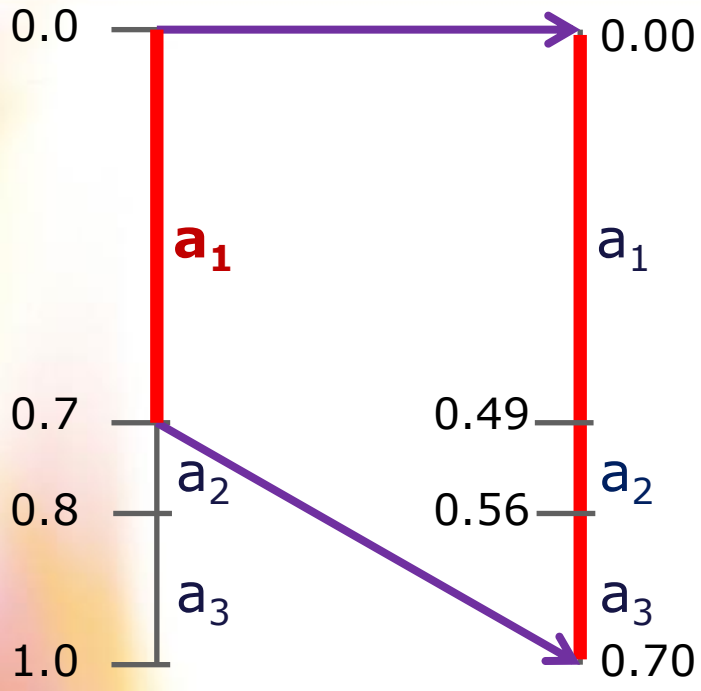
- Exemple
  - Soit un alphabet formé de trois lettre A= $\{a_1, a_2, a_3\}$ avec $P(a_1)=0.7$, $P(a_2)=0.1$ et $P(a_3)=0.2$.
  - → $F_X(1)=0.7$, $F_X(2)=0.8$ et $F_X(3)=1$.

Exemple de partitionnement des sous-intervalles (1)



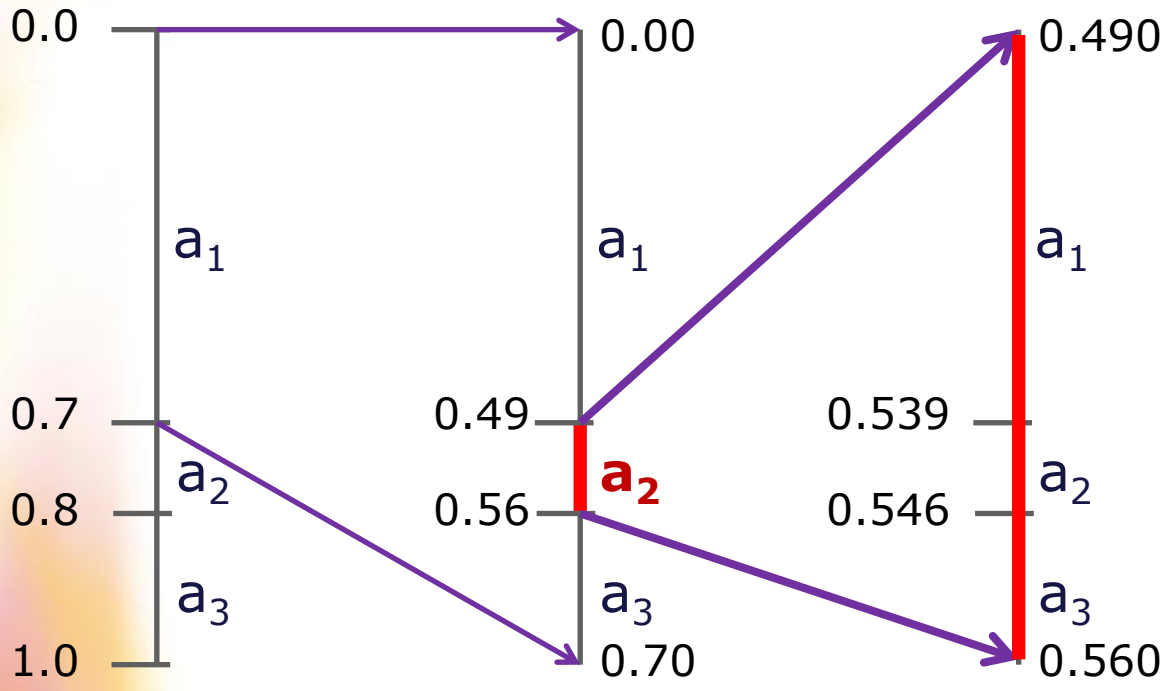
$a_{k=1} \rightarrow [F_X(k-1), F_X(k)) = [F_X(0), F_X(1)) = [F_X(0), F_X(1)) = [0, 0.7)$

Exemple de partitionnement des sous-intervalles (2)



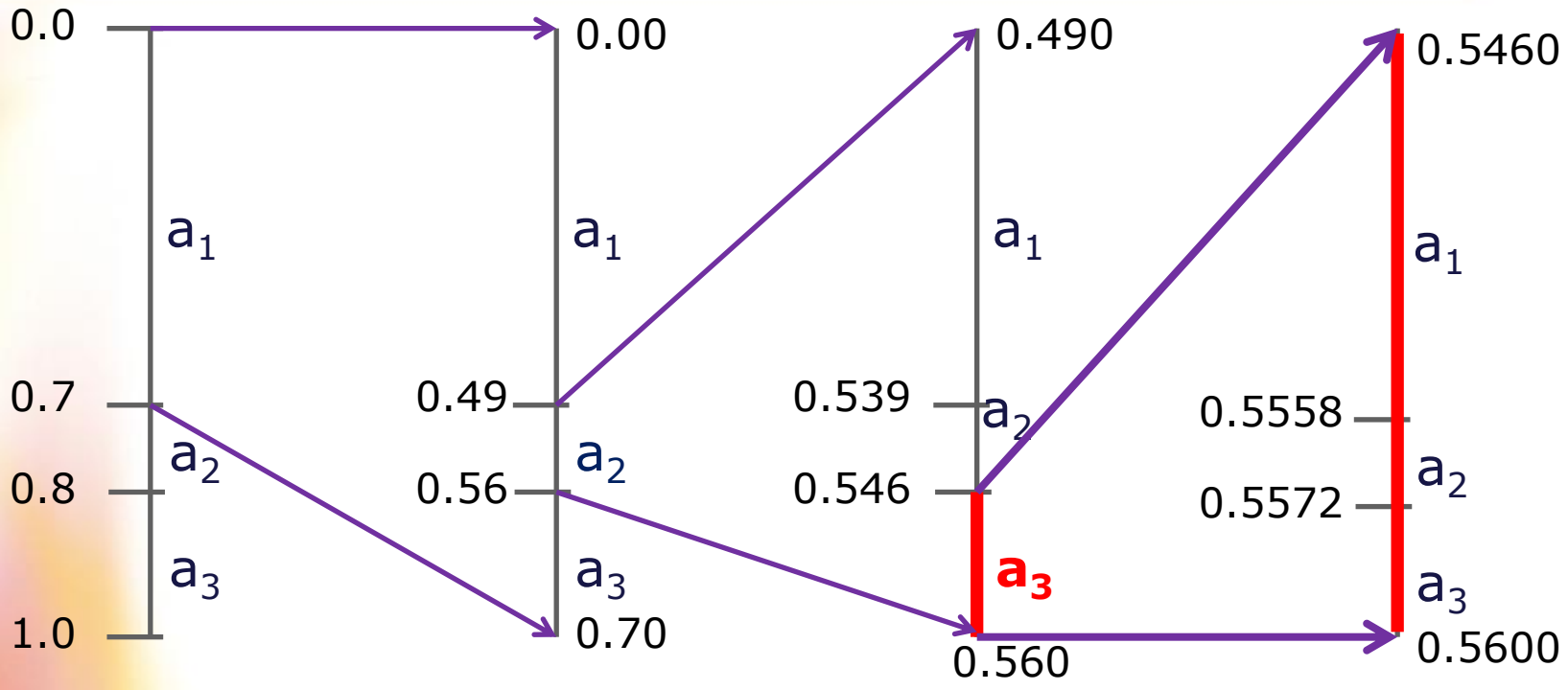
$$[F_X(0) + F_X(1) \times (F_X(1) - F_X(0)),\ F_X(0) + F_X(2) \times (F_X(1) - F_X(0)))$$
$$= [0 + 0.7 \times (0.7 - 0),\ 0 + 0.8 \times (0.7 - 0))$$
$$= [0.7 \times 0.7,\ 0.8 \times 0.7)$$
$$= [0.49,\ 0.56)$$

Exemple de partitionnement des sous-intervalles (3)



$$[l^{(1)} + F_X(1)\times(u^{(1)} - l^{(1)}),\ l^{(1)} + F_X(2)\times(u^{(1)} - l^{(1)}))$$
$$= [0.49 + 0.7\times 0.07,\ 0.49 + 0.8\times 0.07)$$
$$= [0.49 + 0.049,\ 0.49 + 0.056)$$
$$= [0.539,\ 0.546)$$

Exemple de partitionnement des sous-intervalles (4)



$$[l^{(2)} + F_X(1)\times(u^{(2)} - l^{(2)}),\ l^{(2)} + F_X(2)\times(u^{(2)} - l^{(2)}))$$
$$= [0.546 + 0.7\times 0.014,\ 0.546 + 0.8\times 0.014)$$
$$= [0.546 + 0.0098,\ 0.546 + 0.0112)$$
$$= [0.5558,\ \ 0.5572)$$

Partitionnement en sous-intervalles

- Remarques importantes suite à l'exemple
  - L'apparition de chaque nouveau symbole donne naissance à un sous-intervalle disjoint [disjoint: $\Pi = \emptyset$] avec tout autre sous-intervalle qui aurait pu être généré par une autre séquence.
  - Ainsi tout entier appartement à l'intervalle obtenu, peut être associé comme tag à la séquence de symboles.
  - i.e. la limite inférieure de l'intervalle ou encore le point central de l'intervalle.
  - Dans ce qui suit on va considérer le point centrale comme tag de la séquence.

Génération d'un tag pour symbole unique

- Procédure mathématique
  - Pour comprendre la procédure mathématique de la génération d'un tag on commence par une séquence de longueur un.
  - → Un tag est associé à chaque symbole de l'alphabet.
  - Soit une source qui génère une séquence à partir d'un alphabet A={a_1, a_2, …, a_m}. On définit pour un symbole a_i le tag suivant:

$$\overline{T}_X(a_i) = \sum_{k=1}^{i-1} P(x=k) + \frac{1}{2}P(x=i)$$

$$= F_X(i-1) + \frac{1}{2}P(x=i)$$

  - Donc pour chaque symbole a_i, on va avoir un tag unique.

Exemple de génération d'un tag pour un symbole unique

- Exemple
  - Considérons l'expérience de lancement d'un dé juste.
  - Le résultat de cette expérience peut être associé à l'ensemble {1, 2, 3, 4, 5, 6}.
  - Pour un dé juste on a: P(X=k)=1/6 pour k=1,…,6

$$\bar{T}_X(1) = F_X(0) + \frac{1}{2}P(x=1) = 0 + \frac{1}{2}\times\frac{1}{6} = 0.08\overline{3}$$

$$\bar{T}_X(2) = F_X(1) + \frac{1}{2}P(x=2) = \frac{1}{6} + \frac{1}{2}\times\frac{1}{6} = 0.25$$

$$\bar{T}_X(3) = 0.41\overline{6}; \quad \bar{T}_X(4) = 0.58\overline{3}$$

$$\bar{T}_X(5) = 0.75; \quad \bar{T}_X(4) = 0.91\overline{6}$$

Génération d'un pour une séquence de symboles

- Procédure mathématique
  - L'approche d'assignation de tag pour une séquence de longueur un d'une source prenant des valeur de alphabet $A=\{a_1, a_2, \ldots, a_m\}$, peut s'étendre pour traiter des séquences plus longueur n>1.
  - → Il faut établir un ordre pour différencier entre les $\boldsymbol{m^n}$ combinaisons possibles des n symboles.
  - La séquence numéro i parmi les $\boldsymbol{m^n}$ combinaisons possibles est dénoté x_i, à laquelle on associe le tag suivant:

$$\overline{T}_{\mathrm{x}}^{(m)}(\mathrm{x}_i) = \sum_{k=1}^{i-1} P(\mathrm{X}=\mathrm{x}_k) + \frac{1}{2}P(\mathrm{X}=\mathrm{x}_i)$$

  - X est une variable aléatoire de longueur m.
  - $x_i=\{x_i^1, x_i^2, \ldots, x_i^m\}$,

Exemple de génération d'un pour une séquence de symboles

- Procédure mathématique
  - Considérons l'expérience de 2 lancements d'un dé juste.
  - Il y a $6^2=36$ combinaisons possibles qu'on ordonne comme suit: 11,12,…,16, 21,22,…,26, …, …,51,52,…,56, 61,62,…,66.
  - Calculons par exemple le tag de la séquence 13:

$$\bar{T}_{\mathrm{x}}^{(2)}(13) = \sum_{\substack{k<13 \\ k\in\ 11,12,\ldots,66}} P(\mathrm{X}=\mathrm{x}_k) + \frac{1}{2}P(\mathrm{X}=13)$$

$$= P(\mathrm{X}=11) + P(\mathrm{X}=12) + \frac{1}{2}P(\mathrm{X}=13)$$

$$= \frac{1}{36} + \frac{1}{36} + \frac{1}{2}\times\frac{1}{36} = \frac{5}{72}$$

Génération d'un pour une séquence de symboles

- Remarques suite à l'exemple:
  - Pour générer le tag de la séquence numéro 13, on n'a pas besoin de générer le tag des autres séquences de me même longueur n=2.
  - Il faut par contre calculer les probabilités de toutes les séquences dont l'ordre est inférieur a 13: 11 et 12.
  - Le calcul des probabilité des séquences précédentes peut entraîner une complexité de même ordre que la génération de mot-codes pour les autres séquences de même longueur.
  - Nous allons voir par la suite qu'on a besoin juste de la probabilité des symboles individuels pour le calcul d'un tag d'une séquence de symboles.

Délimitation de l'intervalle du tag

- Calcul des bornes inférieure et supérieure de l'intervalle du tag:
  - Lors de la génération du tag on constate que l'intervalle obtenu est disjoint des autres intervalles contenant les tags des autres séquences.
  - Ainsi la détermination d'un tag pour une séquence donnée revient à délimiter l'intervalle en question et ce par le calcul des bornes inférieure et supérieure.
  - Ces deux bornes peuvent être calculées récursivement.
- Exemple:
  - Reprenons l'exemple de lancement du dé non truqué.
  - Déterminons l'intervalle du tag de la séquence 322.

Exemple: bornes de l'intervalle du tag par récursion (1)

- Tag de la séquence 322
  - Observation séquentielle de 3 ensuite 2 et enfin 2.
  - Après chaque observation on calcule les bornes de l'intervalle du tag à ce stade là.
  - Notation: limite supérieure → $u^{(n)}$ [upper limit]

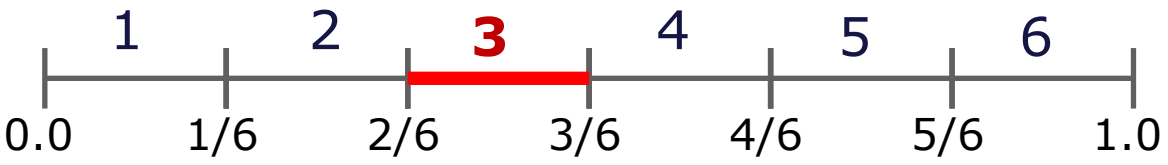
    limite inférieure → $l^{(n)}$ [lower limit]

    n le nombre d'observations à ce moment là.

- Observation 1: **3**

$$u^{(1)} = F_X(3) = \Pr(X \le 3) = \Pr(X = 1 \; ou \; X = 2 \; ou \; X = 3) = \frac{3}{6}$$

$$l^{(1)} = F_X(2) = \Pr(X \le 3) = \frac{2}{6}$$



Exemple: bornes de l'intervalle du tag par récursion (2)

- Observation 2: 3 **2**

$$u^{(2)} = F_X^{\,2}(32) = \Pr(X \le 32)$$

$$= \Pr(X=11) + \Pr(X=12) + \cdots + \Pr(X=16)$$

$$+ \Pr(X=21) + \Pr(X=22) + \cdots + \Pr(X=26)$$

$$+ \Pr(X=31) + \Pr(X=32)$$

[Rappel] Théorème de Bayes: loi de la probabilité totale

$$\sum_{i=1}^{6} \Pr(X = ki) = \sum_{i=1}^{6} \Pr(X = k, X = i) = \Pr(X = k)$$

$$\Rightarrow$$

$$u^{(2)} = \Pr(X=1) + \Pr(X=2) + \Pr(X=31) + \Pr(X=32)$$

$$u^{(2)} = F_X(2) + \Pr(X=31) + \Pr(X=32)$$

Exemple: bornes de l'intervalle du tag par récursion (3)

- Observation 2: 3 **2**

$$\Pr(X = 31) + \Pr(X = 32) = \Pr(3)\Pr(1) + \Pr(3)\Pr(2)$$

$$= \Pr(3)(\Pr(1) + \Pr(2)) = \Pr(3)F_X(2)$$

$$= (F_X(3) - F_X(2))F_X(2)$$

$$\Rightarrow$$

$$u^{(2)} = F_X(2) + (F_X(3) - F_X(2))F_X(2)$$

$$\boxed{u^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(2)}$$

de la même façon on trouve:

$$\boxed{l^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(1)}$$

Exemple: bornes de l'intervalle du tag par récursion (4)

- Observation 2: 3 **2**

$$u^{(1)} = F_X(3) = \frac{3}{6}$$

$$l^{(1)} = F_X(2) = \frac{2}{6}$$

$$u^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(2)$$

$$l^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(1)$$

- Observation 3: 3 2 **2**

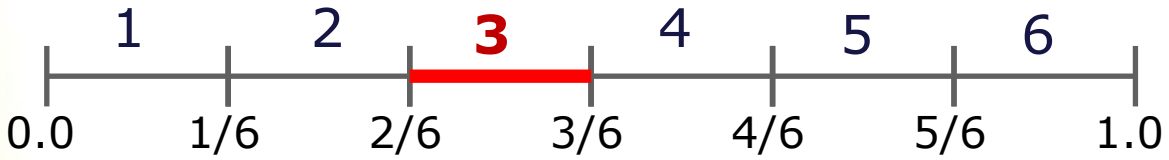
$$u^{(3)} = l^{(2)} + (u^{(2)} - l^{(2)})F_X(2)$$

$$l^{(3)} = l^{(2)} + (u^{(2)} - l^{(2)})F_X(1)$$

Exemple: bornes de l'intervalle du tag par récursion (5)

- Observation 1: **3**

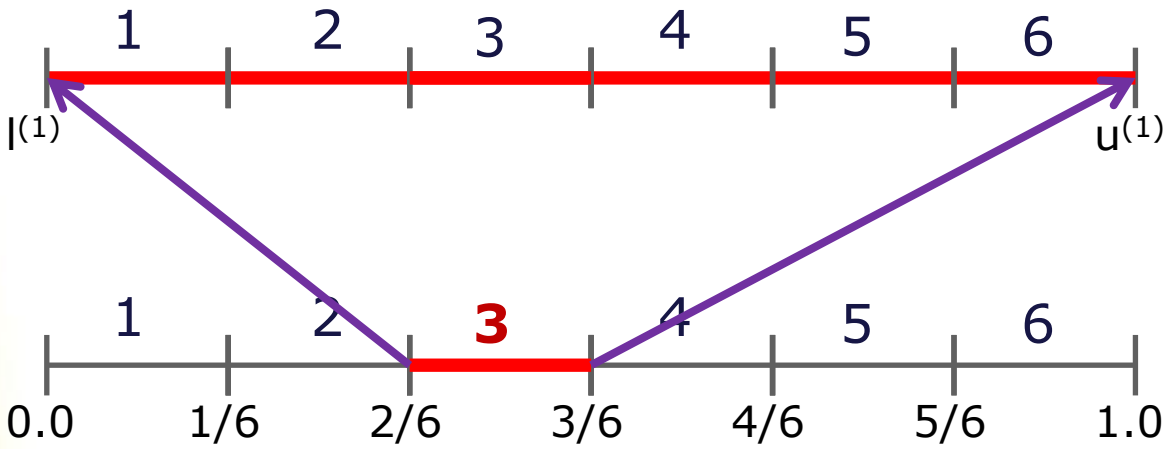
$$u^{(1)} = F_X(3) = \frac{3}{6}, \quad l^{(1)} = F_X(2) = \frac{2}{6}$$



Exemple: bornes de l'intervalle du tag par récursion (6)

- Observation 1: **3**

$$u^{(1)} = F_X(3) = \frac{3}{6}, \quad l^{(1)} = F_X(2) = \frac{2}{6}$$

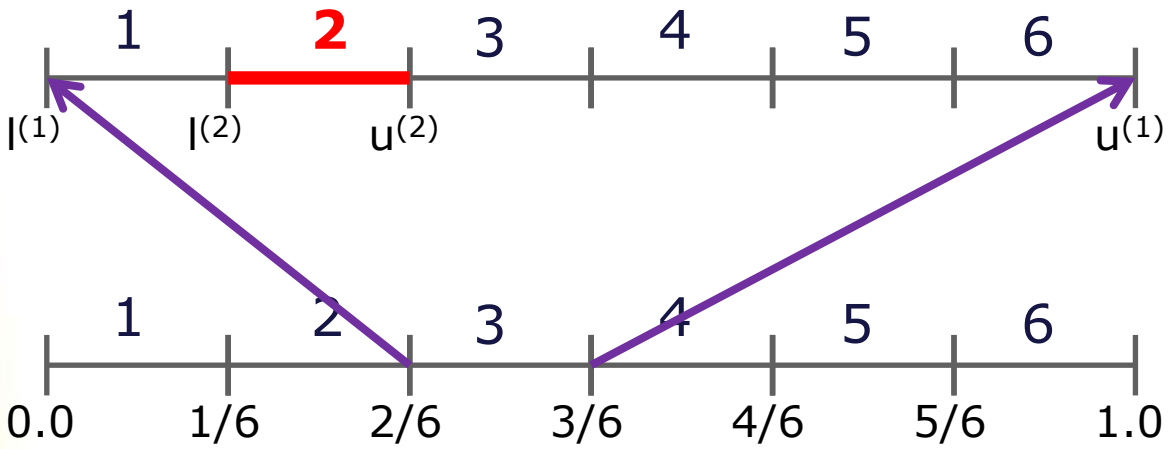


Mise à l'échelle:
× $(u^{(1)}-l^{(1)})$

Exemple: bornes de l'intervalle du tag par récursion (7)

- Observation 2: 3 **2**

$$u^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(2), \quad l^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(1)$$



Mise à l'échelle:
× ($u^{(1)}$-$l^{(1)}$)

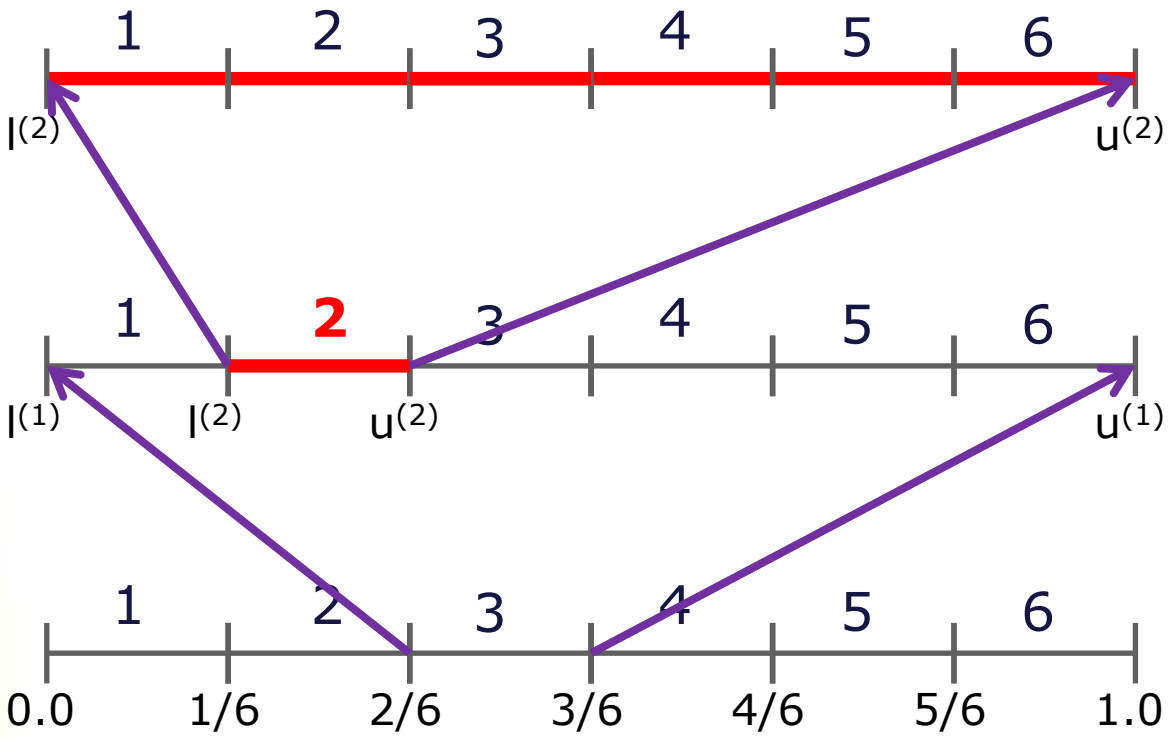
Exemple: bornes de l'intervalle du tag par récursion (8)

- Observation 2: 3 **2**

$$u^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(2), \quad l^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(1)$$

Mise à l'échelle:
× ($u^{(2)}$-$l^{(2)}$)

Mise à l'échelle:
× ($u^{(1)}$-$l^{(1)}$)

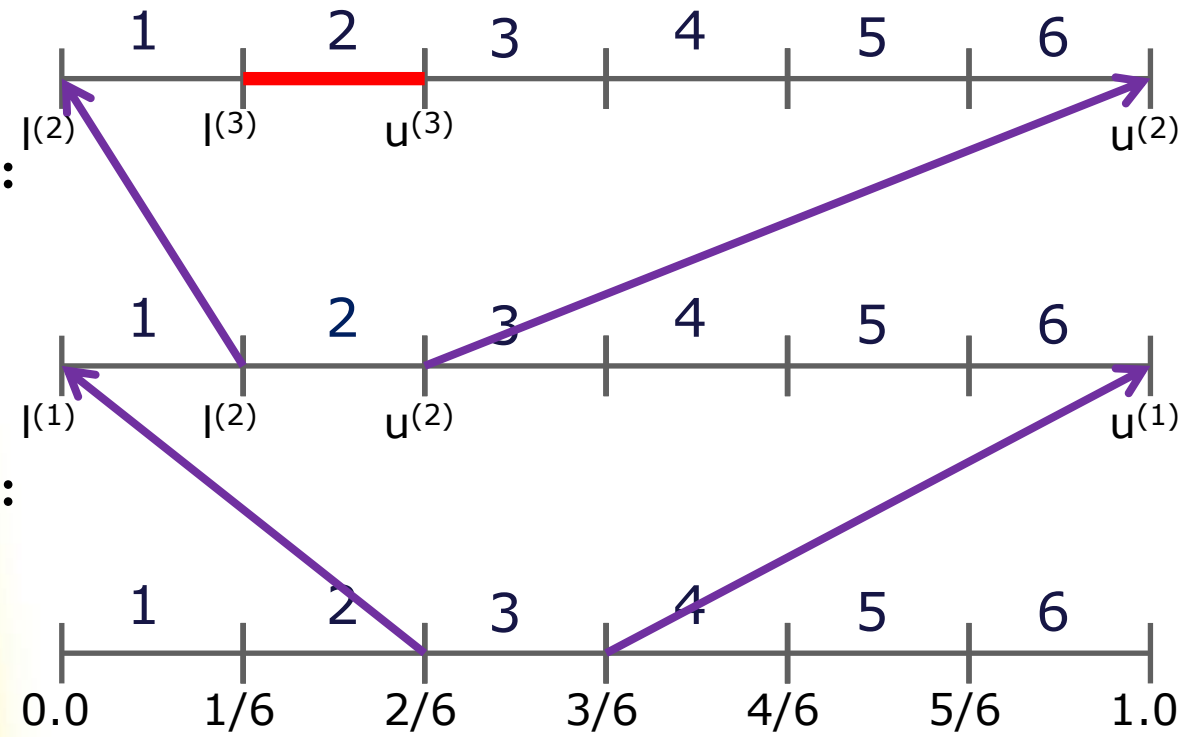


Exemple: bornes de l'intervalle du tag par récursion (9)

- Observation 3: 3 2 **2**

$$u^{(3)} = l^{(2)} + (u^{(2)} - l^{(2)})F_X(2), \quad l^{(3)} = l^{(2)} + (u^{(2)} - l^{(2)})F_X(1)$$

Mise à l'échelle:
× $(u^{(2)}-l^{(2)})$



Mise à l'échelle:
× $(u^{(1)}-l^{(1)})$

Intervalle du tag par récursion

- Règle générale
  - Pour une séquence de symbole x on a:

$$x = (x_1 x_2 \ldots x_n)$$
$$u^{(n)} = l^{(n-1)} + (u^{(n-1)} - l^{(n-1)}) F_X(x_n)$$
$$l^{(n)} = l^{(n-1)} + (u^{(n-1)} - l^{(n-1)}) F_X(x_n - 1)$$

  - Avec cette procédure de récursion, on a pas eu besoin de calculer la probabilité conjointe d'une séquence donnée.
  - On a juste besoin des probabilités individuelle.
- Génération du tag:
  - Pour un tag milieu d'intervalle on a:

$$\overline{T}_X \quad x = \frac{u^{(n)} + l^{(n)}}{2}$$

Exemple génération du tag par récursion (1)

- Exemple: génération du tag de la séquence: 1 3 2 1
  - Considérons la source suivante:

Lettres	a	b	c
Probabilité	0.8	0.02	0.18

  - A partir du modèle de probabilité on a: $F_X(k<0)=0$, $F_X(0)=0$, $F_X(1)=0.8$, $F_X(2)=0.82$, $F_X(3)=1$, $F_X(k>3)=1$.
- Observation 1: 1

[*Rappel*]

$$u^{(n)} = l^{(n-1)} + (u^{(n-1)} - l^{(n-1)})F_X(x_n)$$
$$l^{(n)} = l^{(n-1)} + (u^{(n-1)} - l^{(n-1)})F_X(x_n - 1)$$

$$u^{(1)} = l^{(0)} + (u^{(0)} - l^{(0)})F_X(x_1) = 0 + 1 \times F_X(1) = 0.8$$
$$l^{(1)} = l^{(0)} + (u^{(0)} - l^{(0)})F_X(x_1 - 1) = 0 + 1 \times F_X(0) = 0$$

Exemple génération du tag par récursion (2)

[*Rappel*]

$$u^{(n)} = l^{(n-1)} + (u^{(n-1)} - l^{(n-1)})F_X(x_n)$$
$$l^{(n)} = l^{(n-1)} + (u^{(n-1)} - l^{(n-1)})F_X(x_n - 1)$$

- Observation 1: **1**

$$u^{(1)} = l^{(0)} + (u^{(0)} - l^{(0)})F_X(x_1) = 0 + 1 \times F_X(1) = 0.8$$
$$l^{(1)} = l^{(0)} + (u^{(0)} - l^{(0)})F_X(x_1 - 1) = 0 + 1 \times F_X(0) = 0$$

- Observation 1: 1 **3**

$$u^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(x_3) = 0 + 0.8 \times F_X(3) = 0.8 \times 1 = 0.8$$
$$l^{(2)} = l^{(1)} + (u^{(1)} - l^{(1)})F_X(x_3 - 1) = 0 + 0.8 \times F_X(3 - 1) = 0.8 \times 0.82 = 0.656$$

- Observation 1: 1 3 **2**

$$u^{(3)} = l^{(2)} + (u^{(2)} - l^{(2)})F_X(x_2) = 0.656 + 0.144 \times 0.82 = 0.77408$$
$$l^{(3)} = l^{(2)} + (u^{(2)} - l^{(2)})F_X(x_2 - 1) = 0.656 + 0.144 \times 0.8 = 0.7712$$

- Observation 1: 1 3 2 **1**

$$u^{(4)} = l^{(3)} + (u^{(3)} - l^{(3)})F_X(x_1) = 0.7712 + 0.00288 \times 0.8 = 0.773504$$
$$l^{(4)} = l^{(3)} + (u^{(3)} - l^{(3)})F_X(x_1 - 1) = 0.7712 + 0.00288 \times 0 = 0.7712$$

Exemple génération du tag par récursion (3)

- Séquence: 1 3 2 1

$$u^{(4)} = l^{(3)} + (u^{(3)} - l^{(3)})F_X(x_1) = 0.7712 + 0.00288 \times 0.8 = 0.773504$$
$$l^{(4)} = l^{(3)} + (u^{(3)} - l^{(3)})F_X(x_1 - 1) = 0.7712 + 0.00288 \times 0 = 0.7712$$

- Génération du tag:

$$\bar{T}_X(1321) = \frac{0.7712 + 0.773504}{2} = 0.772352$$

- Remarques:
  - L'intervalle suivant est toujours contenu dans l'intervalle précédent.
  - Cette propriété va servir pour le déchiffrage du tag.
  - L'intervalle du Tag devient de plus en plus petit.
  - Pour remédier à ça on va utiliser une approche de remise à l'échelle.

Déchiffrage du tag

- Déchiffrage:
  - La procédure de génération du tag est relativement simple.
  - Il faut aussi pouvoir déchiffrer le tag avec un coût de calcul minimal.
  - Effectivement lé déchiffrage du tag est aussi simple que sa génération.
- Exemple de déchiffrage du tag: $\overline{T}_X = 0.772352$
  - On reconsidère l'exemple précédent et on va essayer de déchiffrer le tag obtenu, et ce, par imiter l'encodeur.
  - L'intervalle contenant ce tag est un sous-intervalle de tout les intervalles rencontrés dans le processus d'encodage.

Exemple déchiffrage du tag (1)

- Exemple de déchiffrage du tag: $\bar{T}_X = 0.772352$
  - La stratégie consiste à décoder les éléments de la séquence en s'assurant que $l^{(k)}$ et $u^{(k)}$ contiennent toujours le tag.
- Étape 0:
  - $l^{(0)} = 0$ et $u^{(0)} = 1$ contiennent bien le tag 0.772352.
- Étape 1:
  - Après le décodage du premier élément de la séquence x_1 les limites inférieures et supérieures deviennent.

$$u^{(1)} = l^{(0)} + (u^{(0)} - l^{(0)})F_X(x_1) = F_X(x_1)$$
$$l^{(1)} = l^{(0)} + (u^{(0)} - l^{(0)})F_X(x_1 - 1) = F_X(x_1 - 1)$$

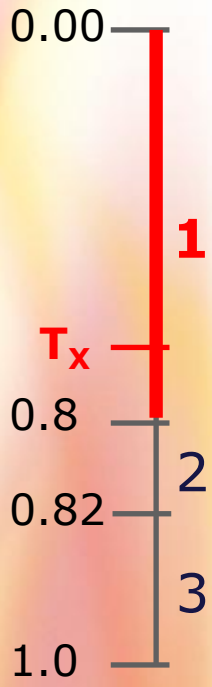
Exemple déchiffrage du tag (2)

- Suite étape 1:

$$u^{(1)} = F_X(x_1), \quad l^{(1)} = F_X(x_1 - 1)$$

$$\bar{T}_X = 0.772352$$

  - Le tag suite à l'observation du premier élément se trouve dans l'intervalle $[F_X(x_1\text{-}1), F_X(x_1)) = [0, 0.8)$.



$$\begin{aligned} &\bar{T}_X = 0.772352 \in [0, 0.8) \\ &\bar{T}_X = 0.772352 \notin [0.8, 0.82) \\ &\bar{T}_X = 0.772352 \notin [0.82, 1) \end{aligned} \Rightarrow x_1 = 1$$

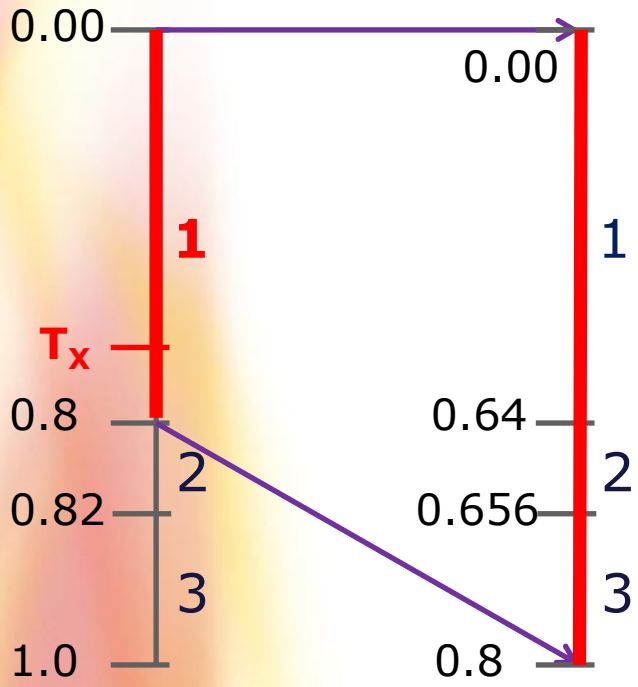
- Séquence décodée: **1**

Exemple déchiffrage du tag (3)

- Remise à l'échelle:

$$\bar{T}_X = 0.772352$$

  - Préparation des nouveaux intervalles pour le second élément.



Remise en échelle des intervalles

$$l^{(1)} + (u^{(1)} - l^{(1)})0.8 = 0 + 0.8 \times 0.8 = 0.64$$

$$l^{(1)} + (u^{(1)} - l^{(1)})0.82 = 0 + 0.8 \times 0.82 = 0.656$$

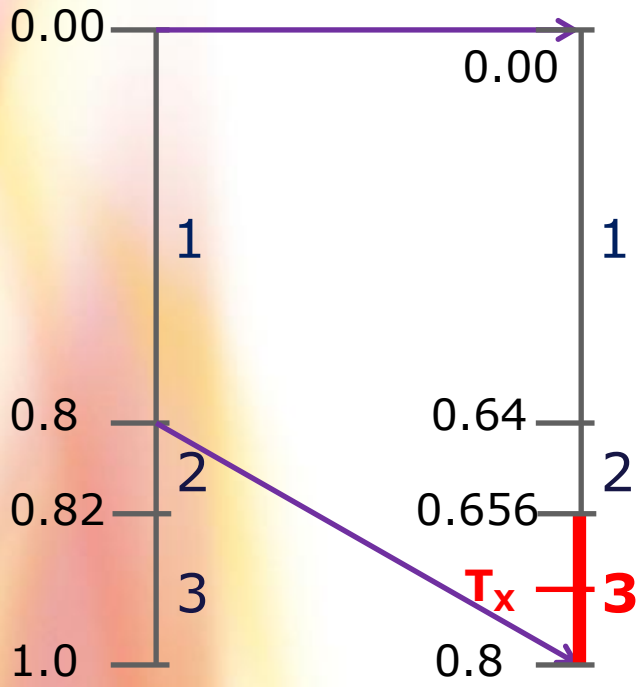
- Séquence décodée: **1**

Exemple déchiffrage du tag (4)

- Étape 2:

$$\bar{T}_X = 0.772352$$

  - Le tag suite à l'observation du premier élément se trouve dans l'intervalle $[F_X(x_2\text{-}1), F_X(x_2))=[0.656,0.8)$.



$$\bar{T}_X = 0.772352 \notin [0, 0.64)$$
$$\bar{T}_X = 0.772352 \notin [0.64, 0.656) \Rightarrow x_2 = 3$$
$$\bar{T}_X = 0.772352 \in [0.656, 0.8)$$

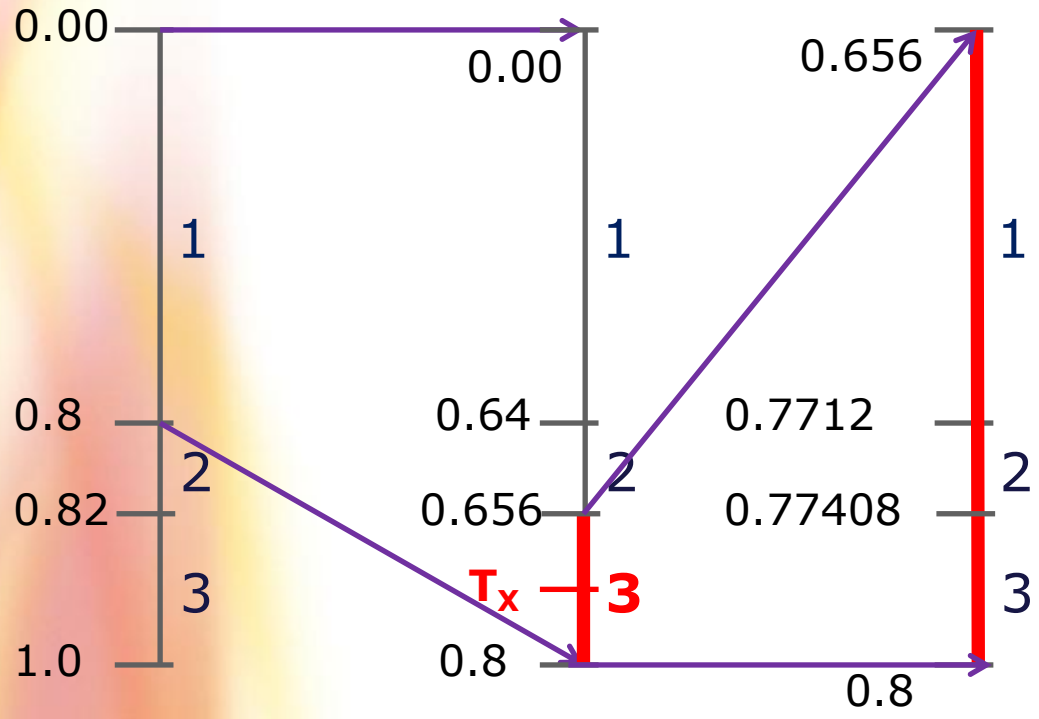
- Séquence décodée: 1 **3**

Exemple déchiffrage du tag (5)

- Remise à l'échelle:

$$\bar{T}_X = 0.772352$$

  - Le tag suite à l'observation du premier élément se trouve dans l'intervalle $[F_X(x_2\text{-}1), F_X(x_2))=[0.656,0.8)$.



Remise en échelle des intervalles

$$l^{(2)} + (u^{(2)} - l^{(2)})0.8$$
$$= 0.656 + 0.144 \times 0.8 = 0.7712$$
$$l^{(2)} + (u^{(2)} - l^{(2)})0.8$$
$$= 0.656 + 0.144 \times 0.82 = 0.77408$$

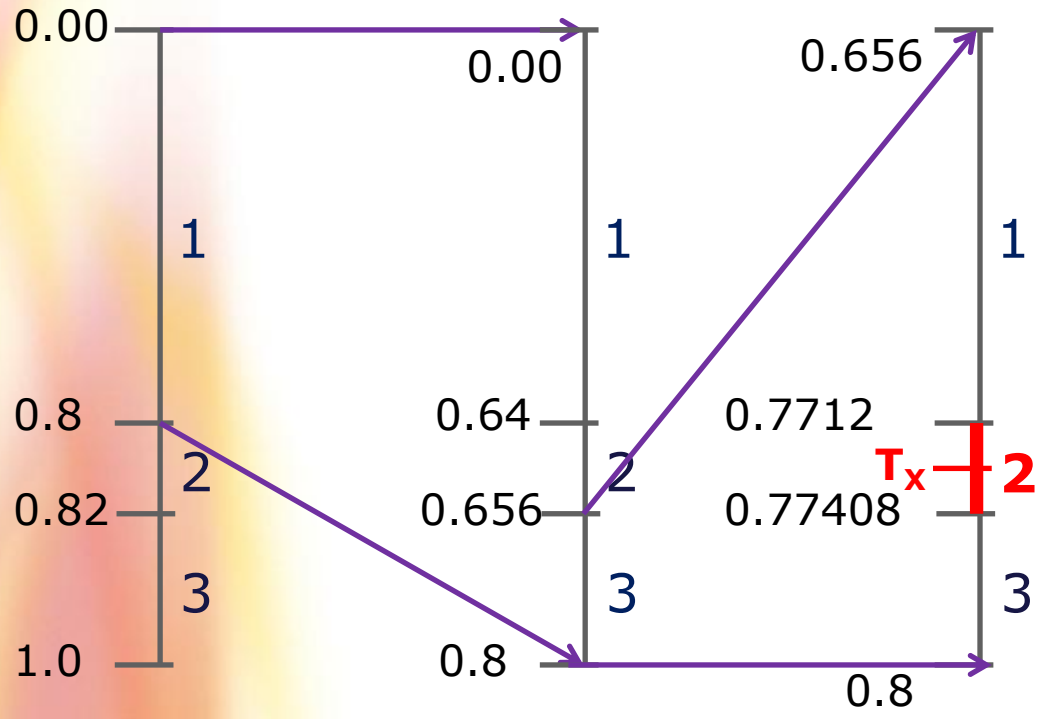
- Séquence décodée: 1 **3**

Exemple déchiffrage du tag (6)

- Étape 3:

$$\begin{aligned} \bar{T}_X &= 0.772352 \notin [0.656, 0.7712\ ) \\ \bar{T}_X &= 0.772352 \in [0.7712,\ 0.77408) \\ \bar{T}_X &= 0.772352 \notin [0.77408, 0.8) \end{aligned} \Rightarrow x_3 = 2$$

$$\bar{T}_X = 0.772352$$



- Séquence décodée: 1 3 **2**

Exemple déchiffrage du tag (7)

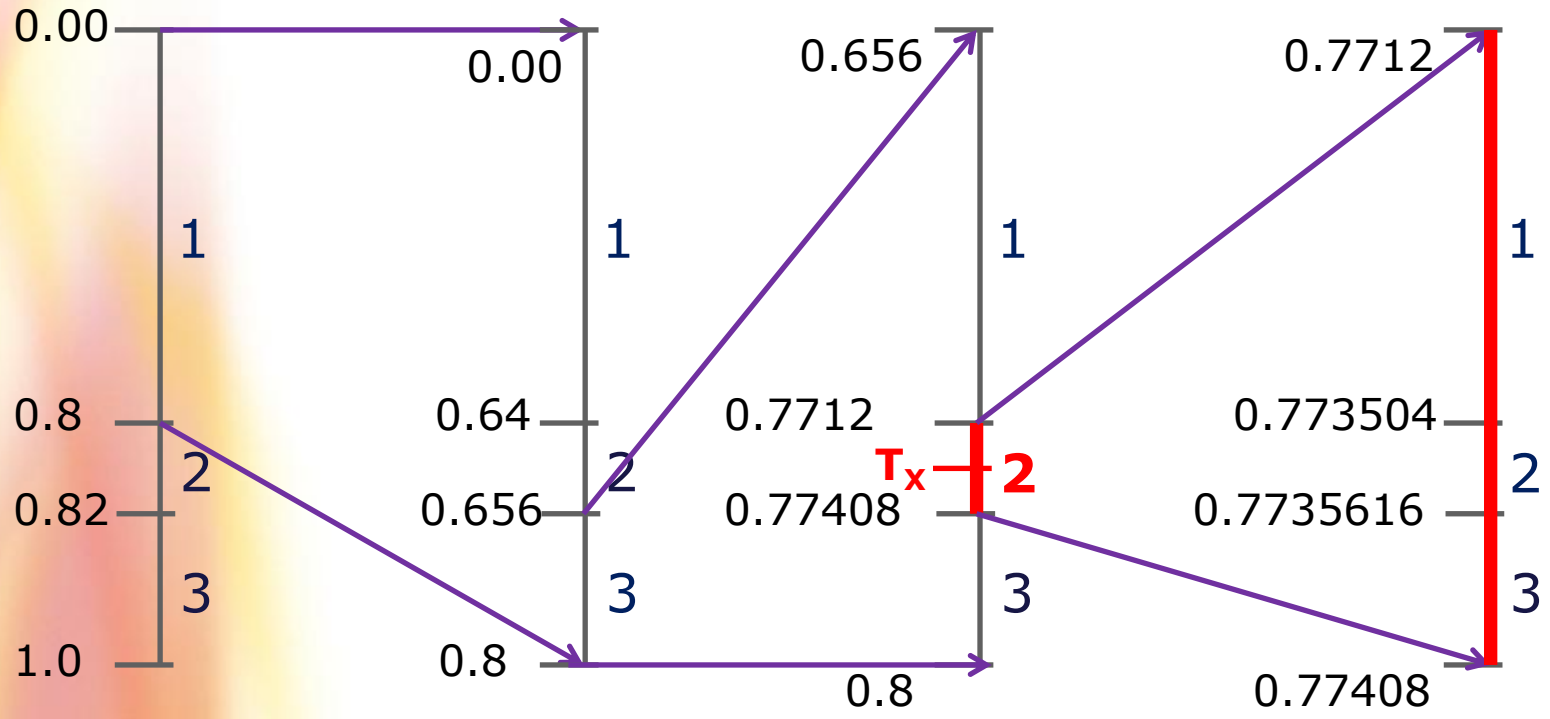
- Étape 3:

Remise en échelle des intervalles

$$l^{(3)} + (u^{(3)} - l^{(3)})0.8 = 0.7712 + 0.0288 \times 0.8 = 0.773504$$

$$l^{(3)} + (u^{(3)} - l^{(3)})0.8 = 0.7712 + 0.0288 \times 0.82 = 0.7735616$$

$$\bar{T}_X = 0.772352$$



- Séquence décodée: 1 3 **2**

Exemple déchiffrage du tag (8)

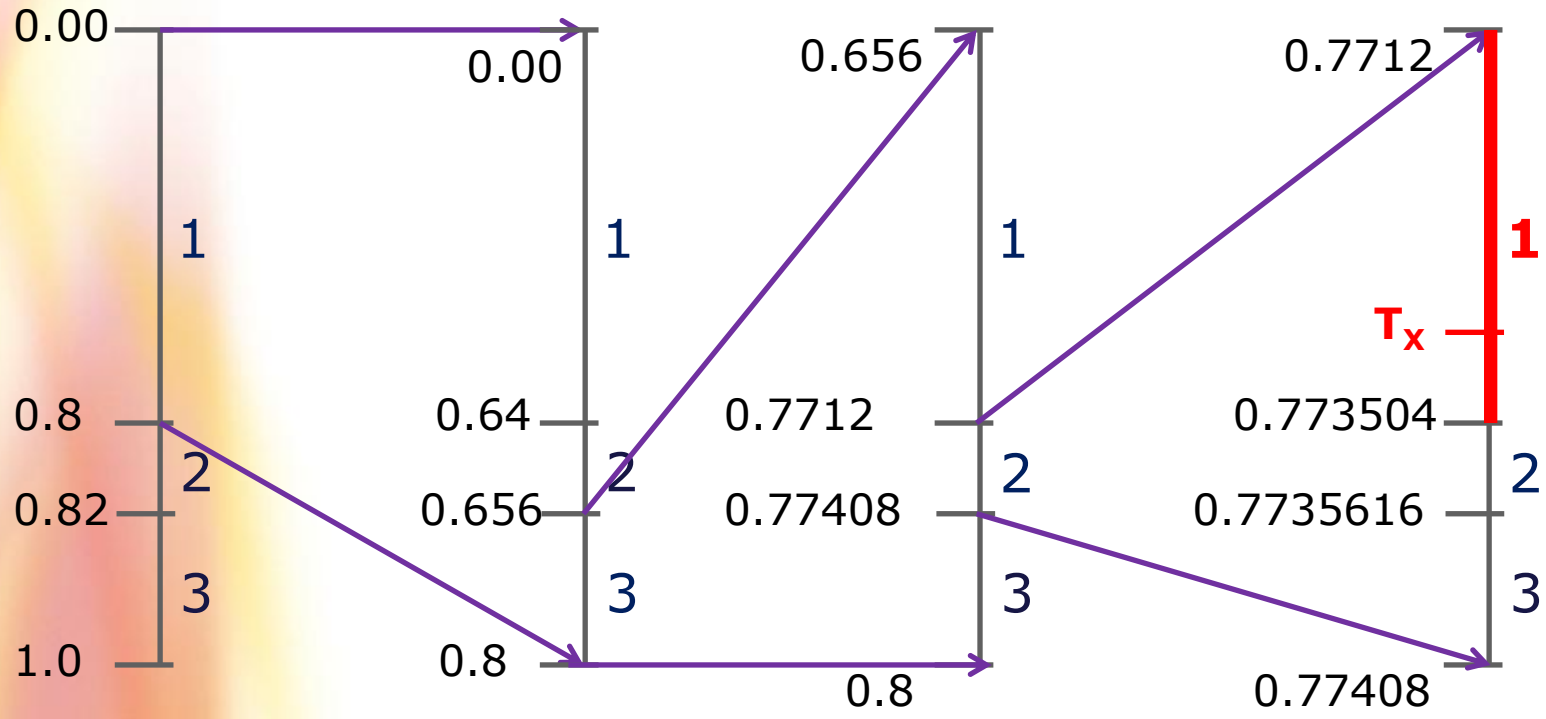
- Étape 3:

$$\bar{T}_X = 0.772352 \in [0.7712, 0.773504)$$
$$\bar{T}_X = 0.772352 \notin [0.773504, 0.7735616) \Rightarrow x_4 = 1$$
$$\bar{T}_X = 0.772352 \notin [0.7735616, 0.77408)$$

$$\bar{T}_X = 0.772352$$

Fin: **1321**

Peut on simplifier plus ?



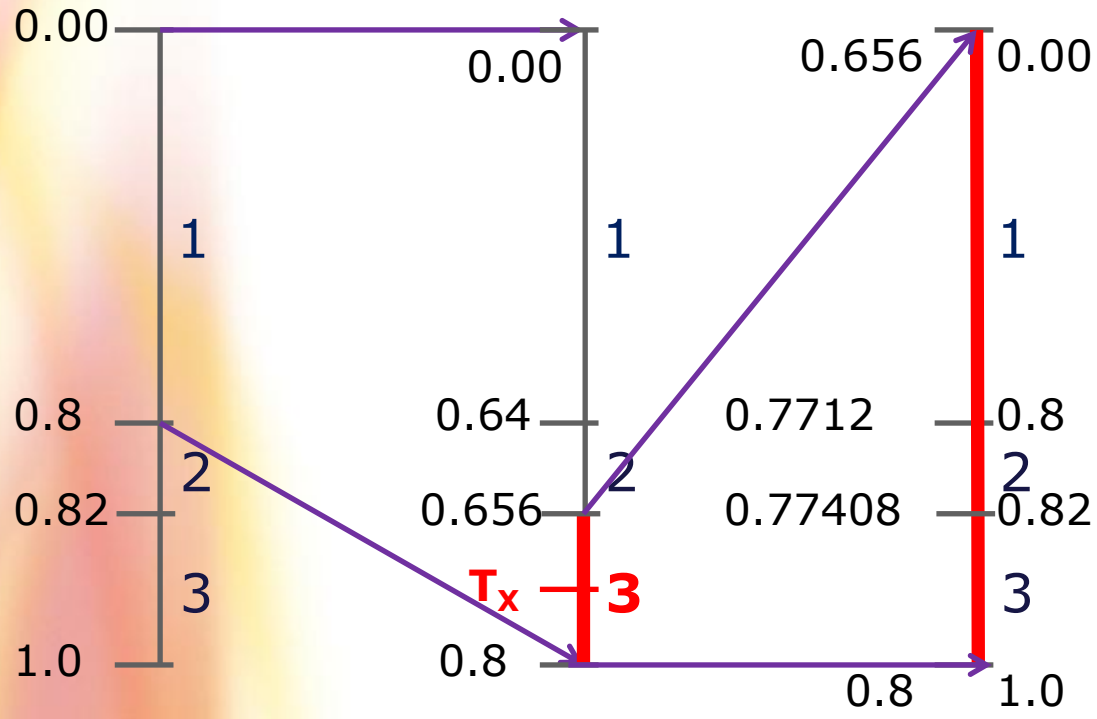
- Séquence décodée: 1 3 2 **1**

Exemple déchiffrage du tag (Simplification)[1]

- On ramène l'intervalle [0.656, 0.8) à [0,1)

$$0.656-0.656\ /0.144=0,\quad 0.7712-0.656\ /0.144=0.8$$
$$0.77408-0.656\ /0.144=0.82,\quad 0.8-0.656\ /0.144=1$$

$$\bar{T}_X = 0.772352$$
$$(\bar{T}_X - 0.652)/0.144 = 0.808$$



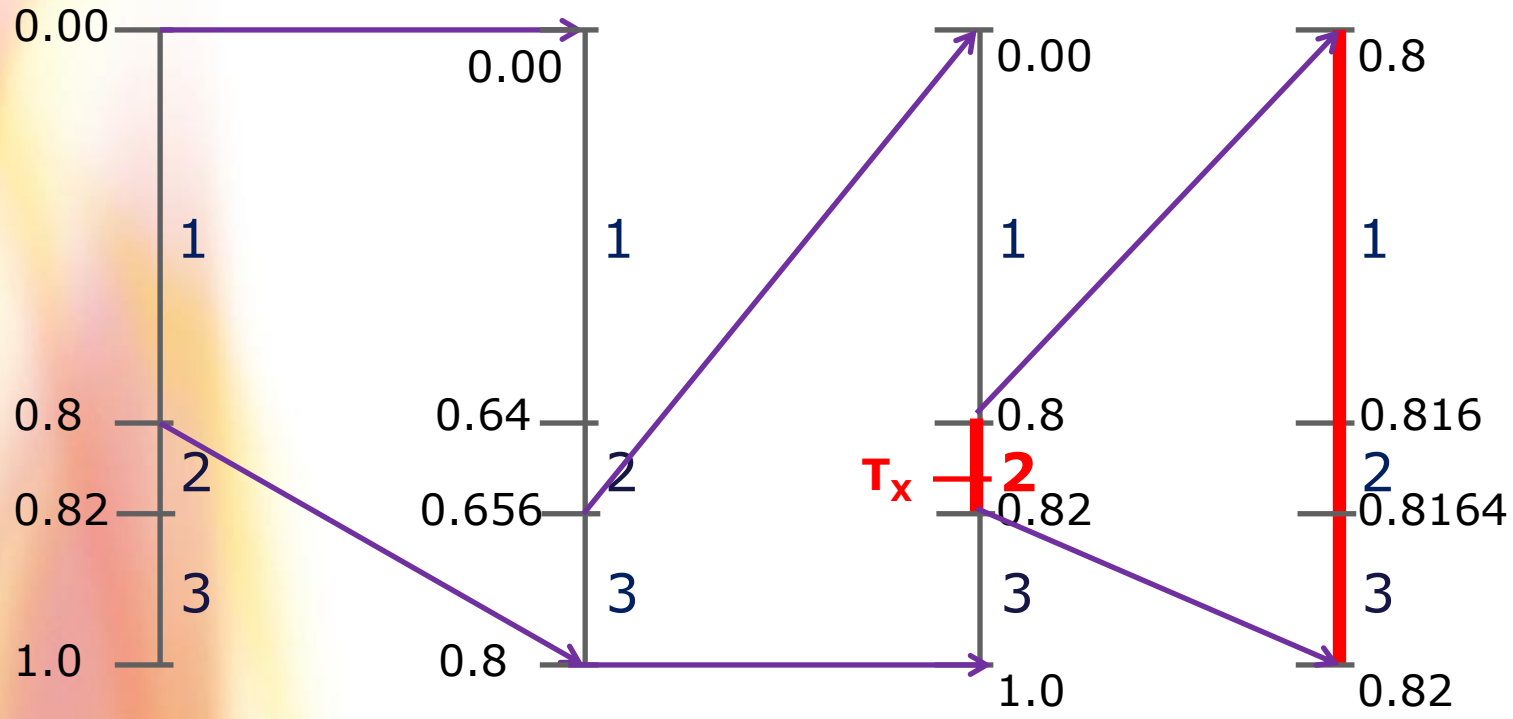
- Séquence décodée: 1 **3**

Exemple déchiffrage du tag (Simplification)[2]

- Remise à l'échelle

$$\bar{T}_X = 0.808$$

$$l^{(3)} + (u^{(3)} - l^{(3)})0.8 = 0.8 + 0.02 \times 0.8 = 0.816$$
$$l^{(3)} + (u^{(3)} - l^{(3)})0.8 = 0.8 + 0.02 \times 0.82 = 0.8164$$



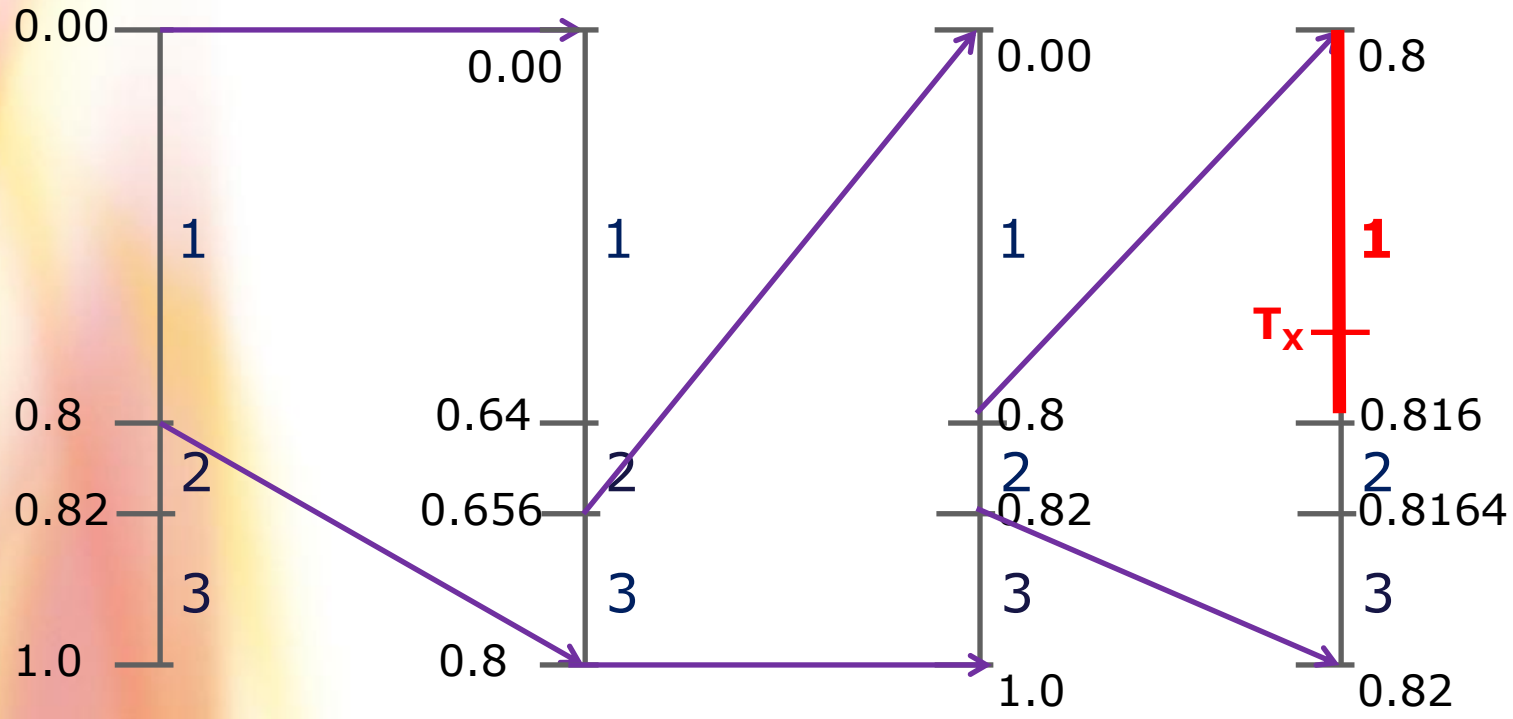
- Séquence décodée: 1 3 **2**

Exemple déchiffrage du tag (Simplification)[2]

- Remise à l'échelle

$$\bar{T}_X = 0.808$$

$$l^{(3)} + (u^{(3)} - l^{(3)})0.8 = 0.8 + 0.02 \times 0.8 = 0.816$$
$$l^{(3)} + (u^{(3)} - l^{(3)})0.8 = 0.8 + 0.02 \times 0.82 = 0.8164$$



- Séquence décodée: 1 3 2 **1**

Génération du code binaire