Codes de symbole

Un codeur (binaire) est une application

$$\begin{array}{rccc} C: & \mathcal{X} & \longrightarrow & \{0,1\}^+ \\ & x & \longrightarrow & c(x) \end{array}$$

Le décodeur D est une application des séquences binaires dans l'alphabet $\mathcal{X}$:

$$\begin{array}{rccc} D: & \{0,1\}^+ & \longrightarrow & \mathcal{X} \\ & c(x) & \longrightarrow & d(c(x)) \end{array}$$

Code (binaire) étendu, C^+

$X_i \in \mathcal{X}, i = 1, 2, \ldots$
$C\ :\ \mathcal{X} \to \{0,1\}^+$: code de symbole (binaire).
Le code étendu C^+ est :

$$\begin{array}{rccc} C^+ : & \mathcal{X}^+ & \longrightarrow & \{0,1\}^+ \\ & x = x_1 x_2 \cdots x_n & \to & c^+(x) = c(x_1)c(x_2)\cdots c(x_n). \end{array}$$

Codes uniquement décodables

C : code (de symbole) pour $X \in \mathcal{X}$
C est uniquement décodable si C^+ est une application inversible:

$$\forall x^{(n)}, y^{(m)} \in \mathcal{X}^+, \qquad x^{(n)} \neq y^{(m)} \Rightarrow c^+(x^{(n)}) \neq c^+(y^{(m)}).$$

Codes de préfixe

C : code pour $X \in \mathcal{X}$
C est un code de préfixe si aucun mot de code n'est le début d'un autre mot de code:

$$\text{Si } c_2 = c_1\, c, \quad \text{où} \quad c_1 \in C(\mathcal{X}), c \in \{0,1\}^+ \Rightarrow c_2 \notin C(\mathcal{X}).$$

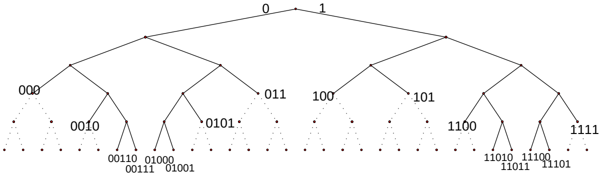
C est un code de préfixe $\Rightarrow C$ est uniquement décodable .

(Par construction, la terminaison de chaque mot de code $c(x_i)$ est bien définie.)

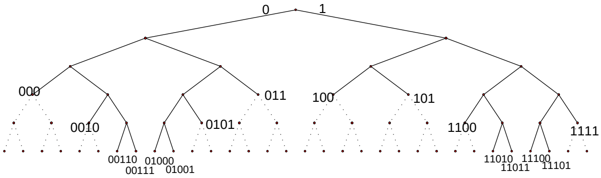
Codes de préfixe (binaires) et arbres (binaires)

À un code (binaire) de préfixe C nous pouvons toujours associer un arbre binaire, où tous les mots de code se situent sur les **feuilles** de l'arbre.

$$C(\mathcal{X}) = \{000, 100, 101, 0010, 0101, 1100, 1111,$$
$$00110, 00111, 01000, 01001, 11010, 11011, 11100, 11101\}\,.$$



C code de préfixe $\Rightarrow$ aucun autre mot de code ne peut être trouvé dans les sous-arbres qui ont comme racine un mot de code (les branches en pointillé dans la Figure).



N_n

$\ell(c) = n \Rightarrow c \in$ niveau n de l'arbre binaire. Le nombre maximal de mots de code de longueur n est

$$N_n = 2^n.$$

Nombre de déscendants au niveau n d'un noeud du niveau $m \leq n$ est 2^{n-m}.

Relation d'ordre dans $\{0,1\}^+$

Choix consistant des étiquettes des branches descendantes (e.g. gauche $\leftrightarrow$ 0, droite$\leftrightarrow$ 1)
$\Leftrightarrow$ séquences ordonnées par l'ordre *lexicographique* héritée de la relation d'ordre dans l'alphabet binaire ($0 < 1$) :

$$c_1, c_2 \in \{0,1\}^n$$

$$c_1 > c_2 \Leftrightarrow \exists i \in [1,n] \quad : \quad c_1(j) = c_2(j), \forall j < i \;\wedge\; c_1(i) > c_2(i)$$

Généralisation à $\{0,1\}^+$:
Soient c_1 et $c_2 \in \{0,1\}^n$, et $c_2 > c_1$. Soit $c' = c_1 c$, $c \in \{0,1\}^m$. Alors

$$c_1 < c' < c_2.$$

Exemple

$c_1, c_2 \in \{0,1\}^6$: $c_1 = 001101, \;\; c_2 = 001011$. Fixons $i = 4$. Alors

$$c_1(j) = c_2(j), i = 1, 2, 3 \;\text{ et :}\qquad c_1(4) = 1 > c_2(4) = 0.$$

$\Rightarrow c_1 > c_2$.

$\forall c$, $c' = c_2 c$ sont plus grandes que c_2, par exemple

$$c' = 00101100001 > c_2 = 001011\ ,$$

et plus petites que c_1 :

$$c' = 00101100001 < c_1 = 001101\ .$$

Inégalité de Kraft

Soient $X \in \mathcal{X}$ une source et C un code de symbole (binaire) uniquement décodable pour X.
Alors

$$\sum_{x\in\mathcal{X}} 2^{-\ell(c(x))} \leq 1.$$

Démonstration

Soit $\ell \geq \max_{x \in \mathcal{X}} \ell(c(x))$. Il existent 2^n séquences (binaires) distinctes de longueur ℓ. Nombre de descendants de $c(x)$ au niveau ℓ : $2^{\ell - \ell(c(x))}$. Comme les descendants sont disjoints, nous devons avoir

$$2^{\ell} \geq \sum_{x \in \mathcal{X}} 2^{\ell - \ell(c(x))} \Rightarrow \sum_{x \in \mathcal{X}} 2^{-\ell(c(x))} \leq 1.$$

kraft

Inégalité de Kraft-McMillan et codes de préfixe

Soit $\{n(x) > 0, x \in \mathcal{X}\}$, des entiers positifs qui vérifient l'inégalité de Kraft-McMillan):

$$\sum_{x \in \mathcal{X}} 2^{-n(x)} \leq 1.$$

Alors, il existe un code de préfixe C avec longueurs $\ell(x) = n(x)$. (Qui est donc, nécessairement, uniquement décodable.)

Nous allons choisir itérativement les mots de code $c(x), x \in \mathcal{X}$, par ordre lexicographique. *Par construction*, le code obtenu sera un code de préfixe.

Démonstration

Admettons que $n(x_1) \leq n(x_2) \leq \cdots \leq n(x_m)$.
Fixons : $c(x_1) = 0^{n(x_1)} = 0 \cdots 0 \in \{0,1\}^{n(x_1)}$, et $c(x_2)$ le plus petit succésseur de $c(x_1)$ de taille $n(x_2)$:

$$c(x_2) = 0^{n(x_1)-1}10^{n(x_2)-n(x_1)} \in \{0,1\}^{n(x_2)} \qquad (n(x_1)\text{-ème} = 1).$$

Pour $i = 3, \ldots, |\mathcal{X}|$

$$c(x_i) = \min_{c \in \{0,1\}^{n(x_i)}} \notin \text{descendants}\,\{c(x_1), \ldots c(x_{i-1})\}\,.$$

$\#\{$ mots de longueur $n_i = n(x_i)$ "libres"$\}$:

$$\begin{aligned} 2^{n_i} - \sum_{j<i} 2^{n_i - n(x_j)} &= 2^{n_i} - 2^{n_i} \sum_{j<i} 2^{n(x_j)} \\ &> 2^{n_i} - 2^{n_i} \sum_{j=1}^{n} 2^{-n(x_j)} > 2^{n_i} - 2^{n_i} = 0, \end{aligned}$$

(inégalité de Kraft)
$\Rightarrow$ il existe **au moins un** mot libre avec longueur $n(x_i)$.

Code complet

Un code C est complet, si la longueur de ses mots de code satisfait l'inégalité de Kraft-McMillan **avec égalité**

$$\sum_{x \in \mathcal{X}} 2^{-\ell(c(x))} = 1.$$

$\Rightarrow$ Pour tous les noeuds de l'arbre binaire correspondante, soit ils ont **les deux descendants** soit ils sont des **feuilles** de l'arbre.

Exemple d'un qui **n'est pas** complet

NotComplete

Code de Shannon

Soit X une source, $x \in \mathcal{X}$, $X \sim p$. Alors, il existe un code de préfixe avec longueurs

$$\ell(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil \qquad \forall x \in \mathcal{X}.$$

Démonstration
Il suffit de vérifier que l'inégalité de Kraft est vérifiée:

$$\sum_{x \in \mathcal{X}} 2^{-\lceil \log 1/p(x) \rceil} \leq \sum_{x \in \mathcal{X}} 2^{-(\log 1/p(x))} = \sum_{x \in \mathcal{X}} 2^{\log p(x)} = 1.$$

L'inégalité découle du fait que $\lceil x \rceil \geq x$.

Longueur moyenne, L_C

Soit C un code de symbole pour la source $X \in \mathcal{X}$. La longueur moyenne de C est

$$L_C(X) = \sum_{x \in \mathcal{X}} p(x)\ell(c(x)).$$

Théorème Codage source (codes de symbole)

Soit $X \in \mathcal{X}$, $X \sim p$, avec entropie $H(X)$. Alors, il existe un code de préfixe C qui code X avec longueur moyenne à moins d'un bit de entropie :

$$H(X) \leq L_C(X) < H(X) + 1.$$

Démonstration (borne supérieure) $L_C(X) < H(X) + 1$

Il suffit de calculer la longueur moyenne du code de préfixe dont nous avons démontré l'existance :

$$\begin{aligned} L_C(X) = \sum_{x\in\mathcal{X}} p(x)\lceil \log \frac{1}{p(x)} \rceil & < \sum_{x\in\mathcal{X}} p(x)\left(\log \frac{1}{p(x)} + 1\right) \\ & = H(X) + 1. \end{aligned}$$

où nous avons utilisé

$$\lceil x \rceil < x + 1.$$

Le code optimal doit avoir une longueur inférieure ou ègale à celui ci.

Redondance

La redondance d'un code C (de préfixe) par rapport à une source $X \sim p$, $X \in \mathcal{X}$, est *bornée inférieurement* par l'entropie relative :

$$\mathcal{R} = L_C(X) - H(X) \geq D(p||q_C) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q_C(x)},$$

où la loi q_C est

$$q_C(x) = \frac{2^{-\ell(c(x))}}{\sum_{x \in \mathcal{X}} 2^{-\ell(c(x))}} \qquad \forall x \in \mathcal{X}.$$

Démonstration

Soit

$$s = \sum_{x\in\mathcal{X}} 2^{-\ell(c(x))} \leq 1, \text{(par l'inégalité de Kraft).}$$

$$\Rightarrow 2^{-\ell(c(x))} = q_C(x)s \Rightarrow \ell(c(x)) = -\log q_C(x) - \log s$$

$$\begin{aligned}\Rightarrow \mathcal{R} &= L_C(X) - H(X) = \sum_{x\in\mathcal{X}} p(x)\ell(c(x)) - \sum_{x\in\mathcal{X}} p(x)\log\frac{1}{p(x)} \\ &= -\sum_{x\in\mathcal{X}} p(x)\log q_C(x) - \log s + \sum_{x\in\mathcal{X}} p(x)\log p(x)\end{aligned}$$

$$\Leftrightarrow \mathcal{R} = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q_C(x)} - \log s = D(p||q_C) - \log s$$

$s \leq 1 \Rightarrow \log s \leq 0$

$$\Rightarrow \mathcal{R} \geq D(p||q_C)$$

Démonstration (borne inférieure) $H(X) \leq L_C(X)$

Nous venons de voir

$$\mathcal{R} = L_C(X) - H(X) \geq D(p||q_C).$$

Comme l'entropie relative est toujours non négative

$$\mathcal{R} \geq 0 \Leftrightarrow L_C(X) - H(X) \geq 0 \Leftrightarrow L_C(X) \geq H(X).$$

Codes optimaux

Pour qu'un code C atteigne la longueur moyenne minimale

$$L_C(X) = H(X),$$

il faut que

$$\log s = 0 \Leftrightarrow s = 1,$$

c'est à dire, le code doit être complet **et**

$$D(p||q_C) = 0 \Leftrightarrow q_C(x) = p(x),$$

c'est à dire, la longueur des mots de code doit satisfaire

$$\ell(c(x)) = \log \frac{1}{p(x)}.$$

Lien entre codes et lois de probabilité

Nous avons établi un lien intime entre *lois de probabilité* et *longueurs de codes* optimaux : pour une loi p fixée, un ensemble de longueurs optimales peut lui être associé :

$$p \rightarrow \qquad \ell(x) = \log \frac{1}{p(x)}, \forall x \in \mathcal{X}.$$

Réciproquement, si nous avons un code C dont les mots ont longueurs $\ell(x)$, nous pouvons déterminer une loi de probabilité p pour laquelle le code est optimal :

$$C \rightarrow \qquad p(x) = 2^{-\ell(c(x))} \qquad \forall x \in \mathcal{X}.$$

Ce lien sera exploité – à travers la notion de *Code Universel* – par le *Principe de Longueur de Description Minimale* pour résoudre le problème de la *sélection de modèles* de compléxité différente.

Codage source (codes de bloc)

Soit X une source de symboles i.i.d, $X_i \in \mathcal{X}$, $X_i \sim p$. Alors $\forall \epsilon > 0$ il existe un code de bloc C_n de longueur moyenne (par symbole source) qui vérifie

$$L_{C_n}(X) \leq H(X) + \epsilon.$$

Démonstration

Application du Théorème du Codage Source à des blocs de taille n de symboles de la source X.
L'entropie d'un ensemble $X^{(n)}$ de n variables i.i.d. est

$$H(X^{(n)}) = nH(X).$$

Nous savons que :

$$L_C(X^{(n)}) \leq H(X^{(n)}) + 1 = nH(X) + 1.$$

La longueur *par symbole* de la source est

$$L_{C_n}(X) = \frac{L_C(X^{(n)})}{n} \leq H(X) + \frac{1}{n}.$$

Il suffit donc de prendre n de façon que

$$\frac{1}{n} \leq \epsilon \Leftrightarrow n \geq \frac{1}{\epsilon},$$

Soit $\mathcal{X} = \mathcal{X}_0$ l'alphabet de la source, $|\mathcal{X}| = m$, et $p_0 = p$ sa loi de probabilité.

$$p(a_1) \geq p(a_2) \geq \cdots \geq p(a_m).$$

Soit C un code optimal (de longueur moyenne minimale). Alors

$$\text{si } p_i > p_j \Rightarrow \ell(c_i) \leq \ell(c_j).$$

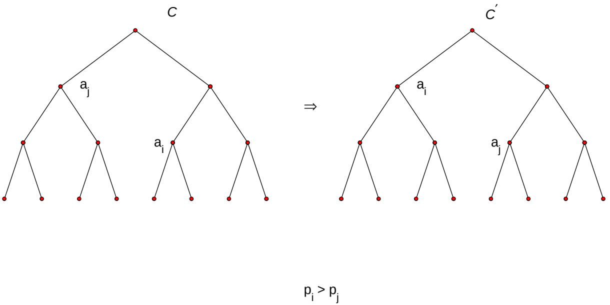
C'est à dire, les symboles les plus probables correspondent à des mots de code plus courts.

Démonstration

(Par absurde.) Admettez qu'il existent des symboles a_i et a_j pour lesquels

$$p_i - p_j > 0 \qquad \text{et} \qquad \ell(c_i) - \ell(c_j) > 0.$$

Nous allons voir que alors le code ne peut pas être optimal. Soit C' : $a_i \leftrightarrow a_j$



$$p_i - p_j > 0 \qquad \ell(c_i) - \ell(c_j) > 0$$

$$c'(a_n) = c(a_n), \forall n \neq i, n \neq j, \qquad c'(a_i) = c(a_j), c'(a_j) = c(a_i)$$

$$\ell(c'_n) = \ell(c_n), n \neq i, n \neq j, \qquad \ell(c'_i) = \ell(c_j), \ell(c'_j) = \ell(c_i)$$

$$\Rightarrow L_{C'}(X) = \sum_{n=1}^{m} p_n \ell(c'_n).$$

$$\begin{aligned}\Rightarrow L_C(X) - L_{C'}(X) &= \sum_{n=1}^{m} p_n \left[\ell(c_n) - \ell(c'_n)\right] \\ &= \sum_{n=i,j} p_n \left[\ell(c_n) - \ell(c'_n)\right] \\ &= p_i \left(\ell(c_i) - \ell(c_j)\right) + p_j \left(\ell(c_j) - \ell(c_i)\right) \\ &= \left(\ell(c_i) - \ell(c_j)\right)\left(p_i - p_j\right) > 0\end{aligned}$$

ce qui contrarie l'optimalité de C ($L_C > L_{C'}$).

Il existe un code C de longueur moyenne minimale tel que

$$\ell(c_1) \leq \ell(c_2) \leq \cdots \leq \ell(c_m).$$

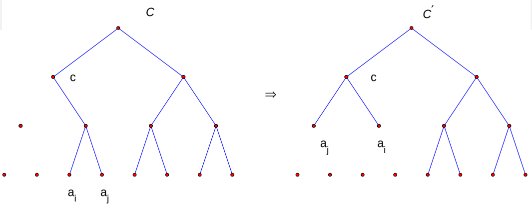
Démonstration
Par hypotèse : $i > j \Leftrightarrow p_i \leq p_j$
Nous venons de voir que C est optimal $\Rightarrow$

$$p_i < p_j \Rightarrow \ell(c_i) \geq \ell(c_j)$$

Si $p_i = p_j$ et $\ell(c_i) < \ell(c_j)$
$\Rightarrow a_i \leftrightarrow a_j \Rightarrow$ longueur moyenne n'est pas modifiée, et nous obtenons un code qui satisfait l'inégalité.
$\Rightarrow$
Il existe un code optimal C où le mot le plus long est $c(a_m)$.

L'arbre correspondant à un code optimal C ne possède pas des noeuds internes incomplets.

Démonstration(Par absurde)
Soit $c \in \{0,1\}^n$ un *noeud interne*. Si les deux descendants de c ne sont pas racines d'arbres qui contiennent des mots de C, nous pouvons construir un autre code de taille plus petite.



c : noeud interne avec une seule branche descendante
$\mathcal{C}_n^d$ l'ensemble des *descendants* de c.
Par hypothèse, il existe au moins un $c_i \in \mathcal{C}_n^d \Rightarrow$

$$c_i = c\ b\ c_i', \qquad b \in \{0,1\}, c_i' \in \{0,1\}^+,$$

$$\ell(c_i) = \ell(c) + 1 + \ell(c_i').$$

C' : maintient tous les mots de$C$, sauf pour $c_i \in \mathcal{C}_n^d$:

$$c_i = c\ b\ c_i' \rightarrow c_i' = c\ c_i', c_i' \in \{0,1\}^+$$

C' est un code de préfixe, et $L_{C'} < L_C$.

Il existe un code (binaire) optimal où les mots c_m et c_{m-1} ne difèrent que dans le dernier bit (ils forment un pair de "jumeaux").

Démonstration
Si c_m n'a pas de "jumeau" le code n'est pas optimal (son prédécesseur serait un noeud interne incomplet).
Si C est optimal il doit exister un mot de code c_i qui difère de c_m uniquement dans le dernier bit $\Rightarrow \ell(i) = \ell(m)$. Par hypothèse, $p_i \geq p_{m-1} \geq p_m$, $\Rightarrow \ell(m-1) = \ell(i) = \ell(m)$
$a_i \leftrightarrow a_{m-1} \rightarrow$ code de la même longueur qui satisfait la proposition.

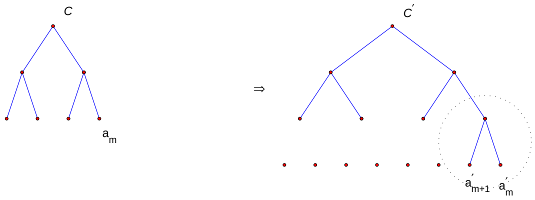
C code optimal pour $X \in \mathcal{X}$, $|\mathcal{X}| = m$, $X \sim p = [p_1 \cdots p_m]$
C' obtenu à partir de C :

$$c'(a_i) = c(a_i), i = 1, \ldots, m-1; \qquad c'(a'_m) = c\ 0, \qquad c'(a'_{m+1}) = c\ 1$$

est un code optimal pour

$$X' \in \mathcal{X}' = \{a_1, \ldots, a_{m-1}, a'_m, a'_{m+1}\}$$

$$X' \sim p' = [p_1 \ldots, p_{m-1}, p'_m, p'_{m+1}]$$
$$p'_m + p'_{m+1} = p_m.$$


C
a_m
$\Rightarrow$
C'
a'_{m+1}
a'_m

Démonstration (Par absurde)

$$L_{C'}(X') = L_C(X) + p'_{m+1} + p'_m.$$

Soit $Q' \neq C'$ un autre code, optimal pour X' :

$$L_{Q'} < L_{C'}$$

Q' est optimal $\Rightarrow q'_m$ et q'_{m+1} ne difèrent que dans le dernier bit.
Soit Q le code pour $X \in \mathcal{X}$, obtenu en associant à a_m le *préfixe commun* de q'_m et q'_{m+1}

$$\Rightarrow L_Q(X) = L_{Q'}(X') - \left(p'_m + p'_{m+1}\right).$$

$$\Rightarrow L_Q(X) < L_{C'}(X') - \left(p'_m + p'_{m+1}\right) = L_C(X)$$

Q optimal $\Rightarrow$ C ne peut pas être optimal

Réduction de $\mathcal{X}$

$X \in \mathcal{X}$, $|\mathcal{X}| = m$, $X \sim p$
$x = a_m,\ y = a_{m-1} \in \mathcal{X}$ les deux éléments *les moins probables* de $\mathcal{X}$
Soit $X_{i+1} \in \mathcal{X}_{i+1}$, $X_{i+1} \sim p^{i+1}$:

$$\begin{array}{rcl} \mathcal{X}_{i+1} & = & \mathcal{X}_i \backslash \{x, y\} \cup \{x'\} \qquad \mathcal{X}_o = \mathcal{X}, i = 1, \ldots, m-1 \\ p^{i+1} & = & \left[p_1^i, \ldots, p_{m-1}^i, (p_{m-1}^i + p_m^i)\right] \qquad p^0 = p, i = 1, \ldots, m \end{array}$$

x' : un nouveau symbole $x' \leftrightarrow (x, y)$.

$$|\mathcal{X}_{i+1}| = |\mathcal{X}_i| - 1 = m - i.$$

Algorithme de Huffman

Application répétée de la *réduction* :

$$\{\mathcal{X}, p\} \xrightarrow{\mathcal{R}_1} \left\{\mathcal{X}_1, p^1\right\} \xrightarrow{\mathcal{R}_2} \left\{\mathcal{X}_2, p^2\right\} \xrightarrow{\mathcal{R}_3} \cdots \xrightarrow{\mathcal{R}_{m-1}} \left\{\mathcal{X}_{m-1}, p^{m-1}\right\},$$

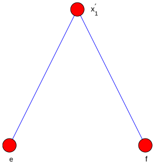
Par construction

$$|\mathcal{X}_{m-1}| = 1 \qquad p^{m-1} = 1.$$

Code optimal pour $\left\{\mathcal{X}_i, p^i\right\} \Rightarrow$ Code optimal pour $\left\{\mathcal{X}_{i-1}, p^{i-1}\right\}$

Code de Huffman

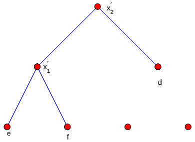
x	$p(x)$
a	0.3
b	0.25
c	0.2
d	0.1
e	0.08
f	0.07



$$\mathcal{X}_1 = \{a, b, c, x_1', d\}, \qquad x_1' \leftrightarrow (e, f)$$

$$p^1 = [0.3, 0.25, 0.2, 0.15, 0.1]$$

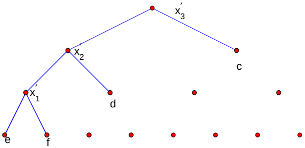
x	$p^1(x)$
a	0.3
b	0.25
c	0.2
x'_1	0.15
d	0.1



$$\mathcal{X}_2 = \{a, b, c, x'_2\}, \qquad x'_2 \leftrightarrow (x'_1, d)$$

$$p^2 = [0.3, 0.25, 0.2, 0.25]$$

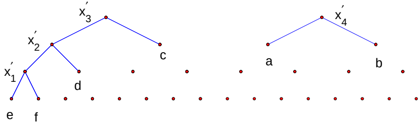
x	$p^2(x)$
a	0.3
b	0.25
x_2'	0.25
c	0.2



$$\mathcal{X}_3 = \{x_3', a, b\}, \qquad x_3' \leftrightarrow (x_2', c)$$

$$p^3 = [0.45, 0.3, 0.25],$$

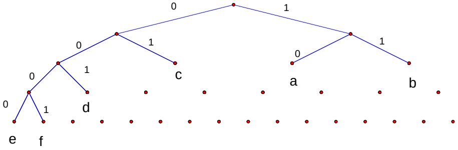
x	$p^3(x)$
x_3'	0.45
a	0.3
b	0.25



$$\mathcal{X}_4 = \{x_4', x_3'\}, \qquad x_4' \leftrightarrow (a, b)$$

$$p^4 = [0.55, 0.45]\,,$$

x	$p^4(x)$
x'_4	0.55
x'_3	0.45



$$\mathcal{X}_4 = \{x'_5\}, \qquad x'_5 \leftrightarrow (x'_4, x'_3)$$

$$p^5 = [1],$$

Code obtenu

x	$p(x)$	$c(x)$	$\ell(c(x))$
a	0.3	10	2
b	0.25	11	2
c	0.2	01	2
d	0.1	001	3
e	0.08	0001	4
f	0.07	0000	4

$$\begin{aligned} L_C(X) &= \sum_{x \in \{a,\cdots,f\}} \ell(c(x))p(x) \\ &= 2(0.3+0.25+0.2)+3\cdot 0.1+4(0.08+0.07) \\ &= 2.38 \text{ bits} \end{aligned}$$

$$H(X) = \sum_{x \in \{a,\cdots,f\}} p(x) \log \frac{1}{p(x)} = 2.377 \text{ bits}$$

$$H(X) < L_C(X) \leq H(X) + 1.$$