Université Paris 8 Année 2022–2023

M1 Mathématiques et applications, parcours ACC

Théorie de l'information – Solutions feuille de TD 3

05/10/2022

Retrouvez le sujet du TD et d'autres exercices à l'adresse :
`www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/theorie-information.html`

($\star$) exercice fondamental ($\star\star$) pour s'entraîner ($\star\star\star$) pour aller plus loin 💻 sur machine

Exercice 1. ($\star$) Arbre binaire d'un code.

On considère le code suivant sur l'alphabet $\mathcal{X} = \{a, b, c, d, e, f, g\}$

a	b	c	d	e	f	g
00	01	100	101	1100	1101	1111

Question 1.– Ce code est-il préfixe ? Est-il uniquement décodable ? Vérifie-t'il l'inégalité de Kraft ?

Question 2.– Décoder le texte encodé suivant, tout en essayant de bien comprendre comment vous arrivez à « séparer » les lettres :

010011110011111100.

Question 3.– Construire l'arbre binaire correspondant à ce code.

Question 4.– Est-il possible de rendre ce code plus efficace, quelle que soit la distribution de la source X sur $\mathcal{X}$? Si oui, que deviendrait le texte encodé de la question 2 ?

Solutions de l'Exercice 1.

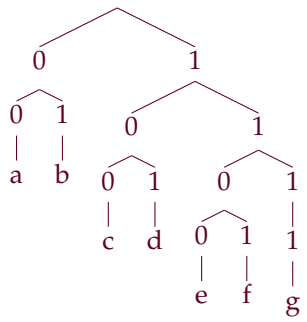
Solution Q1.

- Oui, ce code est préfixe : il n'existe aucune paire de mots distincts (u,v) telle que u est un préfixe de v.
- On peut en déduire qu'il est uniquement décodable, d'après un résultat du cours.
- Toujours d'après un théorème du cours, la séquence des longueurs du code doit donc satisfaire l'inégalité de Kraft. On peut également vérifier :

$$\sum_{i=1}^{7} 2^{-n_i} = 2 \times \frac{1}{4} + 2 \times \frac{1}{8} + 3 \times \frac{1}{16} = \frac{15}{16} < 1\,.$$

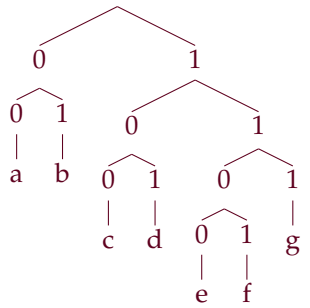
Solution Q2. On obtient le mot « bagage ».

Solution Q3. On obtient l'arbre binaire suivant.



Solution Q4.

On observe qu'il est possible d'obtenir un arbre strictement plus petit, en remontant d'un niveau la feuille portant l'étiquette g. On obtient alors l'arbre ci-contre :



Avec ce code « raccourci », le mot « bagage » s'encoderait en

$$0100111001111100$$

soit un gain de 2 bits.

Exercice 2. ($\star$) Codes préfixes et uniquement décodables.

Parmi les codes suivants, lesquels sont préfixes ? Lesquels sont uniquement décodables ?

1. $\{0, 10, 11\}$
2. $\{0, 01, 11\}$
3. $\{0, 01, 10\}$
4. $\{0, 101, 111, 100\}$
5. $\{00, 10, 100, 11, 110\}$

Solutions de l'Exercice 2.

1. Le code est préfixe, donc uniquement décodable.
2. Le code n'est pas préfixe : 0 est un préfixe de 01. En revanche, il est uniquement décodable. Cela peut être vu en « renversant » les mots de codes. On obtient le code $\{0, 10, 11\}$, qui lui est préfixe donc uniquement décodable. Or, l'opération de « renversement » ne modifie pas le caractère uniquement décodable d'un code. Plus formellement, si $C^+(x) = C(x_1) \dots C(x_n)$ et si l'on note $R(C(x_i))$ le renversé de $C(x_i)$, alors on observe que R est une bijection sur $\{0,1\}^+$, et que $R(C^+(x)) = R(C(x_n)) \dots R(C(x_1))$. Ensuite, le résultat provient du fait que C^+ est injective si et seulement si $R \circ C^+$ est injective.
3. Le code n'est pas préfixe car 0 est un préfixe de 01. Il n'est pas uniquement décodable, car 010 a deux antécédents par C^+ : les mots x_1x_3 et x_2x_1.
4. Le code est préfixe donc uniquement décodable.
5. Le code n'est pas préfixe : 10 est un préfixe de 100. Le code est uniquement décodable mais c'est plus difficile à démontrer. Soit $\boldsymbol{x} \in \{0,1\}^+$. On présente une procédure permettant de déterminer de manière unique les premiers mots qui constituent $\boldsymbol{x}$:
   - (A) si $\boldsymbol{x}$ commence par un 0, alors nécessairement $\boldsymbol{x}$ commence par 00, car 00 est le seul mot du code qui commence par un 0 ;
   - (B) si $\boldsymbol{x}$ commence par un 1, alors :
     - (B1) si ce 1 est suivi d'un nombre impair de 0, on doit décoder ce préfixe de $\boldsymbol{x}$ comme 10 suivi d'une suite de 00 ;
     - (B2) si ce 1 est suivi d'un nombre pair (non nul) de 0, on doit décoder ce préfixe de $\boldsymbol{x}$ comme 100 suivi d'une suite de 00 ;
     - (B3) si ce 1 est suivi d'un autre 1 :
       - (B3a) si $\boldsymbol{x}$ commence par un nombre impair $2m+1$ de 1, alors on peut éliminer m fois le préfixe 11, puis revenir au cas (B)
       - (B3b) si $\boldsymbol{x}$ commence par un nombre pair $2m$ de 1, alors :
         - (B3b1) si $\boldsymbol{x}$ commence par $2m$ fois 1 et un nombre impair $2p+1$ de 0, on ne peut décoder ce préfixe que comme $m-1$ fois 11, puis 110, puis p fois 00 ;
         - (B3b2) si $\boldsymbol{x}$ commence par $2m$ fois 1 et un nombre pair $2p$ de 0, on ne peut décoder ce préfixe que comme m fois 11, puis p fois 00.

Exercice 3. ($\star\star$) Code unaire.

Considérons le code suivant sur l'ensemble discret $\mathbb{N}$. À tout entier $n \in \mathbb{N}$, on associe la séquence de bits :

$$U(n) = \underbrace{0 \cdots\cdots 0}_{\longleftarrow n \longrightarrow} 1 \in \{0,1\}^{n+1}\,.$$

Question 1.– Démontrer que le code U est préfixe.

Question 2.– Sous quelle condition sur la source X, à valeurs dans $\mathbb{N}$, la longueur moyenne du code est-elle finie ?

Question 3.– Déterminer une source d'entropie finie pour laquelle le code U est optimal.

Solutions de l'Exercice 3.

Solution Q1. Pour tous $n < m$, le mot $U(n)$ est plus court que le mot $U(m)$, mais ce n'est pas un préfixe de $U(m)$: en effet, le $(n+1)$-ème bit de $U(n)$ est 1 tandis que le $(n+1)$-ème bit de $U(m)$ est 0..

Solution Q2. Soit $(p_n)_{n\in\mathbb{N}}$ la distribution de la source X. La longueur moyenne de U est égale à

$$\overline{\ell}(U) = \sum_{n=0}^{+\infty} p_n\,\ell(U(n)) = \sum_{n=0}^{+\infty} p_n(n+1) = \sum_{n=0}^{+\infty} np_n + \underbrace{\sum_{n=0}^{+\infty} p_n}_{=1}\,.$$

Donc $\overline{\ell}(U)$ est finie si et seulement si la série de terme général np_n converge, autrement dit si l'espérance de X est finie.

On peut donner des exemples (en n'oubliant pas qu'il faut également satisfaire la contrainte $\sum_n p_n = 1$).

— Les distributions telles que p_n est de la forme $p_n = C \cdot \frac{1}{n^\alpha}$, où $1 < \alpha < 2$ et $C > 0$, donnent une longueur moyenne infinie. Par exemple, c'est le cas de $p_n = \frac{6}{\pi^2} \cdot \frac{1}{n^2}$.

— Au contraire, celles où $p_n = C' \cdot \frac{1}{n^\beta}$ où $\beta > 2$ et $C' > 0$ donnent une longueur moyenne finie. C'est également le cas des distributions géométriques pour lesquelles $p_n = (1-\gamma)\gamma^n$ avec $0 < \gamma < 1$.

Solution Q3. Pour atteindre l'optimalité, on peut chercher X telle que $\overline{\ell}(U) = H(X)$. L'entropie s'écrit $H(X) = \sum_{n\in\mathbb{N}} p_n \log_2 \frac{1}{p_n}$ tandis que la longueur moyenne est de la forme $\overline{\ell}(X) = \sum_{n\in\mathbb{N}} p_n(n+1)$. Il suffit donc de choisir p_n telle que $n+1 = \log_2 \frac{1}{p_n}$, autrement dit :

$$p_n = 2^{-(n+1)}\,.$$

On vérifie également que $\sum_{n\in\mathbb{N}} p_n = 1$ pour que $(p_n)_{n\in\mathbb{N}}$ soit bien la distribution d'une variable aléatoire. D'ailleurs, on remarque qu'une telle variable suit la loi géométrique de paramètre $\gamma = \frac{1}{2}$.

Exercice 4. ($\star\star$) Code Morse.

Le code Morse international permet de transcrire un texte sous forme d'impulsions courtes ou longues. Il a été utilisé pour la télégraphie au XIXème siècle.

L'alphabet du code Morse est retranscrit avec deux symboles : l'impulsion courte « · » et l'impulsion longue « – ».

Voici un dictionnaire des encodages des lettres de l'alphabet courant :

A	$\cdot -$	B	$- \cdot\cdot\cdot$	C	$- \cdot - \cdot$	D	$- \cdot\cdot$	E	$\cdot$
F	$\cdot\cdot - \cdot$	G	$- - \cdot$	H	$\cdot\cdot\cdot\cdot$	I	$\cdot\cdot$	J	$\cdot - - -$
K	$- \cdot - \cdot$	L	$\cdot - \cdot\cdot$	M	$- -$	N	$- \cdot$	O	$- - -$
P	$\cdot - - \cdot$	Q	$- - \cdot -$	R	$\cdot - \cdot$	S	$\cdot\cdot\cdot$	T	$-$
U	$\cdot\cdot -$	V	$\cdot\cdot\cdot -$	W	$\cdot - -$	X	$- \cdot\cdot -$	Y	$- \cdot - -$
Z	$- - \cdot\cdot$								

Question 1.– Le code Morse est-il uniquement décodable ?

Question 2.– Calculer la longueur moyenne du code Morse en supposant que la loi de distribution des lettres est uniforme.

En français, après l'analyse de plusieurs œuvres littéraires, on estime que la distribution des lettres dans un texte suit approximativement la loi dont les probabilités sont données dans la table suivante :

A	8,15%	B	0,97%	C	3,15%	D	3,73%	E	17,39%
F	1,12%	G	0,97%	H	0,85%	I	7,31%	J	0,45%
K	0,02%	L	5,69%	M	2,87%	N	7,12%	O	5,28%
P	2,80%	Q	1,21%	R	6,64%	S	8,14%	T	7,22%
U	6,38%	V	1,64%	W	0,03%	X	0,41%	Y	0,28%
Z	0,15%								

Question 3.–

1. Calculer l'entropie de la langue française.
2. Calculer la longueur moyenne du code Morse pour un texte en français. Commenter.

En anglais, une analyse identique donne la distribution suivante :

A	8,08%	B	1,67%	C	3,18%	D	3,99%	E	12,56%
F	2,17%	G	1,80%	H	5,27%	I	7,24%	J	0,14%
K	0,63%	L	4,04%	M	2,60%	N	7,38%	O	7,47%
P	1,91%	Q	0,09%	R	6,42%	S	6,59%	T	9,15%
U	2,79%	V	1,00%	W	1,89%	X	0,21%	Y	1,65%
Z	0,07%								

Question 4.–

1. Calculer l'entropie de la langue anglaise.
2. Calculer la longueur moyenne du code Morse pour un texte en français. Commenter.

Solutions de l'Exercice 4.

Solution Q1. Non, il n'est pas uniquement décodable. Par exemple, $- \cdot$ peut représenter à la fois « N » et « TE ».

Pour les réponses suivantes, voir le fichier `python` associé.

Solution Q2. La longueur moyenne du code Morse sur une source uniforme est $\simeq 3{,}19$.

Solution Q3. L'entropie de la langue française est $\simeq 3{,}99$, et la longueur moyenne du code Morse sur la langue française est $\simeq 2{,}44$. Ce n'est pas en contradiction avec le cours, car on rappelle que le code Morse n'est pas uniquement décodable.

Solution Q4. L'entropie de la langue anglaise est $\simeq 4{,}16$, et la longueur moyenne du code Morse sur la langue anglaise est $\simeq 2{,}53$. Même commentaire que pour la Question 3.

Exercice 5. ($\star\star\star$) Formalisation des arbres binaires.

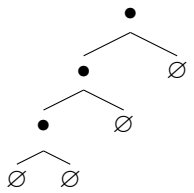
Cet exercice a pour but de démontrer quelques propriétés sur les arbres binaires, qui peuvent être utiles au cours.

Formellement, on peut donner la définition suivante d'un arbre binaire :

— l'arbre vide (sans racine), noté $\varnothing$, est un arbre binaire,
— si G et D sont deux arbres binaires, alors le couple (G, D) est un arbre binaire.

Par exemple, l'arbre binaire $(\varnothing, \varnothing)$ est l'arbre réduit à une racine. C'est l'arbre :

Autre exemple, l'arbre $(((\varnothing, \varnothing), \varnothing), \varnothing)$ peut-être représenté ainsi :



Descendants. Si $T = (G, D)$ est un arbre binaire non-vide, on appelle G le descendant gauche de T et D son descendant droit. Si $G = D = \varnothing$, alors on dit que T est une feuille. Sinon, c'est un nœud interne. On note $n(T)$ son nombre de nœuds internes d'un arbre T et $f(T)$ son nombre de feuilles. Exprimé plus formellement :

— si $T = \varnothing$, alors $f(T) = 0$ et $n(T) = 0$ (par convention)
— si $T = (\varnothing, \varnothing)$, alors $f(T) = 1$ et $n(T) = 0$
— sinon, pour $T = (G, D)$ avec G ou D non-vide, alors on définit par induction $f(T) = f(G) + f(D)$ et $n(T) = 1 + n(G) + n(D)$.

Enfin, on note $v(T) = n(T) + f(T)$ le nombre total de nœuds de T.

Hauteur. La hauteur d'un arbre binaire est définie par une relation de récurrence :

— par convention, l'arbre binaire vide a hauteur $h(\varnothing) = -1$;
— si $T = (G, D)$, alors $h(T) := 1 + \max\{h(G), h(D)\}$.

Pour revenir aux exemples précédents, l'arbre $T = (\varnothing, \varnothing)$ réduit à une racine possède $f(T) = 1$ feuille, aucun nœud interne ($n(T) = 0$) et sa hauteur est $h(T) = 0$. D'autre part, l'arbre $T = (((\varnothing, \varnothing), \varnothing), \varnothing)$ possède $f(T) = 1$ feuille, $n(T) = 2$ nœuds internes et sa hauteur est $h(T) = 2$.

Question 1.– Vérifier que dans le cas général, le nombre de nœuds total $v(T)$ satisfait :

— si $T = \varnothing$, alors $v(T) = 0$;
— si $T = (G, D)$, alors $v(T) = 1 + v(G) + v(D)$.

Question 2.– Démontrer par récurrence que si T est non-vide, alors $h(T) \leq n(T) + 1 \leq 2^{h(T)}$. Donner la forme des arbres pour les cas d'égalité.

Question 3.– Démontrer par récurrence que $1 \leq f(T) \leq 2^{h(T)}$. Donner la forme des arbres pour les cas d'égalité.

On note maintenant $n_1(T)$ le nombre de nœuds internes ayant 1 descendant, et $n_2(T)$ le nombre de nœuds internes ayant 2 descendants.

Question 4.–

1. Quelle est la relation entre $n(T)$, $n_1(T)$ et $n_2(T)$?
2. Donner un exemple d'arbre T avec $n_1(T) = 0$ et $n(T) = 3$.
3. Donner un exemple d'arbre T avec $n_2(T) = 0$ et $n(T) = 3$.

Question 5.– Démontrer par récurrence que $f(T) = n_2(T) + 1$.

Solutions de l'Exercice 5.

Solution Q1. D'après la définition de $v(T) := n(T) + f(T)$ et les relations de récurrence qui régissent n et f, on a bien :

– d'une part $v(\varnothing) = n(\varnothing) + f(\varnothing) = 0 + 0 = 0$.
– d'autre part, si $T = (\varnothing, \varnothing)$, alors $v(T) = n(T) + f(T) = 0 + 1 = 1 = 1 + v(\varnothing) + v(\varnothing)$.
– enfin, si $T = (G, D)$ avec G ou D non-vide, alors $v(T) = n(T) + f(T) = 1 + n(G) + n(D) + f(G) + f(D) = 1 + v(G) + v(D)$.

Solution Q2. On effectue ensuite la récurrence sur la hauteur de l'arbre.

- Si $h(T) = 0$, cela signifie que T est réduit à une feuille, donc T n'a pas de nœud interne. Par conséquent on a bien $0 \leq \underbrace{n(T)}_{=0} + 1 \leq 2^0 = 1$.
- Supposons le résultat vrai pour toute hauteur $\leq h$. Soit $T = (G, D)$ un arbre de hauteur $h + 1$. Alors G et D ont hauteur $\leq h$, et l'un d'entre eux au moins a hauteur h.
  — Si G est vide, alors c'est D qui a hauteur h, et on a $n(T) = n(D) + 1$ et $h(T) = h(D) + 1$. On applique l'hypothèse de récurrence sur D et on obtient le résultat :

$$h(T) = h(D) + 1 \leq n(D) + 2 = n(T) + 1 \leq 2^{h(D)} + 1 \leq 2^{h(D)+1} = 2^{h(T)}.$$

  — Si D, c'est le même raisonnement.

— Sinon, G et D ont hauteur ≥ 0 et on peut leur appliquer les hypothèses de récurrence. On a alors :

$$n(T)+1 = (1+n(G)+n(D))+1 = (1+n(G))+(1+n(D)) \leq 2^h + 2^h \leq 2^{h+1}\,.$$

D'autre part, observons que

$$n(T)+1 = (1+n(G))+(1+n(D)) \geq h(G)+h(D) = \max\{h(G),h(D)\} + \min\{h(G),h(D)\}\,.$$

Le résultat est donc établi si $\min\{h(G),h(D)\} \geq 1$, autrement dit si D et G ne sont pas de hauteur nulle. Maintenant, si $h(D)=0$ (par exemple), cela signifie que c'est un arbre réduit à une feuille, donc il vérifie $n(D)=0=h(D)$. Dans ce cas, on a alors $n(T)+1 = (1+n(G))+(1+n(D)) = 2+n(G) \geq 1+h(G) = 1+\max\{h(G),h(D)\}$.

Pour le cas d'égalité à gauche, on a un arbre linéaire, avec un seul nœud à chaque profondeur. Pour le cas d'égalité à droite, on a un arbre parfait, c'est-à-dire complètement rempli.

Solution Q3. En cours d'écriture

Solution Q4. En cours d'écriture

Solution Q5. En cours d'écriture