Codage et entropie

Théorie de Shannon (1948)

Signal = suite $x_0, x_1, x_2, \dots$ d'éléments d'un ensemble fini $A = (a_1, \dots, a_K)$ l'alphabet.

Ex : • un texte est un signal sur l'alphabet de taille finie (a ... z, A ... Z, _, 0 ... 9, . ; ; ! ?)

• Un texte en français est un signal sur l'alphabet $A = \{$ mots français $\}$. (85 000 mots dans le lexique)

Plutôt que de transmettre chaque symbole de l'alphabet sans codage est-ce que on peut coder chaque symbole de manière à ce que les symboles fréquents aient des codes courts de manière à transmettre la même quantité d'informations en transmettant un message + court ???

On se donne un modèle du message à transmettre c'est une probabilité sur $A = (a_i)_{1 \le i \le K}$ et on se donne $(p_{a_i})_{1 \le i \le k}$ $\quad 0 \le p_{a_i} \le 1$ et $\sum_{i=1}^{K} p_{a_i} = 1$

On suppose qu'on essaye de transmettre

Un message $(X_1, X_2, \dots)$ où les X_i sont des v.a. indep à valeur dans A et de loi donnée par p c'est à dire $\mathbb{P}(X_j = a) = p_a$

Méthode naïve : Si $2^{p-1} < K \leq 2^p$

alors A en bijection avec une partie de $\{0;1\}^p$ on peut coder chaque lettre de A par un élément $c(a)$ de $\{0;1\}^p$

à un message $x_1 \dots x_n$ je transmet $c(x_1) \dots c(x_n)$

le nombre moyen R_n de bits $\{0;1\}$ transmis vérifie $R_n = np$.

Ici $R_n = n(\lfloor \log_2 K \rfloor + 1)$

Et asymptotiquement le nombre moyen de bits transmis par lettre est

$$R = \lim_n \frac{R_n}{n} = \lfloor \log_2 K \rfloor + 1.$$

Objectif : Trouver d'autres codages pour lesquels R_n et R seront + petit.

Def : * $A = (a_i)_{1 \leq i \leq K}$ alphabet fini

* $p = (p_{a_i})_{1 \leq i \leq K}$ proba sur A.

* $p(x_1 \dots x_n) = p_{x_1} \dots p_{x_n}$ proba d'un message

$= \mathbb{P}((X_1 \dots X_n) = (x_1 \dots x_n))$.

* l'entropie de la distribution p :

$$H = -\sum_i p_{a_i} \log_2 p_{a_i}$$

Rem : Motivation pour cette définition :

Imaginons un ensemble $\Omega = \{0;1\}^N$ muni de la proba uniforme $\mathbb{P}(\{x\}) = \frac{1}{2^N}$.

Je me fixe un élément x de Ω

Combien de bits d'information je dois transmettre pour vous dire de quel x il s'agit ? $x = (x_1, x_2, \dots x_N)$

Il faut vous transmettre N informations

$x_1 \in \{0,1\}$, $x_2 \in \{0;1\}$ ..... $N = \log_2 |\Omega|$

$= -\log_2 \mathbb{P}(\{x\})$

$A = \{x \in \Omega : x_1 = \varepsilon_1, \dots, x_p = \varepsilon_p\}$.

Combien de bits d'information je dois transmettre pour préciser qu'un élément donné est dans A ?

Je dois dire $x_1 = \varepsilon_1 \dots x_p = \varepsilon_p$

donc p informations $A = \{\varepsilon_1\} \times \{\varepsilon_2\} \times \dots \{\varepsilon_p\} \times \{0;1\}^{N-p}$

On $\mathbb{P}(A) = \frac{|A|}{|\Omega|} = \frac{2^{N-p}}{2^N} = 2^{-p}$

Il faut $-\log_2 \mathbb{P}(A)$ informations.

En général : $-\log_2 \mathbb{P}(A)$ est le nombre de bits d'informations à transmettre pour dire qu'un éléments est dans A.

Sur mon alphabet

$-\log_2 p_a$ = nbre d'informations à transmettre pour dire qu'on a a (si a représentant une partie d'un $\Sigma = \{0;1\}^N$)

$$H = \sum_{a \in A} p_a (-\log_2 p_a) = \mathbb{E}[-\log_2 X]$$

où X de loi p

= quantité d'information moyenne d'une lettre.

Rem : $\boxed{H = -\sum_{a \in A} p_a \underbrace{\log_2 p_a}_{\leq 0 \text{ car } 0 \leq p_a \leq 1}}$

- $H \geq 0$.
- $(x \log x)'' = (\log x + 1) = \frac{1}{x} \geq 0$ si $x > 0$

donc $\psi(x) = -x \log x$ est concave. et même strictement concave $\frac{1}{x} > 0$ sur $]0;1[$.

Donc : $H = \sum_{a \in A} \psi(p_a) = K \sum_{a \in A} \frac{1}{K} \psi(p_a)$

$\leq K \psi\left(\sum_{a \in A} \frac{1}{K} p_a\right)$ $(K = |A|)$

concavité

$= K \psi\left(\frac{1}{K} \underbrace{\sum_{a \in A} p_a}_{=1}\right)$

$= K\left(-\frac{1}{K} \log_2 \frac{1}{K}\right)$

$= \log_2 K$

Donc on a $\boxed{0 \leq H \leq \log_2 K.}$

Le cas d'égalité à droite a lieu si p_a constant : $p_a = \frac{1}{K}$ $\forall a$.

Donc borne de droite :

$H = \log_2 K \iff \forall a \quad p_a = 1/K$ (proba uniforme)

borne de gauche :

$H = 0 \iff \forall a \quad p_a \log_2 p_a = 0$

$\iff \forall a \quad p_a \in \{0; 1\}$.

$\iff \exists a_0 : \quad p_a = \mathbb{1}_{a=a_0}$

$H = 0 \iff p$ est un dirac.

H mesure la quantité de désordre d'aléa du signal.

$H = 0 \iff$ signal déterministe

$H = \log_2 K \iff$ désordre maximal. loi uniforme.

Prop : $X_1, X_2, \ldots$ v.o. indep $\in A$ de loi p.

$p(X_1, \ldots X_n) = p_{X_1} \cdots p_{X_n}$

On : $(p(x_1 \ldots x_n) = \mathbb{P}(X_1 = x_1, \ldots X_n = x_n))$

$-\frac{1}{n} \log_2 p(X_1, \ldots, X_n) \xrightarrow[n \to \infty]{} H$ avec proba 1.

Rem : Lorsque on regarde un long message $X_1 \ldots X_n$ la proba de l'observer est $\approx 2^{-nH}$

Preuve : repose la loi forte des grands nombres

LFGN : Si $Y_1, Y_2, \dots Y_n$ sont intégrables et indép. avec même loi
Alors $\frac{1}{n}(Y_1 + \dots + Y_n) \xrightarrow[n \to \infty]{} E[Y_i]$ avec proba 1.

Ici : $-\frac{1}{n} \log_2 p(X_1 \dots X_n)$ ⚠ $p_a \in \mathbb{R}$, P_{X_i} v.a.

$= \frac{1}{n}(Y_1 + \dots + Y_n)$

avec $\boxed{Y_i = -\log_2 P_{X_i}}$ les Y_i sont indép

et Y_i intégrable ? $E[|\underbrace{\log_2 P_{X_i}}|]$

$\in \{\log_2 p_a\}_{a \in A}$

et $\mathbb{P}(|\log_2 P_{X_i}| = \infty) = \sum_{\substack{a \\ p_a = 0}} \underbrace{\mathbb{P}(X_i = a)}_{p_a} = 0$

Donc $E[|\log_2 P_{X_i}|]$ prend un nombre fini de valeurs fini avec proba 1 donc $< \infty$

par la LFGN

Donc $-\frac{1}{n} \log_2 p(X_1 \dots X_n) \xrightarrow[\text{proba 1}]{n \to \infty} E[-\log_2 P_{X_i}]$

$= -\sum p_a \log_2 p_a$

$= H$

Notation : Avec grande proba toutes les $X_1 \dots X_n$ que l'on observe avait proba $\approx 2^{-nH}$ d'arriver.

On veut définir des suites "typiques" :

$$T_\varepsilon^n = \left\{(x_1, \dots x_n) \in A^n \,\middle|\, H - \varepsilon \leq -\frac{1}{n} \log_2 p(x_1 \dots x_n) \leq H + \varepsilon\right\}$$

$$= \left\{(x_1, \dots, x_n) \in A^n \,\middle|\, 2^{-n(H+\varepsilon)} \leq p(x_1, \dots, x_n) \leq 2^{-n(H-\varepsilon)}\right\}$$

La proposition dit qu'on est "typique" avec

grande proba :

$$\mathbb{P}(X_1, \dots, X_n \in T_\varepsilon^n) \xrightarrow[n \to \infty]{} 1$$

(La proposition qu'on a montré est plus forte que ça).

En pratique : on ne voit pas n'importe quelle suite de A^n mais seulement celles de T_ε^n

Théorème $(1-\varepsilon)2^{n(H-\varepsilon)} \leq |T_\varepsilon^n| \leq 2^{n(H+\varepsilon)}$ $\forall \varepsilon$ si $n \geq N_\varepsilon$

(cardinal)

Dém : $\exists N_\varepsilon$ tel que $n \geq N_\varepsilon$:

$$1-\varepsilon \leq \mathbb{P}(X_1, \dots, X_n \in T_\varepsilon^n) \leq 1$$

Rappel : $x_1 \dots x_n \in T_\varepsilon^n \iff p_{x_1 \dots x_n} \in [2^{-n(H+\varepsilon)}; 2^{-n(H-\varepsilon)}]$

- $1 \geq \mathbb{P}(X_1, \dots, X_n \in T_\varepsilon^n) = \sum_{x_1 \dots x_n \in T_\varepsilon^n} \underbrace{\mathbb{P}(X_1 \dots X_n = x_1 \dots x_n)}_{p_{x_1 \dots x_n}}$

$$\geq \sum_{x_1 \dots x_n \in T_\varepsilon^n} 2^{-n(H+\varepsilon)}$$

$$= |T_\varepsilon^n| \, 2^{-n(H+\varepsilon)}$$

Donc $|T_\varepsilon^n| \leq 2^{n(H+\varepsilon)}$ ✓

- $1-\varepsilon \leq \mathbb{P}(X_1 \dots X_n \in T_\varepsilon^n) \leq \sum_{x_1 \dots x_n \in T_\varepsilon^n} p_{x_1 \dots x_n}$

$$\leq |T_\varepsilon^n| \, 2^{-n(H-\varepsilon)}.$$

Donc $|T_\varepsilon^n| \geq (1-\varepsilon) 2^{n(H-\varepsilon)}$.

∎

Conséquence : codage : Si on a un message $x_1 \dots x_n$ on le code de la manière suivante

* Si $(x_1 \dots x_n) \in T_\varepsilon^n$ on le code comme
  $0\ \varphi(x_1 \dots x_n)$

$|T_\varepsilon^n| \leq 2^{n(H+\varepsilon)}$

T_ε^n s'injecte dans $\{0;1\}^{\lfloor n(H+\varepsilon)\rfloor+1}$

$\varphi : T_\varepsilon^n \to \{0;1\}^{\lfloor n(H+\varepsilon)\rfloor+1}$

* Si $x_1 \dots x_n \notin T_\varepsilon^n$

$A^n \xrightarrow[\text{inj}]{\psi} \{0;1\}^{n\lfloor \log_2 K\rfloor+1}$ (codage naïf)

on le code comme
$1\ \psi(x_1 \dots x_n)$

Remarques : • Facile à décoder $0\, z_1 \dots z_m \to \varphi^{-1}(z_1 \dots z_m)$

$1\, z_1 \dots z_m \to \psi^{-1}(z_1 \dots z_m)$

• R_n = nbre moyen de bits produits.

$= 1 + (\lfloor n(H+\varepsilon)\rfloor+1)\,\mathbb{P}(X_1 \dots X_n \in T_\varepsilon^n)$

1 ↑ 0 ou 1 $+ n(\lfloor \log_2 K\rfloor+1)\,\mathbb{P}(X_1 \dots X_n \notin T_\varepsilon^n)$

pour n grand $\leq \varepsilon$

$\leq 1 + (\lfloor n(H+\varepsilon)\rfloor+1)\,1 + n(\lfloor \log_2 K\rfloor+1)\,\varepsilon$

$\leq 2 + nH + n\varepsilon\varepsilon$

Donc le nombre moyen de bits par lettre est :

$$R = \lim_n \frac{R_n}{n} \leq H + C\varepsilon.$$

Le codage naïf donnait $R \simeq \log_2 K$.

Ici on a R arbitrairement proche de H et on a vu $H \leq \log_2 R$

On verra qu'on ne peut pas avoir de codage avec $R < H$.

⟶ codage optimal.

Mais * codage pas explicite dépend de ℓ et
* il dépend de n. Si on rajoute des lettres au message il faut le codage depuis le début (pas un codage lettre à lettre)

Codage de Huffmann :

- codage lettre à lettre : on construit

$$\ell : A \longrightarrow \bigcup_{n \geq 0} \{0;1\}^n$$

$\ell(a)$ est le code de a

$x_1 \dots x_n$ codé en $\ell(x_1) \dots \ell(x_n)$

On ne peut pas forcément décoder un tel code : Si $A = \{a, b, c\}$

$\ell(a) = 01$ $\ell(b) = 011$ $\ell(c) = 1$

et que l'on reçoit 011 on ne sait pas si le message initial est ac ou b

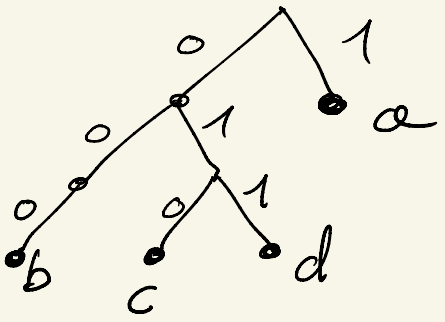
On demande à ℓ d'être "sans préfixe" c'est à dire que $\ell(a_i)$ n'est jamais

préfixe de $c(a_j)$ pour $j \neq i$ $(c(a_j) \neq c(a_i)w \ \forall w)$

Un tel codage peut être représenté sur un arbre binaire :

Si $c(a) = 1$ $c(b) = 000$

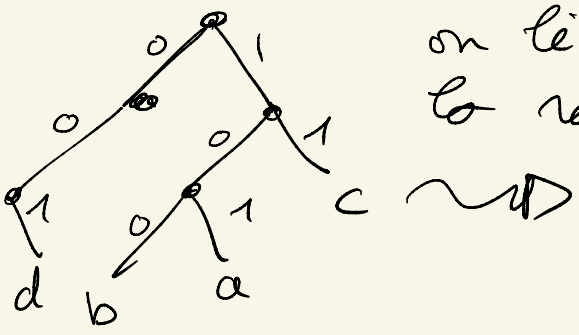
$c(c) = 010$ $c(d) = 011$



On place chaque lettre dans l'arbre à la position de son code

La condition sans préfixe ($\Leftarrow$) aucune lettre n'est descendante d'une autre

Réciproquement : si on se donne un arbre binaire avec les lettres sur les feuilles on en déduit un code sans préfixe :



on lit le code en descendant de la racine :

$c(a) = 101$

$c(b) = 100$

$c(c) = 11$

$c(d) = 001$

Code de Huffman : on a un alphabet $A = (a_i)_{1 \leq i \leq K}$ et une proba $(p_{a_i})_{1 \leq i \leq K}$

on classe les lettres de manière à avoir

$p_{a_1} \leq p_{a_2} \leq \dots\dots$

on pose $a_{12} = \{a_1, a_2\}$ on construit l'arbre de Huffman sur

$a_{12}, a_3, a_4 \dots$ avec $p_{a_{12}} = p_{a_1} + p_{a_2}$

L'arbre final s'obtient en groupant a_1 et a_2

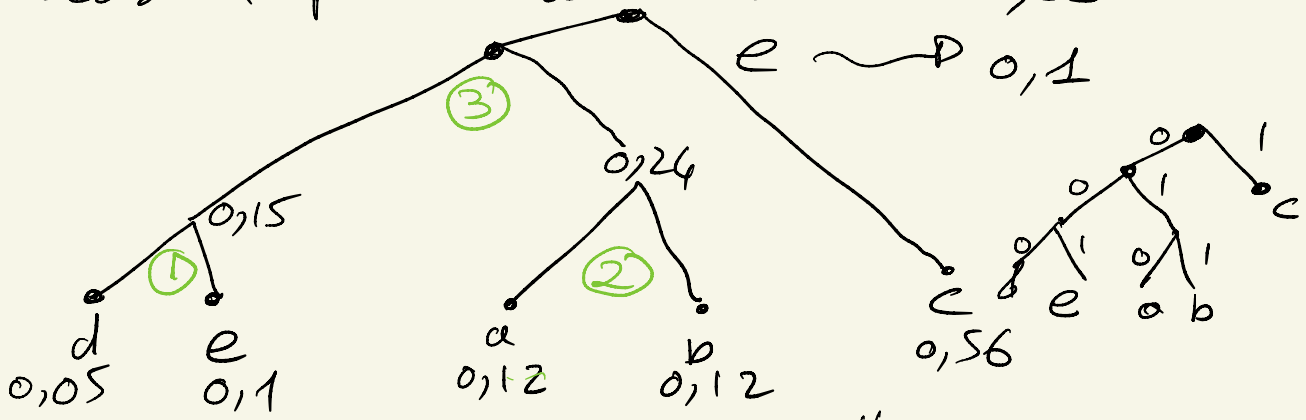
Puis on continue sur le nouvel alphabet

Ex : $A = \{a; b; c; d; e\}$.

$p = \{0,12 ; 0,12 ; 0,56 ; 0,05 ; 0,1\}$

Les deux + petits sont d ⇝ 0,05

e ⇝ 0,1



C : a ⟶ 010

b 011

c 1

d 000

e 001

Une lettre est beaucoup + fréquente "c" ⇝ elle est codé par un code très court.

Exo : $A = \{M, P, R, U, Y, Z\}$.

$p_M = \frac{1}{16} = p_P = p_R = p_Z$ $\quad p_U = \frac{1}{4}$ $\quad p_Y = \frac{1}{2}$

① Calculer H.

② Calculer le codage Huffman et l'arbre associé.

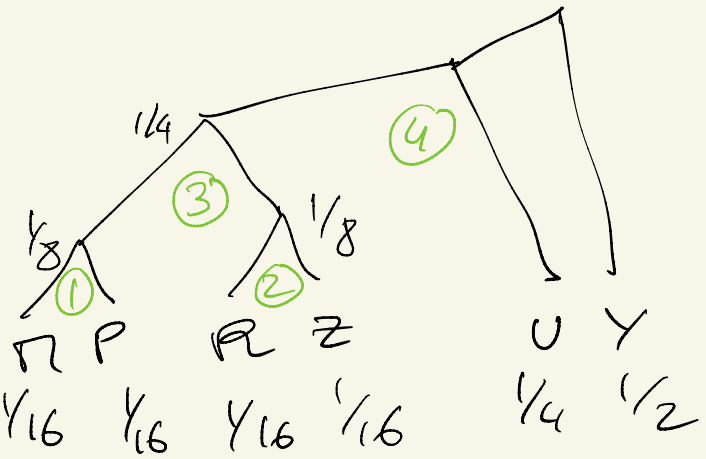
③ Calculer R_1 = nbre moyen de lettre du code d'une lettre aléatoire
$= E[\text{longueur du code de } X_1]$

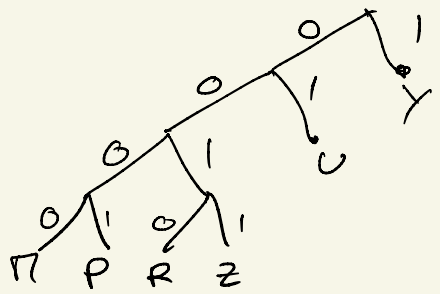
④ Quel est le code de YUTI ?

$\log_2 \frac{1}{2^n} = -n$

$$H = -\sum p_a \log_2 p_a$$
$$= -4 \frac{1}{16} \log_2 \frac{1}{2^4} - \frac{1}{4} \log_2 \frac{1}{2^2} - \frac{1}{2} \log_2 \frac{1}{2}$$
$$= -\frac{1}{4}(-4) - \frac{1}{4}(-2) - \frac{1}{2}(-1) = 1 + \frac{1}{2} + \frac{1}{2}$$
$$= 2.$$

②





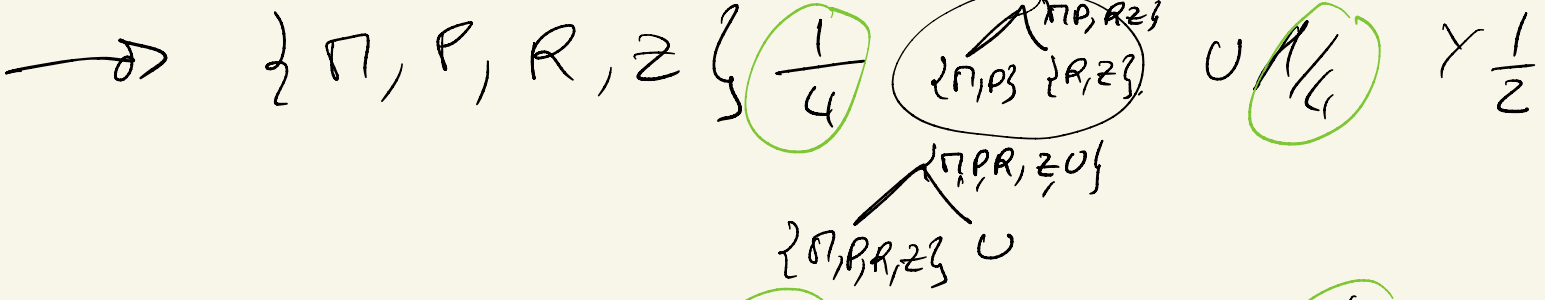
A chaque étape je regroupe les 2 plus les + faibles

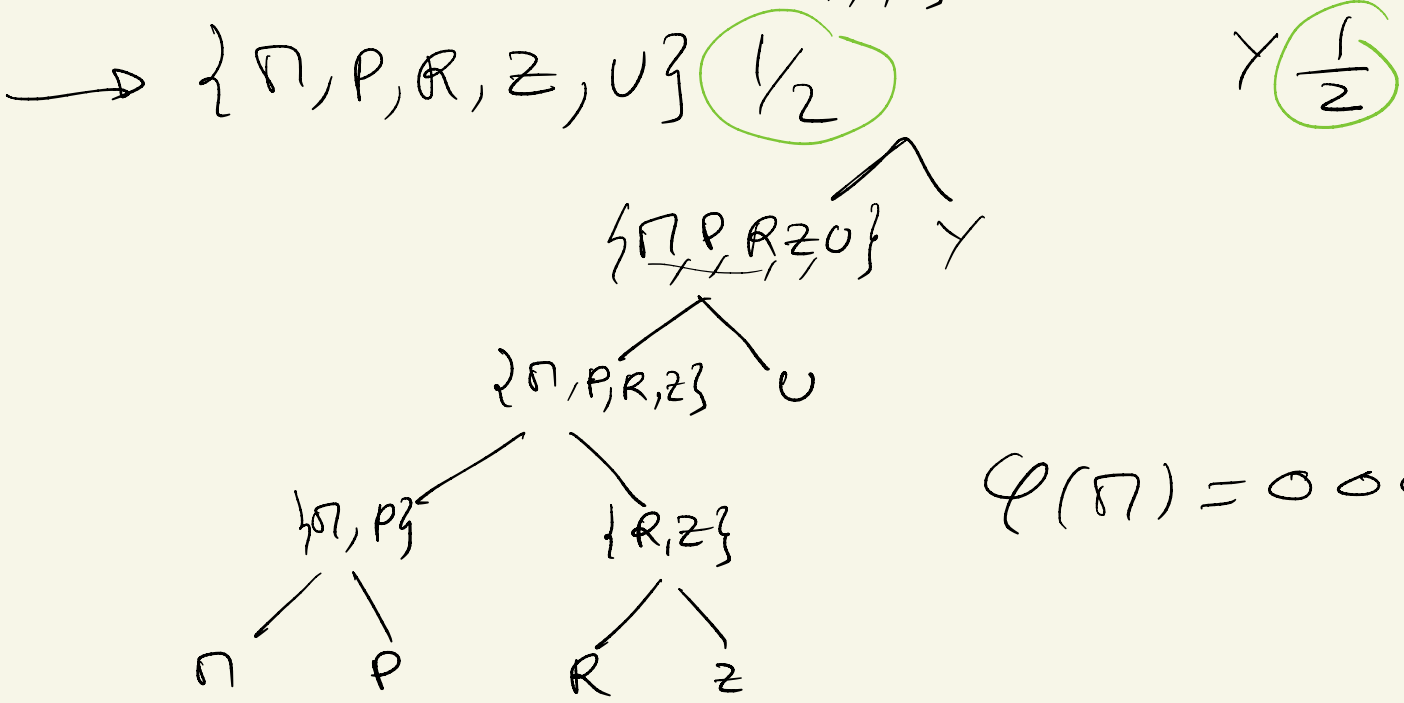
les 2 + petit.

Π 1/16 P 1/16 R 1/16 Z 1/16 U 1/4 Y 1/2

→ {Π, P} 1/8 R 1/16 Z 1/16 U 1/4 Y 1/2

→ {Π, P} 1/8 {R, Z} 1/8 U 1/4 Y 1/2





$\varphi(\Pi) = 0000$

$R_1 = E[\text{Longueur du code}]$

$= \frac{4}{4}\mathbb{P}(X = \Pi, P, R \text{ ou } Z)$

$+ 2\,\mathbb{P}(X = U) + 1 \cdot \mathbb{P}(X = Y)$

$= 4 \times 4 \times \frac{1}{16} + 2 \times \frac{1}{4} + \frac{1}{2} = 2 = H$