# TD 6 : Codage

Correction

## 1 Codes uniquement déchiffrables

**Question 1 : Correction** : Soit $S = \{w_1, w_2, ..., w_n\}$. Le code donnée par $S$ est une fonction :

$$C : \mathcal{X} \to S \text{ avec } |\mathcal{X}| = |S|$$

$C$ est uniquement déchiffrable si la fonction $\bar{C} : \mathcal{X}^* \to S^*$ obtenue par concaténation des codes données par $C$ est injective.

On observe que la définition de $T$ est une définition de plus petit point fixe et que la définition de $T_i$ est une itération de la fonction de ce point fixe. D'où $T = \cup_i T_i$.

$T$ ne contient que les suffixes de mots de $S$ car $T_0$ est constitué de tels suffixes et la définition de chaque $T_i$ construit des suffixes des mots dans $S$ (terme $(s - v)$ avec $s \in S$ et $v \in T_{i-1}$) ou des suffixes des suffixes de $S$ (terme $(v - s)$ avec $v \in T_{i-1}$).

Prouvons les deux lemmes :

(i) On raisonne par récurrence sur $k + \ell$.
$\boxed{k + \ell = 0 :}$ on a $u = \varepsilon \in T$.
$\boxed{k + \ell > 0 :}$ on a $u \in T$ et $u_1 \cdots u_k = uv_1 \cdots v_\ell$, donc plusieurs cas selon la taille de $u$ :
— si $k = 0$ et $\ell > 0$ alors $u_1 \cdots u_k = \varepsilon = uv_1 \cdots v_\ell$ et donc $u = \varepsilon \in T$ (et $v_i = \epsilon$) ;
— pour $k > 0$, si $|u_1| \geq |u|$ alors $u_1 - u \in T$ et $(u_1 - u) \cdots u_k = v_1 \cdots v_\ell$ ; par hypothèse de récurrence (pour $k + \ell - 1$) alors $\epsilon \in T$.
— pour $k > 0$, si $|u| > |u_1|$ alors $u - u_1 \in T$ et $u_2 \cdots u_k = (u - u_1)v_1 \cdots v_\ell$ ; par hypothèse de récurrence (pour $k - 1 + \ell$) alors $\epsilon \in T$.

(ii) Par récurrence sur $i$.
$\boxed{i = 0 :}$ on a $u \in T_0$ et $uu_1 \cdots u_k = v_1 \cdots v_\ell$, qed.
$\boxed{i > 0 :}$ on a $u \in T_{i+1}$ et $uu_1 \cdots u_k = v_1 \cdots v_\ell$
— si $u \in T_i$, par hypothèse de récurrence on obtient la propriété ;
— si $u = u' - v$ avec $u' \in S$ et $v \in T_i$ alors $u' = vu$ et $u'u_1 \cdots u_k = vv_1 \cdots v_\ell$, donc la propriété est obtenue par HR ;
— si $u = v - u'$ avec $u' \in S$ et $v \in T_i$ alors $v = u'u$ et $vu_1 \cdots u_k = u'v_1 \cdots v_\ell$, donc la propriété est obtenue par HR.

Pour conclure, on prouve la propriété : "$\bar{C}$ n'est pas injective ssi $\varepsilon \in T$" :

**Si $T$ contient le mot vide $\varepsilon$ :** alors $\varepsilon$ a été introduit en $T$ par un certain $T_j$. Pour ce $T_j$, il existe $\varepsilon \in T_j$ et $u_1, \cdots, u_k \in S$ tels que $uu_1 \cdots u_k = u_1 \cdots u_k$. Par le lemme (ii), alors il existe $v \in T_0$ (et donc, par définition de $T_0$, $v \neq \varepsilon$) et $u'_1, \ldots, u'_k, v'_1, \ldots, v'_\ell \in S$ tel que $vu'_1 \cdots u'_k = v'_1 \cdots v'_\ell$. Comme $v \in T_0$, il existe $u', v' \in S$ tels que $u' - v' = v$ et $v' \neq u'$. Alors $u'u'_1 \cdots u'_k = v'v'_1 \cdots v'_\ell$ avec $u' \neq v'$, donc pour deux $w_1, w_2 \in \mathcal{X}^*$ avec $w_1[1] \neq w_2[1]$ (car $u' \neq v'$), $\bar{C}(w_1) = \bar{C}(w_2)$, donc $\bar{C}$ n'est pas injective.

**Si $\bar{C}$ n'est pas injective :** alors il existe $u_1, \ldots, u_k, v_1, \ldots, v_\ell \in S$ avec $u_1 \neq v_1$ et $u_1 \cdots u_k = v_1 \cdots v_\ell$. Supposons que $|u_1| > |v_1|$. Alors $(u_1 - v_1)u_2 \cdots u_k = v_2 \cdots v_\ell$ et $u_1 - v_1 \in T_0$. Par le lemme (i), alors $T$ contient $\varepsilon$.

**Question 2 :** **Correction** :

$S_0$ : Oui car $T_0 = \emptyset = T$

$S_1$ : Oui car $T_0 = \{1\} = T$

$S_2$ : Non car $T_0 = \{1\}$, $T_1 = \{1, 0\}$, $T_2 = \{0, 1, \varepsilon\} = T$

$S_3$ : Oui car $T_0 = \{1\} = T$

$S_4$ : Oui car $T_0 = \emptyset = T$

$S_5$ : Oui car $T_0 = \{0\} = T$

$S_6$ : Oui car $T_0 = \{0\} = T$.

**Question 3 :** **Correction** : $T$ est un sous-ensemble de sufixes de $S$. En notant $n = |S|$ et $m = \max\{|u| \mid u \in S\}$, $|T| \leq n \times (m + 1)$. Chaque nouvel élément ajouté à $T$ doit être comparé à chaque élément de $S$ en temps $O(m)$ et le résultat ajouté à $T$ s'il n'y est pas déjà. On a donc au pire $O(|T|^2 \cdot n \cdot m) = O(n^3 \cdot m(m + 1)^2)$ opérations.

# 2 Codage de flux de données

## 2.1 Découpage d'un flux infini

**Question 4 :** **Correction** : On doit montrer que pour toute paire $u, v \in S$, si $u \sqsubseteq v$ alors $u = v$.

Comme $u \sqsubseteq v$ alors $v = uv'$. Par (A), on sait qu'il existe une suite $u_1 \ldots$ en $S$ telle que $v'a^\omega = u_1 \ldots$ pour $a \in \Sigma$ quelconque. De même, il existe une suite $v_1 \ldots$ telle que $a^\omega = v_1 \ldots$. Ainsi, $v \cdot v_1 \ldots = u \cdot v' \cdot a^\omega = u \cdot u_1 \ldots$. Par (B) on a $u = v$.

## 2.2 Code de Elias

**Question 5 :** **Correction** : $|B_0(n)| = \lceil \log n \rceil \sim \log n$. $B_0$ n'est pas un code préfixe, par exemple $3 \neq 7$ mais $B_0(3) = 11 \sqsubseteq 111 = B_0(7)$.

**Question 6 :** **Correction** : $|B_1(n)| = 2|B_0(n)| - 1 = 2\lceil \log n \rceil - 1 \sim 2 \log n$.
Montrons que $B_1$ est un code préfixe. Soient $n, m \geq 1$ tels que $B_1(n) \sqsubseteq B_1(m)$. Soit $i$ le premier indice tel que $B_1(n)[i] = 1$. Alors $B_1(m)[i] = 1$ car $B_1(n) \sqsubseteq B_1(m)$ et on en déduit que $|B_0(n)| = |B_0(m)|$ et aussi $B_0(n) = B_0(m)$, donc $n = m$.

**Question 7 :** **Correction** : (DM)

## 2.3 Codage par rang

**Question 8 :** **Correction** : $N_k[x]$ correspond au nombre de lettres différentes de $x$ depuis la dernière occurence de $x + 1$.

Pour le calcul de $N_{k+1}$, il faut mettre à jour la valeur de $U_{k+1}$ (qui passe à 1) et augmenter de 1 tous les rangs des lettres lues depuis la dernière lecture de $U_{k+1}$. On a ainsi :

$$N_{k+1}[U_{k+1}] = 1$$
$$\forall x \neq U_{k+1}, N_{k+1}[x] = N_k[x] + 1_{N_k[x] < N_k[U_{k+1}]}$$

En effet, si dans $W_k$, la dernière occurence de $U_{k+1}$ est avant $x$, alors $N_k[x] < N_k[U_{k+1}]$ (on considère que les occurrences différentes) et il faut actualiser $(+1)$ l'entrée de $x$ dans le $N_{k+1}$.
**Question 9 :** **Correction** : L'algorithme est donné ci-dessous. Il suppose que les lettres de $\Sigma$ sont ordonnées, $m = |\Sigma|$ et calcule $N_0$ en fonction de cet ordre.

```
pour la i–ème lettre x_i de Σ faire
   N_0(x_i) ← 1 + (m − i)
fin pour
N ← N_0
pour chaque mot V en entrée faire
   r ← C^{-1}(V)
   U ← N^{-1}[r]        % car N_k[x] = N_k[y] ⇒ x = y
   afficher(U)
   maj(N, U)      % avec la formule pour N_{k+1}[U_{k+1}]
fin pour
```

**Question 10 : Correction** : On veut que $C$ soit un code préfixe. En effet, en lisant lettre par lettre, on veut éviter d'avoir $\underbrace{v_1 \cdots v_k}_{\text{correspond à un } k} \cdot v_{k+1} \cdots v_\ell$ correspondent à un $k'$. Or cela n'arrive pas avec un code $C$ préfixe.

**Question 11 : Correction** : $\Delta_k$ représente le nombre de cases entre la dernière lettre $U_k$ et son avant-dernière occurence (le temps depuis la dernière occurrence de $U_k$).

On note $i = \Delta_k$ et $u = U_k$, alors $W_k$ est de la forme $w \cdot u \cdot w' \cdot u$ avec $|w'| = i-1$ et $w'$ ne contient pas la lettre $u$. En posant $P = \{x \mid x \text{ apparaît dans } w'\}$, on a clairement $W_{k-1} \in \Sigma^* \cdot u \cdot P^*$ donc $N_{k-1}[u] \leq 1 + |P| \leq 1 + |w'| = 1 + i = \Delta_k$.

**Question 12 : Correction** : Notons $p = \mathsf{P}(U_k = u)$ (indépendant de $k$).

$$\begin{aligned}
\mathsf{E}(\Delta_k \mid U_k = u) &= \sum_{i=1}^{k} i\mathsf{P}(\Delta_k = i \mid U_k = u) \\
&= \sum_{i=1}^{k} ip(1-p)^{i-1} = p\sum_{i=1}^{k} i(1-p)^{i-1} \\
&\xrightarrow{k\to\infty} p\sum_{i=1}^{\infty} i(1-p)^{i-1} = \frac{p}{(1-(1-p))^2} = \frac{1}{p}
\end{aligned}$$

**Question 13 : Correction** : $n \mapsto |B_1(n)|$ est croissant, donc $|V_k| \leq |B_1(N_{k-1}[U_k])| \leq |B_1(\Delta_k)| = 2\lfloor \log \Delta_k \rfloor + 1$. On note que log est concave, donc $\mathsf{E}(\log .) \leq \log(\mathsf{E}(.))$ ainsi :

$$\begin{aligned}
\mathsf{E}(\log \Delta_k) &= \sum_{u\in\Sigma} \mathsf{P}(U_k = u)\mathsf{E}(\log \Delta_k \mid U_k = u) \\
&\leq \sum_{u\in\Sigma} p_u \left(2 \log \mathsf{E}(\Delta_k \mid U_k = u) + 1\right) \\
&\xrightarrow{k\to\infty} \sum_{u\in\Sigma} p_u \left(2 \log \frac{1}{p_u} + 1\right) \\
&= 2\mathsf{H}(U) + 1
\end{aligned}$$

**Question 14 : Correction** : (DM)

**Question 15 : Correction** : On regroupe les données en entrée par blocs de taille $l$. Ainsi, le flux d'entrée devient $U'_1U'_2\ldots$ avec $U'_i = U_{l(i-1)+1}\ldots U_{li}$. On a $\mathsf{H}(U') = l\mathsf{H}(U)$. Le codage moyen d'une lettre prend ainsi $\mathsf{E}(|V_k|)/l \leq \mathsf{H}(U) + 2\frac{\log(l\mathsf{H}(U)+1)}{l} + \frac{1}{l}$. Pour $l$ suffisamment grand, ce taux est proche de $\mathsf{H}(U)$. En contre partie, il est nécessaire d'attendre l'arrivée de $l$ lettres avant de pouvoir réaliser le codage du bloc, ce qui augmente potentiellement la latence.