Université Laval
Faculté des sciences et de génie
Département d'informatique et de génie logiciel
IFT-4003/7023

Danny Dubé
Hiver 2016
Version: 3 mars

Solutionnaire du travail pratique #1

Réponses

Note: les réponses aux différentes questions ne sont pas toujours uniques.

1.

Code 1 : il est uniquement décodable. Utilisons l'algorithme de test de décodabilité unique. On commence par initialiser DS. En constatant que $C(\mathtt{a})$ est un préfixe de $C(\mathtt{b})$, on insère `101` dans DS. En inspectant le reste des mots de code, on constate qu'on ne peut trouver d'autre *dangling suffix* à insérer dans DS. Ainsi, après l'initialisation, DS est $\{\mathtt{101}\}$. Au cours de la première itération de l'algorithme, on constate qu'on ne peut pas augmenter DS davantage car `101` n'est pas le préfixe d'un mot de code et aucun mot de code n'est préfixe de `101`. Ainsi, DS est arrivé à saturation. On conclut que le code est uniquement décodable car `101` n'est pas un mot de code.

Code 2 : il est uniquement décodable. Utilisons l'algorithme de test de décodabilité unique. On commence par initialiser DS. On constate que tous les mots de code qui débutent par `1` sont des préfixes les uns des autres. Ainsi, après l'initialisation, DS est $\{\mathtt{0}, \mathtt{000}, \mathtt{0000}, \mathtt{00000}\}$. Au cours de la première itération de l'algorithme, on constate que certains *dangling suffixes* sont les préfixes de certains mots de code (ceux commençant par `0`). Ceci amène DS à atteindre $\{\mathtt{0}, \mathtt{01}, \mathtt{000}, \mathtt{0001}, \mathtt{0000}, \mathtt{00001}, \mathtt{00000}, \mathtt{000001}, \mathtt{00000001}\}$. Au cours d'une deuxième itération, on observe qu'on ne peut découvrir de nouveaux *dangling suffixes*. Comme DS est arrivé à saturation et qu'on ne retrouve aucun mot de code dans DS, on conclut que le code est uniquement décodable.

Code 3 : il est uniquement décodable car il est un code préfixe.

Code 4 : il est uniquement décodable. Utilisons l'algorithme de test de décodabilité unique. On commence par initialiser DS. On constate que $C(\mathtt{d})$ est un préfixe de $C(\mathtt{e})$. Comme c'est le seul cas où un mot de code est préfixe d'un autre, on obtient que DS est initialisé à $\{\mathtt{1}\}$. Au cours de la première itération, on ajoute les *dangling suffixes* résultant de l'extraction de `1` des mots de code qui débutent par `1`. Cette itération augmente DS à $\{\mathtt{00}, \mathtt{01}, \mathtt{1}, \mathtt{10}, \mathtt{111}\}$. Au cours de la deuxième itération, on observe que le *dangling suffix* `00` est un préfixe de deux mots de code, ce qui cause l'ajout du nouveau *dangling suffix* `0`. On observe aussi que le *dangling suffix* `01` est un préfixe de trois mots de code, ce qui cause l'ajout d'un *dangling suffix* additionnel: `11`. Il y a encore d'autres cas où des *dangling suffixes* sont des préfixes de mots de code mais tous ces cas ne causent pas l'ajout d'autres *dangling suffixes*. À la fin de la deuxième itération, on a que DS est $\{\mathtt{0}, \mathtt{00}, \mathtt{01}, \mathtt{1}, \mathtt{10}, \mathtt{11}, \mathtt{111}\}$. Au cours d'une troisième itération, on ne découvre plus de nouveaux *dangling suffixes*. Ayant atteint la saturation pour DS, on constate qu'aucun *dangling suffix* n'est un mot de code et, donc, que le code est uniquement décodable.

Code 5 : il n'est pas uniquement décodable car:

$$C(\mathtt{c}) \cdot C(\mathtt{f}) = \mathtt{010}\ \ \mathtt{1001} = \mathtt{0101}\ \ \mathtt{001} = C(\mathtt{d}) \cdot C(\mathtt{b}).$$

2. (a) Oui, il y a le code préfixe suivant.

$$\begin{array}{ll} \mathtt{a}: & \mathtt{000001} \\ \mathtt{b}: & \mathtt{01} \\ \mathtt{c}: & \mathtt{00001} \\ \mathtt{d}: & \mathtt{1} \\ \mathtt{e}: & \mathtt{00000001} \\ \mathtt{f}: & \mathtt{0001} \\ \mathtt{g}: & \mathtt{0000001} \\ \mathtt{h}: & \mathtt{001} \end{array} \qquad \text{ou, présenté autrement,} \qquad \begin{array}{ll} \mathtt{e}: & \mathtt{00000001} \\ \mathtt{g}: & \mathtt{0000001} \\ \mathtt{a}: & \mathtt{000001} \\ \mathtt{c}: & \mathtt{00001} \\ \mathtt{f}: & \mathtt{0001} \\ \mathtt{h}: & \mathtt{001} \\ \mathtt{b}: & \mathtt{01} \\ \mathtt{d}: & \mathtt{1} \end{array}$$

(b) Il n'y a pas de tel code préfixe car les longueurs des mots de code ne respectent pas l'inégalité de Kraft-McMillan.

$$\begin{aligned}
& \tfrac{1}{2^4} + \tfrac{1}{2^2} + \tfrac{1}{2^4} + \tfrac{1}{2^3} + \tfrac{1}{2^2} + \tfrac{1}{2^3} + \tfrac{1}{2^4} + \tfrac{1}{2^3} \\
= {} & \tfrac{1}{2^2} + \tfrac{1}{2^2} + \tfrac{1}{2^3} + \tfrac{1}{2^3} + \tfrac{1}{2^3} + \tfrac{1}{2^4} + \tfrac{1}{2^4} + \tfrac{1}{2^4} \\
= {} & \tfrac{2}{2^2} + \tfrac{2}{2^3} + \tfrac{1}{2^3} + \tfrac{2}{2^4} + \tfrac{1}{2^4} \\
= {} & \tfrac{1}{2^1} + \tfrac{1}{2^2} + \tfrac{1}{2^3} + \tfrac{1}{2^3} + \tfrac{1}{2^4} \\
= {} & \tfrac{1}{2^1} + \tfrac{1}{2^2} + \tfrac{2}{2^3} + \tfrac{1}{2^4} \\
= {} & \tfrac{1}{2^1} + \tfrac{1}{2^2} + \tfrac{1}{2^2} + \tfrac{1}{2^4} \\
= {} & \tfrac{1}{2^1} + \tfrac{2}{2^2} + \tfrac{1}{2^4} \\
= {} & \tfrac{1}{2^1} + \tfrac{1}{2^1} + \tfrac{1}{2^4} \\
= {} & \tfrac{2}{2^1} + \tfrac{1}{2^4} \\
= {} & \tfrac{1}{2^0} + \tfrac{1}{2^4} \\
= {} & 1 + \tfrac{1}{2^4} \\
> {} & 1
\end{aligned}$$

3. (a) On trie les symboles par fréquences d'apparition. Si je ne me trompe pas, tous les points de coupure sont imposés par les fréquences des symboles.

$$
\begin{array}{l}
\texttt{g}:14 \;\; \texttt{c}:13 \;\; \texttt{l}:13 \;\; \texttt{b}:12 \;\; \texttt{j}:11 \;\; \texttt{f}:10 \;\; \texttt{h}:10 \;\; \texttt{i}:9 \;\; \texttt{k}:8 \;\; \texttt{d}:7 \;\; \texttt{e}:5 \;\; \texttt{a}:4 \\
\texttt{g}:14 \;\; \texttt{c}:13 \;\; \texttt{l}:13 \;\; \texttt{b}:12 \;\; \texttt{j}:11 \mid \texttt{f}:10 \;\; \texttt{h}:10 \;\; \texttt{i}:9 \;\; \texttt{k}:8 \;\; \texttt{d}:7 \;\; \texttt{e}:5 \;\; \texttt{a}:4 \\
\texttt{g}:14 \;\; \texttt{c}:13 \mid \texttt{l}:13 \;\; \texttt{b}:12 \;\; \texttt{j}:11 \mid \texttt{f}:10 \;\; \texttt{h}:10 \;\; \texttt{i}:9 \mid \texttt{k}:8 \;\; \texttt{d}:7 \;\; \texttt{e}:5 \;\; \texttt{a}:4 \\
\texttt{g}:14 \mid \texttt{c}:13 \mid \texttt{l}:13 \mid \texttt{b}:12 \;\; \texttt{j}:11 \mid \texttt{f}:10 \mid \texttt{h}:10 \;\; \texttt{i}:9 \mid \texttt{k}:8 \;\; \texttt{d}:7 \mid \texttt{e}:5 \;\; \texttt{a}:4 \\
\texttt{b}:12 \mid \texttt{j}:11 \quad \texttt{h}:10 \mid \texttt{i}:9 \quad \texttt{k}:8 \mid \texttt{d}:7 \quad \texttt{e}:5 \mid \texttt{a}:4
\end{array}
$$

Donc, le code obtenu est le suivant.

g :	000
c :	001
l :	010
b :	0110
j :	0111
f :	100
h :	1010
i :	1011
k :	1100
d :	1101
e :	1110
a :	1111

ou, présenté autrement:

a :	1111
b :	0110
c :	001
d :	1101
e :	1110
f :	100
g :	000
h :	1010
i :	1011
j :	0111
k :	1100
l :	010

$$
\begin{aligned}
L_{\text{moy}} &= \frac{\begin{array}{c}14*3+13*3+13*3+12*4+11*4+10*3+\\ 10*4+9*4+8*4+7*4+5*4+4*4\end{array}}{116} \\
&= \frac{(14+13+13+10)*3+(12+11+10+9+8+7+5+4)*4}{116} \\
&= \frac{50*3+66*4}{116} \\
&= \frac{414}{116} \\
&\approx 3.569
\end{aligned}
$$

(b) L'algorithme de Huffman doit joindre des symboles pour en faire des frères. Par exemple, à la première étape ci-bas, les symboles `a` et `e` sont joints. Nous dénotons par $\texttt{A} \equiv \texttt{a} \swarrow\!\!\wedge\!\!\searrow \texttt{e}$ le fait de joindre ces deux symboles et de les remplacer par un nouveau symbole artificiel nommé `A`. Cette notation signifie que le mot de code de `a` sera formé en ajoutant `0` au mot de code de `A` et que le mot de code de `e` sera formé en ajoutant `1` au mot de code de `A`. De plus, la fréquence de `A` est la somme des fréquences de `a` et de `e`.

$\texttt{a}:4,\ \texttt{e}:5,\ \texttt{d}:7,\ \texttt{k}:8,\ \texttt{i}:9,\ \texttt{f}:10,\ \texttt{h}:10,\ \texttt{j}:11,\ \texttt{b}:12,\ \texttt{c}:13,\ \texttt{l}:13,\ \texttt{g}:14$

$\boxed{\texttt{A} \equiv \texttt{a} \swarrow\!\!\wedge\!\!\searrow \texttt{e}}$

$\texttt{d}:7,\ \texttt{k}:8,\ \texttt{i}:9,\ \texttt{A}:9,\ \texttt{f}:10,\ \texttt{h}:10,\ \texttt{j}:11,\ \texttt{b}:12,\ \texttt{c}:13,\ \texttt{l}:13,\ \texttt{g}:14$

$\boxed{\texttt{B} \equiv \texttt{d} \swarrow\!\!\wedge\!\!\searrow \texttt{k}}$

$\texttt{i}:9,\ \texttt{A}:9,\ \texttt{f}:10,\ \texttt{h}:10,\ \texttt{j}:11,\ \texttt{b}:12,\ \texttt{c}:13,\ \texttt{l}:13,\ \texttt{g}:14,\ \texttt{B}:15$

$\boxed{\texttt{C} \equiv \texttt{i} \swarrow\!\!\wedge\!\!\searrow \texttt{A}}$

$\texttt{f}:10,\ \texttt{h}:10,\ \texttt{j}:11,\ \texttt{b}:12,\ \texttt{c}:13,\ \texttt{l}:13,\ \texttt{g}:14,\ \texttt{B}:15,\ \texttt{C}:18$

$\boxed{\texttt{D} \equiv \texttt{f} \swarrow\!\!\wedge\!\!\searrow \texttt{h}}$

$\texttt{j}:11,\ \texttt{b}:12,\ \texttt{c}:13,\ \texttt{l}:13,\ \texttt{g}:14,\ \texttt{B}:15,\ \texttt{C}:18,\ \texttt{D}:20$

$\boxed{\texttt{E} \equiv \texttt{j} \swarrow\!\!\wedge\!\!\searrow \texttt{b}}$

$\texttt{c}:13,\ \texttt{l}:13,\ \texttt{g}:14,\ \texttt{B}:15,\ \texttt{C}:18,\ \texttt{D}:20,\ \texttt{E}:23$

$\boxed{\texttt{F} \equiv \texttt{c} \swarrow\!\!\wedge\!\!\searrow \texttt{l}}$

$\texttt{g}:14,\ \texttt{B}:15,\ \texttt{C}:18,\ \texttt{D}:20,\ \texttt{E}:23,\ \texttt{F}:26$

$\boxed{\texttt{G} \equiv \texttt{g} \swarrow\!\!\wedge\!\!\searrow \texttt{B}}$

$\texttt{C}:18,\ \texttt{D}:20,\ \texttt{E}:23,\ \texttt{F}:26,\ \texttt{G}:29$

$\boxed{\texttt{H} \equiv \texttt{C} \swarrow\!\!\wedge\!\!\searrow \texttt{D}}$

$\texttt{E}:23,\ \texttt{F}:26,\ \texttt{G}:29,\ \texttt{H}:38$

$\boxed{\texttt{I} \equiv \texttt{E} \swarrow\!\!\wedge\!\!\searrow \texttt{F}}$

$\texttt{G}:29,\ \texttt{H}:38,\ \texttt{I}:49$

$\boxed{\texttt{J} \equiv \texttt{G} \swarrow\!\!\wedge\!\!\searrow \texttt{H}}$

$\texttt{I}:49,\ \texttt{J}:67$

$\boxed{\texttt{K} \equiv \texttt{I} \swarrow\!\!\wedge\!\!\searrow \texttt{J}}$

$\texttt{K}:116$

Donc, le code obtenu est le suivant.

j :	000
b :	001
c :	010
l :	011
g :	100
d :	1010
k :	1011
i :	1100
a :	11010
e :	11011
f :	1110
h :	1111

ou, présenté autrement:

a :	11010
b :	001
c :	010
d :	1010
e :	11011
f :	1110
g :	100
h :	1111
i :	1100
j :	000
k :	1011
l :	011

$$\begin{aligned}
L_{\text{moy}} &= \frac{\begin{array}{c}4*5+12*3+13*3+7*4+5*5+10*4+\\ 14*3+10*4+9*4+11*3+8*4+13*3\end{array}}{116} \\
&= \frac{(12+13+14+11+13)*3+(7+10+10+9+8)*4+(4+5)*5}{116} \\
&= \frac{63*3+44*4+9*5}{116} \\
&= \frac{410}{116} \\
&\approx 3.534
\end{aligned}$$

4.

(a) Pour construire le mot de code pour le nombre i, on doit déterminer le numéro du groupe de naturels, $q = \lfloor i/3 \rfloor$, et identifier le membre du groupe, $r = i - 3q \in \{0, 1, 2\}$. Le numéro de groupe est encodé à l'aide du code unaire vu en classe. Le numéro q se traduit en le mot de code $\texttt{0}^q\texttt{1}$, en $q+1$ bits. Le numéro de membre est encodé à l'aide d'un code préfixe. Les numéros $r = 0$, $r = 1$ et $r = 2$ se traduisent en les mots de code `0`, `10` et `11`, respectivement. Soit $l_3(i)$ la longueur du mot de code pour le naturel i. Nous avons donc que, lorsque i est un multiple de 3, $l_3(i) = q+2$, sinon, $l_3(i) = q+3$. Maintenant, soit L_3 la longueur moyenne du mot de code pour un naturel tiré au hasard.

$$
\begin{aligned}
L_3 &= \sum_{i=0}^{\infty} p_i \cdot l_3(i) \\
&= \sum_{q=0}^{\infty} \sum_{r=0}^{2} p_{3q+r} \cdot l_3(3q+r) \\
&= \sum_{q=0}^{\infty} \left(p_{3q} \cdot l_3(3q) + p_{3q+1} \cdot l_3(3q+1) + p_{3q+2} \cdot l_3(3q+2)\right) \\
&= \sum_{q=0}^{\infty} \left(\frac{1}{5} \left(\frac{4}{5}\right)^{3q} \cdot (q+2) + \frac{1}{5} \left(\frac{4}{5}\right)^{3q+1} \cdot (q+3) + \frac{1}{5} \left(\frac{4}{5}\right)^{3q+2} \cdot (q+3) \right) \\
&= \frac{1}{5} \cdot \sum_{q=0}^{\infty} \left((q+2) + \frac{4}{5} \cdot (q+3) + \left(\frac{4}{5}\right)^2 \cdot (q+3) \right) \cdot \left(\frac{4}{5}\right)^{3q} \\
&= \frac{1}{125} \cdot \sum_{q=0}^{\infty} \left(25 \cdot (q+2) + 5 \cdot 4 \cdot (q+3) + 4^2 \cdot (q+3)\right) \cdot \left(\frac{64}{125}\right)^q \\
&= \frac{1}{125} \cdot \sum_{q=0}^{\infty} (61 \cdot q + 158) \cdot \left(\frac{64}{125}\right)^q \\
&= \frac{61}{125} \cdot \sum_{q=0}^{\infty} q \cdot \left(\frac{64}{125}\right)^q + \frac{158}{125} \cdot \sum_{q=0}^{\infty} \left(\frac{64}{125}\right)^q \\
&= \frac{61}{125} \cdot \frac{\frac{64}{125}}{(1 - \frac{64}{125})^2} + \frac{158}{125} \cdot \frac{1}{1 - \frac{64}{125}} \\
&= 61 \cdot \frac{64}{(125-64)^2} + 158 \cdot \frac{1}{125-64} \\
&= 61 \cdot \frac{64}{61^2} + 158 \cdot \frac{1}{61} \\
&= \frac{64+158}{61} \\
&= \frac{222}{61} \\
&\approx 3.639
\end{aligned}
$$

(b) Le code de Golomb avec paramètre $m = 4$ est plus simple. On décompose un nombre naturel i en son numéro de groupe, $q = \lfloor i/4 \rfloor$, et son numéro de membre du groupe, $r = i - 4q$. Comme dans l'exercice précédent, le numéro de groupe s'encode en unaire en $q + 1$ bits. Toutefois, le numéro de membre s'encode systématiquement en binaire en 2 bits. Ainsi, la longueur de l'encodage de i, $l_4(i)$, est de $q + 3$ bits.

$$
\begin{aligned}
L_4 &= \sum_{i=0}^{\infty} p_i \cdot l_4(i) \\
&= \sum_{q=0}^{\infty} \sum_{r=0}^{3} p_{4q+r} \cdot l_4(4q+r) \\
&= \sum_{q=0}^{\infty} \sum_{r=0}^{3} \frac{1}{5} \left(\frac{4}{5}\right)^{4q+r} \cdot (q+3) \\
&= \frac{1}{5} \cdot \sum_{q=0}^{\infty} (q+3) \cdot \left(\frac{4}{5}\right)^{4q} \cdot \sum_{r=0}^{3} \left(\frac{4}{5}\right)^{r} \\
&= \frac{1}{5} \cdot \sum_{q=0}^{\infty} (q+3) \cdot \left(\frac{4}{5}\right)^{4q} \cdot \left(1 + \frac{4}{5} + \left(\frac{4}{5}\right)^2 + \left(\frac{4}{5}\right)^3\right) \\
&= \frac{1}{5} \cdot \left(1 + \frac{4}{5} + \left(\frac{4}{5}\right)^2 + \left(\frac{4}{5}\right)^3\right) \cdot \sum_{q=0}^{\infty} (q+3) \cdot \left(\frac{4}{5}\right)^{4q} \\
&= \frac{1}{5} \cdot \frac{1}{5^3} \cdot \left(5^3 + 5^2 \cdot 4 + 5 \cdot 4^2 + 4^3\right) \cdot \sum_{q=0}^{\infty} (q+3) \cdot \left(\frac{256}{625}\right)^{q} \\
&= \frac{125 + 100 + 80 + 64}{625} \cdot \left(\sum_{q=0}^{\infty} q \cdot \left(\frac{256}{625}\right)^{q} + \sum_{q=0}^{\infty} 3 \cdot \left(\frac{256}{625}\right)^{q}\right) \\
&= \frac{369}{625} \cdot \left(\frac{\frac{256}{625}}{(1 - \frac{256}{625})^2} + \frac{3}{1 - \frac{256}{625}}\right) \\
&= 369 \cdot \left(\frac{256}{(625 - 256)^2} + \frac{3}{625 - 256}\right) \\
&= 369 \cdot \left(\frac{256}{369^2} + \frac{3}{369}\right) \\
&= \frac{256}{369} + 3 \\
&\approx 3.694
\end{aligned}
$$

5. On fait croître le dictionnaire de mots sources tant qu'il y a au plus 32 entrées.

a	2/7
b	1/7
c	3/7
d	1/7

a	2/7
b	1/7
ca	6/49
cb	3/49
cc	9/49
cd	3/49
d	1/7

aa	4/49
ab	2/49
ac	6/49
ad	2/49
b	1/7
ca	6/49
cb	3/49
cc	9/49
cd	3/49
d	1/7

aa	4/49
ab	2/49
ac	6/49
ad	2/49
b	1/7
ca	6/49
cb	3/49
cca	18/343
ccb	9/343
ccc	27/343
ccd	9/343
cd	3/49
d	1/7

aa	4/49
ab	2/49
ac	6/49
ad	2/49
ba	2/49
bb	1/49
bc	3/49
bd	1/49
ca	6/49
cb	3/49
cca	18/343
ccb	9/343
ccc	27/343
ccd	9/343
cd	3/49
d	1/7

aa	4/49
ab	2/49
ac	6/49
ad	2/49
ba	2/49
bb	1/49
bc	3/49
bd	1/49
ca	6/49
cb	3/49
cca	18/343
ccb	9/343
ccc	27/343
ccd	9/343
cd	3/49
da	2/49
db	1/49
dc	3/49
dd	1/49

aa	4/49
ab	2/49
ac	6/49
ad	2/49
ba	2/49
bb	1/49
bc	3/49
bd	1/49
ca	6/49
cb	3/49
cca	18/343
ccb	9/343
ccc	27/343
ccd	9/343
cd	3/49
da	2/49
db	1/49
dc	3/49
dd	1/49

aa	4/49
ab	2/49
aca	12/343
acb	6/343
acc	18/343
acd	6/343
ad	2/49
ba	2/49
bb	1/49
bc	3/49
bd	1/49
ca	6/49
cb	3/49
cca	18/343
ccb	9/343
ccc	27/343
ccd	9/343
cd	3/49
da	2/49
db	1/49
dc	3/49
dd	1/49

aa	4/49
ab	2/49
aca	12/343
acb	6/343
acc	18/343
acd	6/343
ad	2/49
ba	2/49
bb	1/49
bc	3/49
bd	1/49
caa	12/343
cab	6/343
cac	18/343
cad	6/343
cb	3/49
cca	18/343
ccb	9/343
ccc	27/343
ccd	9/343
cd	3/49
da	2/49
db	1/49
dc	3/49
dd	1/49

aaa	8/343
aab	4/343
aac	12/343
aad	4/343
ab	2/49
aca	12/343
acb	6/343
acc	18/343
acd	6/343
ad	2/49
ba	2/49
bb	1/49
bc	3/49
bd	1/49
caa	12/343
cab	6/343
cac	18/343
cad	6/343
cb	3/49
cca	18/343
ccb	9/343
ccc	27/343
ccd	9/343
cd	3/49
da	2/49
db	1/49
dc	3/49
dd	1/49

aaa	8/343
aab	4/343
aac	12/343
aad	4/343
ab	2/49
aca	12/343
acb	6/343
acc	18/343
acd	6/343
ad	2/49
ba	2/49
bb	1/49
bc	3/49
bd	1/49
caa	12/343
cab	6/343
cac	18/343
cad	6/343
cb	3/49
cca	18/343
ccb	9/343
ccca	54/2401
cccb	27/2401
cccc	81/2401
cccd	27/2401
ccd	9/343
cd	3/49
da	2/49
db	1/49
dc	3/49
dd	1/49

6. Les tableaux suivants indiquent l'état de l'intervalle avant et après le traitement de chaque caractère de l'entrée à coder. L'état du compteur `E3.count` est aussi indiqué. Enfin, l'intervalle résultant du codage du symbole est redimensionné si nécessaire. Les éventuels bits émis par le redimensionnement sont indiqués.

(a)

# CAR.	I	II	III	IV
INT./`E3.count` AVANT	$[0,\ 1)/0$	$[\frac{2}{5},\ \frac{4}{5})/0$	$[\frac{13}{50},\ \frac{9}{10})/2$	$[\frac{4}{125},\ \frac{68}{125})/0$
INT. `a`	$[0,\ \frac{2}{5})$	$[\frac{2}{5},\ \frac{14}{25})$	$[\frac{13}{50},\ \frac{129}{250})$	$[\frac{4}{125},\ \frac{148}{625})$
INT. `b`	$[\frac{2}{5},\ \frac{4}{5})$	$[\frac{14}{25},\ \frac{18}{25})$	$[\frac{129}{250},\ \frac{193}{250})$	$[\frac{148}{625},\ \frac{276}{625})$
INT. `c`	$[\frac{4}{5},\ 1)$	$[\frac{18}{25},\ \frac{4}{5})$	$[\frac{193}{250},\ \frac{9}{10})$	$[\frac{276}{625},\ \frac{68}{125})$
INT./`E3.count` APRÈS	$[\frac{2}{5},\ \frac{4}{5})/0$	$[\frac{18}{25},\ \frac{4}{5})/0$	$[\frac{129}{250},\ \frac{193}{250})/2$	$[\frac{4}{125},\ \frac{148}{625})/0$
REDIM. & BITS		$E_2:$ `1`	$E_2:$ `100`	$E_1:$ `0`
INT./`E3.count`		$[\frac{11}{25},\ \frac{3}{5})/0$	$[\frac{4}{125},\ \frac{68}{125})/0$	$[\frac{8}{125},\ \frac{296}{625})/0$
REDIM. & BITS		$E_3:\ \epsilon$		$E_1:$ `0`
INT./`E3.count`		$[\frac{19}{50},\ \frac{7}{10})/1$		$[\frac{16}{125},\ \frac{592}{625})/0$
REDIM. & BITS		$E_3:\ \epsilon$		
INT./`E3.count`		$[\frac{13}{50},\ \frac{9}{10})/2$		

(b)

# CAR.	I	II	III	IV
INT./`E3.count` AVANT	[0, 31]/0	[13, 25]/0	[8, 31]/2	[2, 21]/0
INT. `a`	[0, 12]	[13, 17]	[8, 16]	[2, 9]
INT. `b`	[13, 25]	[18, 22]	[17, 26]	[10, 17]
INT. `c`	[26, 31]	[23, 25]	[27, 31]	[18, 21]
INT./`E3.count` APRÈS	[13, 25]/0	[23, 25]/0	[17, 26]/2	[2, 9]/0
REDIM. & BITS		$E_2:$ `1`	$E_2:$ `100`	$E_1:$ `0`
INT./`E3.count`		[14, 19]/0	[2, 21]/0	[4, 19]/0
REDIM. & BITS		$E_3:\ \epsilon$		
INT./`E3.count`		[12, 23]/1		
REDIM. & BITS		$E_3:\ \epsilon$		
INT./`E3.count`		[8, 31]/2		