

Codage d'Huffman

mardi 5 novembre 2013

Exemple

$\Sigma :$	a	b	c	d	e	f
f:	45	13	12	16	9	5

mardi 5 novembre 2013

Le problème

obj: ´tant donn´ un t xt sur un alphas t Σ ,
l **c de** succinct m nt n binair .

- **donn´ s:** un alphas t Σ t un fonction d «fr´qu nc » $f : \Sigma \rightarrow \mathbb{N}$
- **r´ultat:** un codag $\phi : \Sigma \rightarrow \{0,1\}^*$ t l
qu $\sum_{a \in \Sigma} f(a) \cdot |\phi(a)|$ soit minimal.

mardi 5 novembre 2013

Exemple

$\Sigma :$	a	b	c	d	e	f
f:	45	13	12	16	9	5

Solution 1:

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

Solution 1:

a	b	c	d	e	f
000	001	010	011	100	101

$$\sum f(a) \cdot |\phi(a)| = 45 \times 3 + 13 \times 3 + \dots = 300$$

mardi 5 novembre 2013

Codage...

Ici on se limite aux codes préfixes, i.e. :
 aucun $\phi(x)$ n'est préfixe d'un $\phi(y)$

Propriété:

Il existe un code préfixe optimal.

Avantage:

L'encodage est très simple !

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

Solution 1:

a	b	c	d	e	f
000	001	010	011	100	101

$$\sum f(a) \cdot |\phi(a)| = 45 \times 3 + 13 \times 3 + \dots = 300$$

Solution 2:

a	b	c	d	e	f
0	101	100	111	1101	1100

$$\sum f(a) \cdot |\phi(a)| = 45 \times 1 + 13 \times 3 + \dots = 224$$

mardi 5 novembre 2013

Exemple de décodage

a	b	c	d	e	f
0	101	100	111	1101	1100

1 0 1 0 1 0 0 1 1 1 1 0 1 0 1 0 1

mardi 5 novembre 2013

Example: Package

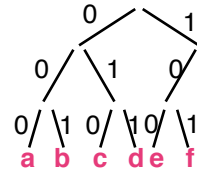


1010101



Codes préfixes et arbres binaires

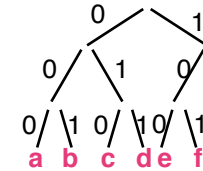
a	b	c	d	e	f
000	001	010	011	100	101



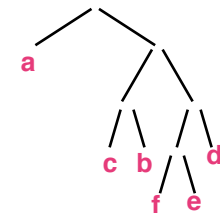
mardi 5 novembre 2013

Codes préfixes et arbres binaires

a	b	c	d	e	f
000	001	010	011	100	101



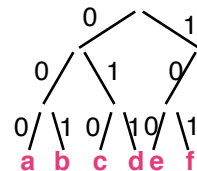
a	b	c	d	e	f
0	101	100	111	1101	1100



mardi 5 novembre 2013

Codes préfixes et arbres binaires

a	b	c	d	e	f
000	001	010	011	100	101



a	b	c	d	e	f
0	101	100	111	1101	1100

mardi 5 novembre 2013

Codes préfixes et arbres binaires

Un cod préfix optimal st toujours r pr's nt' sous la form d'un arbr binaire **local m nt compl t**.

Coût d'un arbre T selon f:

$$B_f(T) = \sum_{a \in F(T)} f(a) \cdot \text{prof}_T(a)$$

$F(T)$: feuilles de T

$\text{prof}_T(a)$: profondeur de a dans T

- On écrira $B(T)$ lorsque f est fixée dans le contexte.
- On étend $B()$ aux codages: $B(\phi) = \sum f(a) \cdot |\phi(a)|$

mardi 5 novembre 2013

e ee e e e
e e

Arbres

On consid`r l s primitiv s suivant s sur l s
arbr s: e e e

- **f uill (a)** = cr´ un f uill ´tiqu t´ par «a»
- **arbr (t₁,t₂)** = cr´ un arbr av c t₁ comm fils
gauch t t₂ comm fils droit
- **fg(t)** = r tourn l fils gauch
- **fd(t)** : r tourn l fils droit

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

mardi 5 novembre 2013

Algorithme d'Huffman

e e
n := | Σ |
FP := **Fil Priorit´** ({ (f uill (a),f(a)) | a $\in \Sigma$ })
Pour i = 1 à n-1:
 (t₁,f₁) := **Extraire Min**(FP)
 (t₂,f₂) := **Extraire Min**(FP)
 Ajour r(FP,(arbr (t₁,t₂),f₁+f₂))
(T,f) := **Extraire Min**(FP)
r tourn r T

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

(f,5) t (,9):

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

(f,5) t (,9): $\begin{matrix} / \\ \text{f} \end{matrix} \backslash \begin{matrix} \\ \text{e} \end{matrix} 14$

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

(f,5) t (,9): $\begin{matrix} / \\ \text{f} \end{matrix} \backslash \begin{matrix} \\ \text{e} \end{matrix} 14$ + (a,45) (b,13) (c,12) (d,16)

(c,12) t (b,13):

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

(f,5) t (,9): $\begin{matrix} / \\ \text{f} \end{matrix} \backslash \begin{matrix} \\ \text{e} \end{matrix} 14$ + (a,45) (b,13) (c,12) (d,16)

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

(f,5) t (,9): $\begin{matrix} / \\ \text{f} \end{matrix} \backslash \begin{matrix} \\ \text{e} \end{matrix} 14$ + (a,45) (b,13) (c,12) (d,16)

(c,12) t (b,13): $\begin{matrix} / \\ \text{c} \end{matrix} \backslash \begin{matrix} \\ \text{b} \end{matrix} 25$

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

$(f,5) \text{ t } (,9):$ $\begin{array}{c} / \backslash \\ f \quad e \end{array} 14$ + (a,45) (b,13) (c,12) (d,16)

$(c,12) \text{ t } (b,13):$ $\begin{array}{c} / \backslash \\ c \quad b \end{array} 25$ + (a,45) (d,16) $\begin{array}{c} / \backslash \\ f \quad e \end{array} ,14$

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

$(f,5) \text{ t } (,9):$ $\begin{array}{c} / \backslash \\ f \quad e \end{array} 14$ + (a,45) (b,13) (c,12) (d,16)

$(c,12) \text{ t } (b,13):$ $\begin{array}{c} / \backslash \\ c \quad b \end{array} 25$ + (a,45) (d,16) $\begin{array}{c} / \backslash \\ f \quad e \end{array} ,14$

$\begin{array}{c} / \backslash \\ f \quad e \end{array} ,14 \text{ t } (d,16):$ $\begin{array}{c} / \backslash \\ / \backslash \\ f \quad e \quad d \end{array} 30$

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

$(f,5) \text{ t } (,9):$ $\begin{array}{c} / \backslash \\ f \quad e \end{array} 14$ + (a,45) (b,13) (c,12) (d,16)

$(c,12) \text{ t } (b,13):$ $\begin{array}{c} / \backslash \\ c \quad b \end{array} 25$ + (a,45) (d,16) $\begin{array}{c} / \backslash \\ f \quad e \end{array} ,14$

$\begin{array}{c} / \backslash \\ f \quad e \end{array} ,14 \text{ t } (d,16):$

mardi 5 novembre 2013

Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

$(f,5) \text{ t } (,9):$ $\begin{array}{c} / \backslash \\ f \quad e \end{array} 14$ + (a,45) (b,13) (c,12) (d,16)

$(c,12) \text{ t } (b,13):$ $\begin{array}{c} / \backslash \\ c \quad b \end{array} 25$ + (a,45) (d,16) $\begin{array}{c} / \backslash \\ f \quad e \end{array} ,14$

$\begin{array}{c} / \backslash \\ f \quad e \end{array} ,14 \text{ t } (d,16):$ $\begin{array}{c} / \backslash \\ / \backslash \\ f \quad e \quad d \end{array} 30$ + (a,45) $\begin{array}{c} / \backslash \\ c \quad b \end{array} ,25$

mardi 5 novembre 2013


Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

$(f,5) \text{ t } (e,9):$  14 + (a,45) (b,13) (c,12) (d,16)

$(c,12) \text{ t } (b,13):$  25 + (a,45) (d,16) (,14)

$(\text{f e},14) \text{ t } (d,16):$  30 + (a,45) (,25)

$(\text{c b},25) \text{ t } (\text{f e},30):$ 

mardi 5 novembre 2013


Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

$(f,5) \text{ t } (e,9):$  14 + (a,45) (b,13) (c,12) (d,16)

$(c,12) \text{ t } (b,13):$  25 + (a,45) (d,16) (,14)

$(\text{f e},14) \text{ t } (d,16):$  30 + (a,45) (,25)

$(\text{c b},25) \text{ t } (\text{f e},30):$  55 + (a,45)

mardi 5 novembre 2013


Exemple

Σ :	a	b	c	d	e	f
f:	45	13	12	16	9	5

$(f,5) \text{ t } (e,9):$  14 + (a,45) (b,13) (c,12) (d,16)

$(c,12) \text{ t } (b,13):$  25 + (a,45) (d,16) (,14)

$(\text{f e},14) \text{ t } (d,16):$  30 + (a,45) (,25)

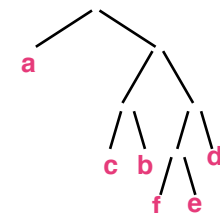
$(\text{c b},25) \text{ t } (\text{f e},30):$  55

mardi 5 novembre 2013

Exemple

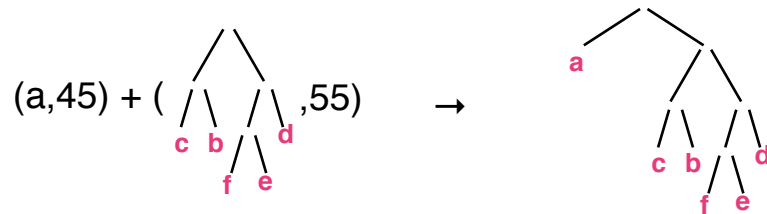
$(a,45) + ($

\rightarrow



mardi 5 novembre 2013

Exemple



Algorithme d'Huffman

Lemme 1:

Étant donnés (Σ, f) et $x, y \in \Sigma$ telles que x et y aient des fréquences minimales, alors il existe un code préfixe optimal ϕ avec $\phi(x)=w.0$ et $\phi(y)=w.1$

NB: $\phi(x)$ et $\phi(y)$ ont la même longueur et, x et y ont le même père dans l'arbre binaire associé à ϕ .

mardi 5 novembre 2013

mardi 5 novembre 2013

Algorithme d'Huffman

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

mardi 5 novembre 2013

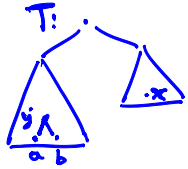
Lemme 1- preuve

- soit T l'arbre associé à un code optimal.
- soit a, b deux feuilles de T , de même père et situées à la prof. max dans T



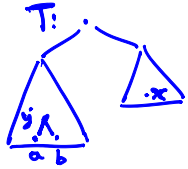
Lemme 1- preuve

- soit T l'arbre associé à un code optimal.
- soit a, b deux feuilles de T , de même père et situées à la prof. max dans T



Lemme 1- preuve

- soit T l'arbre associé à un code optimal.
- soit a, b deux feuilles de T , de même père et situées à la prof. max dans T



Lemme 2 - preuve

$$\text{pf}(x) = \text{pf}(y) = \text{pf}(z_{\text{new}}) - 1$$

mardi 5 novembre 2013

Lemme 2 - preuve

$$B(T) = \sum f(a) \cdot \text{pf}_T(a)$$

$$\text{pf}(x) = \text{pf}(y) = \text{pf}(z_{\text{new}}) - 1$$

$$B(T') = B(T) - f(x) \cdot \text{pf}(x) - f(y) \cdot \text{pf}(y) + (f(x) + f(y)) \cdot \text{pf}(z_{\text{new}})$$

mardi 5 novembre 2013

Lemme 2 - preuve

$$B(T) = \sum f(a) \cdot \text{pf}_T(a)$$

$$\text{pf}(x) = \text{pf}(y) = \text{pf}(z_{\text{new}}) - 1$$

mardi 5 novembre 2013

Lemme 2 - preuve

$$B(T) = \sum f(a) \cdot \text{pf}_T(a)$$

$$\text{pf}(x) = \text{pf}(y) = \text{pf}(z_{\text{new}}) - 1$$

$$B(T') = B(T) - f(x) \cdot \text{pf}(x) - f(y) \cdot \text{pf}(y) + (f(x) + f(y)) \cdot \text{pf}(z_{\text{new}})$$

$$B(T') = B(T) - (f(x) + f(y)) (\text{pf}(x) - \text{pf}(z_{\text{new}}))$$

mardi 5 novembre 2013

Lemme 2 - preuve

$$B(T) = \sum f(a).pf_T(a)$$

$$pf(x)=pf(y)=pf(z_{new})-1$$

$$B(T') = B(T) - f(x).pf(x) - f(y).pf(y) + (f(x)+f(y)).pf(z_{new})$$

$$B(T') = B(T) - (f(x)+f(y)) (pf(x) - pf(z_{new}))$$

$$B(T') = B(T) - f(x) - f(y)$$

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

mardi 5 novembre 2013

Lemme 2 - preuve

$$B(T) = \sum f(a).pf_T(a)$$

$$pf(x)=pf(y)=pf(z_{new})-1$$

$$B(T') = B(T) - f(x).pf(x) - f(y).pf(y) + (f(x)+f(y)).pf(z_{new})$$

$$B(T') = B(T) - (f(x)+f(y)) (pf(x) - pf(z_{new}))$$

$$B(T') = B(T) - f(x) - f(y)$$

Si T' n'est pas optimal, il existe T'' optimal pour (Σ', f')

Et remplacer z par (x,y) dans T'' donne T''' tq

$$B(T''') = B(T'') + f(x) + f(y) < B(T) !$$

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma|$

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma| = 2$: ok

- $|\Sigma| = n+1$

Soit ϕ_{algo} le code renvoyé par l'algo et ϕ_{opt} un code optimal.

Soient x et y les deux lettres choisies par l'algo avec des priorités min.

Par le Lemme 1, on peut supposer $\phi_{\text{opt}}(x)=w.0$ et $\phi_{\text{opt}}(y)=w.1$

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma| = 2$: ok

- $|\Sigma| = n+1$

Soit ϕ_{algo} le code renvoyé par l'algo et ϕ_{opt} un code optimal.

Soient x et y les deux lettres choisies par l'algo avec des priorités min.

Par le Lemme 1, on peut supposer $\phi_{\text{opt}}(x)=w.0$ et $\phi_{\text{opt}}(y)=w.1$

Par le Lemme 2, on sait que ϕ' définie par:

$$\phi'(u) = \phi_{\text{opt}}(u) \text{ si } u \in \Sigma \setminus \{x, y, z_{\text{new}}\} \text{ ET } \phi'(z_{\text{new}}) = w$$

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma| = 2$: ok

- $|\Sigma| = n+1$

Soit ϕ_{algo} le code renvoyé par l'algo et ϕ_{opt} un code optimal.

Soient x et y les deux lettres choisies par l'algo avec des priorités min.

Par le Lemme 1, on peut supposer $\phi_{\text{opt}}(x)=w.0$ et $\phi_{\text{opt}}(y)=w.1$

Par le Lemme 2, on sait que ϕ' définie par:

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma| = 2$: ok

- $|\Sigma| = n+1$

Soit ϕ_{algo} le code renvoyé par l'algo et ϕ_{opt} un code optimal.

Soient x et y les deux lettres choisies par l'algo avec des priorités min.

Par le Lemme 1, on peut supposer $\phi_{\text{opt}}(x)=w.0$ et $\phi_{\text{opt}}(y)=w.1$

Par le Lemme 2, on sait que ϕ' définie par:

$$\phi'(u) = \phi_{\text{opt}}(u) \text{ si } u \in \Sigma \setminus \{x, y, z_{\text{new}}\} \text{ ET } \phi'(z_{\text{new}}) = w$$

est optimal pour $(\Sigma', f') \dots$!

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma| = 2$: ok

- $|\Sigma| = n+1$

Soit ϕ_{algo} le code renvoyé par l'algo et ϕ_{opt} un code optimal.

Soient x et y les deux lettres choisies par l'algo avec des priorités min.

Par le Lemme 1, on peut supposer $\phi_{\text{opt}}(x)=w.0$ et $\phi_{\text{opt}}(y)=w.1$

Par le Lemme 2, on sait que ϕ' définie par:

$$\phi'(u) = \phi_{\text{opt}}(u) \text{ si } u \in \Sigma \setminus \{x, y, z_{\text{new}}\} \text{ ET } \phi'(z_{\text{new}}) = w$$

est optimal pour (Σ', f') [...] !

Et de plus: $B(\phi') = B(\phi_{\text{opt}}) - f(x) - f(y)$

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma| = 2$: ok

- $|\Sigma| = n+1$

Soit ϕ_{algo} le code renvoyé par l'algo et ϕ_{opt} un code optimal.

Soient x et y les deux lettres choisies par l'algo avec des priorités min.

Par le Lemme 1, on peut supposer $\phi_{\text{opt}}(x)=w.0$ et $\phi_{\text{opt}}(y)=w.1$

Par le Lemme 2, on sait que ϕ' définie par:

$$\phi'(u) = \phi_{\text{opt}}(u) \text{ si } u \in \Sigma \setminus \{x, y, z_{\text{new}}\} \text{ ET } \phi'(z_{\text{new}}) = w$$

est optimal pour (Σ', f') [...] !

Et de plus: $B(\phi') = B(\phi_{\text{opt}}) - f(x) - f(y)$

De son côté, l'algorithme calcule aussi un code ϕ'_{algo} pour (Σ', f') et par **hypothèse d'induction**, il est optimal, donc: $B(\phi'_{\text{algo}}) = B(\phi')$

Et on a: $B(\phi'_{\text{algo}}) = B(\phi_{\text{algo}}) - f(x) - f(y)$ (car $\phi_{\text{algo}}(x)=w'.0$ et $\phi_{\text{algo}}(y)=w'.1$)

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma| = 2$: ok

- $|\Sigma| = n+1$

Soit ϕ_{algo} le code renvoyé par l'algo et ϕ_{opt} un code optimal.

Soient x et y les deux lettres choisies par l'algo avec des priorités min.

Par le Lemme 1, on peut supposer $\phi_{\text{opt}}(x)=w.0$ et $\phi_{\text{opt}}(y)=w.1$

Par le Lemme 2, on sait que ϕ' définie par:

$$\phi'(u) = \phi_{\text{opt}}(u) \text{ si } u \in \Sigma \setminus \{x, y, z_{\text{new}}\} \text{ ET } \phi'(z_{\text{new}}) = w$$

est optimal pour (Σ', f') [...] !

Et de plus: $B(\phi') = B(\phi_{\text{opt}}) - f(x) - f(y)$

De son côté, l'algorithme calcule aussi un code ϕ'_{algo} pour (Σ', f') et par **hypothèse d'induction**, il est optimal, donc: $B(\phi'_{\text{algo}}) = B(\phi')$

mardi 5 novembre 2013

Théorème:

L'algorithme d'Huffman donne un code préfixe optimal.

Preuve: par induction sur $|\Sigma|$

- $|\Sigma| = 2$: ok

- $|\Sigma| = n+1$

Soit ϕ_{algo} le code renvoyé par l'algo et ϕ_{opt} un code optimal.

Soient x et y les deux lettres choisies par l'algo avec des priorités min.

Par le Lemme 1, on peut supposer $\phi_{\text{opt}}(x)=w.0$ et $\phi_{\text{opt}}(y)=w.1$

Par le Lemme 2, on sait que ϕ' définie par:

$$\phi'(u) = \phi_{\text{opt}}(u) \text{ si } u \in \Sigma \setminus \{x, y, z_{\text{new}}\} \text{ ET } \phi'(z_{\text{new}}) = w$$

est optimal pour (Σ', f') [...] !

Et de plus: $B(\phi') = B(\phi_{\text{opt}}) - f(x) - f(y)$

De son côté, l'algorithme calcule aussi un code ϕ'_{algo} pour (Σ', f') et par **hypothèse d'induction**, il est optimal, donc: $B(\phi'_{\text{algo}}) = B(\phi')$

Et on a: $B(\phi'_{\text{algo}}) = B(\phi_{\text{algo}}) - f(x) - f(y)$ (car $\phi_{\text{algo}}(x)=w'.0$ et $\phi_{\text{algo}}(y)=w'.1$)

D'où: $B(\phi_{\text{algo}}) = B(\phi_{\text{opt}})$

mardi 5 novembre 2013