Analyse Numérique - 1

mercredi 22 janvier 2014
09:19

Calcul numérique par machine non exact
->Pourquoi ne pas tout faire via calcul exact, analytique?

=>Certains problèmes mathématiques ne sont pas solvables analytiquement.
i.e polynômes degré>5, Equa° diff. des n corps avec n>=3, systèmes linéaires.

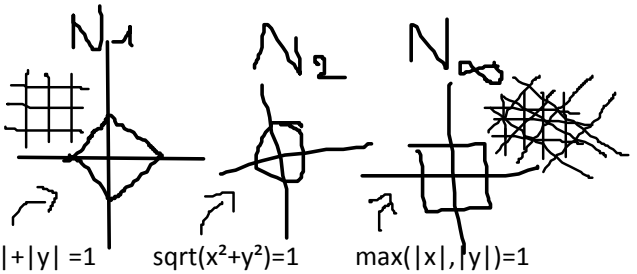
On utilise le calcul numérique pour approcher les solutions.

http://sifoci.eisti.fr

TP Scilab 1:

```
sqrt(2)
sqrt(2)*sqrt(2)

//preuve du pb du calcul num
format("v",25)
sqrt(2)*sqrt(2)
sqrt(2)*sqrt(2)-2

help ieee
//1/0
//0/0
5
ans*ans

m=10^6
sqrt(m)

y=%e
log(y)

sin(%pi)
%eps
1+%eps
//1+eps est le plus petit nombre directement à droite de 1: La précision machine.
2^(-52)

%i
x=3-4*%i
%i*%i
abs(x)
//vect ligne
v=[3.8,-4,%pi/6]
v=[3.8 -4 %pi/6]
//vect colonne
v=[3.8 ;-4; %pi/6]
size(v)

//transposition
w=v'
x=[1,2]
y=[2,3,4]
z=[x,y]

//remplir automatiquement entre les bornes
v=[4:9]
//pas
v=[4:1:9]
v=[4:2:9]
v=[0:0.1:1]

//10 nombres répartis sur une échelle d'additions
t=linspace(0,1,11)
//10 nombres répartis sur une échelle logarithmique
//(On multiplie toujours pas la même chose pour avancer)
```

```
t=logspace(1,2,10)

u = sqrt(2)*[1:2:8]'
//remplir de 1
s=ones(u)
//5 cases remplies de valeurs entre 0 et 1
t=rand(1,5)
//remplir de 0
zeros(u)

v=rand(s)
M=[1,2;3,4]
//Ecrire une matrice sur 2 lignes
N=[11,12,13,...
14,15,16]
diag([5 4 3])
diag(M)
diag(diag(M))

//mat.identité
eye(6,6)

B=eye(6,7)
A=ones(3,7)
C=[A;(-6)*B]

D=zeros(2,5)
E=rand(D)

//Construire une matrice (4,9) dont la première ligne est composée de 1
//et les autres nulles.
A=ones(1,9)
B=zeros(3,9)
C=[A;B]

//Construire une (3,5) dont la première colonne est formée de 2,
//la deuxième des nombres 1 à 3, et le reste de -1.
A=2*ones(3,1)
B=[1;2;3]
B=[1:3]'
C=-ones(3,1)
D=[A,B,C]

u=2*%pi*rand()
w=[cos(u) sin(u)]
//norme w = 1
norm(w)
norm(w,2)
norm(w,1)
norm(w,'inf')
//La norme permet de voir l'erreur. On n'utilise que 2,1, et inf.
t=[0:%pi:2*%pi]
v=sin(t)
//La valeur max de v. est dans m et elle est atteinte à l'indice k.
[m,k]=max(v)
sign(v)
```



|x|+|y| =1 V.V.neumann | sqrt(x²+y²)=1 Dist.euclidienne | max(|x|,|y|)=1 V.Moore

Norme: $racine\ n-ième(|x|^p+|y|^p)\ \ avec\ p\geq 1$

Analyse Numérique - 2

mardi 4 février 2014
10:31

Recap IEEE:

Simple prec: 32bit - signe 1, exposant 8, mantisse 23
Double prec: 64bit - signe 1, exposant 11, mantisse 52

(-1)^Signe * Mantisse ^ (Exposant-01111111)
(-1)^Signe * Mantisse ^ (Exposant-127) (simple preci)
PROTIP: **127 = 255/2**

On va de -126 jusqu'à 127 sur 8 bits.
Sur 11 bits, on va de -1022 jusqu'à 1023.

8bit	Décimal	
00000000	0	Réservé pour les exceptions (Non normalisé + zéro)
00000001		
.		
.		
.		
01111111	127	
10000000	128	
.		
.		
.		
11111111	255	Réservé pour les exceptions (Inf + NaN)

Paramètre	Simple précision	Simple précision étendue	Double précision	Double précision étendue	Quadruple (Extension (du standard)	Étendue (Introduite (par Intel)
Nombre de bits de la mantisse plus le signe (p)	24	≥ 32	53	≥ 64	113	64
Nombre de chiffres décimaux exacts ($p/\log_2 10$)	7.22	≥ 9.63	15.95	≥ 19.26	34.01	19.26
Type du codage	Bit caché	Non spécifié	Bit caché	Non spécifié	Bit caché	Bit explicite
Nombre de bits pour la mantisse ($p-1$)	23	≥ 31	52	≥ 63	112	64
$E_{\max}$ $E_{\min}$	+127 -126	$\geq$ +1023 $\leq$ -1022	+1023 -1022	$\geq$ +16383 $\leq$ -16382	+16383 -16382	+16383 -16382
Biais de l'exposant	+127	Non spécifié	+1023	Non spécifié	+16383	+16383
Nombre de bits pour l'exposant	8	≥ 11	11	≥ 15	15	15
Nombre de bits pour le signe	1	1	1	1	1	1
Nombre de bits pour le format	32	≥ 43	64	≥ 79	128	80
Valeur max décimale ($2E_{\max}+1$)	3.4028E+38	$\geq$ 1.7976E+308	1.7976E+308	$\geq$ 1.1897E+4932	1.1897E+4932	1.1897E+4932
Valeur min décimale ($2E_{\min}$)	1.1754E-38	$\leq$ 2.2250E-308	2.2250E-308	$\leq$ 3.3621E-4932	3.3621E-4932	3.3621E-4932
Valeur min décimale dénormalisée ($2E_{\min}-p+1$)	1.4012E-45	$\leq$ 1.0361E-317	4.9406E-324	$\leq$ 3.6451E-4951	6.4751E-4966	1.8225E-4951

TABLE 2.1 – Les formats du standard IEEE-754

Langage	Simple précision	Double précision	Double étendue	Quadruple précision	Etendue
SCILAb		Par défaut	En interne 80 bits		
FORTRAN	`REAL*4`	`REAL*8`		`REAL*16`	`REAL*10`
C, C++	`float`	`double`		`long double`	`long double`

TABLE 2.2 – Déclarations des formats selon le langage de programmation

2.1.1 Format simple précision

Les caractéristiques de ce format sont les suivantes :

Les exposants `00000000` et `11111111` sont à usage reservé et ne sont pas utilisés pour la conversion. Par conséquent nous avons $E_{\min} = -\text{biais}+1$ et $E_{\max} = \text{biais}$.
Un mot de 32 bits pour lequel tous ses bits sont à 0, représente la valeur décimale 0,0 . Il est évident que, en fonction de la valeur du signe s, on a un zéro positif et un zéro négatif.
Pour l'infini nous avons les représentations suivantes :

- `0 11111111 00000000 00000000000000000` $= +\infty = 7F800000_{16}$
- `1 11111111 00000000 00000000000000000` $= -\infty = FF800000_{16}$.

Exercice 1.1

Base B= 10

Mantisse p places

Exposant E

Emin<=E<=Emax

Nombre de valeurs normalisées pouvant être représentées:
[(Emax-Emin+1)-2]*B^p

Exercice 1.2

0.125*10^6 + 0.437*10^12
0.000000125*10^12 + 0.437*10^12
=0.437000125*10^12
=0.437*10^12 (car on a seulement 3 chiffres sig.)
Aucune précision.

Exercice 1.3

L'addition en représentation finie n'est pas associative.

Analyse Numérique - 3

mardi 11 février 2014
10:28

Exercice 2.5

Système de représentation de nombres réels avec 3 bits d'exposant et 3 bits de mantisse.

1) Exposants:
000 (réservé)
001 ->-2
010 ->-1
011 -> 0
100 ->+1
101 ->+2
110 ->+3
111 (réservé)

$B = 2^{q-1} - 1 = 3$

Mantisses:
000
001
011
100
101
110
111

1.(mantisse)*2^(exposant) 6 exposants possibles, 8 mantisses possibles
=> 6*8 = 48 nombres possibles.

2) Peut-on avoir (m,e) et (m',e') qui codent le même nombre?
Y'a-t-il vraiment 48 nombres codables distincts?

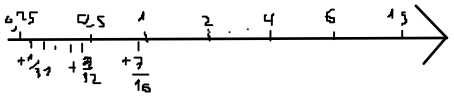
Non, car l'exposant règle un intervalle sur lequel on code le nombre (pas exponentiel)
Tandis que la mantisse donne le nombre, de 1.000 à 1.111. (<2) (pas arithmétique)

1.111 = 1 + 0.5 + 0.25 + 0.125

3)
Plus petit nombre: 1.000 * 2^001 = 2^(-2) = 0,25
Plus grand nombre: 1.111 * 2^110 = 1.875 * 2^3 = 15

4)
Précision machine = 1/32, 1/16, 1/8…

5)



Exposant Mantisse	001 2^(-2)	010 2^(-1)	011 1	100 2	101 4	110 6
000	0.25	0.5	1	2	4	8
001	0.25 + 1/32	0.5 + 1/16	1 + 1/8	2 + 1/4	4 + 1/2 = 4,5	8+1 = 9
010	0.25 + 2/32	0.5 + 2/16	1 + 2/8	2 + 2/4	4 + 1 = 5	8+2 = 10
…	…	…	…	…	…	…
110	0.25 + 6/32	0.5 + 6/16	1 + 6/8	2 + 6/4	4 + 6/2 = 7	8+6 = 14
111	0.25 + 7/32	0.5 + 7/16	1 + 7/8	2 + 7/4	4 + 7/2 = 7,5	8+7 = 15

Exercice 2.2

(-1)^Signe * Mantisse ^ (Exposant-01111111)
(-1)^Signe * Mantisse ^ (Exposant-127) (simple preci)
PROTIP: **127 = 255/2**

Coder 0.1 en binaire puis le décoder.

IEEE : 0 | 000001 | 01111110

Binaire: 0.1 = 1/16 + 1/32 + 1/256 + 1/512 + …

Exercice 2.6

```
while  i +1 <> 1 do
i=i/2;
End;
Return i*2;
```

dssd

Analyse Numérique - 4

mardi 18 février 2014
08:38

Exercice 3.2
Soit a² - b².

On peut l'exprimer de deux façons, qui donnent un résulat identique avec des chiffres exacts:
A1: (a*a)-(b*b) A2:(a+b)(a-b)

Si a et b sont déjà inexacts, quelle va être l'inexactitude avec a², b², ou des opérations entre a et b? Quelle opération faut-il effectuer pour limiter les erreurs?

On calcule Δy pour A1 et A2: (voir equation 3.19):

Calcul A1:
ϕ1: s <- a*a ϕ1: $\binom{a}{b} = \binom{a*a}{b} = \binom{s}{b}$
ϕ2: t <- b*b ϕ2: $\binom{s}{b} = \binom{s}{b*b} = \binom{s}{t}$
ϕ3: w <- s-t ϕ3: $\binom{s}{t} = [s-t]$
L'algorithme est réecrit sous forme vectorielle.

ϕ = ϕ3 ∘ ϕ2 ∘ ϕ1 ψ1=(ϕ3 ∘ ϕ2)
= (ϕ3 ∘ ϕ2) ∘ ϕ1 ψ2=ϕ3
= ψ1 ∘ ϕ1
= ψ2 ∘ (ϕ2 ∘ ϕ1)

J[ϕ(x)] =[$\frac{d\phi(x)}{da}, \frac{d\phi(x)}{db}$] $= [2a, -2b]$

Ψ1 (k=1):
$(\phi 3 \circ \phi 2)\binom{s}{b} = s - b^2$
J[ψ1] = $[\frac{d\psi 1}{ds}, \frac{d\psi 1}{db}] = [1, -2b]$

H1 = $\begin{bmatrix} \eta 1 & 0 \\ 0 & 0 \end{bmatrix}$ $On\ calcule\ l'erreur\ lors\ de\ l'opération\ a * a.$

Ψ2 (k=2):
$\phi 3\binom{s}{t} = \ s - t$
J[ψ2] = $\left[\frac{d\psi 2}{ds}, \frac{d\psi 2}{db}\right] = [1, -1]$

H2 = $\begin{bmatrix} 0 & 0 \\ 0 & \eta 2 \end{bmatrix}$ $On\ calcule\ l'erreur\ lors\ de\ l'opération\ b * b.$

Ψ3 (k=3):
H3= [η3] $La\ dernière\ opération(soustraction)implique\ tout\ de\ même\ une\ erreur.$

D'où le calcul de Δ1:
Δy ≈ $[2a, -2b]\begin{bmatrix} \Delta a \\ \Delta b \end{bmatrix}$
$+ [1, -2b]\begin{bmatrix} \eta 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} s \\ b \end{bmatrix}$ <- Algo à i=1
$+[1, -1]\begin{bmatrix} 0 & 0 \\ 0 & \eta 2 \end{bmatrix}\begin{bmatrix} s \\ t \end{bmatrix}$ <- Algo à i=2
$+[\eta 3](a^2 - b^2)$ <- Algo à i=3

Δy ≈ 2aΔa $- 2b\Delta b + a^2\eta 1 - b^2\eta 2 + \eta 3(a^2 - b^2)$

Calcul A2

ϕ1: s <- a+b ϕ1: $\binom{a}{b} = \begin{pmatrix} a \\ b \\ a+b \end{pmatrix}$
ϕ2: t <- a-b ϕ2: $\begin{pmatrix} a \\ b \\ s \end{pmatrix} = \binom{t}{s}$
ϕ3: w <- s*t ϕ3: $\binom{t}{s} = [s * t]$

Ψ1 = $\phi 3 \circ \phi 2\begin{pmatrix} a \\ b \\ s \end{pmatrix} = (a - b) * s$
J[ψ1] = $[s, -s, a - b]$

H1 = $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \eta 1 \end{bmatrix}$

Ψ2 = $\phi 3\binom{t}{s} = s * t$
J[ψ2] = $[s, t]$

H2 = $\begin{bmatrix} 0 & 0 \\ 0 & \eta 2 \end{bmatrix}$

H3= [η3]

Calcul de Δ2:

$$\Delta y \approx [2a, -2b]\begin{bmatrix}\Delta a\\ \Delta b\end{bmatrix} + [s, -s, a-b]\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & \eta 1\end{bmatrix}\begin{pmatrix}a\\ b\\ s\end{pmatrix} + [s,t]\begin{bmatrix}0 & 0\\ 0 & \eta 2\end{bmatrix}\begin{pmatrix}t\\ s\end{pmatrix} + [\eta 3](a^2-b^2)$$

$$\Delta y \approx 2a\Delta - 2b\Delta b + \eta 1(a^2-b^2) + \eta 2(a^2-b^2) + \eta 3(a^2-b^2)$$

<u>Comparaison Finale</u>

Les ε représentent l'erreur.

$$|E(A1,x)| = |a^2\varepsilon 1 - b^2\varepsilon 2 + (a^2-b^2)\varepsilon 3| \leq (a^2+b^2+|a^2-b^2|)\varepsilon < -Factorisation\ de\ \Delta 1$$
$$|E(A2,x)| \leq 3|a^2-b^2|\varepsilon < -Factorisation\ de\ \Delta 2$$

A2 est meilleur que A1 si:

$$3|a^2-b^2| \leq (a^2+b^2+|a^2-b^2|)$$

$$1/3 \leq (\frac{a}{b})^2 \leq 3$$

Analyse Numérique - 5

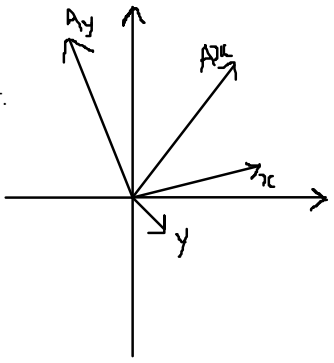
mardi 18 mars 2014
08:56

Etude de transformées de matrices.

Soit A la matrice d'une transformation linéaire T.
Soit x un vecteur.

Ax est la transformée du vecteur par T.

Si Ax = x', alors A(2x) = 2x'
Si Ax = x' et Ay = y' alors A(x+y) = Ax+Ay = x' + y'
A(αx+βy) = αx'+βy' etc.



De la même manière que l'on a des suites de réels convergentes,
On voudrait définir des suites de matrices convergentes.
==> Définir comment se déforme un vecteur par une application/transformation codée par une matrice
==> Appliquer cette transfo de manière répétée.

Si une transfo renvoie tous les vecteurs dans une boîte plus petite, on peut itérer pour atteindre des intervalles 2D de plus en plus petits.

Exercice 4.2

```
A = [2 1 1;2 3 2;1 1 2];
//norme standard (2 sous scilab)
norm(A)
//norme 1
norm(A,1)
//norme inf
norm(A,'inf')
```

La norme par défaut est la norme 2.
N1=5, N2=5,2035273, Ninf=7.

Exo 2 Feuille

$$|\|(A+\Delta A)+(B+\Delta B)-(A+B)\|p \leq |\|(A+\Delta A)-A\||p + |\|(B+\Delta B)-B\||p$$
$$= |\|A\||p * \varepsilon A + |\|B\||p * \varepsilon B$$
$$\leq (|\|A\||p + |\|B\||p) * \max(\varepsilon A, \varepsilon B)$$

$\varepsilon A = \frac{|\|\Delta A\||p|}{|\|A\||p}$ $\varepsilon B = \frac{|\|\Delta B\||p}{|\|B\||p}$

Le nombre condition pour l'addition est $\frac{|\|A\||p+|\|B\||p}{|\|A+B\||p}$

Donc le problème est mal conditionné si $(|\|A\||p + |\|B\||p) \gg |\|A + B\||p$

Exo 3 Feuille

$$|\|(A+\Delta A)(B+\Delta B)-AB\||p \leq |\|A\|| \, |\|\Delta B\|| + |\|\Delta A\|| \, |\|B\|| + |\|\Delta A\|| |\|\Delta B\||$$
$$|\|A\|| \, |\|B\|| * \varepsilon B + |\|A\|| |\|B\|| * \varepsilon A + |\|A\| \varepsilon A| * ||B|| \varepsilon B$$

$De\ même, si\ |\|A\|| |\|B\|| \gg |\|AB\||\ alors\ le\ produit\ est\ mal\ conditionné$

Analyse Numérique - 7

mercredi 9 avril 2014
11:08

On cherche le cas idéal: des matrices diagonales.

Ainsi, on n'a qu'a stocker les coefficients diagonaux, et en multiplier deux revient à une multiplication terme à terme.
(une multiplication de deux matrices standard a un coût n^3)

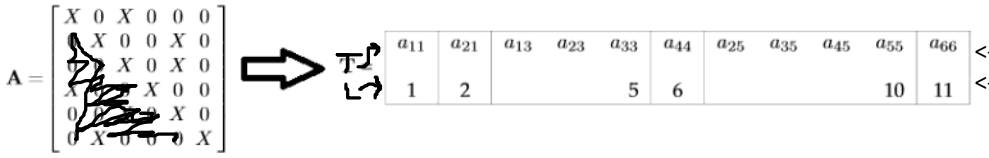
Exercice 1
Soit Ax = b un SEL (système d'équations linéaires)
Montrer avec P une matrice de permutation, que même permuté le SEL reste correct.

P est une matrice qui permute les sommets :

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{matrix} * \begin{matrix} x \\ y \\ z \end{matrix} = \begin{matrix} x \\ z \\ y \end{matrix} \qquad \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{matrix} * \begin{matrix} x \\ y \\ z \end{matrix} = \begin{matrix} z \\ x \\ y \end{matrix}$$

Ici inverse de P = P ici inverse de P = transposée

Exercice 3



a_{11}	a_{21}	a_{13}	a_{23}	a_{33}	a_{44}	a_{25}	a_{35}	a_{45}	a_{55}	a_{66}
1	2			5	6				10	11

<-Liste des coefficients
<- Liste des indices des coefs. diagonaux

Montrer que le coefficient a_ij (avec j>=i) est à l'indice k = L(j)-(j-i) dans le tableau T.
K doit être supérieur à L(j-1).

Analyse Numérique - 10

jeudi 15 mai 2014
09:20

Exo Sup 0.1

Si A est de rang 1, chacune de ses colonnes est un multiple de toutes les autres

$u = \begin{matrix} 1 \\ 2 \\ 4 \\ -1 \end{matrix}$ $v = \begin{matrix} 2 \\ 1 \\ 1 \end{matrix}$

Exo 10.5 p.173

A =

1. 1. 0.
0. 0. 1.

A(T) =

1. 0.
1. 0.
0. 1.

A*A(T) =

2. 0.
0. 1.

λ1=1, λ2=2

A(T)*A =

1. 1. 0.
1. 1. 0.
0. 0. 1.

$\text{Det}(\lambda I - A(T)*A) = \begin{vmatrix} \lambda-1 & 1 & 0 \\ 1 & \lambda-1 & 0 \\ 0 & 0 & \lambda-1 \end{vmatrix} = (\lambda-1)\begin{vmatrix} \lambda-1 & 1 \\ 1 & \lambda-1 \end{vmatrix} = (\lambda-1)((\lambda-1)^2-1) = (\lambda-1)(\lambda-2)\lambda$

Image et Noyau de A et A(T):

Dim(E) = dim(Im(A)) + dim(ker(A))

Ker(A) = {x appartennt à R3 tq Ax = 0}. $A*x = \begin{matrix} x+y \\ z \end{matrix}$

Ker(A) = Vect$\left(\begin{matrix} 1 \\ -1 \\ 0 \end{matrix}\right)$ dim(ker(A) = 1.

D'où dim(Im(A)) = 2. Im(A)={y appartenant à R2 tq Ax=y}

Im(A) = Vect$\left(\begin{vmatrix} 1 \\ 0 \end{vmatrix}, \begin{vmatrix} 0 \\ 1 \end{vmatrix}\right)$

Im(A(T)) = Vect$\left(\begin{vmatrix} 1 \\ 1 \\ 0 \end{vmatrix}, \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix}\right)$

Dim = 2.

Or, A(T): R2 -> R3. Donc dim(ker)=0.

GL2

mardi 28 janvier 2014
10:23

UML Distilled: Fowler 2004
Software Engineering: Sommerville 2007

Pierre/Benjamin/Thibault

Créer un git/SVN -> http://www.commentcamarche.net/faq/13094-utiliser-git-sous-windows
http://msysgit.github.io/

Intégrer une documentation (rapport technique etc écrit avant le développement)

Préparation au développement:

***Analyse des besoins, en particulier des besoins non fonctionnels et fonctionnels.**

$ Besoins non-fonctionnels:
Qualités globales que l'on attend du logiciel mais qui ne correspondent pas aux fonctionnalités propres du logiciel.
(**Présence très limitée pour le projet GL2**)
Ces besoins sont souvent les plus critiques.
Exemples: **Besoins en comportement:*
Performances, fiabilité, facilité d'utilisation/interface graphique, domaine d'action, portabilité…
**Besoins organisationnels:*
Standards, processus de développement: Méthode d'organisation, Algorithmes…
Gestion du temps: Plannings, calendrier des livrables.
**Besoins externes:*
Interopérabilité: Interaction avec d'autres systèmes
Contraintes légales: Code utilisé, open source, GNU/GPL, etc.
Contraintes éthiques: Acceptabilité par les utilisateurs
Performances acceptables, minimisation des pertes de données, etc. sont des exemples précis de besoins non fonctionnels.
Il ne suffit pas de simplement énoncer les besoins, il faut quantifier, donner des bornes.
Par exemple, interface utilisable par des enfants de moins de 5 ans, valeurs min/max des données traitées et des temps de réponse…
Transactions par seconde, temps de réponse, nombre d'octets pris en mémoire, temps moyen avant un échec…
Ces bornes sont déterminées par des prototypes.

$ Besoins fonctionnels:
Fonctionnalités nécessaires au logiciel, ce qu'il doit faire et les services qu'il doit rendre.
Ces besoins appartiennent à deux types, utilisateur ou système.
Exemples: *L'utilisateur doit pouvoir rechercher dans un ensemble par défaut ou des sous-ensembles de BDD
*L'utilisateur doit pouvoir garder une trace du résultat de ses requêtes, par exemple par impression(requête utilisateur)
*Chaque recherche laisse un log (requête système)

La description des besoins fonctionnels doit être:
Abstraite: Ne pas influencer l'implémentation
Claire: Pouvoir en discuter avec le client
Précise: Ils doivent pouvoir servir à détailler les besoins non-fonctionnels.
Hiérarchisée: Poser des priorités.
Ne jamais faire apparaître des éléments liés à des langages ou des styles de programmation.
=>Ne pas parler du "comment", juste du "quoi"

La description de ces besoins doit être précisée par:
*La quantification
*La spécification de tests de contrôle/validation (ex: fixer une taille de données sur laquelle on peut atteindre le seuil de performance raisonnable
*La description de contraintes ou difficultés techniques (**ici, java imposé**)
*L'enonciation de risques: Prévision de ce qui pourrait mal se passer(ex: Choix techniques sans bonne connaissance(**ici, java**), ou utilisation d'outils sans maintenance)
*Leur priorité dans le développement.

En général, la liste de besoins se met en place à l'aide d'itérations successives au fil des discussions avec le client.

Conclusion:
Etablir une liste des besoins fonctionnels et non fonctionnels, ainsi que leur description et analyse, avec à chaque fois:
-Quantifications
-Tests de contrôle et de validation
-Contraintes et problèmes techniques
-Risques et parades, niveaux de difficulté
-Priorités dans le développement.

***Elaboration de schémas/représentations graphiques, mettant en avant les besoins et structures du logiciel prévu.** (Ex: Diagrammes UML, dessin interface graphique)

***Planification du développement, calendrier et gestion du temps.**
-> Prendre en charge les temps d'apprentissage des technologies utilisées (Java, UML, MVC, etc)

====> Passer 2 semaines sur les besoins, par la suite finaliser le cahier des charges avec les schémas UML, et commencer à coder vers fin Février.

Liste de besoins fonctionnels et non fonctionnels
Description et analyse de ces besoins, avec si nécessaires:
*Quantifications
*Priorités dans le développement
*Tests de contrôle et de validation
*Contraintes et problèmes techniques
*Risques et parades, niveaux de difficulté.
Scénarios, représentations, schémas, prototypes papier
Planning DdGantt

Réaliser un scénario:
Environnement et acteurs ou le logiciel est requis.

Un scénario induit des besoins fonctionnels, et met en scène les besoins non-fonctionnels.

Ils peuvent être des tests de variation.
Pour justifier les besoins, on peut décrire des ensembles de scénarios.

Moyens de représentation:
Représentent le système et ses composants.
Fort efficace, en particulier pour aider à la mise en place des besoins fonctionnels:

-Souvent graphiques
-Souvent multiples
-Protoypes papier, schémas de structure, UML, etc.

UML très utilisé en entreprise en tant que vecteur de communication entre l'ingénieur et le client.

Au minimum:
-Etablir la liste des tâches par rapport aux besoins
-Etablir un calendrier en utilisant l'analyse des besoins et répartir les tâches
-Déterminer les versions et phases successives du programme (Livrables):
Quelque chose que le client peut voir et tester.

Diagramme de Gantt

Prolog - 1

mercredi 19 mars 2014
14:17

Propositions | Prédicats

Eléments de la logique

-Noms
-Propriétés

couleurs('livre') = bleue
f:E -> F

-Propriétés
couleur('livre', bleue) = vraie
E -> {vraie,fausse}

-Propositions
- Vraie
- Composée

P -> Q == ¬ P V Q

Exercice 2:

p: x>0
q: y>0
r: Afficher (…)

P1: (pΛq)→r = ¬ (pΛq)Vr = ¬ P V ¬q V r
P2: p → (q → r) = p → (¬q V r) = ¬P V ¬ q V r = P1

OK.

Exercice 5

Constantes a, b
Variables x,y,z
Relation binaire f
U = {O, *}

I: a → aI = O
b → bI = *
x → ϕI (x) = O
y → ϕI(y) = O
z → ϕI(z) = *
$fI = \{(O,*),(*,*)\}$

$\neg f(b,a)$	$\neg fI(*,O)$	VRAI
$f(a,b)\ \Lambda\ f(b,a)$	$fI(O,*)\ \Lambda\ fI(*,O)$	FAUX
$f(a,b)\ V\ f(b,a)$	$fI(O,*)\ V\ fI(*,O)$	VRAI
$f(a,b)\ \rightarrow\ f(b,a)$	$fI(O,*)\ \rightarrow\ fI(*,O)$	FAUX
$f(a,b)\ \leftrightarrow\ \neg f(b,a)$	$fI(O,*)\ \leftrightarrow\ f(*,O)$	VRAI

TD 1
Exercice 1:

$P\ -{>}\ (q\ V\ (r\ \leftrightarrow\ (r\ \rightarrow\ \neg p)))$

P	Q	R	¬P	$R\ \rightarrow\ \neg P$	$R\ \leftrightarrow\ (R \rightarrow \neg P)$	$Q\ V\ (R\ \leftrightarrow\ (R\ \rightarrow \neg P))$	P -> A
0	0	0	1	1	0	0	1
0	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1
0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	1	0	1	1
1	1	1	0	0	0	1	1

P	Q	P->Q
0	0	1
0	1	1
1	0	0
1	1	1

Connecteurs : {P, Δ, q, ->, <->}

V: OU
Λ: ET
¬: NOT
->: IMPLY (alors)
⊥ : FAUX =0
T : VRAI = 1

Toto est grand et lolo est son ami

Exercice 3

(p ΛqΛr → ⊥) Λ (¬qΛs → p) Λ (T → r)

Exercice 6

$\forall x\ \exists yp\ (x,y) \rightarrow \exists y \forall xp(x,y)\ \rightarrow fcn$

1) Elimination du connecteur ↔
2) Elimination du connecteur →
3) Transfert du connecteur ¬ au niveau le plus intérieur
4) Renommage des variables de sorte que chaque variable apparaisse une seule fois sous la portée d'un quantificateur
5) Suppression des quantificateurs non opérationnel
6) Suppression des quantificateurs existentiels par application de la fonction de Skolan
7) Transformation en forme prénexe: Tous les quantificateurs universels se trouvent au début de la formule
8) Suppression des quantificateurs universels

e

Prolog - 2

mercredi 9 avril 2014
16:28

Voir fichiers td3.pl et fib.pl.

```
?-length(L,9).                          Crée une liste de 9(noms de variables interne genre _Gxxx)
L=[X11,X12,X13,X21,X22,X23,X31,X32,X33]. Définition de variables
?-numlist(1,9,L).                       Prend les nombres de 1 à 9 et les mets dans une liste
?-between(1,9,X).                       Prend les nombres de 1 à 9 et les donne un par un
?-time(fib(25,R)).                      Affiche le temps pris par l'opération
fib_iteratif_aux(_X,Y,1,Y).             Le _ précise que nous n'avons pas besoin de la variable
```

```
:- begin_tests(mestests).     Début gamme tests
test(somme) :-                Préciser le prédicat sur lequel on teste
    ____,                     test
    ____.                     test
:- end_tests(mestest).        Fin gamme tests
```

Prolog - 3

vendredi 18 avril 2014
09:26

`findall(Motif, But, liste)` : Trouve tous les motifs tel que but et les stocke dans liste.

`between(x,y,i)` : Equivalent d'un for de x à y incrémenté de i.

`!` : Cut. Si le prédicat précédent est vrai, on continue, sinon on passe à la définition suivante.

`apropos(predicat)` : aide.

```
functor(f(a,b),F,A).
nth0(N,[a,b,c],X).
nth1(N,[a,b,c],X).
display("toto").
name(toto,X).
name(X,"toto").
```

Stats Descriptives - 1

vendredi 4 avril 2014
09:23

Nature:
Qualitatif: Si non quantifiable
Quantitatif: Si quantifiable

Echelle de mesure:
Nominal: Non triable
Dichotomique: Binaire
Ordinal: Triable

Discret: Entier
Continu: Réel

1. Type de caractères

Pour chaque caractère, déterminer sa nature et préciser son échelle de mesure.

	Qualitatif			Quantitatif	
	nominal	dichotomique	ordinal	discret	continu
Nbre d'enfants dans un foyer					
Age					
Département					
Note sous forme A, B, C, D, E, F					
Date de naissance					
Classe d'âge					
Note sur 20					
Abstentionniste					
Sexe					

Exercice 2

1°)Dans le cas d'un caractère dichotomique, codé en 0 ou 1, la moyenne donne le nombre de 1 sur le total, c'est-à-dire la proportion d'individus ayant le caractère 1.

2°)

- Si le caractère de départ est quantitatif discret, il n'y a rien à faire: seulement considérer les valeurs du caractère comme les modalités ordonnées d'un caractère qualitatif.
- Si le caractère est continu, il faut regrouper la population par intervalles ordonnés. Cette transformation est à éviter si possible, car on perd de l'information.

3°)Lorsqu'on transforme un caractère ordinal en caractère quantitatif discret, on remplace les modalités par des nombres. On impose ainsi des relations entre les écarts des différentes modalités, relations qui n'existaient pas auparavant:
Ecart entre 1ère et 3ème modalité égal (ou double ou moitié) de l'écart entre 4ème et 6ème modalité.

4°)L'échelle de richesse permet de savoir que:
a)On peut transformer un caractère en caractère moins riche
b)Tout traitement appliqué à un caractère peut être appliqué aux caractères plus riches

Exercice 3

1°)
a) On désire étudier X = consommation de livres dans les foyers français
X dépend de Y = catégorie socio-professionnelle(agriculteur,cadre,etc)
Et de Z = nombre d'enfants scolarisés (0,1,2,3)

Pour fabriquer un échantillon de 1000 par strates, on a besoin de connaître les proportions de foyers agriculteurs à 0 enfants, à 1 enfants, de foyers employés à 0, à 1, etc.

Il faut connaître la proportion dans la population de chaque couple de modalité (yi,zi)

Zi/yi	Agriculteur	Employé	Cadres
0	1	13	
1	2	20	
2			
3			

Le nombre de strates sera égal à
(modalités de Y) * (modalités de Z).

b) L'inconvénient est que l'on multiplie le nombre de strates, et qu'on réduit donc le nombre d'individus à trier au hasard dans chaque strate.

2°) Population = individus numérotés de 1 à N.
Echantillon souhaité de taille: n
Population divisée en k strates.
Fonction S qui à chaque individu i appartenant à [|1;N|] associe s(i) appartenant à [|1;k|] numéro de sa strate.
Effectif de chaque strate e(j) (j appartient à [|1;k|])

Stats Descriptives - 2

mercredi 16 avril 2014
16:07

TD3

Exercice 1:

a) Dernière ligne=distribution marginale du caractère Y = sexe
Dernière colonne = distribution marginale du caractère X = niveau d'étude

On peut obtenir la distribution en fréquence: fij = nij/ntotal
On peut obtenir les profils-lignes en divisant par le total de chaque ligne.
On peut obtenir les profils-colonnes en divisant par le total de chaque colonne.

b) n1. = effectif total des individus ayant la modalité n1 pour X.
Ici: n1. = 94 =nombre d'individus ayant bac+3.

N01 = effectif total des individus ayant la modalité y1 pour Y.
Ici: n.1=65= nombre de femmes.

fij=frequence des individus ayant X = ième modalité et Y=jème modalité.
f22= 11/131=8.3% de cette population ont bac+5 et sont des hommes.
F31= 4/131.

c) Distribution marginale de X:

X	Bac+3	Bac+5	Bac+8
Effectif	94	27	10

Total: 131

Distribution marginale de Y:

Y	Feminin	Masculin
Effectif	65	66

Total: 131

d)

Bac+3	Bac+5	Bac+8
45	16	4

e) fi/j = fréquence des indices avec X=xi parmi ceux avec Y=yj.

= nij/n.j = $\frac{nij}{n.j} = \frac{nij}{n} * \frac{n}{n.j} = \frac{\frac{nij}{n}}{\frac{n.j}{n}} = \frac{fij}{f.j}$

fi.=$\frac{ni.}{n} = fréquence\ marginale\ de\ la\ modalité\ xi\ de\ X.$

Exercice 2:

1) Deux variables: -Famille politique, 3 modalités
-Opinion sur la cohabitation, deux modalités.

Théorie des Langages - 2

vendredi 11 avril 2014
09:13

L est le langage des mots qui se terminent par b
L={b,ab,bb,aab,abb,bab,bbb,...}

Il existe un algorithme qui prend un automate non deterministe quelconque et retourne un automate déterministe qui connait le même langage.

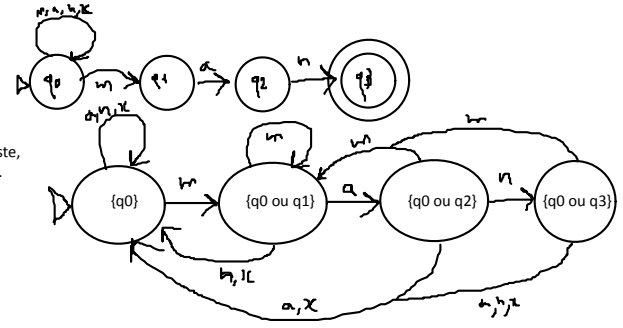
Soit L le langage des mots sur E = {m,a,b,x} qui se terminent par man.

1)Donner un automate non détrministe le plus simple possible qui reconnaît L.
2)Donner un automate déterministe qui reconnaît L.

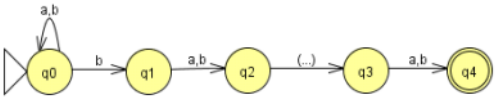
L={man,aman,nman,xman,mman, mnman,15 autres mots de taille 5, etc...}
1)
Etape Q0: Boucler jusqu'à trouver m
Q1: Voir si a
Q2: voir si n

Q3: Fini!

2)Pour passer à un automate déterministe, détailler tous les états à chaque étape.



Soit E = {a,b}
L1 est le langage des mots qui se terminent par b.
L2 est _____ dont l'avant-dernière lettre est un b.
L3 est _____ dont l'antépénultième lettre est un b.
Lk est _____ dont la kième lettre avant la fin est un b.

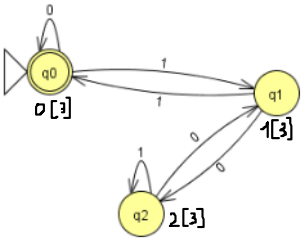


Version non-déterministe de Lk: automate à k+1 états.

Soit E = {0,1}
Donner un automate déterministe qui reconnaît le langage des nombres(en binaire) divisibles par 3.

L = {0,00,11,000,011,110,...}

Ajout d'un 0 à X en binaire: 2X
Ajout d'un 1: 2X + 1



Modulos etc.
Tests: 11,110,1001,1100,1111,10010

Théorie des Langages - 5

vendredi 9 mai 2014
10:59

Données: Grammaire G de type 2 et un mot w

Question: w appartient-il à L(G)?

(1) S —> b A
(2) S —> a B
(3) A —> b A A
(4) A —> a S
(5) A —> a
(6) B —> a B B
(7) B —> b S
(8) B —> b

T={a,b} Elements du langage (**Terminaux**)
N= {S,A,B} Elements groupant d'autres (**Non-terminaux**)
S=S Début
P={,} Possibilités

Définir L(G):

ε = elt vide
L(G) = {ε,a,b,aa,ab,ba,b,aaa,aab,aba,abb,baa,bab,bba,bbb,aaaa,abba,baab,...}

Dans L
Pas dans L

On peut en déduire que:
S = $\{w \mid |w_a| = |w_b|\}$ *Langage des mots ayant autant de a que de b*
A = $\{w \mid |w_a| = |w_b| + 1\}$ *Langage des mots ayant un b de plus*
B = $\{w \mid |w_a| + 1 = |w_b|\}$ *Langage des mots ayant un a de plus*

Sous-mot Taille 6	S (a1a2b3b4a5b6)	S -> A1 B				
Sous-mot Taille 5	A(4') (a1a2b3b4a5)	B (a2b3b4a5b6)	B -> A1 B2			
Sous-mot Taille 4	S(2') (a1a2b3b4)	S(2') (a2b3b4a5)	B2(12') (b3b4a5b6)	B2 -> B B		
Sous-mot Taille 3	A(règle 4') (a1a2b3)	B(règle 6') (a2b3b4)	B(règle 7') (b3b4a5)	B(règle 7') (b4a5b6)	B -> B1 S	
Sous-mot Taille 2	A2(règle 11') (a1a2)	S(règle 2') (a2b3)	B2(règle 12') (b3b4)	S(règle 1') (b4b5)	S(règle 2') (a5b6)	S -> A1 B
Sous-mot Taille 1	A,A1 (a1)	A,A1 (a2)	B,B1 (b3)	B,B1 (b4)	A,A1 (a5)	B,B1 (b6)
	a1	a2	b3	b4	a5	b6

(a1a2b3) = a1(a2 b3) OU (a1 a2) b3
= A,A1 S OU A B,B1
(a1a2b3b4) = a1(a2b3b4) OU (a1a2)(b3b4)
OU (a1a2b3) b4
Etc.

Mise en forme normale de Chomsky:

Données: Une grammaire de type 2

Question: Trouver une grammaire en forme normale de Chomsky équivalente i.e dont toutes règles sont de la forme:
A -> a (un terminal)
A -> BC (deux non-terminaux)

Etape 1:
On ajoute les deux règles suivantes:
A1 -> a
B1 -> b

Etape 2:
A -> b(AA)
B -> a(BB)
A2 -> AA (A2 est donc le langage des mots ayant deux b de plus)
B2 -> BB (etc)

Etape 3:
Réecrire la grammaire avec les règles ajoutées:

(1') S -> B1 A
(2') S -> A1 B1
(3') A -> B1 A2
(4') A -> A1 S
(5') A -> a
(6') B -> A1 B2
(7') B -> B1 S
(8') B -> b
(9') A1 -> a
(10') B1 -> b
(11') A2 -> AA
(12') B2 -> BB

Le résultat est bien sous forme normale de Chomsky équivalente.

Algo II - Ford-Fulkerson

mardi 8 avril 2014
11:19

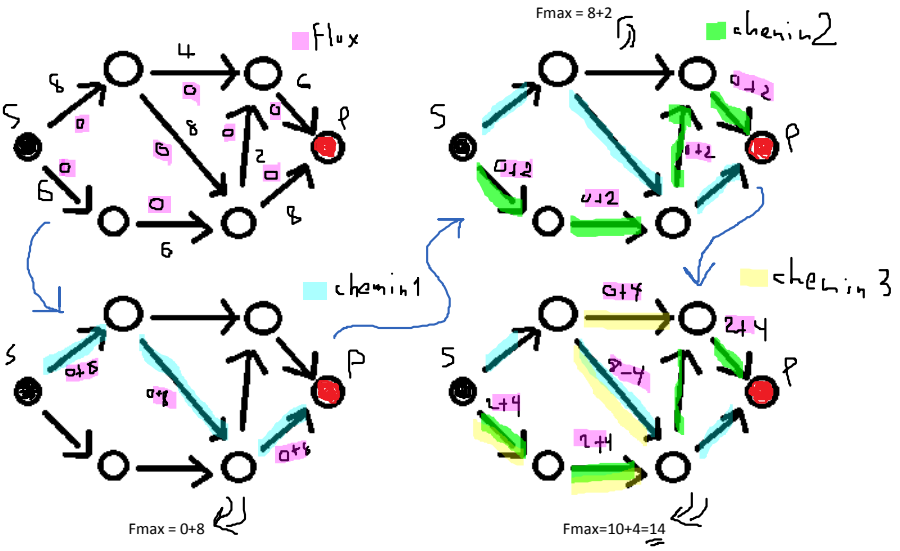
Soit un graphe ayant des arcs de capacité C.
L'algo de Ford-F permet de trouver le chemin optimal évitant la saturation de cette capacité pour un flux F.
Il nous donne le flux max.

<u>DEROULEMENT:</u>
Depuis le sommet S, aller de sommet en sommet
Toujours vérifier si le flux est strictement inférieur à la capacité avant de marquer. En chemin direct, on marque par +.

Dés qu'on arrive à P, on a une chaîne améliorante.
On prend min{eps-;eps+} eps+ si chemin direct, eps- sinon.
On prend la capacité minimale de tous les arcs de la chaîne: C'est le flux maximal pour le chemin. Pour tous les flux du chemin, on leur ajoute ce flux maximal.
On enlève les marquages, et on pose
Flux Maximal Général = FluxMax Général + FluxMax Chemin.

On réitére en repartant de S. Certains arcs sont désormais saturés(flux pleins), il faut essayer ainsi de trouver un autre chemin.

Quand on ne trouve plus de chemin direct(flux saturés) il faut faire un marquage indirect:
Choisir un arc non marqué et avec un flux non nul et le marquer par -
Une fois le flux maximal du chemin calculé, pour tous les flux marqués d'un -, on leur soustrait ce flux maximal.



Théorie de l'information - 1

mardi 15 avril 2014
16:29

Entropie: H= $-\sum_1^n Pi * \log 2\ Pi$
L'entropie représente l'incertitude.

Log2(x) = log(x)/log(2)

TD1 p.14:

Exercice 1: Calcul de l'entropie d'une source.

X une source d'alphabet [1,2,3,4,5].
Ici n = 5.

1) H = -0.1 log2(0.1) -0.2 log2(0.2) - 0.3log2(0.3) - 0.15log2(0.15) -0.25log2(0.25) = 2.23 bits/symbole
2) H = -4* 0.05*log2(0.05) -0.8log2(0.8) = 1.12193 bits/symbole
3) H = log2(n) = log2(5) = 2,32 bits/symbole

Exercice 2

1) L'alphabet de la source est [pile] de taille 1. H = log2(1) = 0
2) Alphabet: [pair,impair] de taille 2. H = log2(2) = 1 bit/symbole
3)

Xi	Pique	Trèfle	Cœurs	Carreau
Pi	3/10	4/10	2/10	1/10

H = -0.3*log2(0.3)-0.4*log2(0.4)-0.2*log2(0.2)-0.1*log2(0.1) = 1.85 bits/symbole.

Exercice 3

1) H(X) = -0.1log2(0.1) - 0.2log2(0.2) -0.3log2(0.3) - 0.15log2(0.15) -0.25log2(0.25) = 2.23 bits/symbole
2) p=P(A) = 0.1+0.3+0.25 = 0.65
   q=P(B)= 1-p = 0.35
   ===> H(p,q) = -0.65log2(0.65)-0.35log2(0.35) = 0.93 bits/symbole
3) Ha? Hb?
   A:

X	1	3	5
Pa(X)	0.1/0.65	0.3/0.65	0.25/0.65

   B:

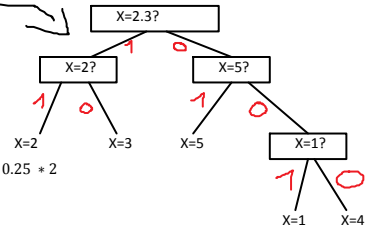
X	2	4
Pb(X)	0.2/0.35	0.15/0.35

   Ha = 1.46 bit/symbole
   Hb = 0.98 bit/symbole

   Vérification propriété de groupe: H(X) = H(p,q) +pHa + qHb?
   2.23 = 0.93 + 0.65*1.46 + 0.35*0.98
   OK.

Exercice 4

0) Non car elle ne partage pas Omega en 2 sous-ensembles A et B équiprobables pour ainsi maximiser l'entropie.
1) H(X) = -0.1*log2(0.1)-0.2log2(0.2)-0.3log2(0.3)-0.15log2(0.15)-0.25log2(0.25)
   = 0.23 bits/s
2) Propriété de groupe: H(X) = H(p,q) +pHa+qHb
3) Max H(p,q) = log2(p) = 1bit/symbole (p=0.5)
4) Meilleure question: Est-ce que le symbole X appartient à A={2,3}? Car p(X appartient à A) = p(X=2)+p(X=3) =0.2=0.3=0.5
5) Arbre:



6) Table de code:

X	1	2	3	4	5
mi(mot-code)	001	11	10	000	01
li (longueur)	3	2	2	3	2

7) Nombre moyen de questions:
   L = = $\sum_{i=1}^{s} pi * li = 0.1 * 3 + 0.2 * 2 + 0.3 * 2 + 0.15 * 3 + 0.25 * 2$
   = 2.25 questions

Théorie de l'information - 2

mardi 29 avril 2014
16:33

Exam année dernière:
Exo 1 - Codage optimal de Huffman, p33.

1°) H = 192 bits/symbole

2°)

Si	C	D	B	A	E
Pi	0.5	0.2	0.15	0.1	0.05

Si	C	D	B	AE
Pi	0.5	0.2	0.15	0.15

Si	C	BAE	D
Pi	0.5	0.3	0.2

Si	C	BAED
Pi	0.5	0.5

4°) L = 0.4+0.45+0.5+0.4+0.2 = 1.95 bits/symbole

5°) t = 8-1.95/

Théorie de l'information - 3

mardi 6 mai 2014
16:01

Ne pas compter les retours à la ligne.

$$\begin{bmatrix} B & B & B & B & N & B & B & B & B \\ B & B & B & N & N & N & B & B & B \\ B & B & G & G & G & G & G & B & B \\ B & G & G & G & N & G & G & G & B \\ G & G & G & N & N & N & G & G & G \\ B & G & G & G & N & G & G & G & B \\ B & B & G & G & G & G & G & B & B \\ B & B & B & N & N & N & B & B & B \\ B & B & B & B & N & B & B & B & B \end{bmatrix}$$

(4,B)(1,N)(7,B)
(3,N)(5,B)
(5,G)(3,B)
(3,G)(1,N)(3,G)(1,B)
(3,G)(3,N)(3,G)
(1,B)(3,G)(1,N)(3,G)(3,B)
(5,G)(5,B)
(3,N)(7,B)
(1,N)(4,B)

Total: 25

Exercice 1

8 bits par pixel en non compressé.

1)

B	G	N
40	28	13
40/81	28/81	13/81

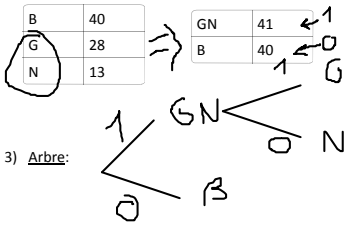
Ni
Pi

2) Entropie: H= $\frac{40}{81}\log 2\left(\frac{40}{81}\right) - (\ldots) - \frac{13}{81} log2\left(\frac{13}{81}\right) = 1{,}46 \frac{bits}{symbole}$ $comparés\ à\ 8 \frac{bits}{pixel}$.

Méthode 1: Huffman

B	40
G	28
N	13

GN	41
B	40



3) Arbre:

Table de code:

Si	B	G	N
Mi	0	11	10
Li	1	2	2
Pi	40/81	28/81	13/81

4) Taille du code:

(40*1)+(28*2)+(13*2)=122 bits
Taille image non compressée = 9*9 pixels * 8 bits = 648 bits

Taux de Compression:

$$\frac{NC - C}{NC} = \frac{648 - 122}{648} = 81\%$$

Méthode 2: RLE

1)

2) Longueur du code: 25 couples * 16 bits = 400 bits
3) Taux de compression: $\frac{648-400}{648} = 38\%$

Méthode 3: RLE + Huffman

Couple	3,G = a	1,N = b	3,N = c	4,B = d	7,B = e	5,B = f	5,G = g	3,B = h	1,B = i
Pi	6/25	4/25	3/25	2/25	2/25	2/25	2/25	2/25	2/25

A	6
B	4
C	3
D	2
E	2
F	2
G	2
H	2
I	2

A	6
B	4
HI	4
C	3
D	2
F	2
G	2

Toujours réunir les 2 derniers groupes.

A	6
B	4
HI	4
FG	4
DE	4
C	3

DEC	7
A	6
B	4
HI	4
FG	4

HIFGDEC	15
AB	10

Table de code:

Couple	3,G = a	1,N = b	3,N = c	4,B = d	7,B = e	5,B = f	5,G = g	3,B = h	1,B = i
Pi	6/25	4/25	3/25	2/25	2/25	2/25	2/25	2/25	2/25
Mi	01	00	101	1001	1000	1101	1100	1111	1110
Li	2	2	3	4	4	4	4	4	4

Taille du code:
10 couples de 2 bits + 3 couples de 3 bits + 12 couples de 4 bits = 77 bits

Taux de Compression:

$$\frac{NC - C}{NC} = \frac{648 - 77}{648} = 88\%$$

Exercice Annexe: 8 bits/couleur en non compressé

1)

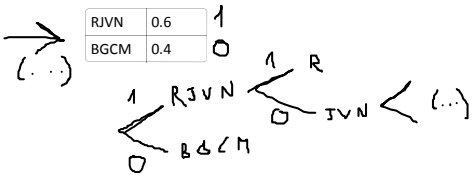
$$-0.1\log_2(0.1) - 0.2\log_2(0.2) - 0.05\log_2(0.05) - 0.1\log_2(0.1) - 0.3\log_2(0.3) - 0.1\log_2(0.1) - 0.05\log_2(0.05) - 0.1\log_2(0.1)$$

= 2.76 bits/symbole

2) Arbre Huffman:

R	0.3
B	0.2
N	0.1
J	0.1
V	0.1
G	0.1
C	0.05
M	0.05

RJVN	0.6
BGCM	0.4



3) Table

Si	R	B	N	G	J	V	C	M
Pi	0.3	0.2	0.1	0.1	0.1	0.1	0.05	0.05
Mi	11	01	100	001	1011	1010	0001	1111
Li	2	2	3	3	4	4	4	4

4) Taille du code: L= Somme Pi*Li = 2.8 bits/symbole

5) Taux compression: $\frac{8-2.8}{8} = 65\%$

Théorie de l'information - 4

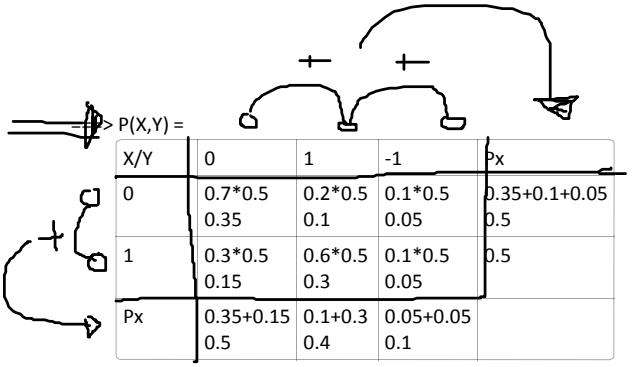
mardi 13 mai 2014
15:20

Exercice 1

P(Y|X) =

X/Y	Y1 = 0	Y2 = 1	Y3 = -1
X1 = 0	0.7	0.2	0.1
X2 = 1	0.3	0.6	0.1

Distribution conditionelle



P(X,Y) =

X/Y	0	1	-1	Px
0	0.7*0.5 0.35	0.2*0.5 0.1	0.1*0.5 0.05	0.35+0.1+0.05 0.5
1	0.3*0.5 0.15	0.6*0.5 0.3	0.1*0.5 0.05	0.5
Px	0.35+0.15 0.5	0.1+0.3 0.4	0.05+0.05 0.1	

Distribution Marginale

P(X|Y)=

X/Y	0	1	-1
0	0.35/0.5 0.7	0.1/0.4 0.75	0.05/0.1 0.5
1	0.15/0.5 0.3	0.3/0.4 0.75	0.05/0.1 0.5

Distribution conditionelle

4) H(X) = log2(2) = 1 bit/symbole
   H(Y) = -0.5*log2(0.5) -0.4*log2(0.4)-0.1*log2(0.1) = 1.36 bits/symbole

   H(X,Y) = 2.23 bits/Symbole
   H(X|Y) = 0.86 bits/symb
   H(Y|X) = 1.23 bits/sy

5) I(X,Y) = I(Y,X) = H(X) - H(X|Y)
   = H(Y) - H(Y|X)
   = 0.13 bit/symbole

Exercice 2

Non symétrique.

P(Y|X) =

X/Y	Y1 = 0	Y2 = -1	Y3 = 1
X1 = 0	0.8	0.2	0
X2 = 1	0	0.2	0.8

Ananu TP2

mardi 27 mai 2014
10:58

```
//Obtention de l'image originale
a = read('im34.txt',-1,640);
ap = uint8(255*a);
figure();
ShowImage(ap,'Image originale');

amono = matrix(a,1,-1);
Afft=fft(amono);
Amod = abs(amono);

Afft=fft(Amod);
Asub=Afft(:,2:160000);

//On prend un % de fréquences
Asub2=Asub(:,1:60000);
Aifft=ifft(Asub2);
plot(Aifft);
Aifft=abs(Aifft);
```

B10