Grenoble INP
ENSIMAG

Annales d'examens 1ère année

Année universitaire 2009/2010

NOVEMBRE 2009

ANALYSE POUR L' INGENIEUR 1 – 2 PAGES

ARCHITECTURE 1 – 4 PAGES

PROBABILITES APPLIQUEES 1 – 2 PAGES

THEORIE DES LANGUANGES 1 – 3 PAGES

Grenoble INP-Ensimag **Analyse pour l'ingénieur**
1ère année

Examen à mi-parcours
Lundi 16 novembre 2009 - 2h

Documents manuscrits et polycopié de cours autorisés.
Le nom du groupe doit être noté en évidence sur la première page
NB :
(i) Toutes les intégrales considérées dans cet énoncé le sont au sens de Lebesgue.
(ii) Un soin particulier devra être apporté à la rédaction et à la justification des calculs.

Exercice 1
Soit $(X, \mathcal{T}, \mu)$ un espace mesuré et f une fonction mesurable sur X ($X \subset R^d$). Pour $n \in \mathbb{Z}$, on pose :

$$A_n = \{x \in X \ ; \ 2^{n-1} \leq |f(x)| < 2^n\}$$

Montrer que f est intégrable sur X si et seulement si f est finie presque partout sur X et :

$$\sum_{n \in \mathbb{Z}} 2^n \mu(A_n) < +\infty$$

Exercice 2 Soit f une fonction Lebesgue-intégrable sur $\mathbb{R}$. Calculer la limite :

$$\lim_{n \to +\infty} \int_0^{+\infty} e^{-n \sin^2 x} f(x) \, dx$$

Exercice 3 On pose

$$F(x) = \int_0^{+\infty} \frac{e^{-xt^2}}{1+t^2} dt$$

1. Montrer que la fonction F est définie, continue sur $\mathbb{R}_+$, et dérivable sur $\mathbb{R}_+^*$.
2. Calculer $\lim_{x \to +\infty} F(x)$.
3. Montrer que F est solution d'une équation différentielle linéaire du premier ordre.
4. Intégrer cette équation différentielle et en déduire la valeur de $\int_0^{+\infty} e^{-t^2} dt$.

Exercice 4 Soit

$$f(x,y) = ye^{-(1+x^2)y^2}$$

Montrer que f est intégrable sur $\mathbb{R}_+^2$ et que : $\int_0^{+\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$

Exercice 5
Soit $(X, \mathcal{T}, \mu)$ un espace mesuré (X un ouvert de $\mathbb{R}$), et soit $f_n : X \to \overline{\mathbb{R}}_+$ une suite de fonctions mesurables positves. On définit $s_k = \sup_{n \geq k} f_n$, puis la fonction $\limsup f_n$ comme $x \to \lim_k s_k(x)$. De la même manière, on définit $g_k = \inf_{n \geq k} f_n$ et la fonction $\liminf f_n$ comme $x \to \lim_k g_k(x)$.

1. Montrez que si f_n est mesurable alors s_k et g_k sont mesurables.

2. Ecrire $\limsup f_n$ et $\liminf f_n$ comme combinaison de sup et d'inf, en déduire qu'elles sont mesurables.

3. Montrer alors que :

$$\int_X \liminf f_n \, d\mu = \lim_{k \to +\infty} \int_X g_k \, d\mu$$

4. Montrer ensuite que :

$$\int_X g_k \; d\mu \leq \inf_{n \geq k} \int_X f_n \; d\mu$$

5. Puis finalement que :

$$\int_X \liminf f_n \; d\mu \leq \liminf \int_X f_n \; d\mu$$

avec $\liminf \int_X f_n d\mu = \lim\limits_{k \to +\infty} \inf\limits_{n \geq k} \int_X f_n d\mu$

Architecture 1 :
Circuits Numériques et Éléments d'Architecture
Partiel de mi-semestre

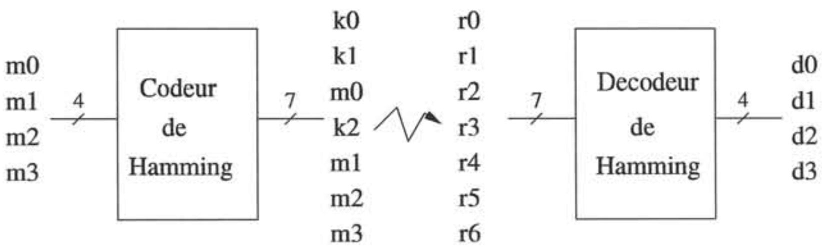
ENSIMAG 1A

Année scolaire 2009–2010

- Durée : 2h. Tous documents et calculatrices autorisés.
- Le barème est donné à titre indicatif.
- Les exercices sont indépendants et peuvent être traités dans le désordre.

Ex. 1 : Circuits pour un codeur/décodeur de Hamming (6pts)

Le code de Hamming est utilisé en transmission de données pour détecter et corriger des erreurs.



Les données à transmettre sont groupées par blocs de 4 bits appelés messages m de 4 bits $(m_0 \cdots m_3)$. Pour permettre la correction, on ajoute à chaque bloc, 3 bits de redondance k $(k_0 k_1 k_2)$ obtenus par :

$$\begin{cases} k_0 = & (m_0 + m_1 + m_2) \mod 2 \\ k_1 = & (m_0 + m_2 + m_3) \mod 2 \\ k_2 = & (m_1 + m_2 + m_3) \mod 2 \end{cases}$$

Le bloc issu du codeur est un vecteur de 7 bits agencé comme suit : $(k_0 k_1 m_0 k_2 m_1 m_2 m_3)$. Ce vecteur pouvant subir des erreurs de transmission. On appellera $R = (r_0 \cdots r_6)$ le vecteur reçu par le décodeur et $D = (d_0 \cdots d_3)$ le message décodé.

Question 1 Rappeler la table de vérité de l'addition à deux entrées modulo 2 et la porte élémentaire correspondante.

Question 2 Construire un circuit implantant le codeur de Hamming présenté avec des portes à deux entrées de la question précédente.

Question 3 Pour réaliser un décodeur de Hamming, il est nécessaire de calculer le syndrome S du vecteur reçu par le décodeur définit par les équations suivantes :

$$\left\{\begin{array}{ll} s_0 = & (r_0 + r_2 + r_4 + r_6) \mod 2 \\ s_1 = & (r_1 + r_2 + r_5 + r_6) \mod 2 \\ s_2 = & (r_3 + r_4 + r_5 + r_6) \mod 2 \end{array}\right.$$

Construire un circuit implantant le syndrome toujours avec les mêmes portes à 2 entrées.

Question 4 Par construction, le syndrome renvoie la position de l'erreur s'il y en a une et 0 sinon[1]. En supposant que le vecteur reçu par le décodeur ait au plus une erreur, on a :
si $S = 0$, R est correct et $D = (r_2 r_4 r_5 r_6)$
si $S = i + 1$, $R' = (r_0 \cdots \bar{r}_i \cdots r_6)$ est correct et $D = (r'_2 r'_4 r'_5 r'_6)$
En utilisant le bloc syndrome de la question précédente, construire un circuit implantant le décodeur.

Ex. 2 : Gestion de la température par hysteresis (8pts)

On cherche à réaliser un système permettant d'assurer le maintien à une température quasi-constante d'un poulailler industriel, afin d'assurer aux volatiles une durée de vie compatible avec les normes Européennes. Ce système repose sur la disponibilité d'un climatiseur pouvant produire du chaud, du froid, ou ne rien faire, en fonction de la température courante de l'entrepôt (qui dépend de la température externe et du degré d'agitation des gallinacées).

Le système est basé sur le principe de l'hysteresis, c'est-à-dire que le comportement lors de l'accroissement de la température est différent du comportement lorsque celle-ci baisse. Trois températures de référence sont nécessaires au fonctionnement du système : $T_{min} < T_{nom} < T_{max}$. La production du chaud ou du froid, indiquée respectivement par un signal C à 1 ou F à 1, se passe comme suit :

- lorsque la température T devient inférieure à T_{min}, il y a production de chaud : $C \leftarrow 1$;
- la production de chaud s'arrête, *i.e.* $C \leftarrow 0$, lorsque la température devient supérieure ou égale à la valeur nominale T_{nom} ;
- lorsque la température devient supérieure ou égale à T_{max}, il y a production de froid $F \leftarrow 1$;
- la production de froid cesse lorsque la température devient inférieure à la valeur nominale T_{nom}.

Question 1 Peut-on représenter les valeurs de C et F en fonction de la température sous la forme de tables de vérité ? Justifiez.

Le système exploite un capteur de température qui fournit une information encodée sur 2 bits comme précisé ici :

b_1	b_0	cas
0	0	$T < T_{min}$
0	1	$T_{min} \leq T < T_{nom}$
1	1	$T_{nom} \leq T < T_{max}$
1	0	$T_{max} \leq T$

On cherche à formaliser le comportement du système comme un automate de Moore, en faisant l'hypothèse que la période de l'horloge de l'automate est très inférieure au temps qu'il faut pour que la température T du poulailler passe de T_{min} à T_{max} ou inversement, à cause de l'inertie thermique.

Question 2 Précisez quelles sont les entrées et les sorties de l'automate.

[1] Le code permet de corriger au plus une erreur.

Question 3 Donnez le nombre d'états de l'automate et représentez le graphe de l'automate en y incluant les conditions de transitions sous forme *d'intervalle de températures* sur les arcs et les valeurs des sorties dans les états. Transformez ensuite les intervalles de températures en condition booléennes.

Question 4 Proposez un codage logarithmique des états de façon à ce que les expressions des sorties C et F soient simples et donnez la table de transition qui précise les sorties et l'état futur en fonction des entrées et de l'état courant. On note Q_i les sorties du registre représentant l'état courant et D_i les entrées du registre, représentant l'état futur.

Question 5 Donnez les expressions simplifiées de C, F et des D_i, et faites le schéma en portes correspondant.

Ex. 3 : Construction algorithmique du décaleur à barillet (6pts)

On s'intéresse ici à la généralisation du décaleur à barillet vu en TD, opérateur qui réalise le décalage d'un nombre x codé sur n bits par un nombre $p < n$ vers la gauche ou vers la droite.

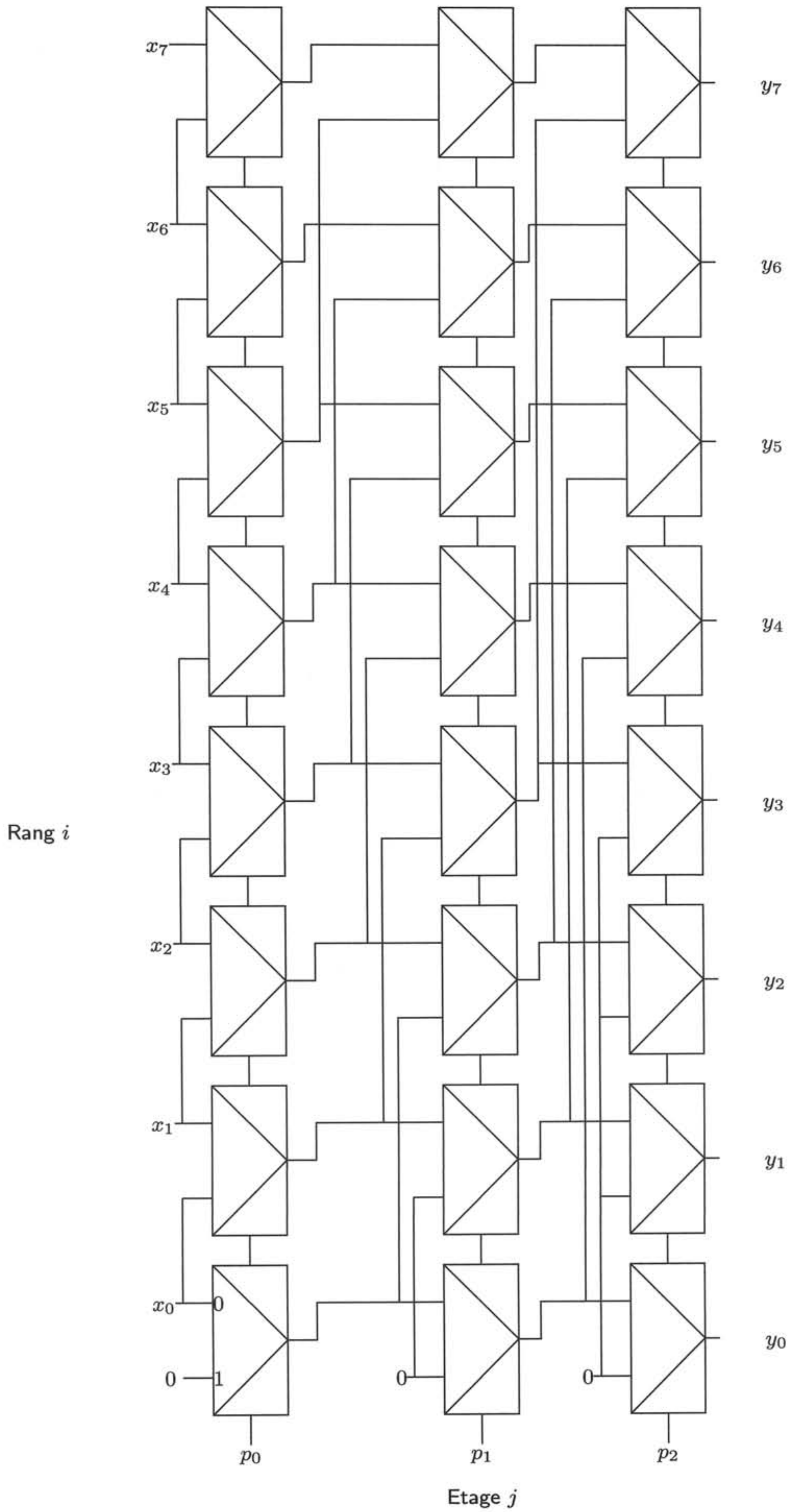
On prend n égal à une puissance entière de 2 ($n = 2^k$) et donc p est codé sur k bits. Le décaleur à barillet est alors composé de k étages de n cellules. Comme vu en TD, une cellule est un MUX à 2 entrées si on fait un décalage à gauche ou à droite, et un ensemble de MUX quand on offre des décalages plus compliqués. L'étage $j, 0 \leq j \leq k-1$ effectue le décalage de 2^j positions si l'entrée $p_j = 1$ et de 0 position sinon. La $i^{\text{ème}}$ cellule génère le bit de poids i du résultat du décalage. Il s'agit de trouver la formule analytique précisant la cellule (i, j) (cellule de rang i de l'étage j du décaleur à barillet) dont la sortie est utilisée pour attaquer les entrées des cellules de l'étage suivant en fonction du type de décalage que l'on cherche à faire.

Question 1 On s'intéresse au décalage vers la gauche. Pour mémoire, le schéma d'un décaleur à gauche pour $n = 8$ est fourni en annexe, tous les MUX étant placées avec l'entrée 1 en bas et 0 en haut comme indiqué sur la cellule de coordonnées $(0, 0)$. Chaque cellule est composée d'un MUX1b $(2 \rightarrow 1)$. En fonction de n, i et j, spécifiez les coordonnées de la cellule dont la sortie est connectée à l'entrée 0 ainsi que celles de la cellule dont la sortie est connectée à l'entrée 1 de la cellule (i, j).

Question 2 On s'intéresse au décalage logique vers la droite. Chaque cellule est composée d'un MUX1b $(2 \rightarrow 1)$. En fonction de n, i et j, spécifiez les coordonnées de la cellule dont la sortie est connectée à l'entrée 0 ainsi que celles de la cellule dont la sortie est connectée à l'entrée 1 de la cellule (i, j).

Question 3 A partir des résultats des deux questions précédentes, on va traiter les décalages logiques à gauche et à droite en fonction d'une entrée d telle que $d = 1$ pour le décalage à droite. Donnez le schéma de la cellule avec les coordonnées des cellules dont la sortie attaque ses entrées.

Question 4 On veut également traiter le décalage arithmétique à droite (entrée a = 1 si on veut effectuer un décalage arithmétique quand d = 1). Donnez le schéma de la cellule avec les coordonnées des cellules dont la sortie attaque ses entrées.



x_7
x_6
x_5
x_4
x_3
x_2
x_1
x_0
0
1
0
0
0
y_7
y_6
y_5
y_4
y_3
y_2
y_1
y_0
p_0
p_1
p_2
Rang i
Etage j

Ensimag 1^re année **2009-2010**

Probabilités Appliquées
Novembre 2009

Durée 2h00 - Documents et calculatrices non-autorisés. Veuillez inscrire votre numéro de groupe sur la copie.

Exercice I (10pts) - Soit n un entier strictement positif et `p[]` un tableau de probabilités de dimension n, c'est à dire n nombres positifs dont la somme est égale à 1. On considère l'algorithme suivant :

```
Répéter
I <- dé(n); U <- Random
Jusqu'à (U < p[I])
X <- I
```

où `dé(n)` est un générateur de loi uniforme sur $\{1, \ldots, n\}$ et `Random` est un générateur de loi uniforme sur $(0, 1)$.

1-3pts) Montrer que

$$\mathrm{P}(\mathtt{U} < \mathtt{p[I]}) = \frac{1}{n}.$$

En déduire la probabilité de l'événement $(X = i)$ pour tout $i = 1, \ldots, n$.

2-2pts) Quelle est la loi du nombre d'appels de `dé(n)` dans cet algorithme ? Quelle est l'espérance de cette loi ?

3-3pts) On considère une constante c telle que $c \geq \max$ `p[i]` et on modifie l'algorithme de la manière suivante

```
Répéter
I <- dé(n) ; U <- Random
Jusqu'à (c * U < p[I])
X <- I
```

Calculer la probabilité que la condition `(c * U < p[I])` se réalise. En déduire que l'algorithme simule la même loi que précédemment.

4-2pts) Comment doit on choisir c pour minimiser la durée moyenne d'exécution de cet algorithme ? (Justifier)

Exercice II (3pts) On tire U un nombre au hasard uniformément dans l'intervalle $(0,1)$, puis on lance une pièce équilibrée dont le résultat est indépendant de U. Si c'est pile, on définit $X = U$ sinon, $X = \sqrt{U}$. Soit $t \in [0,1]$, calculer la probabilité de l'événement $(X \leq t)$.

Exercice III (7pts) - On joue à pile (P) ou face (F) avec une pièce équilibrée n fois de suite. On définit X_n comme le nombre de fois que l'on obtient le motif FF (compté avec les recouvrements). Par exemple, dans la réalisation suivante,

FFPFPFFFPFFPF

nous avons $n = 13$ et $X_n = 4$ (le motif FFF contribue deux fois au nombre total).

1-2pts) Pour $i = 1, \ldots, n$, on note $\mathrm{FF}_i =$ (le motif FF apparaît à l'issue du lancer i). Montrer que les événements (FF_i) ne sont pas indépendants.

2-1pt) Décrire X_n comme une variable étagée.

3-1pt) Calculer $\mathrm{E}[X_n]$

4-1pt) Justifier que la loi de X_n n'est pas la loi binomiale.

5-2pts) En raisonnant par récurrence, démontrer que

$$\mathrm{P}(X_n = 0) = f_n/2^n, \quad n \geq 2$$

où f_n est la suite de Fibonacci, $f_n = f_{n-1} + f_{n-2}$, $n \geq 4$, initialisée par $f_2 = 3$ et $f_3 = 5$.

Grenoble INP - Ensimag 1ère année novembre 2009

Théorie des langages 1

Durée : 2h.
Documents : tous documents autorisés.

Ce sujet est constitué de 3 exercices notés respectivement sur 10 points, 5 points et 5 points. Le barème est donné à titre indicatif.

Exercice 1 - Langage et grammaire (10 points)

Soit la grammaire $G = (V_T, V_N, S, R)$ avec $V_T = \{0, 1\}$, $V_N = \{S, A, B\}$ et R l'ensemble des règles :

$$S \to 0 \mid S0 \mid A1 \qquad A \to 1 \mid S1 \mid B0 \qquad B \to A0 \mid B1$$

▷ **Question 1** (1 point)
Quel est le type de la grammaire G ?

▷ **Question 2** (2 points)
On rappelle que, pour tout non-terminal $X \in V_N$, le langage $L(X)$ est défini par $L(X) = \{w \mid w \in V_T^* \wedge X \Longrightarrow^* w\}$. On a $L(G) = L(S)$ avec S l'axiome. On pose $c_1 = 10$, $c_2 = 11$ et $c_3 = 100$. Montrer $c_1 \in L(B)$, $c_2 \in L(S)$ et $c_3 \in L(A)$.

▷ **Question 3** (2 points)
On définit la fonction $Val : V_T^+ \to \mathbb{N}$ par :

$$\begin{array}{ll} Val(0) = 0 & Val(1) = 1 \\ Val(w0) = Val(w) \times 2 & Val(w1) = Val(w) \times 2 + 1 \end{array}$$

Et on pose :

$$\begin{array}{l} E_S = \{w \mid w \in V_T^+ \ \wedge \ \exists n \, . \, Val(w) = 3 \times n\} \\ E_A = \{w \mid w \in V_T^+ \ \wedge \ \exists n \, . \, Val(w) = 3 \times n + 1\} \\ E_B = \{w \mid w \in V_T^+ \ \wedge \ \exists n \, . \, Val(w) = 3 \times n + 2\} \end{array}$$

À quels ensembles appartiennent les chaînes c_1, c_2 et c_3 ?

▷ **Question 4** (2 points)
On veut prouver, par récurrence sur la longueur des dérivations que :

$$L(S) \subseteq E_S \ \wedge \ L(A) \subseteq E_A \ \wedge \ L(B) \subseteq E_B$$

Dans la suite w désigne une chaîne de V_T^*.

1. Montrer que $(S \Longrightarrow^1 w) \Rightarrow w \in E_S$
2. Montrer que $(A \Longrightarrow^1 w) \Rightarrow w \in E_A$
3. Montrer que $(S \Longrightarrow^{n+1} w) \Rightarrow w \in E_S$, en utilisant l'hypothèse de récurrence :

$$\begin{array}{ll} & [(S \Longrightarrow^n w) \Rightarrow w \in E_S] \\ \wedge & [(A \Longrightarrow^n w) \Rightarrow w \in E_A] \\ \wedge & [(B \Longrightarrow^n w) \Rightarrow w \in E_B] \end{array}$$

On ne prouvera pas les cas $A \Longrightarrow^{n+1} w$ et $B \Longrightarrow^{n+1} w$.

▷ **Question 5** (3 points)
On veut maintenant prouver, par récurrence sur la longueur de la chaîne w (notée $|w|$) que :

$$E_S \subseteq L(S) \ \wedge \ E_A \subseteq L(A) \ \wedge \ E_B \subseteq L(B)$$

1. Montrer que $(w \in E_S \ \wedge \ |w| = 1) \Rightarrow w \in L(S)$
2. Montrer que $(w \in E_A \ \wedge \ |w| = 1) \Rightarrow w \in L(A)$
3. Montrer que $(w \in E_S \wedge |w| = n+1) \Rightarrow w \in L(S)$. On utilisera l'hypothèse d'induction suivante :

$$\begin{array}{ll} & [(|w| = n \ \wedge \ w \in E_S) \Rightarrow w \in L(S)] \\ \wedge & [(|w| = n \ \wedge \ w \in E_A) \Rightarrow w \in L(A)] \\ \wedge & [(|w| = n \ \wedge \ w \in E_B) \Rightarrow w \in L(B)] \end{array}$$

On ne prouvera pas les cas $(w \in E_A \wedge |w| = n+1)$ et $(w \in E_B \wedge |w| = n+1)$.

Exercice 2 - Modélisation de langages (5 points)

On s'intéresse à un sous-ensemble des instructions Ada décrit par la grammaire G suivante :

$$\begin{array}{ll} S \rightarrow S\ I\ ; & S \rightarrow I\ ; \\ I \rightarrow \mathbf{i} & I \rightarrow \mathbf{if\ c\ then}\ S\ \mathbf{else}\ S\ \mathbf{end\ if} \end{array}$$

avec $V_N = \{S, I\}$, S étant l'axiome, et $V_T = \{;, \mathbf{i}, \mathbf{c}, \mathbf{if}, \mathbf{then}, \mathbf{else}, \mathbf{end}\}$. Le non-terminal S dénote une suite d'instructions et I une instruction. On utilise les terminaux **i** et **c** pour ne pas avoir à décrire les instructions autres que la conditionnelle, ni les conditions.

▷ **Question 1** (2 points)
Donner l'arbre de dérivation de la chaîne :

i ; if c then i ; else i ; end if ;

▷ **Question 2** (3 points)
Modifier la grammaire G afin de décrire les instructions conditionnelles générales, c'est-à-dire contenant une suite éventuellement vide de elsif et où la partie else peut être absente :

if c then i ; elsif c then i ; elsif ... end if

Pour cela on étend le vocabulaire terminal par l'élément **elsif**. On pourra introduire de nouveaux non-terminaux, à condition de les définir.

Exercice 3 - Calculs sur les grammaires (5 points)

Soit $G = (V_T, V_N, S, R)$ une grammaire hors-contexte. On donne les définitions suivantes :

- un symbole A de V_N est dit *productif* si et seulement si il existe une dérivation $A \Longrightarrow^* w$ avec $w \in V_T^*$.
- un symbole A de V_N est dit *accessible* si et seulement si il existe une dérivation $S \Longrightarrow^* w_1 A w_2$ avec $w_1, w_2 \in (V_T \cup V_N)^*$.

▷ **Question 1** (2 points)
Soit la grammaire $G = (\{a, b, c, d\}, \{S, A, B, C, D\}, S, R)$ avec R défini par :

$$\begin{array}{lllll} S \to AB & S \to \epsilon & A \to aA & A \to D & \\ B \to bB & B \to aS & C \to c & C \to cC & D \to dA \end{array}$$

Donner l'ensemble Pr des symboles productifs de G ainsi que l'ensemble Ac des symboles accessibles.

▷ **Question 2** (3 points)
Soit $G = (V_T, V_N, S, R)$ une grammaire hors-contexte quelconque.

1. Donner une méthode permettant de calculer, à partir des règles de la grammaire, l'ensemble Pr des non-terminaux productifs.
2. Même question pour l'ensemble Ac des non-terminaux accessibles.
3. En utilisant les résultats précédents, comment peut-on décider simplement si $L(G) \neq \emptyset$?

JANVIER 2010

ALGORITHMIQUE 1 – 8 PGAES

ANALYSE POUR L' INGENIEUR 1 – 2 PAGES

ARCHITECTURE 1 – 6 PAGES

ECONOMIE GENERALE - 4 PGAES

INTRODUCTION AUX RESEAUX – 5 PGAES

PROBABILITES APPLIQUEES 1 – 2 PAGES

THEORIE DES LANGUANGES 1 – 3 PAGES

Algorithmique 1

Durée : 3h
Machines électroniques interdites **Tous documents papiers autorisés**

Les trois parties du sujet sont indépendantes.

Veuillez respecter les notations introduites dans l'énoncé. Il est inutile de paraphraser l'énoncé dans vos réponses, mais des explications avec dessins sur votre code sont les bienvenues. Sauf dans les questions où le style récursif est explicitement demandé, vous devez répondre aux questions du sujet en écrivant des procédures et fonctions itératives.

I - Tables de hachage (8 points)

Dans cet exercice, on souhaite implémenter une table de hachage dans un paquetage similaire à celui du TPL2. On suppose que le type des données à stocker dans la table s'appelle `Donnee` et qu'il est défini par un paquetage `Donnees` fourni :

```
package Donnees is

   type Donnee is ... ;

   function Hache(S: Donnee) return Natural is
   -- fonction de hachage

end Donnees ;
```

Ce paquetage `Donnees` définit aussi la fonction de hachage à utiliser dans la table de hachage. L'interface de la table de hachage est la suivante :

```
with Donnees ; use Donnees;
package TableHach is

   RedimC: constant := 4 ; -- constante de redimensionnement de la table

   function NbDonnees return Natural ;
   -- retourne le nb de données dans la table.

   function EstPresente(S: Donnee) return Boolean ;
   -- Indique si S est présente.

   procedure Inserer(S: Donnee) ;
   -- requiert EstPresente(S)=False.
   -- garantit S a été ajoutée dans la table.

end TableHach ;
```

Par rapport au TPL2, on veut ici coder la table de hachage comme un tableau de listes **circulaires** simplement chainées avec une sentinelle de tête qui sert donc aussi de sentinelle de queue. Concrètement, on suppose fourni le squelette de paquetage à la figure 1 page 4. Cette structure de données permet d'optimiser un peu le code de `EstPresente` en évitant une comparaison de pointeur à chaque étape de la boucle (voir le code fourni).

Dans les questions suivantes, on vous demande de compléter l'implémentation du paquetage page 4, de manière à ce que **le code fourni** (en particulier pour `NbDonnees` et `EstPresente`) **fonctionne correctement tel quel** (vous ne devez donc pas modifier ce code).

Comme dans le TPL2, on implémente ici une table de hachage avec redimensionnement dynamique. Comme dans le TPL2, la variable `NbEnt` représente le nombre de données dans la table et `RedimC` est la constante de redimensionnement : initialement, `Dico'Length` vaut `RedimC`, puis ensuite une certaine puissance de `RedimC`. Le redimensionnement doit permettre ici de préserver **l'invariant** suivant :

- Si `Dico'Length = RedimC`, alors `0 <= NbEnt <= Dico'Length`.
- Si `Dico'Length > RedimC` alors `Dico'Length <= NbEnt <= RedimC*Dico'Length`.

▷ Question 1 (2 points) :

Écrire le code de `InitDico` qui initialise les variables globales. En particulier, elle initialise le tableau `Dico` comme un tableau à `RedimC` sentinelles telles que chacune est son propre suivant.

▷ Question 2 (1 point) :

Écrire le code de la procédure auxiliaire `InsererEntree` qui est destinée à être utilisée dans `ChangeTaille` et `Inserer`. Cette procédure insère la cellule `E` à sa place dans `Dico`. Elle requiert que `E` soit une cellule correctement allouée et initialisée qui ne soit pas déjà une cellule de `Dico`. Elle ne modifie ni `NbEnt`, ni `Limite`, ni la valeur des sentinelles de `Dico` : elle modifie uniquement des champs `Suiv` de cellules `EntreeCell`.

▷ Question 3 (4 points) :

Écrire la procédure auxiliaire `ChangeTaille` qui augmente la taille de `Dico`. Cette procédure requiert `NbEnt = Limite`. Elle alloue un nouveau tableau de taille `NbEnt` et modife `Limite` en conséquence. Elle réutilise les cellules de l'ancien `Dico` (y compris les sentinelles) et en crée de nouvelles (les nouvelles sentinelles) : elle ne doit pas libérer de cellule de type `EntreeCell` mais ne doit pas laisser de cellules inutilisées en mémoire. Elle libère l'ancien tableau `Dico` (via la procédure `Liberer` définie dans le squelette du paquetage).

▷ Question 4 (1 point) :

Écrire le code de `Inserer` en utilisant les deux procédures auxiliaires précédentes.

```
with Ada.Unchecked_Deallocation ;
package body TableHach is

   -- Déclarations de type
   type EntreeCell ;
   type Entree is access EntreeCell ;

   type EntreeCell is record
      Val: Donnee ;
      H: Integer ; -- doit être égale a Hache(Val)
      Suiv: Entree ;
   end record ;

   type TableEntree is array(Natural range <>) of Entree ;
   type TableEntreeDyn is access TableEntree ;
   procedure Liberer is new Ada.Unchecked_Deallocation(TableEntree,
                                                        TableEntreeDyn) ;

   -- Variables Globales
   Dico: TableEntreeDyn ;       -- table de hachage
   NbEnt: Natural := 0 ;        -- nbre d'entrées dans la table
   Limite: Positive ;           -- doit correspondre a RedimC*Dico'Length

   function NbDonnees return Natural is
   begin
      return NbEnt ;
   end ;

   function EstPresente(S:Donnee) return Boolean is
      Senti: Entree := Dico(Hache(S) mod Dico'Length) ;
      Cour: Entree := Senti.Suiv ;
   begin
      Senti.Val := S ; -- mise-à-jour de la sentinelle
      while Cour.Val /= S  loop
         Cour := Cour.Suiv ;
      end loop ;
      return Cour /= Senti ;
   end ;

   procedure InitDico is                  -- A COMPLETER question 1
   procedure InsererEntree(E: Entree) is  -- A COMPLETER question 2
   procedure ChangeTaille is              -- A COMPLETER question 3
   procedure Inserer(S: Donnee) is        -- A COMPLETER question 4

begin
   InitDico ;
end TableHach ;
```

FIG. 1 – Squelette de l'implémentation du paquetage TableHach

II - Longueur d'une liste chaînée (4 points)

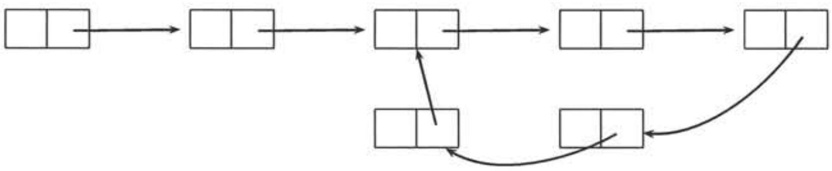
Dans cet exercice, on considère des listes chaînées définies par les types suivants :

```
type Cellule ;
type Liste is access Cellule ;
type Cellule is record
  D: Donnee ;  -- type des données dans les listes
  Suiv: Liste ;
end record ;
```

Dans les deux questions qui suivent, on étudie deux façons d'implémenter la fonction `Longueur` suivante. Cette fonction requiert que tous les pointeurs accessibles depuis `L` soient **null** ou alloués. Si le chaînage des champs `Suiv` à partir de `L` se termine par **null**, alors la fonction retourne le nombre de cellules de la liste. Sinon, cela signifie qu'il y a un cycle dans `L`, et dans ce cas, la fonction termine et lève l'exception `Cycle`.

```
Cycle: exception ;
function Longueur(L: Liste) return Natural ;
```

En effet, une liste dont le chaînage n'est pas terminé par **null** a nécessairement la forme suivante : un préfixe (ci-dessous les deux première cellules) suivi d'une liste qui boucle sur elle-même.



▷ **Question 5 (2 points) :**

Dans un premier temps, on suppose que le type `Donnee` possède un bit nommé `Couleur` qui va permettre à `Longueur` de reconnaître des cellules qu'elle a déjà rencontrées :

```
type Donnee is record
  Val: ... ;  -- type quelconque
  Couleur: Boolean ;
end record ;
```

La fonction `Longueur` requiert que les bits `Couleur` de toutes les cellules du chaînage aient initialement la même valeur. En sortie, elle garantit que les bits de toutes les cellules du chaînage ont été inversés (et qu'ils sont donc toujours égaux entre eux). Ainsi, si après un appel à `Longueur` on souhaite remettre la liste `L` dans son état initial, il suffit de faire un deuxième appel à `Longueur`. **Implémenter** la fonction `Longueur` en respectant cette spécification additionnelle sur le bit de couleur.

▷ **Question 6 (2 points) :**

Maintenant, on ne fait plus d'hypothèse sur le type `Donnee` : on demande de coder `Longueur` sans utiliser de bit de couleur. L'idée est ici d'utiliser deux pointeurs `Lievre` et `Tortue` de type `Liste` qui parcourent le chaînage de `L` à des vitesses différentes : pendant que `Tortue` avance d'un champ `Suiv`, `Lievre` avance de deux. Ainsi, soit `Lievre` arrive en premier sur le pointeur **null**, soit `Lievre` finit par rattraper `Tortue` parce que la liste a un cycle.

III - Codes de Gray (10 points)

Étant donné M une constante entière strictement positive, le but de cet exercice est de construire une procédure affichant l'énumération des 2^M entiers binaires de taille M sous formes d'un codage de Gray. Ici, on considère les entiers binaires comme des mots dans l'alphabet $\{0, 1\}$. Un codage de Gray sur M bits est simplement défini comme une suite de 2^M mots (e.g. d'entiers binaires) qui contient uniquement des mots distincts de taille M, et tels que 2 mots successifs dans la suite ne diffèrent que d'un bit. Par définition, un codage de Gray sur M bits contient donc tous les mots de taille M. Voir les exemples de la figure 2.

000	010
001	011
011	001
010	000
110	100
111	101
101	111
100	110

FIG. 2 – Exemples de 2 codes de Gray sur 3 bits sur chacune des colonnes ci-dessus.

Comme M est strictement positif, les mots de taille M sont non vides, et une suite de Gray sur M bits est une suite non vide de mots non vides.

Pour la suite de l'exercice, on introduit les notations suivantes sur les mots non vides et les suites de mots non vides :

- si b est un bit, on note $\neg\, b$ son complémentaire. Ainsi, $\neg 1$ vaut 0 et $\neg 0$ vaut 1.
- un mot non vide est soit un bit (0 ou 1), soit de la forme $b.m$ où b est un bit et m un mot non vide. Ainsi, 1.(0.0) correspond au mot 100.
- Si m est un mot non vide, on note $|m|$ la longueur de m. La fonction $|m|$ est définie par récurrence sur la structure des mots non vides :

  si m est un bit
  alors retourner 1
  sinon on a $m = b.m'$ avec m' un mot non vide,
  retourner $|m'| + 1$
- si s_1 et s_2 sont deux suites de mots, on note $s_1; s_2$ leur concaténation. Par exemple, le résultat de $(00; (01; 10)); (10; 11)$ vaut $00; 01; 10; 10; 11$. La concaténation est associative.
- si s est une suite non vide de mots, alors on note prem(s) le premier mot de s et der(s) le dernier mot de s.
- si s est une suite de mot et b un bit, l'opération $b \triangleright s$ retourne la suite des mots de s préfixés du bit b. Par exemple, le résultat de $1 \triangleright (00; 01; 10)$ vaut $100; 101; 110$.

Avant d'écrire une procédure pour afficher des codages de Gray, on s'entraîne sur un système d'énumération plus simple. Soit F la fonction mathématique paramétrée par un entier strictement positif n définie par récurrence sur n de la façon suivante :

> F(n) est la suite définie par:
> si $n = 1$
> alors retourner 0 ; 1
> sinon,
> soit $s = F(n-1)$,
> retourner 1 $\triangleright$ s ; 0 $\triangleright$ s

La suite $F(n)$ énumère les 2^n entiers binaires sur n bits, mais pas sous forme d'un codage de Gray dès que $n \geq 2$. Ainsi, dans $F(2) = 10; 11; 00; 01$ le deuxième mot et le troisième mot diffèrent sur 2 bits.

$\triangleright$ **Question 7 (0,5 points) :**

Donner la suite correspondant à $F(3)$.

On souhaite écrire une procédure Ada récursive qui calcule F. On représente les mots binaires par des tableaux de type `TabBits` définis ci-dessous. Pour éviter d'avoir à manipuler explicitement des suites de mots, on affiche les mots en colonne au fur et à mesure des calculs, un par ligne en utilisant la procédure `Affiche` définie ci-dessous :

```
type TabBits is array(Positive range <>) of Boolean ;

procedure Affiche(T: TabBits) is
begin
   for I in T'Range loop
      if T(I) then
         Put("1") ;
      else
         Put("0") ;
      end if;
   end loop ;
   New_Line ;
end Affiche;
```

Étant donné une constante entière Ada `M` fixée, on introduit une variable globale `T`.

```
subtype Indice is Integer range 1..M ;
T: TabBits(Indice) ;
```

$\triangleright$ **Question 8 (2 points) :**

Écrire une procédure **récursive** `FRec` telle que `FRec(1)` affiche la suite calculée par $F(M)$.

```
procedure FRec(N:Indice) ;
```

Cette procédure utilise `T` pour mémoriser le mot en cours de calcul et correspond à l'affichage de 2^{M+1-N} mots. Plus précisément, `FRec(N)` calcule dans `T(N..M)` les mots successifs de la suite $F(M - N + 1)$, le reste du tableau `T(1..N-1)` n'étant pas modifié.

On imite maintenant la démarche de la question 8 pour les codes de Gray. La première étape consiste à trouver une définition récursive calculant un code de Gray. On remarque que les codes de Gray satisfont la propriété suivante :

> *Si s et s' sont deux codes de Gray sur M bits tels que* $\mathrm{der}(s) = \mathrm{prem}(s')$, *alors, pour tout bit b, $(b \triangleright s);(\neg\, b \triangleright s')$ est un code de Gray sur $M+1$ bits.*

Par exemple, si s vaut $00;01;11;10$ et s' vaut $10;00;01;11$, alors pour tout bit b, $(b\triangleright s);(\neg\, b\triangleright s')$ est un code de Gray sur 3 bits (distinct de ceux donnés figure 2).

▷ Question 9 (3 points) :

En utilisant cette propriété et en s'inspirant de la définition précédente de "$|.|$" (calculant la longueur des mots non vides), définir par récurrence une fonction mathématique G paramétrée par un mot non vide m qui calcule un code de Gray sur $|m|$ bits commençant par m. Les codes de Gray calculés par $G(000)$ et $G(010)$ doivent correspondre à ceux de la figure 2 page 6. On utilisera les notations sur les mots non vides et les suites de mots introduites par l'énoncé.

▷ Question 10 (1 point) :

Donner le code de Gray produit par $G(101)$.

On adapte maintenant la démarche de la question 8 pour afficher la suite des mots calculée par G. Dans le calcul de G, on a besoin de ne mémoriser que le dernier mot calculé qui correspond exactement à la variable globale `T` de la question 8.

▷ Question 11 (2,5 points) :

Écrire une procédure **récursive** `BinGrayRec` telle que `BinGrayRec(1)` affiche le code de Gray calculé par $G(m)$ où m est le mot initialement représenté par `T`.

```
procedure BinGrayRec (N:Positive) ;
-- requiert N<=M+1
```

Ainsi, un appel à `BinGrayRec(M+1)` se contente d'afficher la valeur courante de `T`. Sinon, pour $N \neq M+1$, `BinGrayRec(N)` calcule dans `T(N..M)` les mots successifs du code de G sur la valeur initiale de `T(N..M)`, le reste du tableau `T(1..N-1)` n'étant pas modifié.

▷ Question 12 (1 point) :

Combien `BinGrayRec(1)` effectue-t-elle d'affectations de booléens dans le tableau `T` (en fonction de `M`) ? Comparer avec `FRec(1)` et commenter cette différence.

Grenoble INP-Ensimag **Analyse pour l'ingénieur**
1ère année

Examen final
Mardi 26 janvier 2010 - 2h

Documents manuscrits, polycopié du cours et calculatrice autorisés.
Le nom du groupe doit être noté en évidence sur la première page

Exercice 1

1. Calculer les transformées de Fourier des fonctions suivantes :

$$f_0(x) = \frac{x^k}{k!}e^{-a|x|}U(x) \quad f_1(x) = \frac{1}{a-2i\pi x} \quad f_2(x) = \left(\frac{\sin(px)}{px}\right)^n$$

où $U(x) = 1$ si $x \geq 0$ et 0 sinon.

2. Calculer la transformée de Fourier de la fonction $x \to x\ e^{-x}U(x)$, où $U(x)$ est la fonction valant 1 si $x \geq 1$ et 0 si $x < 0$.
En déduire, par parité, la transformée de Fourier de la fonction $x \to |x|\ e^{-|x|}$.

3. Calculer l'intégrale suivante

$$\int_{\mathbb{R}} \frac{x^2}{(x^2+a^2)^2}\,dx$$

4. Soit f une fonction intégrable sur $\mathbb{R}$. Calculer la limite :

$$\lim_{n\to+\infty} \int_{-\infty}^{+\infty} \cos(nx)f(x)\,dx$$

Exercice 2 Soient f et g deux fonctions de $L^2(\mathbb{R})$. On appelle intercorrélation de f et g la fonction de τ :

$$\mathcal{C}_{fg}(\tau) = \int_{-\infty}^{+\infty} f(x)g(x-\tau)dx$$

1. Montrer que $\tau \to \mathcal{C}_{fg}(\tau)$ est définie pour tout $\tau \in \mathbb{R}$.

2. Montrer que si de plus f et g sont intégrables sur $\mathbb{R}$, alors $\tau \to \mathcal{C}_{fg}(\tau)$ est une fonction intégrable sur $\mathbb{R}$.

3. Montrer que

$$\mathcal{C}_{fg}(\tau) = \int_{-\infty}^{+\infty} \hat{f}(\nu)\overline{\hat{g}(\nu)}e^{2i\pi\nu\tau}d\nu$$

En déduire que si f et g sont intégrables sur $\mathbb{R}$, alors la transformée de Fourier de $\tau \to \mathcal{C}_{fg}(\tau)$ est :

$$\hat{\mathcal{C}}_{fg}(\nu) = \hat{f}(\nu)\overline{\hat{g}(\nu)}$$

4. L'autocorrélation d'une fonction $f \in L^2(\mathbb{R})$ est définie par la fonction $\mathcal{C}_{ff}$ (intercorrélation de f avec elle-même).
Montrer que la fonction $\tau \to \mathcal{C}_{ff}(\tau)$ est paire, et vérifie :

$$\forall\tau \in \mathbb{R},\ |\mathcal{C}_{ff}(\tau)| \leq \mathcal{C}_{ff}(0)$$

Que représente $\mathcal{C}_{ff}(0)$ pour le signal f ?

5. Montrer que $\lim_{\tau\to+\infty} \mathcal{C}_{ff}(\tau) = 0$

6. Calculer la fonction d'autocorrélation et la transformée de Fourier de cette fonction d'autocorrélation pour les fonctions suivantes :

(a) $f(x) = \frac{1}{a}e^{-\frac{x}{a}}$ si $x > 0$ et 0 sinon ($a > 0$).

(b) $f(x) = 1$ si $-\frac{a}{2} < x < \frac{a}{2}$ et 0 sinon ($a > 0$).

Exercice 3 *Partie finie de* $\frac{1}{x^2}$
Pour $\varphi \in \mathcal{D}(\mathbb{R})$ on définit :

$$\langle Pf(\frac{1}{x^2}), \varphi\rangle = \lim_{\varepsilon \to 0^+} \left(\int_{|x|>\varepsilon} \frac{\varphi(x) - \varphi(0)}{x^2}\, dx \right)$$

1. Montrer que $Pf(\frac{1}{x^2})$ est une distribution sur $\mathbb{R}$. On montrera auparavant qu'elle est bien définie sur $\mathcal{D}(\mathbb{R})$

2. Montrer que

$$\frac{d}{dx} Vp(\frac{1}{x}) = -Pf(\frac{1}{x^2})$$

où $Vp(\frac{1}{x})$ est la distribution définie par :

$$\langle Vp(\frac{1}{x}), \varphi\rangle = \lim_{\varepsilon \to 0^+} \left(\int_{|x|>\varepsilon} \frac{\varphi(x)}{x} dx \right)$$

(cette distribution a été étudiée en TD, inutile de redémontrer ses propriétés)

Exercice 4 $\mathcal{D}'(\mathbb{R})$ désigne l'ensemble des distributions sur $\mathbb{R}$.

1. Soit $f_n(x) = ne^{-n^2x^2}$. Calculer la limite dans $\mathcal{D}'(\mathbb{R})$ de la suite de distributions-fonctions f_n.

2. Calculer dans $\mathcal{D}'(\mathbb{R})$:

$$\left(\frac{d}{dx} - \alpha \right) U(x) e^{\alpha x}$$

où $\alpha \in \mathbb{R}$ et $U(x) = 1$ si $x \geq 0$ et 0 sinon.

Architecture 1 : Circuits Numériques et Éléments d'Architecture
Examen final

ENSIMAG 1A

Année scolaire 2009–2010

- Durée : 3h. Tous documents et calculatrices autorisés.
- Le barème est donné à titre <u>indicatif</u>.
- Les exercices sont indépendants et peuvent être traités dans le désordre.

Ex. 1 : Schémas à base de transistors et analyse de masques (4 pts)

Question 1 Donnez un schéma CMOS dual (sans porte de passage) de la fonction booléenne majorité $f = \overline{x \cdot y + x \cdot z + y \cdot z}$ en 12 transistors (ou encore mieux en 10).

Question 2 Donnez le schéma en transistor du masque fourni en dernière page, sachant que les ports d'entrée sont A et B, et la sortie Q. Précisez la fonction réalisée.

Ex. 2 : Conception Partie-Contrôle/Partie Opérative (8 pts)

Tracer des lignes et des cercles sur un écran (qui est une matrice discrète) peut se faire avec des méthodes mathématiques classiques qui font appel à des fonctions mathématiques qu'on ne sait pas implanter efficacement (ni en matériel, ni en logiciel). Des méthodes spécifiques ont donc été développées pour le tracer de lignes et de cercles. Nous nous intéresserons ici à la méthode d'Andres dont l'algorithme pour tracer le $\frac{1}{8}$ème de cercle du premier octant ($x \geq 0, y \geq 0, x \geq y$) est donné ci-dessous (les 7 autres octants pouvant être obtenus par symétrie).

```
x = 0;
y = rayon;
d = rayon - 1;
while (y >= x) {
    tracer_le_point(x, y);
    if (d >= 2 * x) {
        d = d - 2 * x - 1;
        x = x + 1;
    } else if (d <= 2 * (rayon - y)) {
        d = d + 2 * y - 1;
        y = y - 1;
    } else {
        d = d + 2 * (y - x - 1);
        y = y - 1;
        x = x + 1;
    }
}
```

Pour les besoins de l'exercice, la fonction `tracer_le_point` consiste simplement à indiquer à l'aide du signal `trace` que les données dans `x` et `y` sont valides. L'entrée `rayon` n'est valide qu'au début de l'algorithme.

On va utiliser la méthode présentée en cours et mise en œuvre en TD pour proposer une implantation de cette architecture sous la forme PC/PO.

Dans un premier temps, on va réaliser une version naïve de l'implantation qui ne fait pas d'optimisation de performance mais qui minimise le matériel utilisé.

Question 1 Définissez l'ensemble des instructions à réaliser ; déduisez-en le type des unités fonctionnelles (registres ou opérateurs) et le nombre minimal de chacun d'eux, avec la contrainte que l'on souhaite passer au plus un cycle par ligne d'instruction de l'algorithme tel qu'il est écrit. On cherchera à utiliser les opérateurs les plus simples pour chaque opération.

On rappelle aussi que pour un soustracteur, mettre l'entrée de « carry in » (retenue entrante) à 0 permet de produire en sortie le résultat $A - B - 1$ (en supposant que les entrées soient A et B). De même, on pourra supposer l'existence de sorties z (valant 1 quand le résultat est nul) et $sign$ (valant 1 quand le résultat est négatif) sur les opérateurs utilisés.

Question 2 Proposez une partie opérative interconnectant ces unités fonctionnelles et pensez à faire ressortir les fils de contrôle des unités (sélecteurs de mux, we des registres, etc).

Question 3 Proposez une partie contrôle qui pilote cette partie opérative.

Si l'on se dote de suffisamment de ressources, on peut effectuer un grand nombre d'opérations en parallèle sans changer la structure du programme.

On peut remarquer que les lignes (7,8), (10,11) et (13,14,15) sont mutuellement exclusives. On se propose donc de paramétrer les opérations à effectuer en fonction du résultat des conditions directement dans le chemin de données (ce que l'on a appelé le câblage des conditions dans le cours).

Question 4 Réécrivez le programme en utilisant la notion d'affectation concurrente présentée en cours, par exemple $(a, b) \leftarrow (3, 1)$ affecte 3 dans a et 1 dans b.

Question 5 Afin de mettre en évidence les opérations que l'on désire faire en parallèle, reformuler le code avec une affectation conditionnelle ternaire. Pour mémoire, l'affectation conditionnelle binaire vue en cours s'écrit

`x = condition? valeur si vraie : valeur si fausse.`

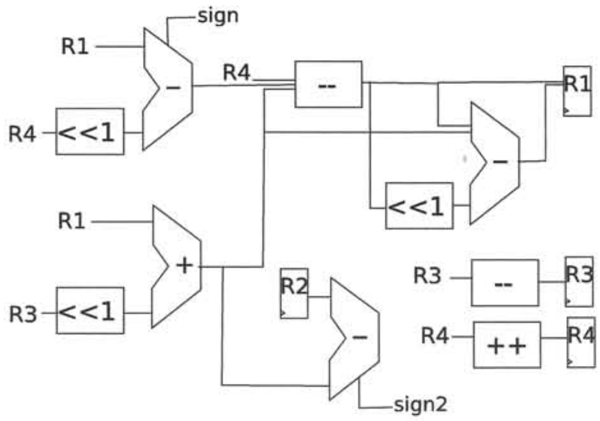
On introduit l'affectation conditionnelle ternaire avec 3 conditions (c1, c2 et défaut) mutuellement exclusives :

`x = [c1|c2]? valeur si c1 vraie : valeur si c2 vraie : valeur sinon`

On peut laisser une action vide s'il n'y a pas de changement dans la condition, et on peut bien sur coupler cela avec l'affectation concurrente.

Question 6 En complétant le schéma fourni ci-dessous (multiplexeurs, signaux de commandes, comptes-rendus, noms des registres, logique manquante...), proposez une partie opérative avec câblage de la condition.

Question 7 Proposez une partie contrôle qui pilote cette PO.



Ex. 3 : Conception de processeur (8 pts)

Cet exercice utilise le processeur vu en TD et TP. On se propose d'ajouter les instructions d'appel à (`jmpf`) et de retour de (`retf`) fonction.

Le principe de l'appel à une fonction consiste à sauver en mémoire l'adresse de la prochaine instruction à exécuter, puis de sauter à l'adresse cible du jmpf. La zone mémoire dans laquelle est sauvée l'adresse s'appelle la « pile », dont le sommet est identifié par un registre spécifique, le pointeur de pile. A chaque fois que l'on enregistre une nouvelle valeur de PC dans la pile, le pointeur de pile doit être incrémenté pour permettre de sauvegarder la valeur de PC lors de l'appel suivant. L'instruction `retf` doit restaurer dans PC la valeur située en sommet de pile, et décrémenter le pointeur de pile.

On rappelle le codage des instructions supportées par ce processeur :

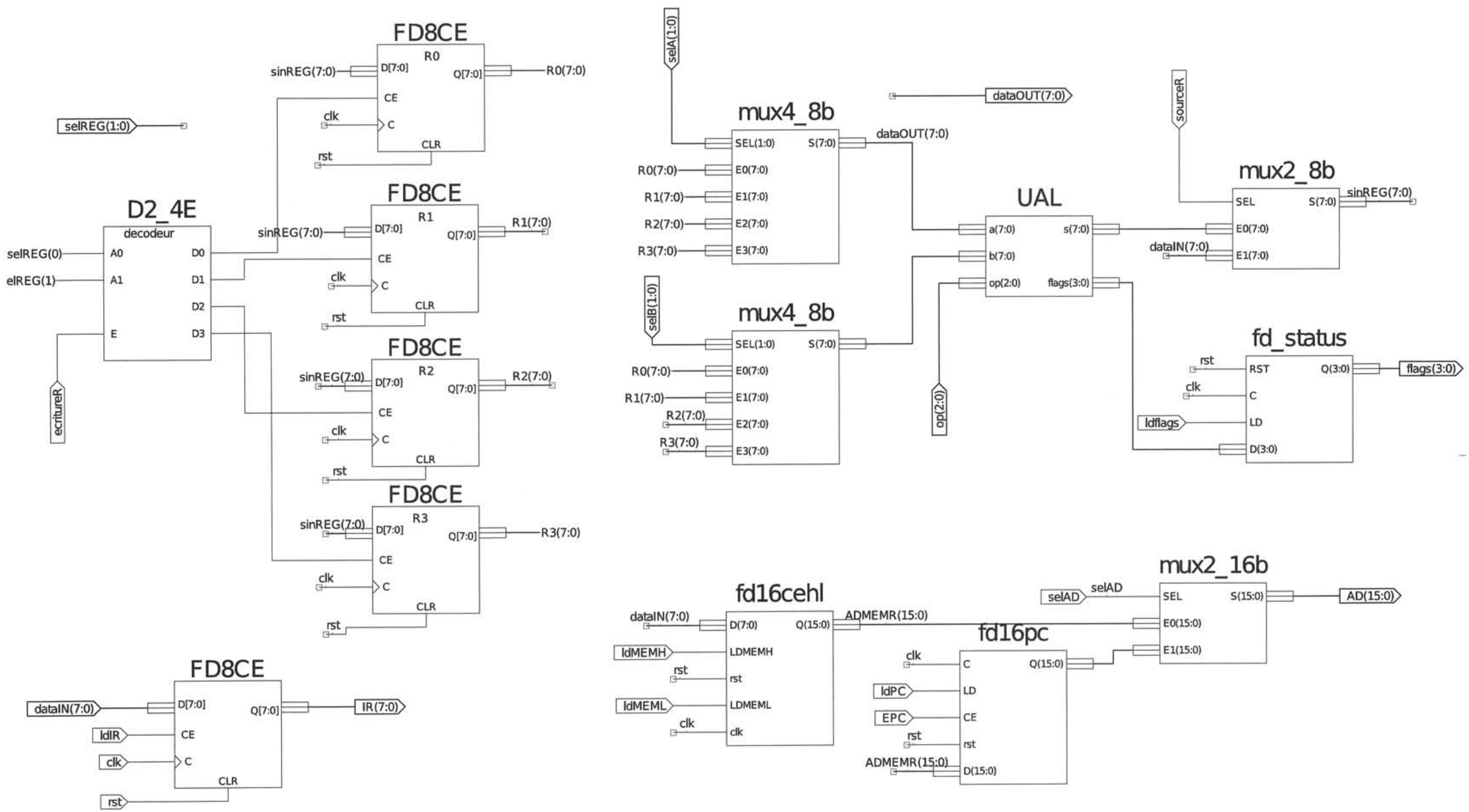
Instruction	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Opérations de la forme : $rd := rd\ op\ rs$								
`or rs, rd`	0	0	0	0	rs_1	rs_0	rd_1	rd_0
`xor rs, rd`	0	0	0	1	rs_1	rs_0	rd_1	rd_0
`and rs, rd`	0	0	1	0	rs_1	rs_0	rd_1	rd_0
`add rs, rd`	0	1	0	0	rs_1	rs_0	rd_1	rd_0
`sub rs, rd`	0	1	0	1	rs_1	rs_0	rd_1	rd_0
Opérations de la forme : $rd := op\ rd$								
`not rd`	0	0	1	1	0	0	rd_1	rd_0
`shl rd`	0	1	1	0	0	0	rd_1	rd_0
`shr rd`	0	1	1	1	0	0	rd_1	rd_0
Chargement : $rd := MEM(AD)$								
`ld AD, rd`	1	0	0	0	0	0	rd_1	rd_0
	ADH							
	ADL							
Stockage : $MEM(AD) := rs$								
`st rs, AD`	1	1	0	0	0	0	rs_1	rs_0
	ADH							
	ADL							
Branchement inconditionnel : $PC := AD$								
`jmp AD`	1	0	0	0	0	1	0	0
	ADH							
	ADL							

Question 1 Précisez un codage possible ainsi que la structure des instructions `jmpf` et `retf`.

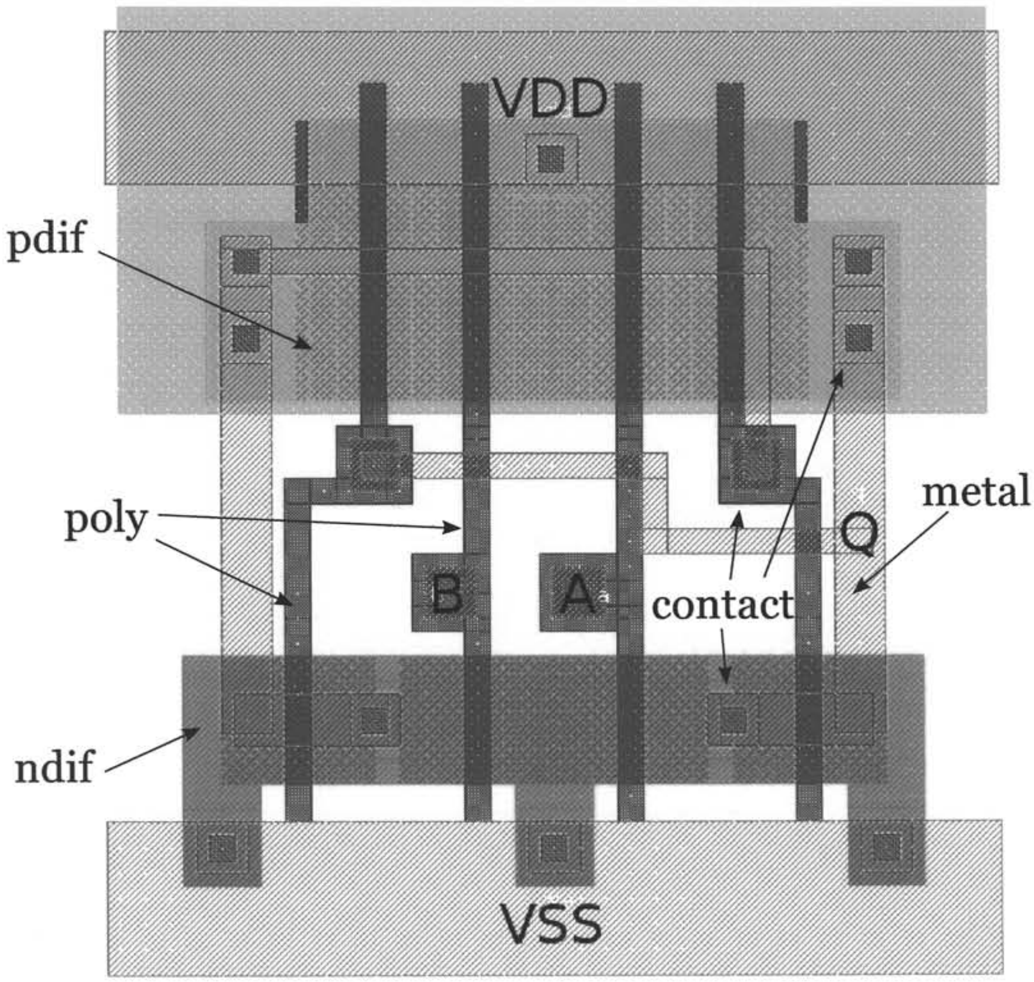
Question 2 Étant donnée la PO en annexe, précisez les ajouts (registres et interconnexions) qui sont nécessaires pour permettre appels et retours de fonctions. Vous pouvez redessiner la partie de la PO qui se trouve modifiée. Pour éviter de se servir de l'UAL, on pourra utiliser des registres à incrément/décrément.

Question 3 Rappelez l'automate de la PC vu en TD, et proposez des modifications afin que ce dernier supporte l'instruction `jmpf`.

Question 4 Proposez des modifications de la PC afin de supporter l'instruction `retf`.



FD8CE
R0
sinREG(7:0)
D[7:0]
Q[7:0]
R0(7:0)
CE
C
CLR
clk
rst
selREG(1:0)
D2_4E
decodeur
selREG(0)
elREG(1)
A0
A1
E
D0
D1
D2
D3
ecritureR
FD8CE
R1
R1(7:0)
FD8CE
R2
R2(7:0)
FD8CE
R3
R3(7:0)
FD8CE
dataIN(7:0)
IR(7:0)
ldIR
mux4_8b
selA(1:0)
SEL(1:0)
E0(7:0)
E1(7:0)
E2(7:0)
E3(7:0)
S(7:0)
dataOUT(7:0)
mux4_8b
selB(1:0)
UAL
a(7:0)
b(7:0)
op(2:0)
s(7:0)
flags(3:0)
mux2_8b
sourceR
SEL
E0(7:0)
E1(7:0)
S(7:0)
sinREG(7:0)
fd_status
RST
LD
D(3:0)
Q(3:0)
ldflags
flags(3:0)
mux2_16b
selAD
E0(15:0)
E1(15:0)
S(15:0)
AD(15:0)
fd16cehl
D(7:0)
LDMEMH
LDMEML
Q(15:0)
ldMEMH
ldMEML
ADMEMR(15:0)
fd16pc
D(15:0)
ldPC
EPC


VDD
pdif
poly
B
A
contact
metal
Q
ndif
VSS

ENSIMAG - 1ère Année – 2009/2010

ECONOMIE

DS du 27 janvier 2010

Durée : 2 heures
Tous documents autorisés sauf mobiles et ordinateurs portables

Travail sur l'un des textes joints, au choix.

Votre devoir ne doit pas dépasser quatre pages, soit une copie double.

1- **Synthèse du texte**

   - Thème général de l'article (une phrase)
   - Question (problématique) soulevée par le texte (une phrase)
   - Réponse (une phrase)
   - Principaux éléments évoqués pour répondre à la question (quelques phrases).

2- Lister les **éléments du texte** que l'auteur ne développe pas et qu'il serait nécessaire d'expliciter **pour une bonne compréhension de l'article** : des notions, des mécanismes, des raisonnements, des références historiques, etc.

3- Choisir, dans 2), **deux éléments** qui vous paraissent particulièrement importants et **rédiger**, pour chacun d'eux, **l'explication** qui vous paraît nécessaire à sa bonne compréhension. (Exclure de ce choix la seule définition de notions).

4- **Conclusion** : En quelques phrases, présentez ce qui vous paraît constituer les points forts ainsi que les limites éventuelles de l'article.

Texte 1

Une dirigeante de la Fed prédit une période prolongée de taux bas

Reuters - Le lundi 4 janvier 2010 - David Lawder, version française Gwénaelle Barzic.

La Réserve fédérale américaine (Fed[1]) estime que la reprise devrait se poursuivre en 2010 mais elle devra maintenir des taux « *exceptionnellement bas* » pendant une période prolongée afin de soutenir l'emploi, déclare une responsable de la Fed.

« *Le taux d'utilisation des ressources va probablement rester inférieur à la normale pendant encore quelques temps ce qui devrait contribuer à contenir l'inflation* », souligne Elizabeth Duke, gouverneur de la Fed, dans un discours préparé en vue de sa participation à une conférence économique à Raleigh en Caroline du Nord.

« *Dans l'environnement actuel, la Fed estime toujours que les conditions économiques nécessitent des taux exceptionnellement bas pendant une période prolongée* », explique-t-elle.

« *Ce type de politique accommodante est nécessaire pour permettre à terme un retour à des niveaux d'activité réelle et de chômage plus satisfaisants dans le contexte d'une stabilité des prix* », ajoute-t-elle.

La responsable de la Fed a souligné que les dernières statistiques sur la production industrielle et les dépenses des ménages laissaient penser que l'activité économique avait continué à se redresser à un « *rythme solide* » au cours des derniers mois de l'année 2009.

« *Je m'attends à voir une reprise modérée continue de l'activité économique en 2010 soutenue par la poursuite de la guérison des marchés financiers et par une politique monétaire accommodante* », prédit-elle.

Elizabeth Duke a toutefois insisté sur les difficultés persistantes du marché immobilier américain, plombé par un nombre élevé de saisies et un accès difficile au crédit pour les promoteurs et les ménages.

Les prêts sont également difficiles à obtenir pour les entreprises, a-t-elle ajouté, en indiquant que ses prévisions de croissance seraient fonction du niveau de redressement des marchés financiers et d'amélioration de l'accès au crédit.

En outre, les entreprises vont hésiter avant d'embaucher à nouveau du personnel, a-t-elle expliqué, car les pressions pour contrôler les coûts et gagner en efficacité devraient rester fortes tant que la reprise ne se sera pas solidement installée.

« *Par conséquent, je m'attends à ce que les entreprises commencent à réembaucher cette année mais elles devraient le faire de manière prudente afin de conserver leurs économies de coûts et leurs gains d'efficacité* », a-t-elle dit. « *Même si le taux de chômage devrait commencer à reculer cette année, il devrait rester élevé par rapport à la moyenne habituelle* », a-t-elle ajouté.

---

[1] Banque Centrale des USA.

Texte 2

Sarkozy se bat contre le PIB

AFP - Lundi 14 septembre 2009

Oui, il faut intégrer le « *bien-être* » dans la mesure de la croissance. C'est le rapport Stiglitz, prix Nobel d'économie, qui l'affirme.

Nicolas Sarkozy a plaidé lundi pour un changement de la mesure du progrès économique et social, dénonçant la « *religion du chiffre* » actuelle. « *La France se battra pour que toutes les organisations internationales modifient leurs systèmes statistiques (...) La France proposera à ses partenaires européens que l'Europe donne l'exemple. La France va adapter son propre appareil statistique en conséquence.* », a promis le président de la République dans un discours à l'université de la Sorbonne.

Il s'exprimait à l'occasion de la réception du rapport de la « Commission de mesure de la performance économique et du progrès social » dirigée par les économistes américain Joseph Stiglitz, prix Nobel, indien Amartya Sen et français Jean-Paul Fitoussi.

Dans ce rapport, cette commission préconise l'élaboration d'un « *système statistique qui complète les mesures de l'activité marchande par des données relatives au bien-être des personnes* ». Mais rien de précis n'est avancé sur les moyens de mesurer une donnée aussi subjective et abstraite. Dans son discours à la Sorbonne, Joseph Stiglitz, ancien conseiller de Bill Clinton à la Maison Blanche, a reconnu que « *ce rapport n'est qu'un premier pas* » et qu'on était encore loin du « but ».

Changer la mesure

Pour autant, Nicolas Sarkozy avance qu'il « *y a depuis longtemps un problème avec ce que nous calculons et avec la manière dont nous l'utilisons* ». « *Pendant des années, les statistiques ont affiché une croissance économique de plus en plus forte (...) jusqu'à ce qu'il apparaisse que cette croissance, en mettant en péril l'avenir de la planète, détruisait davantage qu'elle ne créait* », a-t-il jugé.

« *Le problème vient de ce que le monde, la société, l'économie ont changé et que la mesure n'a pas assez changé* », a poursuivi le président, « *c'est un fossé très dangereux parce que le citoyen a fini par penser qu'on le trompe* ». « *Dans le monde entier, les citoyens pensent qu'on leur ment, que les chiffres sont faux et pire, qu'ils (les chiffres) sont manipulés* », a-t-il enchaîné, « *rien n'est plus destructeur pour la démocratie* ». Nicolas Sarkozy a notamment suggéré d'introduire dans les statistiques « *les services que l'on se rend à l'intérieur d'une famille* », « *les loisirs* » ou « *la qualité du service public* ».

« ***La religion du chiffre*** »

Outre la « *religion du chiffre* », le président a pourfendu « *un système fondé sur des moyennes, car la moyenne, c'est une façon de ne jamais parler des inégalités* », et « *la religion du marché qui, par principe, a raison* ».

« *Le marché, dans lequel je crois, n'est pas porteur de sens (...) de responsabilité (...) de projet (...) de vision. Les marchés financiers encore moins* », a-t-il insisté, « *à force de faire comme si toute la vérité était dans le marché, eh bien on a finit par le croire* », a conclu Nicolas Sarkozy.

Constituée en février 2008 à l'initiative de Nicolas Sarkozy et composée de 25 experts, la Commission avait pour mission d'identifier et de pallier les limites du produit intérieur brut (PIB) comme indicateur de performance économique et de progrès social.

Grenoble INP – ENSIMAG
Année 2009-2010

Introduction aux réseaux

Examen du 29/01/2010. Durée 3h00
Polycopié du cours, des TD-TP et notes personnelles autorisés

Le barème est indicatif. Le problème et chacun des 4 exercices sont indépendants.

Problème (14 points)

Un Message impérial

L'Empereur - dit-on - t'a envoyé, à toi en particulier, à toi, sujet pitoyable, ombre devant le soleil impérial chétivement enfouie dans le plus lointain des lointains, à toi précisément, l'Empereur de son lit de mort a envoyé un message.[...] Le messager s'est mis en route tout de suite[...], il se fraye un chemin à travers la foule; s'il rencontre de la résistance, il désigne sa poitrine où est le signe du soleil; il avance facilement, comme nul autre. Mais la foule est grande et elle n'en finit pas d'habiter partout. Si l'espace s'ouvrait devant lui, comme le messager volerait. Et bientôt tu entendrais le battement magnifique de ses poings à ta porte. Mais hélas, que ses efforts restent vains ! Et il est toujours à forcer le passage à travers les appartements du palais central; jamais il ne les franchira, et s'il surmontait ces obstacles, il n'en serait pas plus avancé; dans la descente des escaliers, il aurait encore à se battre; et s'il parvenait jusqu'en bas, il n'en serait pas plus avancé, il lui faudrait traverser les cours; et après les cours, le second palais qui les entoure, et de nouveau des escaliers et des cours, et de nouveau un palais; et ainsi de suite durant les siècles des siècles; et si enfin il se précipitait par l'ultime porte - mais jamais, jamais cela ne pourrait se produire – il trouverait devant lui la Ville Impériale, le centre du monde [...]. Là, personne ne pénètre, même pas avec le message d'un mort. Mais toi, tu es assis à ta fenêtre, et dans ton rêve tu appelles le message quand vient le soir.
Franz Kafka, Lors de la construction de la muraille de Chine.

Le nouvel empereur de Chine (de la dynastie Qin, avant notre ère), jugeant désastreux l'état des communications dans son empire, confie la conception d'un réseau moderne de communication à deux de ses très éminentes sociétés savantes :

- l'École de l'Harmonie Mathématique du Ciel
- la Céleste Académie des Communications du Lointain.

Après un mois d'intense labeur, ces deux institutions présentent leurs solutions à l'empereur. Celui-ci reste perplexe devant les options très éloignées prises par l'EHMC et la CACL. Il vous charge, vous son très humble conseiller, d'étudier comparativement les deux systèmes et de rendre votre recommandation avant le coucher du soleil. Vous vous mettez sur le champ au travail.

Données du problème posé

Les textes à transmettre sont constitués d'idéogrammes. Chaque nom de ville peut être codé avec un idéogramme. Dans les villes importantes il y a également des nœuds terminaux du réseau. Par exemple, dans la capitale se trouve la chancellerie de l'empereur, d'où émanent ses messages, et les destinataires importants peuvent avoir un bureau pour recevoir ou écrire des messages. Les noms des destinataires et des émetteurs tiennent chacun sur 2 idéogrammes.

Analyse de la solution proposée par l'École de l'Harmonie Mathématique du Ciel

Il s'agit d'un système de communication optique. En renvoyant les rayons du soleil au moyen d'un miroir, on peut envoyer des signaux à un point éloigné. De celui-ci, on peut voir les éclats lumineux et retransmettre le signal au point suivant. Ce système –appelé héliographe- nécessite l'installation de stations héliographiques situées sur des tours de guet.

Répondez précisément et concisément à chacune des questions suivantes. (Bien évidemment, les questions portent sur cet héliographe, pas sur Internet !)

Question 1. *Il faut d'abord convertir le message à envoyer en une succession d'éclats lumineux. Comment s'appelle cette opération en théorie des réseaux ?*

On peut envisager deux façons pour relayer le message d'une tour à la suivante.

A. Comme dans le télégraphe de Chappe, dès qu'un opérateur est averti d'un début d'émission en amont, il se positionne pour retransmettre chaque éclat avec les mêmes caractéristiques à la tour en aval.
B. Sur une tour, un opérateur recevra d'abord le message de l'amont avant de se tourner vers l'aval pour le retransmettre à son tour.

Question 2. *En référence au cours sur les réseaux, comment appelleriez-vous la façon A, et la façon B ?*

Enfin, si la propagation du rayon solaire est très rapide, les temps de réaction des opérateurs humains sont certes plus longs. Par ailleurs, dans la façon B, l'opérateur est indisponible pour l'amont pendant qu'il transmet vers l'aval.

Question 3. *Toujours en référence au cours, quelle fonction de réseau doit-on mettre en œuvre pour gérer ces temps de réaction et ces indisponibilités ?*

Aussi, afin d'éviter des attentes inutiles lors de transmissions consécutives, chaque tour de guet est occupée en fait par plusieurs opérateurs d'héliographie, de sorte qu'un message arrivant de l'amont est observé par un premier opérateur tourné vers la tour amont, qui note le message sur une ardoise qu'il remet à un autre opérateur qui le retransmet vers l'aval pendant que le premier opérateur reste disponible pour observer le message suivant.

Délais

Le cours identifiait quatre sources possibles de délai : attente, propagation, traitement et transmission.

Question 4. *Essayez d'identifier les sources de délai correspondantes dans votre héliographe, et indiquez quels pourraient être les ordres de grandeur typiques pour vos stations d'héliographie, dans la façon B ?*

Les idéogrammes du chinois doivent être transcrits en des successions d'éclats lumineux, sans ambiguïté malgré la distance et les imprécisions de manipulation. Avec une technique de conversion robuste et des opérateurs entraînés, <u>un idéogramme peut être transmis en 20 secondes</u>.

La configuration des terrains et les contraintes de visibilité conduisent à ne pouvoir espacer les stations d'héliographie que de <u>5 km en moyenne</u>. Le coût élevé de ce dispositif conduit donc à n'équiper que quelques grands axes reliant les principales villes de l'empire.

Dans la façon B, on convient de découper les textes longs en messages de 20 idéogrammes. Cependant pour faciliter la reconnaissance des frontières de messages lors d'une longue émission, on rajoute lors de la transmission un idéogramme de début de message et un idéogramme de fin de message, ce qui permet de synchroniser les opérateurs. La transmission optique d'un message de 20 idéogrammes utiles se fait donc avec 22 idéogrammes.

Question 5. *En supposant qu'il n'y a pas d'attente, quel est le délai mis par un message pour traverser une station d'héliographie (délai entre la fin de l'observation en amont et la fin de l'émission en aval)?*

Question 6. *On veut calculer le délai entre le début d'émission (à la source) et la fin de la réception (à destination) d'un texte de M x 20 idéogrammes sur une distance d (exprimée en km, d étant un multiple de 5). Écrivez la formule donnant ce délai en minutes en fonction de M et de d.*

Jusqu'à présent, on n'a considéré qu'une transmission isolée « en ligne droite », sur les artères de transmission reliant les grandes villes. Dans la façon B, à l'intérieur des 20 mots d'un message, les 5 premiers mots servent à assurer l'acheminement. Le premier mot identifie la ville de destination, les 2 suivants le destinataire du message dans la ville, et les 2 autres identifient le nom de l'expéditeur du message.

Question 7. *La prise en compte du routage a deux influences majeures sur votre formule de délai (Question 6) : lesquelles ?*

Analyse de la solution proposée par la Céleste Académie des Communications du Lointain

La Céleste Académie des Communications du Lointain propose la réalisation d'un service de communications impériales par pigeons voyageurs. En voici les *caractéristiques techniques*.

- En cette époque, les messages sont gravés sur des os. Un pigeon peut emporter un petit os sur lequel les scribes arrivent à graver 20 idéogrammes.
- Ainsi chargé, un pigeon peut parcourir plusieurs centaines de km, mais on a choisi une moyenne de 200 km entre les colombiers. La vitesse moyenne d'un pigeon sur un tel parcours est de 60 km/h.
- Entre deux villes éloignées qu'on veut relier, on intercale des colombiers ruraux servant de relais tous les 200 km.
- Un pigeon va d'un colombier au suivant (qui est son colombier d'attache, d'où on le renverra dans les jours qui suivent par voie terrestre au colombier précédent, prêt pour une nouvelle mission). Lors de son arrivée, le message qu'il porte est pris en charge dans les 5 minutes qui suivent et confié à un autre pigeon qui le portera au colombier encore suivant vers la destination du message.

Dans chaque nœud urbain, un colombier standard accueille jusqu'à 1000 pigeons (250 pour les colombiers ruraux). Un personnel suffisant y est associé pour s'occuper des pigeons et des messages. L'approvisionnement en os ne pose pas de difficultés. Un pigeon reste 2 jours au repos, et le matin du troisième jour, un cavalier le ramène au colombier voisin, où il arrive le soir même.

Dans un premier temps, on suppose qu'il n'y a pas de perte, et que tous les messages arrivent dans l'ordre.

Identification de l'architecture des protocoles

Répondez précisément et concisément à chacune des questions suivantes. (Bien évidemment, les questions portent sur le réseau de pigeons voyageurs, pas sur Internet !)

Question 8. *En quoi consiste (ici) la fonction de codage, et sur quels (types de) nœuds (terminal, urbain, rural) doit-on la mettre en œuvre ? Même question pour la fonction de segmentation, et pour la fonction de routage.*

Vous structurez les protocoles mis en œuvre sur chaque nœud en couche selon les principes établis des architectures qu'on appellera dans plusieurs siècles de type OSI et TCP/IP. Au-dessus de la couche physique assurée par les pigeons, vous définissez 3 couches : liaison, réseau et transport.

Question 9. *Associez les fonctions des questions 1 à 3 aux couches normalisées (il peut y avoir des couches vides) et indiquez quelles couches sont implantées sur chaque type de nœud.*

Délais

Le cours identifiait quatre sources de délai : attente, propagation, traitement et transmission.

Question 10. *Quelles sources de délai sont pertinentes pour le réseau de pigeons voyageurs, et quelles sont les valeurs que vous leur associez selon les caractéristiques techniques?*

Limitations, débits utiles

Pour la transmission de textes longs, on découpe le texte en messages, qu'on numérote sur 2 caractères chacun en base 100 (soit 10 000 valeurs).

Question 11. *En prenant en compte les caractéristiques techniques, quelle est la longueur maximum (en nombre de caractères) d'un document que l'on peut transmettre d'un point du réseau à un autre avant d'encourir une brusque réduction du débit?*

Question 12. *En vous inspirant d'une analogie du réseau téléphonique, proposez deux solutions pour doubler cette capacité maximum : l'une jouant sur la transmission, l'autre sur la commutation.*

Pertes

En raison de la présence de rapaces, certains pigeons n'arrivent pas à destination (1 sur 10 en moyenne). Certains nœuds devront donc garder des copies pour pouvoir retransmettre. Il faut 20 minutes pour graver une copie.

Question 13. *Compte tenu des caractéristiques de votre réseau, dans quelle couche proposez vous de gérer les pertes par retransmission, et pourquoi ?*

Comparaison des délais des deux solutions

Comme à la Question 6, on veut calculer le délai entre le début d'émission (à la source) et la fin de la réception (à destination) d'un texte de M x 20 idéogrammes sur une distance d (exprimée en km, d étant un multiple de 5).

Question 14. *Calculer le délai pour la solution à pigeons voyageurs*

Question 15. *Comparez qualitativement les deux réseaux pour le délai de transmission : dans quelles conditions un réseau est-il plus avantageux que l'autre ?*

L'évaluation comparative que vous venez de faire suppose que chaque réseau ne met en œuvre qu'une seule technologie.

Question 16. *Si vous aviez à concevoir un réseau avec une partie "cœur de réseau" et une partie "réseau d'accès", comment répartiriez-vous l'usage de ces deux technologies entre les deux parties de réseau ?*

Exercices (6 points)

Voix sur IP

Question 17. *La voix téléphonique numérisée peut être transmise par paquet sur les réseaux Internet. Rappelez le principe de la paquétisation de la voix téléphonique. Quels en sont les avantages et inconvénients ?*

Réseau d'accès

Question 18. *Dans une récente décision, l'ARCEP a demandé aux opérateurs désireux d'installer des réseaux de fibre optique jusque chez l'abonné de câbler systématiquement deux fibres verticales dans les immeubles. Expliquez cette décision.*

Flux générés par les SMS

Dans la journée du 1er janvier 2020, les abonnés français ont envoyé un total de 530 millions de SMS. Pour le réseau de Orange France, qui a environ 50% de part de marché, cela représente 265 millions de SMS à écouler.

Question 19. *En supposant que chaque SMS est encapsulé avec une entête de 32 octets, calculez le volume maximal total traité par le réseau Orange.*

Question 20. *En supposant que le quart de ce volume maximal total a été envoyé entre 0h et 2h du matin le premier janvier, évaluez le flux moyen que cela représente en entrée du serveur de SMS de Orange France.*

Mél

Pierre Dupont, enseignant à l'Ensimag (dont la boîte aux lettres est sur ensiens.imag.fr), connecté sur le webmail.ensimag.fr envoie un message à Roland.Groz@polytechnique.org . Cette adresse, sur un serveur à Palaiseau, renvoie (« forward ») tous les messages vers la boîte aux lettres de Roland Groz Roland.Groz@imag;fr sur le serveur heka.imag.fr (il n'y a pas de BAL pour R. Groz à Palaiseau). On supposera que le chemin entre l'Ensimag et Polytechnique est aussi direct que celui qui reliait l'Ensimag au GIPSA-lab dans l'exemple vu en TD.

Question 21. *Indiquez la succession des MTA impliqués dans le transfert du message.*

Question 22. *Quelles requêtes DNS successives doivent être faites pour assurer cet acheminement, et par quelles machines ?*

Ensimag 1^re^ année **2009-2010**

Probabilités Appliquées
Janvier 2010

Durée 2h00 - Documents et calculatrices non-autorisés. Veuillez inscrire votre numéro de groupe sur la copie.

Exercice I (6 pts) - Soit N une variable aléatoire de loi de Poisson de paramètre $\lambda > 0$ que l'on conditionne à rester strictement positive.

1-1pt) Montrer que
$$\forall n \geq 1\,, \quad \mathrm{P}(N = n) = \frac{1}{e^{\lambda} - 1} \frac{\lambda^n}{n!}\,.$$

2-1pt) Déterminer la fonction génératrice de la variable N.

3-2pt) Calculer $E[N]$ et $Var(N)$.

4-2pt) Soit M une variable aléatoire de même loi que N, indépendante de N. Déterminer la fonction génératrice de la variable
$$L = M + N.$$
En déduire la valeur de la probabilité $\mathrm{P}(L = \ell)$, pour tout $\ell \geq 2$.

Exercice II (6 pts) - Soit $(X_i)_{i\geq 0}$ une suite de variables aléatoires indépendantes de loi exponentielle de paramètre 1 et N une variable de loi de Poisson de paramètre λ indépendante des (X_i).

1-2pts) Soit $n \geq 0$, et
$$Z_n = \min(X_0, \ldots, X_n)$$
Pour tout $t > 0$, calculer la probabilité $\mathrm{P}(Z_n > t)$. En déduire que la variable aléatoire Z_n suit la loi exponentielle de paramètre $n + 1$.

2-1pt) On pose
$$Z_N = \min(X_0, \ldots, X_N)\,.$$
Pour tout $t > 0$, calculer la probabilité $\mathrm{P}(Z_N > t)$.

3-1pt) En déduire la densité de la variable Z_N.

4-1pt) En plus du générateur aléatoire de loi uniforme sur (0,1), on dispose d'un générateur de loi de Poisson de paramètre quelconque. Ecrire un algorithme de mélange pour simuler la loi obtenue à la question précédente

5-1pt) Calculer $E[Z_N]$.

Exercice III (4 pts) - Soient U et V des variables aléatoires indépendantes de loi uniforme sur $[0,1]$. On pose $Y_1 = V\mathbf{1}_{(U<V)}$.

1-1pt) Grâce à un argument de symétrie, montrer que

$$2E[Y_1] = E[\max(U,V)].$$

2-1pt) Calculer $E[Y_1]$.

3-1pt) Montrer que

$$E[\min(U^2,V^2)] + E[\max(U^2,V^2)] = \frac{2}{3}.$$

4-1pt) Soit $Y_2 = V\mathbf{1}_{(U<V)} + V^2\mathbf{1}_{(U\geq V)}$. Calculer $E[Y_2]$.

Exercice IV (4pts) - On considère la loi de fonction de répartition F définie ci-dessous

$$\forall t \in \mathbb{R}, \quad F(t) = \begin{cases} 0 & \text{si } t \leq 0\ ; \\ t^2 & \text{si } 0 \leq t < 1/2\ ; \\ 1 & \text{si } t \geq 1/2\ . \end{cases}$$

1-2pts) Ecrire un algorithme de simulation pour cette loi.

2-1pt) Prouver que votre algorithme est correct.

3-1pt) Calculer l'espérance de cette loi.

Théorie des langages 1

Durée : 3h.
Documents : tous documents autorisés.

Nb : le barème est donné à titre indicatif; la clarté de la rédaction sera prise en compte dans la notation.

Exercice 1 - Transformations d'automates (4 points)

Soit $Q = \{q_0, \ldots, q_9\}$. On considère l'automate $A = (Q, \{a, b\}, \{q_0\}, \delta, \{q_6, q_7\})$, où la fonction de transition δ est définie dans le tableau ci-dessous :

δ	a	b	ε
q_0	q_0, q_1	q_5	-
q_1	-	-	q_2, q_3
q_2	q_3	-	-
q_3	q_4	-	-
q_4	q_0	-	-
q_5	q_8	q_6	-
q_6	q_8	q_7	-
q_7	-	q_6	-
q_8	q_8	q_9	-
q_9	q_8	-	-

▷ **Question 1** (3 points) Construire un automate minimal équivalent à A, en développant les étapes de la construction.

▷ **Question 2** (1 point) Donner une expression régulière correspondant à l'automate obtenu.

Exercice 2 - Expressions de types (9 points)

On s'intéresse aux expressions de type d'un langage inspiré d'ADA. On considère d'abord la grammaire G suivante :

$$
\begin{array}{rcl}
\texttt{decl\_type} & \rightarrow & \texttt{type\_simple} \mid \texttt{type\_tab} \\
\texttt{type\_tab} & \rightarrow & \textbf{array(}\texttt{contenu}\textbf{) of } \texttt{type\_simple} \\
\texttt{contenu} & \rightarrow & \texttt{elements bornes} \mid \texttt{bornes} \\
\texttt{elements} & \rightarrow & \texttt{type\_simple } \textbf{range} \\
\texttt{type\_simple} & \rightarrow & \textbf{integer} \mid \textbf{character} \\
\texttt{bornes} & \rightarrow & \textbf{cst .. cst}
\end{array}
$$

Cette grammaire est définie sur les vocabulaires terminaux et non-terminaux suivants :

$$
\begin{array}{rcl}
V_T & = & \{\textbf{array}, \textbf{)}, \textbf{(}, \textbf{of}, \textbf{range}, \textbf{integer}, \textbf{character}, \textbf{cst}, \textbf{..}\}, \\
V_N & = & \{\texttt{decl\_type}, \texttt{type\_tab}, \texttt{contenu}, \texttt{elements}, \texttt{type\_simple}, \texttt{bornes}\},
\end{array}
$$

et a pour axiome le non-terminal `decl_type`.

▷ **Question 1** (2 points) Tracer l'arbre de dérivation engendrant

array(integer range cst .. cst) of character

Prouver que $\mathcal{L}(G)$, le langage engendré par la grammaire G, est régulier.

Dans ce nouveau langage, on veut maintenant pouvoir déclarer des tableaux dont les éléments peuvent également être des tableaux (et non plus seulement des types simples comme en ADA). On cherche donc à construire une nouvelle grammaire G' qui permet d'engendrer

array(cst..cst) of array(character range cst..cst) of integer,

c'est-à-dire un tableau dont les éléments sont des tableaux d'entiers.

▷ **Question 2** (2,5 points) Modifier la grammaire G pour obtenir la grammaire G'. De quel type est le langage $\mathcal{L}(G')$? Justifier.

On souhaite étendre la nouvelle grammaire G' afin de pouvoir déclarer comme types les types simples, les tableaux et les enregistrements. Cette nouvelle grammaire a donc comme vocabulaires terminaux et non-terminaux

$$\begin{aligned} V'_T &= V_T \cup \{\textbf{record}, \textbf{end}, \textbf{idf}\}, \\ V'_N &= V_N \cup \{\texttt{type\_enr}, \texttt{liste\_champ}, \texttt{champ}\}, \end{aligned}$$

et contient les règles supplémentaires suivantes :

$$\begin{aligned} \texttt{type\_enr} &\rightarrow \textbf{record}\ \texttt{liste\_champ}\ \textbf{end} \\ \texttt{liste\_champ} &\rightarrow \texttt{champ;}\ \texttt{liste\_champ} \mid \varepsilon \\ \texttt{champ} &\rightarrow \textbf{idf :}\ \texttt{decl\_type} \end{aligned}$$

Les règles correspondant au non-terminal `decl_type` deviennent

$$\texttt{decl\_type} \rightarrow \texttt{type\_simple} \mid \texttt{type\_tab} \mid \texttt{type\_enr}$$

▷ **Question 3** (1,5 point) Quelle(s) règle(s) doit-on modifier pour garantir qu'un enregistrement contient au moins un champ ?

▷ **Question 4** (3 points) Le langage engendré par cette nouvelle grammaire est-il régulier ? Justifier (on pourra admettre que le langage $\{a^n b^n \mid n \geq 1\}$ n'est pas régulier).

Exercice 3 - Opérations sur les langages (7 points)

Soit $A = (Q, V, \delta, I, F)$ un automate **sans ε-transitions.** On considère l'automate $\mathcal{I}(A) = (Q, V, \delta', I, F)$, où la fonction de transition δ' est définie pour tout $q \in Q$ et pour tout $a \in V$ par :

$$\delta'(q, a) = \{q' \in Q \mid \exists q_1 \in \delta(q, a),\ \exists a' \in V :\ q' \in \delta(q_1, a')\}.$$

▷ **Question 1** (1 point) Soient $Q = \{q_0, \ldots, q_3\}$ et $A = (Q, \{a, b\}, \{q_0\}, \delta, \{q_2\})$, où la fonction de transition δ est définie dans le tableau ci-dessous :

δ	a	b
q_0	q_1	$-$
q_1	q_1	q_2
q_2	q_3	q_3
q_3	$-$	q_2

Construire l'automate $\mathcal{I}(A)$. Quelles sont les expressions régulières correspondant à A et à $\mathcal{I}(A)$?

▷ **Question 2** (1,5 points) On considère maintenant un automate A sans ε-transitions, et l'automate $\mathcal{I}(A)$ correspondant. Soit $(p_0, v_1, p_1) \cdots (p_{n-1}, v_n, p_n)$ un chemin de trace $v_1 v_2 \cdots v_n$ dans $\mathcal{I}(A)$. Prouver par induction sur la longueur des chemins qu'il existe des symboles $v'_1, \ldots, v'_n \in V$ et des états $p'_1, \ldots, p'_n$ tels que

$$(p_0, v_1, p'_1)(p'_1, v'_1, p_1) \cdots (p_{n-1}, v_n, p'_n)(p'_n, v'_n, p_n)$$

est un chemin dans A.
On prêtera une attention particulière à la rédaction de la preuve par induction, et à la définition de l'hypothèse d'induction.

▷ **Question 3** (1,5 points) Soit L le langage reconnu par A, et soit L' le langage reconnu par $\mathcal{I}(A)$. Prouver que

$$L' \subseteq \{v_1 v_3 v_5 \ldots \cdots v_{2n-1} \in V^* \mid \exists v_2, v_4, \ldots, v_{2n} \in V : \; v_1 v_2 \cdots v_{2n-1} v_{2n} \in L\}.$$

▷ **Question 4** (3 points) Prouver l'inclusion réciproque.

MAI 2010

ALGORITHMIQUE 1 – 7 PGAES

ANALYSE APPLIQUEES – 2 PAGES

LOGICIEL DE BASE – 6 PAGES

LOGIQUE POUR L'INFORMATIQUE - 3 PGAES

METHODES NUMERIQUES - 2 PGAES

POLITIQUE ECONOMIQUE - 6 PGAES

PROBABILITES APPLIQUEES – 2 PAGES

RECHERCHE OPERATIONNELLE - 2 PGAES

THEORIE DES LANGUANGES 2 – 3 PAGES

TRAITEMENT du SIGNAL - 2 PGAES

TRANSMETTRE L'INFORMATION – 3 PGAES

Algorithmique et structures de données : examen de première session

Ensimag 1A

Année scolaire 2009–2010

Consignes générales :
- Durée : 3 heures.
- Calculatrices, portables et tous instruments électroniques interdits. Livres interdits. Autres documents autorisés.
- Le barème est donné à titre indicatif.
- Les trois exercices sont indépendants et peuvent être traités dans le désordre.
- La syntaxe Ada ne sera pas un critère déterminant de l'évaluation des copies. En d'autres termes, les correcteurs n'enlèveront pas de point pour une syntaxe Ada inexacte mais compréhensible (pensez à bien commenter votre code !).
- **Merci d'indiquer votre numéro de groupe de TD et de rendre votre copie dans le tas correspondant à votre groupe.**

Exercice 1 : Diviser pour régner *(7 pts)*

Question 1 ***(1,5 pts)*** On suppose disposer d'un type `Matrice` qui représente des matrices carrées d'ordre 2, et d'une fonction de multiplication de ces matrices appelée `Mult`.

```
type Matrice is array(1..2,1..2) of Natural ;

function Mult(M1,M2: Matrice) return Matrice ;
```

Écrire une fonction `Power` qui calcule M^N en appliquant un principe "*diviser pour régner*". Donner une approximation du coût asymptotique de votre algorithme en fonction de N. On attend ici une solution *asymptotiquement meilleure* que l'algorithme naïf itérant $N-1$ multiplications successives de M.

```
function Power(M: Matrice; N: Positive) return Matrice ;
```

Question 2 ***(3 pts)*** On définit la suite de Fibonacci $(F_n)_{n\in\mathbb{N}}$ par récurrence sur n de la manière suivante :
- $F_0 = 0$ et $F_1 = 1$.
- pour $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$.

Pour $n \geq 1$, on définit un vecteur V_n en dimension 2 par :

$$V_n = \begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix}$$

1. Établir une relation de récurrence multiplicative entre V_{n+1} et V_n.
2. En déduire un algorithme pour calculer F_n lorsque $n \geq 2$, basé sur la fonction `Power` de la question 1.
3. Quel est le nombre d'opérations élémentaires sur les entiers (addition et multiplications confondues) effectuées par cet algorithme ? Que pensez-vous de ce résultat ?

Question 3 ***(2,5 pts)*** En utilisant un principe "*diviser pour régner*", décrire un algorithme qui calcule simultanément le minimum et le maximum d'un ensemble non ordonné à n éléments avec $n \geq 1$, et dont le nombre de comparaisons soit majoré par

$$\left\lceil \frac{3 \times 2^{\lceil \log_2(n) \rceil}}{2} \right\rceil - 2$$

Attention, on demande ici un **majorant exact et pas une approximation asymptotique**. Justifier soigneusement le calcul de coût. Comparer ce résultat au coût de l'algorithme itératif naïf.

Exercice 2 : Complexité en moyenne *(2 pts)*

Question 4 ***(2 pts)*** On considère le problème du tri par ordre croissant d'un tableau `T` d'éléments 2 à 2 **distincts** de taille N. Il existe donc une unique permutation qui permet de trier le tableau. On considère que chaque permutation de $[1..N]$ a la même probabilité d'être la permutation du tri. Calculer exactement le nombre **moyen** d'appels à la procédure `Swap` dans l'algorithme `BubbleSort` ci-dessous.

```
type Tab is array(1..N) of Element ;

procedure Swap(X,Y: in out Element) is
   Aux: Element := X ;
begin
   X:=Y ;
   Y:=Aux ;
end ;

procedure BubbleSort(T: in out Tab) is
   Swapped: Boolean ;
   I: Natural := N ;
begin
   loop
      Swapped:=False ;
      for J in 1..I-1 loop
         if T(J) > T(J+1) then
            Swap(T(J),T(J+1)) ;
            Swapped := True ;
         end if ;
      end loop ;
      exit when not Swapped ;
      I:=I-1 ;
   end loop ;
end ;
```

Exercice 3 : Arbres binaires *(11 pts)*

Dans l'ensemble de cet exercice, si A est un arbre binaire, on note $\mathcal{N}(A)$ son nombre de nœuds et $\mathcal{H}(A)$ sa hauteur. Par convention, si A est l'arbre vide, alors $\mathcal{N}(A) = 0$ et $\mathcal{H}(A) = -1$.

1 Étude de formes d'arbres binaires

Définition 1 (arbre optimal) *Un arbre binaire A est dit* optimal *ssi :*

$\mathcal{H}(A) \geq 0$ *implique* $\mathcal{N}(A) \geq 2^{\mathcal{H}(A)}$

Question 5 ***(1 pts)*** Montrer que si A est un arbre non vide qui satisfait la définition 1 alors

$$\mathcal{H}(A) = \lfloor \log_2 \mathcal{N}(A) \rfloor$$

Expliquer la définition ci-dessus : en quel sens un arbre qui satisfait la définition 1 est optimal ?

Définition 2 (arbre quasi-complet) *Un arbre binaire A est dit* quasi-complet *ssi il satisfait la définition inductive suivante :*
- *A est l'arbre vide*
- *ou, A est non vide et ses deux fils F_1 et F_2 sont quasi-complets tels que :*

$$|\mathcal{N}(F_1) - \mathcal{N}(F_2)| \leq 1$$

Question 6 ***(1,5 pts)*** Est-ce que A quasi-complet implique A optimal ? Inversement, est-ce que A optimal implique A quasi-complet ? Justifiez vos réponses.

2 Étude du coût de l'union d'ABR

On suppose donné un type `Element` muni d'une relation d'ordre total. On définit ici un type `OrdSet` pour représenter des ensembles finis dont les éléments sont des valeurs de `Element`. Dans cet exercice, une valeur de type `OrdSet` est un couple formé d'un arbre binaire de recherche de type `ABR` (pas forcément équilibré) contenant les éléments de l'ensemble, et du nombre d'éléments dans l'ensemble. Ce type `ABR` est défini classiquement comme un type pointeur sur des cellules de type `Noeud`. L'arbre vide (*i.e.* l'ensemble vide) est représenté par le pointeur `null`.

```
type Noeud ;
type ABR is access Noeud ;

type OrdSet is record
  A: ABR ;
  Card: Natural ;
end record ;

type Noeud is record
```

Question 9 *(2 pts)* Proposer un algorithme pour programmer la fonction `Union` qui construit l'union ensembliste de `S1` et `S2` avec un nombre de comparaisons *dans le pire cas* linéaire en fonction `S1.Card+S2.Card`.

```
function Union(S1,S2: OrdSet) return OrdSet ;
```

On ne demande pas de programmer précisément cette fonction, mais de bien décrire son principe et le calcul de son coût.

Question 10 *(0,5 pts)* On veut maintenant étendre la structure de donnée `ABR` de manière à ce que les arbres construits soient des ABR équilibrés AVL. Décrire (sans les programmer) les modifications qu'il faut apporter sur la structure de données `Noeud` et aux procédures et fonctions précédentes. Est-ce que les coûts de `Import` et `Union` sont modifiés ?

Question 11 *(1,5 pts)* On considère la procédure `UnionEnPlace` ci-dessous : elle place dans `S1` l'union de `S1` et `S2`, et détruit tous les éléments de `S2`.

```
procedure UnionEnPlace(S1, S2: in out OrdSet) is
   Aux: OrdSet ;
   X: Element ;
   DejaPresent: Boolean ;
begin
   if S1.Card < S2.Card then
      -- echange de S1 et S2.
      Aux:=S1;
      S1:=S2 ;
      S2:=Aux ;
   end if ;
   -- S1.Card >= S2.Card ;
   while S2.A /= null loop
      RemoveMin(S2.A,X) ;
      Insert(S1.A,X,DejaPresent) ;
      if not DejaPresent then
         S1.Card := S1.Card+1 ;
      end if ;
   end loop ;
   S2.Card := 0 ;
end ;
```

où les procédures suivantes sont supposées fournies et implémentées de manière efficace sur des AVLs :

```
procedure RemoveMin(A: in out ABR; Min: out Element) ;
-- requiert: A AVL qcq non nul.
-- garantit: Min minimum de l'ABR initial, supprimé de A.

procedure Insert(A: in out ABR; X: Element; DP: out Boolean) ;
-- requiert: A AVL qcq.
-- garantit:
```

```
  Val: Element ;
  FilsG, FilsD: ABR ;
end record ;
```

Dans le type `Noeud` ci-dessus :
- le champ `Val` contient l'élément à la racine de l'arbre
- le champ `FilsG` représente l'ensemble des éléments strictement inférieurs à `Val`
- le champ `FilsD` représente l'ensemble des éléments strictement supérieurs à `Val`

Ainsi, la fonction `NbElems` qui retourne le nombre d'éléments que contient un ensemble de type `ABR` est définie par :

```
function NbElems(A: ABR) return Natural is
begin
   if A=null then
      return 0 ;
   else
      return NbElems(A.FilsG)+NbElems(A.FilsD)+1 ;
   end if ;
end ;
```

Finalement, une valeur de type `S: OrdSet` vérifie l'invariant suivant : `S.Card=NbElems(S.A)` (`S.A` peut être nul).

Hypothèse 1 (coût des comparaisons) *Pour simplifier les analyses de coût, on suppose que le coût d'une* comparaison sur le type `Element` *est du même ordre que le coût d'une* comparaison sur les `Integer`, *lui-même du même ordre que celui d'une* comparaison de pointeurs.

Hypothèse 2 *Dans toute la suite, on fait l'hypothèse que le passage d'une tranche de tableau en paramètre d'entrée (en mode* `in`, `out` *ou* `in out`*) d'une procédure ou d'une fonction Ada se fait à coût constant (indépendant de la taille du tableau ou de la tranche).*[1]

Question 7 ***(2 pts)*** Écrire en Ada une fonction `Export` qui prend un `OrdSet` `S` et retourne l'ensemble des éléments de `S` sous la forme d'un tableau strictement croissant du type `Tab`.

```
type Tab is array(Positive range <>) of Element ;
function Export(S:OrdSet) return Tab ;
```

On demande d'écrire une implémentation qui fait un nombre de comparaisons linéaire en fonction de `Card(S)`. Justifier que votre implémentation vérifie cette propriété.

Question 8 ***(2,5 pts)*** Écrire en Ada une fonction `Import` qui réalise l'inverse de `Export` : sous la précondition que le tableau `T` est trié par ordre strictement croissant, elle retourne un `OrdSet` formé de l'ensemble des éléments de `T`. Il faut implémenter cette fonction de manière à ce que l'arbre retourné soit *quasi-complet* (voir définition 2).

```
function Import(T:Tab) return OrdSet ;
-- requiert T strictement croissant
```

Justifier que l'arbre retourné est quasi-complet. Calculer le nombre de comparaisons (d'entiers) effectuées en fonction de `T'Length`.

[1] Cette hypothèse semble vérifiée par le compilateur GNAT.

```
--    DP=True => X absent initialement dans A, et ajouté dans l'AVL
--    DP=False => X présent initialement dans A, et AVL non modifié
```

En supposant que la fonction `Union` travaille sur des AVL comme à la question 10, comparer cette fonction `Union` et la procédure `UnionEnPlace` ci-dessus pour le nombre de comparaisons dans le pire cas : préciser dans quelles conditions, le coût de quel algorithme est meilleur que l'autre (asymptotiquement).

Ensimag 1ère année **Mai 2010**
Analyse appliquée
Cours et TD autorisés
Durée : 3h00

Exercice 1 Soient E et F des espaces vectoriels normés et $T : E \to F$ une application linéaire.

1) Montrer que (i) est équivalent à (ii):

(i) pour tout sous ensemble borné B de E, $\overline{T(B)}$ est compact;

(ii) pour toute suite bornée $(x_n)_{n\in\mathbb{N}}$ de E, il existe une suite extraite $(x_{n_k})_{k\in\mathbb{N}}$ telle que la suite $(T(x_{n_k}))_{k\in\mathbb{N}}$ converge dans F.

2) On suppose que T vérifie (i). Montrer que T est continue.

Exercice 2 Soit E un espace vectoriel normé. Si $A \subset E$, $A \neq \emptyset$ on pose

$$f(x) = \inf_{a\in A} ||x-a||\,, \quad x \in E\,.$$

1) Montrer que

$$|f(x) - f(y)| \leq ||x-y|| \quad \forall\, x,\, y \in E\,.$$

2) On suppose que $A \subset E$, $A \neq \emptyset$ est fermé. Soit $B \subset E$ un compact tel que $A \cap B = \emptyset$. Montrer que

$$\inf_{b\in B} f(b) > 0\,.$$

Exercice 3 Soit H un espace de Hilbert sur $\mathbb{R}$. On définit $f : H \times H \to \mathbb{R}$ par $f(x,y) = |<x,y>|$. Etudier la différentiabilité de f.

Problème Soit H un espace de Hilbert sur $\mathbb{R}$. Soit $B : H \times H \to \mathbb{R}$ une application bilinéaire vérifiant les deux conditions suivantes :

i) B est continue, c'est à dire : il existe une constante $C > 0$ telle que

$$|B(x,y)| \leq C||x||\,||y|| \quad \forall\, x,\, y \in H\,;$$

ii) il existe une constante $\alpha \in]0, C[$ telle que

$$B(x,x) \geq \alpha ||x||^2 \quad \forall\, x \in H\,.$$

Soit $K \subset H$ un convexe, fermé et non vide. Si $L \in \mathcal{L}(H, \mathbb{R}) = H'$, on veut montrer le résultat suivant :

$$\exists!\, x \in K \text{ tel que } B(x, y - x) \geq L(y - x) \quad \forall\, y \in K\,. \tag{1}$$

1) Montrer qu'il existe $z \in H$ unique tel que

$$L(y) = <y, z> \quad \forall\, y \in H\,.$$

2) Montrer que pour tout $x \in H$ fixé, il existe un élément de H noté $T(x)$ tel que

$$B(x, y) = <y, T(x)> \quad \forall\, y \in H\,.$$

2a) Montrer que $x \to T(x)$ définit bien une application de H dans H.

2b) Montrer que T est linéaire.

2c) Montrer que T est continue et que

$$||T(x)|| \leq C||x|| \quad \forall\, x \in H\,,$$

où C désigne la constante de i).

2d) Montrer que

$$<x, T(x)> \geq \alpha ||x||^2 \quad \forall\, x \in H\,.$$

3) Montrer que (1) est équivalent à

$$\exists!\, x \in K \text{ tel que } <y - x, T(x)> \geq <y - x, z> \quad \forall\, y \in K\,. \tag{2}$$

4) Soit $\theta > 0$. Montrer que (2) est équivalent à

$$\exists!\, x \in K \text{ tel que } <\theta z - \theta T(x), y - x> \leq 0 \quad \forall\, y \in K\,. \tag{3}$$

5) Pour tout $y \in K$, on pose $S(y) = P_K(\theta z - \theta T(y) + y)$, où P_K désigne l'application projection de H sur K. Montrer qu'on peut choisir $\theta > 0$ de telle sorte qu'il existe $k(\theta) \in]0, 1[$ tel que

$$||S(y_1) - S(y_2)|| \leq k(\theta)||y_1 - y_2|| \quad \forall\, y_1\,, y_2 \in K\,.$$

En déduire que, si θ est ainsi choisi, il existe $x \in K$ unique tel que $S(x) = x$.

6) Conclure.

7) Supposons qu'on cherche à calculer x par une méthode itérative. Quel est le choix optimal de θ permettant d'accélérer la convergence des itérations?

Logiciel de Base : examen de première session

ENSIMAG 1A

Année scolaire 2009–2010

Consignes générales :
- Durée : 2h. Tous documents et calculatrices autorisés.
- Le barème est donné à titre indicatif.
- Les exercices sont indépendants et peuvent être traités dans le désordre.
- **Merci d'indiquer votre numéro de groupe de TD et de rendre votre copie dans le tas correspondant à votre groupe.**

Consignes relatives à l'écriture de code C et assembleur Pentium :
- Pour chaque question, une partie des points sera affectée à la clarté du code et au respect des consignes ci-dessous.
- Pour les questions portant sur la traduction d'une fonction C en assembleur, on demande d'indiquer en commentaire chaque ligne du programme C original avant d'écrire les instructions assembleur correspondantes.
- Pour améliorer la lisibilité du code assembleur, on utilisera systématiquement des constantes (directives `.set` ou `.equ`) pour les déplacements relatifs à `%ebp` (*i.e.* paramètres des fonctions et variables locales). Par exemple, si une variable locale s'appelle `var` en langage C, on y fera référence avec `var(%ebp)`.
- Sauf indication contraire dans l'énoncé, on demande de traduire le code C en assembleur de façon systématique, sans chercher à faire la moindre optimisation : en particulier, **on stockera les variables locales dans la pile** (pas dans des registres), comme le fait le compilateur C par défaut.
- On respectera les conventions de gestions des registres Intel vues en cours, c'est à dire :
  - `%eax`, `%ecx` et `%edx` sont des registres *scratch* ;
  - `%ebx`, `%esi` et `%edi` ne sont pas des registres *scratch*.

Ex. 1 : Exercice préliminaire *(6 pts)*

Il est fortement recommandé de commencer par traiter cet exercice qui porte sur des notions utiles dans la suite de l'examen.

Soit la fonction C suivante :

```
void echanger(int *a, int *b)
{
    int t = *a;
    *a = *b;
    *b = t;
}
```

Question 1 Traduisez cette fonction en assembleur sans chercher à l'optimiser.

On rappelle que l'instruction `enter $8, $0` est exactement équivalente à la séquence d'instructions :

```
pushl %ebp
movl %esp, %ebp
subl $8, %esp
```

Soit le type de données structuré et la fonction assembleur suivante :

```
    .text
/*
struct rationnel_t {
    int n;
    unsigned d;
};
*/
    .set n, 0
    .set d, 4
// void afficher_rat_asm(struct rationnel_t q)
    .globl afficher_rat_asm
afficher_rat_asm:
    .set q, 8
    enter $0, $0
    //
    cmpl $0, (q + d)(%ebp)
    jne else
    //
    pushl $ch_err
    call printf
    addl $4, %esp
    jmp fin
else:
    //
    pushl (q + d)(%ebp)
    pushl (q + n)(%ebp)
    pushl $ch_aff
    call printf
    addl $12, %esp
```

```
30  fin:
31      leave
32      ret
33      .data
34  ch_err:     .asciz "Erreur !\n"
35  ch_aff:     .asciz "q = %d / %u\n"
36
```

Question 2 Donner le code C correspondant à cette fonction assembleur. Pour vous aider, on a laissé les // dans le code fourni (lignes 15, 18 et 24) qui délimitent la traduction assembleur de chaque ligne de C originale.

Soit maintenant la fonction C suivante, qui se base sur le même type de données que précédemment :

```
void afficher_ptr_rat(struct rationnel_t *q)
{
    if (q->d == 0) {
        printf("Erreur !\n");
    } else {
        printf("q = %d / %u\n", q->n, q->d);
    }
}
```

Question 3 Traduire cette fonction en assembleur sans chercher à l'optimiser (on demande une traduction littérale du code C donné, vous n'avez notamment pas le droit d'appeler le code assembleur de la question précédente, mais vous pouvez bien sûr vous en inspirer).

Ex. 2 : Tableaux et listes chaînées *(6 pts)*

Dans cet exercice, on va travailler sur une fonction

`void decouper(int tab[], struct cellule_t **liste)`

qui prend en entrée un tableau d'entiers signés et construit en sortie une liste chaînée composée de certains des entiers du tableau initial.

Les cellules de la liste chaînée produite en sortie seront du type :

```
struct cellule_t {
  int val;
  struct cellule_t *suiv;
};
```

Soit alors une réalisation en assembleur de cette fonction :

```
    .text
/*
struct cellule_t {
    int val;
    struct cellule_t *suiv;
```

```
};
*/
    .set val, 0
    .set suiv, 4
// void decouper_asm(int tab[], struct cellule_t **liste);
    .globl decouper_asm
decouper_asm:
    .set tab, 8
    .set liste, 12
    enter $16, $0
    .set sent, -8
    .set queue, -12
    .set i, -16
    //
    leal sent(%ebp), %eax // equivalent a : movl %ebp, %eax ; addl $x, %eax
    movl %eax, queue(%ebp)
    //
    movl $0, i(%ebp)
for:
    //
    movl tab(%ebp), %eax
    movl i(%ebp), %ecx
    cmpl $0, (%eax, %ecx, 4)
    je fin_for
    //
    movl tab(%ebp), %eax
    movl i(%ebp), %ecx
    cmpl $0, (%eax, %ecx, 4)
    jng else
    //
    pushl $8
    call malloc
    addl $4, %esp
    movl queue(%ebp), %edx
    movl %eax, suiv(%edx)
    //
    movl queue(%ebp), %eax
    movl suiv(%eax), %eax
    movl %eax, queue(%ebp)
    //
    movl tab(%ebp), %eax
    movl i(%ebp), %ecx
    movl (%eax, %ecx, 4), %eax
    movl queue(%ebp), %edx
    movl %eax, val(%edx)
    //
    movl queue(%ebp), %eax
    movl $0, suiv(%eax)
else:
    //
    addl $1, i(%ebp)
    jmp for
fin_for:
    //
    movl queue(%ebp), %eax
    movl $0, suiv(%eax)
```

```
        //
        movl (sent+suiv)(%ebp), %eax
        movl liste(%ebp), %edx
        movl %eax, (%edx)
        leave
        ret
```

Question 1 Expliquer en 5 lignes maximum ce que fait la fonction `decouper_asm`.

Question 2 Donner le code C équivalent à la fonction `decouper_asm`. N'oubliez pas les déclarations des variables utilisées.

Question 3 Ecrire une implantation assembleur optimisée de la fonction `decouper_asm`. On cherchera donc à minimiser le nombre d'accès mémoire en utilisant au mieux les registres disponibles.

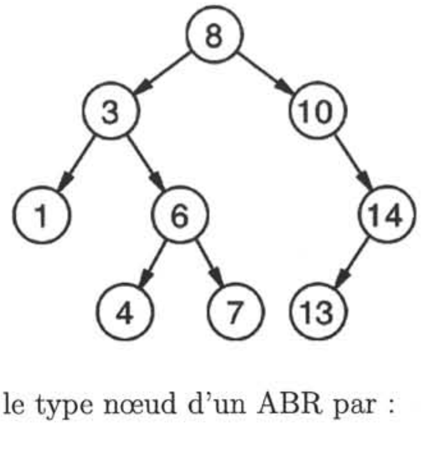
Ex. 3 : Arbres binaires de recherche *(8 pts)*

On va travailler dans cet exercice sur des arbres binaires de recherche (ABR) dont les nœuds contiennent une simple valeur entière (naturelle). On rappelle les principales propriétés de cette structure de données :

- chaque nœud de l'arbre a au plus 2 fils ;
- le sous-arbre gauche d'un nœud N donné ne contient que des nœuds dont la valeur est strictement inférieure à la valeur de N ;
- le sous-arbre droit d'un nœud N donné ne contient que des nœuds dont la valeur est strictement supérieure à la valeur de N ;
- les sous-arbres gauche et droit d'un nœud donné sont aussi des arbres binaires de recherche.

Il vient naturellement de ses propriétés que chaque clé est unique (*i.e.* il n'existe pas plusieurs nœuds de même valeur).

La figure ci-dessous représente un arbre binaire de recherche typique :



En C, on défini simplement le type nœud d'un ABR par :

```
struct noeud_t {
    unsigned val;
    struct noeud_t *fg;
    struct noeud_t *fd;
};
```

Question 1 Ecrire en C une fonction **récursive**

```
int rechercher(unsigned v, struct noeud_t *arbre)
```

qui renvoie faux (*i.e.* 0) si la valeur `v` n'est pas présente dans l'ABR `arbre` et vrai (*i.e.* 1) sinon.

Question 2 Traduire cette fonction de recherche en assembleur sans chercher à l'optimiser.

Soit la fonction C suivante :

```
void arbre_vers_tab(struct noeud_t *arbre, unsigned *tab[])
{
    if (arbre != NULL) {
        arbre_vers_tab(arbre->fg, tab);
        **tab = arbre->val;
        (*tab)++;
        arbre_vers_tab(arbre->fd, tab);
    }
}
```

Question 3 Traduire cette fonction en assembleur sans chercher à l'optimiser.

Question 4 Décrire en 2 lignes maximum le contenu du tableau `tab` en sortie de cette fonction.

EXAMEN DE "LOGIQUE POUR L'INFORMATIQUE"

Ensimag 1ère Année - mai 2010

DURÉE: 3 heures.

SEULS DOCUMENTS AUTORISÉS: Poly "Une approche simple de la logique pour l'Informatique et la formalisation de l'inférence" et notes personnelles.

- S.V.P. **avant de commencer** lisez les 7 points ci-dessous.

- Cet examen comporte 9 questions. Toutes les questions sont indépendantes.
- L'ordre des questions n'a, en principe, aucun rapport avec leur difficulté.
- ***S.V.P.** rédigez vos réponses de façon similaire à celle des corrigés du poly.*
- S.V.P. **séparez** (par une ligne horizontale de la largeur de la feuille) et **numérotez clairement** les réponses aux différentes questions.
- **S.V.P.** respectez le nombre de lignes suggérées pour les réponses.
- Les réponses à l'aide des méthodes ***autres*** que celles demandées explicitement seront ***ignorées***.
- On tiendra compte dans la correction de la clarté, de la simplicité et de la concision des réponses ainsi que de la qualité de la présentation.

Question 1

Dans la fbf ($\star$) ci-dessous $P[x]$, $\psi[x]$ et $\phi[x]$ dénotent respectivement une formule atomique et des fbf dont 'x' est la seule variable.

($\star$) $\forall x(\psi[x] \vee \phi[x]) \Rightarrow \forall x(\ (P[x] \vee \psi[x]) \wedge (\neg P[x] \vee \phi[x])\ )$

Est-elle valide ?

Justifiez votre réponse en utilisant la méthode des tableaux sémantiques.

Question 2

Étant donné le schéma propositionnel suivant

$P_1 \wedge (\bigwedge_{i=1}^{n} P_i \Leftrightarrow P_{i+2}) \wedge \neg P_n$

Fixer $n = 2$

i) Dire s'il est sat. ***En utilisant la méthode des tableaux***.

ii) Dire s'-il est valide. ***En utilisant la réponse donnée en i).***

Question 3

Soit le raisonnement formalisé en L1O suivant ('x' et 'y' dénotent, comme d'habitude, des variables) :

1. $\forall x \exists y Q(x,y)$
2. $\forall x \forall y. Q(x,y) \Rightarrow P(x,y)$

---

3. $\forall x \exists y P(x,y)$

Pouvez-vous prouver que ce raisonnement est correct (ou incorrect) ***en utilisant la méthode des tableaux sémantiques avec unification ?***

Question 4

Monsieur (ou Mme) A propose

pour prouver que le raisonnement de la question 3 est correct (ou incorrect) ***en utilisant la méthode de résolution*** :

Skolémiser la conclusion, puis nier le résultat, ce qui donne disons, la formule $3'$. et après essayer de prouver l'insatisfaisabilité de l'ensemble de clauses obtenu à partir de 1., 2. et $3'$. *Appliquer cette façon de faire.*

et **Monsieur (ou Mme) B** propose :

Nier la conclusion, puis skolémiser le résultat, ce qui donne disons, la formule $3''$. et après essayer de prouver l'insatisfaisabilité de l'ensemble de clauses obtenu à partir de 1., 2. et $3''$. *Appliquer cette façon de faire.*

Qui a raison ?

a) Monsieur (ou Mme) A

b) Monsieur (ou Mme) B

c) Les deux

d) Aucun(e) des deux

Justifiez. Maxi 5 lignes.

Question 5

Rappel : Une *procédure de décision* pour une classe d'ensembles finis de clauses $\mathcal{C}$ est un algorithme permettant de décider (toujours) si $S \in \mathcal{C}$ est insatisfaisable ou satisfaisable.

L'assertion suivante est-elle **vraie** ou **fausse** ?

La méthode de résolution est une procédure de décision pour la classe de Herbrand (poly page 178).

Justifier (Maxi 3 lignes).

Question 6

On suppose qu'une liste L peut contenir deux occurrences du même élément.

Compléter (i.e. remplacer les ' ... ') le programme **prolog** définissant le prédicat:

`sans-dupl(L,M)` : la liste M contient tous les élements de la liste L, mais aucun de des éléments de M n'est dupliqué.

```
sans-dupl([ ], [ ]) → ;
sans-dupl([X | L ], M) → ...
sans-dupl([X | L ], [X | M ]) → ...
```

Question 7

Pouvez-vous donner une fbf **4-valide** de la L1O ?

Aide : Inspirez-vous de l'exemple 47, page 144 du poly

Question 8

On veut définir dans $\mathbb{N}$ *l'ensemble des nombres premiers* avec une fbf de la L1O.

Sachant que le domaine du discours est $D = \mathbb{N}$:

Compléter (i.e. remplacer les '...') l'expression ci-dessous (où 'x', 'y', 'z' dénotent des variables) pour la transformer en la fbf de la L1O demandée et donner la signification des symboles de fonction ('a', 'f' et 'g') et du symbole de prédicat ('E').

$\neg E(x, f(a)) \ \wedge \ \forall y \forall z \, [E(x, g(y, z)) \ \Rightarrow ...]$

Question 9

On dit qu'une formule atomique F est plus générale qu'une autre formule atomique G ssi il existe une substitution σ telle que $\sigma F = G$.

Étant donné les 6 formules atomiques ci-dessous :

1. $P(f(x, x))$
2. $P(f(x, f(x, f(x, x))))$
3. $P(f(x, f(x, f(y, y))))$
4. $P(f(x, f(y, f(y, x))))$
5. $P(f(f(x, y), f(y, x)))$
6. $P(f(f(f(x, y), x), y))$

identifier celles qui sont plus générales que la formule atomique ci-dessous ('a' dénote une constante) :

$P(f(a, f(a, f(a, a))))$

Fin-Examen

ENSIMAG 1ère année (2009-10) Méthodes Numériques

Examen du 12 Mai 2010

Durée : 3h.

Les seuls documents autorisés sont les notes du cours et des travaux dirigés de méthodes numériques. Les calculatrices sont interdites.

La qualité de la rédaction et de la présentation sera prise en compte pour l'évaluation des copies.

Exercice 1 Soit $A \in M_n(\mathbb{R})$ une matrice symétrique définie positive. On cherche à résoudre le système linéaire
$$A^2 x = b, \tag{1}$$
avec $b, x \in \mathbb{R}^n$.

1. On considère une première méthode qui consiste à calculer A^2 puis résoudre (1) par la méthode de Cholesky. Donner un équivalent du coût de cet algorithme (nombre d'opérations arithmétiques élémentaires) lorsque $n \to +\infty$.

2. Montrer qu'on peut résoudre (1) sans calculer A^2, en utilisant la factorisation de Cholesky de A. Donner un équivalent du coût de cet algorithme lorsque $n \to +\infty$, et comparer son efficacité à celle de la méthode précédente lorsque n est grand.

Exercice 2 On considère une fonction $f : \mathbb{R}^n \to \mathbb{R}$ de classe C^2. On suppose que f admet un minimum local en $\bar{x} \in \mathbb{R}^n$, et que les valeurs propres de la matrice hessienne $H_f(\bar{x})$ sont strictement positives. Pour calculer numériquement $\bar{x}$ on considère l'algorithme du gradient à pas fixe
$$x_{k+1} = x_k - \rho \nabla f(x_k), \tag{2}$$
où $\rho > 0$ est un paramètre. On note par la suite $N_\rho(x) = x - \rho \nabla f(x)$.

1. Exprimer la matrice jacobienne $DN_\rho(\bar{x})$ en fonction de $H_f(\bar{x})$.

2. Montrer que pour tout ρ assez petit, il existe $\eta > 0$ tel que si $\|x_0 - \bar{x}\| < \eta$ alors $x_k \to \bar{x}$ quand $k \to +\infty$.

Exercice 3 **sur les "splittings" de matrices**

Dans toute la suite, A est une matrice (n, n) inversible à éléments réels, et b est donné dans $\mathbb{R}^n$. Soit $M - N = A$ un splitting de A (M inversible) auquel est associée l'itération suivante pour résoudre $Ax = b$:
$$\begin{cases} x^0 \text{ donné dans } \mathbb{R}^n \\ x^{k+1} = Tx^k + h \end{cases} \qquad \text{avec } T = M^{-1}N \text{ et } h = M^{-1}b.$$

On supposera le splitting convergent ($\rho(T) < 1$).

1. Montrer que la donnée de A et T définit M et N de façon unique (on explicitera M et N en fonction de A et T).

2. On considère alors deux splittings de A:

$$A = M_1 - N_1 = M_2 - N_2 \qquad\qquad (M_1, M_2 \text{ inversibles })$$

et on pose: $T_1 = M_1^{-1}N_1$; $T_2 = M_2^{-1}N_2$; $h_1 = M_1^{-1}b$; $h_2 = M_2^{-1}b$.
On supposera les deux splittings convergents: $\rho(T_1) < 1$ et $\rho(T_2) < 1$.
Pour résoudre $Ax = b$, on considère l'itération:

$$(I) \qquad \begin{cases} x^0 \text{ donné dans } \mathbb{R}^n \\ M_1 x^{k+\frac{1}{2}} = N_1 x^k + b \qquad (\text{qui définit } x^{k+\frac{1}{2}}) \\ M_2 x^{k+1} = N_2 x^{k+\frac{1}{2}} + b \qquad (\text{qui définit } x^{k+1}). \end{cases}$$

Montrer que le passage de x^k à x^{k+1} s'exprime sous forme d'une itération linéaire:

$$x^{k+1} = Tx^k + h. \tag{$*$}$$

On explicitera T et h, ainsi qu'une condition nécessaire et suffisante pour la convergence de cette itération.

3. Montrer par un exemple (2,2) qu'il est possible d'avoir $\rho(T_1) < 1$, $\rho(T_2) < 1$ et $\rho(T) \geq 1$. Qu'en concluez-vous?

4. On note S la norme de matrice induite par une norme vectorielle de $\mathbb{R}^n$, notée $\|\ \|$:

$$S(Z) = \sup_{x \neq 0} \frac{\|Zx\|}{\|x\|}.$$

Monter que si $S(T_1) < 1$ et $S(T_2) < 1$ alors l'itération $(*)$ est convergente.

5. On va maintenant supposer que

$$\begin{cases} M_1 \text{ et } N_2 \text{ commutent} \quad (M_1 N_2 = N_2 M_1) \\ M_1 + N_2 \text{ est inversible} \end{cases}$$

et l'on reconsidère l'itération (I).
Montrer que l'on a:

$$(M_1 + N_2)^{-1} M_1 M_2 x^{k+1} = (M_1 + N_2)^{-1} N_2 N_1 x^k + b.$$

6. En déduire alors que l'itération (I) correspond à un splitting $A = M - N$ avec:

$$M = (M_1 + N_2)^{-1} M_1 M_2 \quad \text{ et } \quad N = (M_1 + N_2)^{-1} N_2 N_1.$$

Retrouver l'expression de $T = M^{-1}N$ en fonction de M_1, M_2, N_1, N_2.

ENSIMAG - 1ère année – 2009/2010

Politique Economique

DS du 19 mai 2010

Durée : 2 heures
Documents autorisés, sauf mobiles et portables

Travail demandé :

Choisissez un des deux textes joints et rédigez soit un **commentaire libre**, soit un « **commentaire dirigé** » en répondant aux questions posées à la fin du texte de votre choix.

Si vous choisissez le **commentaire libre**, vous le construirez de façon classique (introduction, développement, conclusion). Seule la forme de l'introduction vous est imposée. Il vous est demandé en effet de structurer l'introduction comme suit :

- Très brève synthèse du texte (thème, question posée, réponse, principal argument)
- Annonce justifiée de l'idée que vous allez développer et du plan que vous allez suivre.

N.B. : Veillez à faire ponctuellement référence au texte, à mobiliser vos connaissances, et à soigner la rédaction.

Texte 1

Du krach des tulipes à la bulle Internet

Pierre-Antoine Delhommais, Le Monde, 5 juillet 2002

« *Ce que nous savons avec certitude, c'est que les épisodes spéculatifs ne se terminent jamais en douceur. Il est sage de prédire le pire, même s'il est, selon la plupart des gens, peu probable.* » Tous ceux qui, pour de petites sommes ou des montants considérables, de façon directe ou indirecte, possèdent des actions, payent cher aujourd'hui la confirmation du jugement formulé par l'économiste américain John Kenneth Galbraith. Avec le « *techno-krach* », l'accélération de la chute des cours, les faillites, les fraudes, la panique, la colère des uns, les lamentations des autres, les pertes de tous, l'histoire financière est en passe de s'enrichir d'une de ces grandes catastrophes qui la rythment et la marquent, comme le firent au XVIIe celle des bulbes de tulipe en Hollande, au XVIIIe siècle la faillite de la South Sea Company, l'effondrement du système de John Law, ou, plus près de nous, les crises boursières de 1929 ou de 1987. Rien de vraiment neuf, sur terre, en matière de folie spéculative. Ce qui nous apparaît comme une nouveauté effroyable s'est déjà produit des centaines de fois. Mêmes exagérations, même irrationalité, mêmes sanctions, mêmes désillusions.

Le premier point commun concerne les chiffres, faits pour donner le vertige. Par rapport à son sommet atteint au mois de mars 2000, la Bourse américaine du Nasdaq, qui accueille les entreprises high-tech, a perdu les trois quarts de sa valeur : autrement dit, ce sont, sur ce seul marché, près de 5 000 milliards de dollars qui sont partis en fumée, l'équivalent de trois fois le produit national brut de la France.

Deux remarques, toutefois, pour atténuer cette peur du vide et relativiser l'actuelle débâcle. La première est que ces chutes sont à la juste mesure des envolées qui avaient précédé. Le Nasdaq n'avait-il pas progressé de 150 % dans les dix-huit mois qui précédèrent le début de sa chute ? Pour la seule année 1999, le fabricant finlandais de téléphones Nokia, chouchou des boursicoteurs avisés, n'avait-il pas vu son cours grimper de 246 % ?
La seconde est qu'il y a eu dans le passé des surévaluations plus manifestes et des chutes plus lourdes encore qu'aujourd'hui. Au plus fort de la bulle spéculative, en 1636, un seul bulbe de tulipe valait un carrosse, deux chevaux et tout leur harnachement. Quant à l'indice Times de la Bourse de New York, qui valait 238 points en septembre 1929, il n'en valait plus que 36 en juin 1932. Pour prétendre atteindre pareille dégringolade, il faudrait que l'indice CAC 40 tombe jusqu'à 1 000 points ! Il reste de la marge.

Autre similitude avec les crises financières précédentes, la multiplication des scandales. Après l'annonce de la faillite frauduleuse du courtier en énergie Enron viennent d'être révélées les manipulations comptables de WorldCom, deuxième compagnie américaine de téléphonie longue distance. Et la liste n'est probablement pas close. Le krach de 1987 avait précipité la chute de l'établissement financier à la mode, Drexel Burnham Lambert, et de l'inventeur des junk-bonds, Michael Milken ; la crise de 1929 poussa au suicide Ivar Kreuger, homme d'affaires suédois réputé mais aussi faussaire d'obligations - on prit soin d'annoncer la nouvelle de sa mort après la fermeture de Wall Street - et conduisit à la prison de Sing Sing Richard Whitney, le président de la Bourse de New York. Tous ces scandales sont toutefois la conséquence de l'effondrement des cours et non l'inverse, comme on pourrait presque le penser aujourd'hui, comme l'avait noté dès la fin du XIXe siècle le financier britannique

Walter Bagehot : « *Chaque grande crise révèle les spéculations excessives de nombreuses maisons que personne ne soupçonnait auparavant.* » Et il ajoutait, de façon visionnaire compte tenu de la complicité des cabinets d'audit dans les affaires Enron et WorldCom : « *Une bonne chose dans les crises, c'est qu'elles révèlent ce que les vérificateurs ne réussissent pas à trouver.* »

La désignation de boucs émissaires est aussi une tradition bien établie et respectée lors de l'éclatement des bulles. En 1987, le krach fut pour partie imputé aux programmes automatiques d'achats ou de ventes des titres par ordinateur (program trading) - les autorités boursières instaurèrent par la suite, avec un succès très relatif, des coupe-circuits censés limiter leur impact. En 1929, les vendeurs à découvert se retrouvèrent sur le banc des accusés avant d'être blanchis par une enquête des autorités boursières new-yorkaises. Soixante-treize ans plus tard, Michel Bon, le président de France Télécom, et Jean-Marie Messier, celui de Vivendi Universal, s'en prennent à leur tour, sans originalité, à cette technique, et dénoncent les impopulaires « *hedge funds* » qui la pratiquent assidûment. Il s'agit avant tout pour ces dirigeants, aux stratégies industrielles contestées, de détourner la fureur d'actionnaires individuels qui cherchent de leur côté à se présenter, sans vraiment convaincre, comme les victimes innocentes d'un capitalisme cruel. Eux aussi sont à la recherche des coupables qui allègent dangereusement leur portefeuille boursier. Ils en trouvent chez les analystes de mauvais conseil. Sans doute, à la fin des années 1990, n'a-t-il pas manqué de gourous, grassement payés par des maisons de courtage aux profits directement indexés sur la hausse des Bourses, pour justifier l'envolée des cours et pour expliquer qu'une nouvelle ère, radieuse, s'était ouverte. Mais de telles erreurs – parfois proches de la campagne d'intoxication - sont une constante de l'histoire financière. La plus célèbre reste celle d'Irving Fisher, professeur à l'université de Yale et surnommé par certains l'Einstein de l'économie en raison de ses découvertes sur la masse monétaire, qui, en septembre 1929, crut pouvoir affirmer : « *Le prix des actions a atteint ce qui paraît être un haut plateau permanent.* »

Mais il ne manquait pas non plus de voix pour mettre en garde, en dépit des innovations technologiques, contre « *l'exubérance irrationnelle* » des Bourses, selon la formule du président de la Réserve fédérale américaine, Alan Greenspan, ou pour dénoncer, chez de nombreux chefs d'entreprise, une frénésie de fusions et d'acquisitions, au prix d'un endettement colossal. Mais tous ceux qui lancèrent des appels à la prudence furent taxés, par les mêmes qui aujourd'hui se plaignent de ne pas avoir été avertis, d'incompétence et d'obscurantisme. Comme ils l'ont toujours été à chaque grande crise spéculative. « *Il n'y a aucune chance, tandis que la Bourse s'approche de l'abîme, analyse M. Galbraith, que ceux qui sont concernés s'aperçoivent de la nature de leur illusion et ainsi se protègent eux-mêmes et leur système. Les fous peuvent communiquer leur folie ; ils ne peuvent la percevoir et décider d'être raisonnables.* »

Pas plus aujourd'hui qu'hier, pas plus en 2002 qu'en 1929 ou au XVIIe siècle, il faut chercher ailleurs que dans le marché lui-même le responsable de ses propres égarements et ailleurs que dans l'irrationalité des investisseurs la cause des orgies spéculatives avec leurs lendemains nauséeux. Défendre l'utilité économique de la Bourse, ce n'est pas pour autant l'absoudre de tous ses péchés. « *Je peux mesurer le mouvement des corps, mais je ne peux pas mesurer la folie des hommes.* » Isaac Newton, qui venait de perdre 20 000 livres sterling lors de l'éclatement de la bulle boursière sur la South Sea Company, ne pouvait dire plus juste.

Questions sur le texte 1 :

1) Rédigez une brève **synthèse** du texte en respectant la forme suivante :

-Thème général de l'article
- Question posée
- Réponse
- Principaux arguments (les lister uniquement)

Rappel : dans la synthèse, il ne faut mentionner aucune information « extérieure » au texte

2) Dans le dernier paragraphe du texte, on lit la phrase suivante : *« Défendre l'utilité économique de la Bourse, ce n'est pas pour autant l'absoudre de tous ses péchés. »*.
Commentez (expliquez) le sens de cette proposition.

3) Le texte évoque à plusieurs reprises l'irrationalité des « marchés » (paragraphes 1 ;6 ;7).
Discutez la pertinence de l'emploi de ce « terme ».

4) Courte conclusion personnelle : apports et limites du texte.

N.B. :Veillez à mobiliser vos connaissances et à soigner la rédaction.

Texte 2

Paul Krugman, prix Nobel américain, vante la « *réussite économique* » de l'Europe.

Le Monde, 11 janvier 2010

Le prix Nobel américain d'économie Paul Krugman a vanté lundi « *la réussite économique de l'Europe* » pour contrer les arguments des parlementaires républicains convaincus que la réforme de la santé voulue par Barack Obama conduira au naufrage économique des Etats-Unis. Cette réforme vise à fournir une couverture santé à au moins 31 millions des 36 millions d'Américains qui en sont dépourvus.
« *La véritable leçon que nous donne l'Europe est en fait le contraire de ce que les conservateurs affirment : l'Europe est un succès économique, et ce succès montre que la social-démocratie fonctionne*, écrit M. Krugman dans sa chronique régulière publiée dans le New York Times. *Aux Américains qui l'ont visité : Paris avait-il l'air pauvre ou arriéré ? Et Londres, et Francfort ?* », demande M. Krugman, qui milite fermement en faveur de l'adoption aux Etats-Unis d'une réforme du système de santé qui rapprocherait ce dernier de celui de l'Europe, où « *les dépenses sociales sont largement plus* élevées ».
La croissance moyenne du PIB américain a atteint en moyenne 3 % par an depuis 1980 – « *Quand notre politique prenait un virement marqué à droite, contrairement à l'Europe* » –, contre 2,2 % pour les quinze pays de l'Union européenne d'avant l'élargissement de 2004, reconnaît M. Krugman. Cependant, « *depuis 1980, le PIB par tête – soit ce qui sert à mesurer le niveau de vie – a augmenté à peu près au même rythme en Amérique et dans l'UE : 1,95 % par an ici, contre 1,83 % là-bas* », ajoute-t-il.

L'économiste regrette « *le dogme économique en vigueur* » aux Etats-Unis, « *chez de nombreux démocrates aussi bien que chez tous les républicains, par essence* », selon lequel « *une social-démocratie à l'européenne ne peut être qu'un désastre absolu* ». En matière d'avancement technique, l'Internet à haut débit « *est tout aussi répandu en Europe qu'il l'est aux Etats-Unis, et il y est beaucoup plus rapide et bien moins cher* », ajoute M. Krugman, qui enseigne à l'université de Princeton, dans le New Jersey (dans le nord-est des Etats-Unis).
Si l'on peut considérer que l'Amérique fait mieux que l'Europe en matière d'emploi, avec un taux de chômage officiel habituellement bien plus faible, « *le taux d'emploi*[1] *chez les adultes de 25 à 54 ans était de 80 % dans l'Europe des Quinze [et de 83 % en France]. C'est à peu près la même chose qu'aux Etats-Unis* », écrit-il. « *Les Européens sont moins susceptibles que nous de travailler quand ils sont jeunes ou vieux, mais est-ce foncièrement une mauvaise chose ?* », ajoute son plaidoyer. D'autant que, selon lui, « *les Européens sont également plutôt productifs : ils font moins d'heures* ».

Dans une deuxième chronique intitulée « *Europe's OK; the Euro isn't* » (« *L'Europe va bien, pas l'euro* »), Paul Krugman nuance néanmoins son enthousiasme en soulignant qu'il a toujours considéré la création de l'Euro comme une « *erreur* ». « *Il est important de le rappeler car, si je pense que l'Europe s'en sort mieux que ce que les Américains imaginent, cela ne signifie pas que tout ce qu'ils font est bien* », conclut-il.

---

[1] Le taux d'emploi d'une population donnée s'exprime comme le rapport entre le nombre de personnes employées appartenant à cette population et le montant total de cette population.

Questions sur le texte 2 :

1) Rédigez une brève **synthèse** du texte en respectant la forme suivante :

-Thème général de l'article
- Question posée
- Réponse
- Principaux arguments (les lister uniquement)

Rappel : dans la synthèse, il ne faut mentionner aucune information « extérieure » au texte

2) Quels arguments pourriez-vous évoquer pour comprendre la proposition de Krugman selon laquelle la création de l'Euro fut « *une erreur* » ?

3) Que pourraient répondre, alors, les « Politiques » qui ont voulu et fait l'Euro ?

4) Courte conclusion personnelle : apports et limites du texte.

N.B. :Veillez à mobiliser vos connaissances et à soigner la rédaction.

Ensimag 1re année **2009-2010**

Probabilités Appliquées 2
Mai 2010

Durée 3h00 - Documents et calculatrices non-autorisés. Veuillez inscrire votre numéro de groupe sur la copie.

Exercice I (6 pts) - Soit $\lambda > 0$. On considère un couple (X, Y) de loi définie par la densité

$$\forall (x,y) \in \mathbb{R}^2, \quad f(x,y) = \begin{cases} \lambda^2 e^{-\lambda y} & \text{si } (x,y) \in \Delta \\ 0 & \text{sinon,} \end{cases}$$

où $\Delta = \{(x,y) \in \mathbb{R}^2;\, 0 < x < y\}$.

1-1pt) Déterminer la loi de X. Soit $x > 0$. Montrer que la loi conditionnelle de Y sachant $X = x$ est identique à celle de $x + X_2$, où X_2 est de loi exponentielle de paramètre λ.

2-1pt) Soient X_1 et X_2 deux variables aléatoires indépendantes de loi exponentielle de paramètre λ. On pose

$$\begin{cases} X = X_1 \\ Y = X_1 + X_2 \end{cases}$$

Montrer que le couple (X, Y) admet f pour densité.

3-1pt) Déduire de la question précédente l'espérance conditionnelle de Y sachant $X = x$, $x > 0$.

4-1pt) Montrer que $\mathrm{E}[XY|X = x] = x(x + 1/\lambda)$, $x > 0$.

5-1pt) Déduire des questions précédentes la covariance $\mathrm{Cov}(X, Y)$.

6-1pt) Déterminer la loi de Y.

Exercice II (6 pts) - On considère le couple gaussien (X, Y) de moyenne nulle et de matrice de covariance

$$K = \begin{pmatrix} 1 & c \\ c & 4 \end{pmatrix},$$

où $-2 < c < 2$ et $c \neq 0$.

1-1pt) Déterminer les lois marginales de X et de Y

2-1pt) Déterminer la loi de la variable $Z = X - Y/c$.

3-1pt) Montrer que X et $X - Y/c$ sont deux variables aléatoires indépendantes.

4-2pts) Déduire de la question précédente, l'espérance et la variance conditionnelles $\mathrm{E}[Y|X = x]$ et $\mathrm{Var}(Y|X = x)$, pour tout $x \in \mathbb{R}$.

5-1pt) Proposer un algorithme de simulation du couple (X, Y) et vérifier sa validité.

Exercice III (4 pts) - Une variable aléatoire réelle X suit la loi de Cauchy si elle admet pour densité la fonction

$$\forall x \in \mathbb{R}, \quad f_X(x) = \frac{1}{\pi(1 + x^2)} .$$

1-1pt) La variable X est-elle d'espérance finie ?

2-2pts) Soient $X_1, \ldots, X_n$ des variables indépendantes de loi de Cauchy. Déterminer la loi de la variable

$$M_n = \frac{X_1 + \cdots + X_n}{n} .$$

Indication : on pourra utiliser le résultat suivant

$$\forall t \in \mathbb{R}, \quad \int_{-\infty}^{+\infty} e^{itx} \frac{1}{\pi(1 + x^2)} dx = e^{-|t|} .$$

3-1pt) La loi des grands nombres s'applique-t-elle à la variable M_n ?

Exercice IV (4pts) - Un individu se livre à un jeu de hasard avec des nombres entiers de la manière suivante. Il part de 1, puis choisit un entier X_1 au hasard entre 2 et n (n est un entier fixé). Si $X_1 < n$, il choisit alors un entier X_2 au hasard entre $X_1 + 1$ et n. Sinon le jeu s'arrête. Le joueur itère le processus, en choisissant au hasard un entier plus grand que le nombre présent, jusqu'à atteindre n. On note m_k le nombre moyen de pas nécessaires pour atteindre n en partant de $n - k$, pour tout $0 \leq k \leq n - 1$. On pose $m_0 = 0$.

1-2pt) Montrer, pour tout $k > 0$, que

$$m_k = 1 + \frac{1}{k}(m_{k-1} + m_{k-2} + \cdots + m_0) .$$

2-1pt) Calculer m_4.

3-1pt) Expliciter une relation liant m_k et m_{k-1}. En déduire la valeur de m_{n-1}.

EXAMEN du 12 mai 2010. Durée: 3h. 2 pages numérotées.
Documents manuscrits ou polycopiés autorisés. Aucun livre.

Il est important de bien expliquer ce que vous faites. Le total des points sera de 22 points donc il n'est pas nécessaire de tout faire pour avoir une bonne note, par contre il sera enlevé des points pour une rédaction trop succincte.

Veuillez noter sur votre copie le nom de votre enseignant : BIENIA ou CANI ou NADDEF ou SZIGETI

EXERCICE 1 :

Considérons une grille $n \times n$ comme un graphe $\boldsymbol{T}$. (Attention : il y a n^2 sommets !)
Le coût d'une arête horizontale est égal à *1* et le coût d'une arête verticale est égal à *2*.

a) Calculer le nombre d'arêtes d'un arbre couvrant de T.

b) Exécuter l'algorithme de Kruskal pour trouver un arbre couvrant de $\boldsymbol{T}$ de coût minimum.

c) Calculer le nombre d'arbres couvrant de $\boldsymbol{T}$ de coût minimum.

d) Y a-t-il un arbre couvrant de $\boldsymbol{T}$ de coût minimum qui contient $n-1$ sommets de degré 4 ?

e) Existe-t-il un arbre couvrant de $\boldsymbol{T}$ qui contient $O(n^2)$ sommets de degré 4 ?

EXERCICE 2 :

Une agence de location de résidences de vacances dispose d'un appartement pour lequel elle a reçu de la part de clients potentiels des propositions de locations. Le tableau ci-dessous précise les périodes demandées ainsi que les montants des loyers (en kiloeuros = 1 000 €) :

offres	dates	loyers	durée	offres	dates	loyers	durée
A	*31 mai – 21 juin*	*1 k€/sem*	*3 sem.*	***D***	*28 juin – 26 juillet*	*1 k€/sem*	*4 sem.*
B	*07 juin – 19 juillet*	*1 k€/sem*	*6 sem.*	***E***	*12 juillet – 09 août*	*1 k€/sem*	*4 sem.*
C	*14 juin – 05 juillet*	*2 k€/sem*	*3 sem.*	***F***	*02 août – 15 août*	*1 k€/sem*	*2 sem.*

a) L'agence veut sélectionner (accepter ou refuser) les locations de façon à maximiser la somme des loyers perçus au titre des locations. ***Modéliser*** ce problème comme recherche du plus long chemin dans un graphe orienté que l'on explicitera (préciser ses sommets, ses arcs et leur valuation). ***Résoudre*** en utilisant formellement un algorithme vu en cours. Citer les algorithmes adaptés.

b) L'agence veut maximiser l'occupation de l'appartement (en semaines). ***Modéliser*** et résoudre ce nouveau problème.

EXERCICE 3 :

Le but de cet exercice est de démontrer le *lemme de Farkas* : $\boldsymbol{Ax = b, x \geq 0}$ admet une solution si et seulement s'il n'existe pas de vecteur $\boldsymbol{y}$ tel que $\boldsymbol{y^T A \geq 0}$ et $\boldsymbol{y^T b}<0$, en utilisant le théorème fort de la dualité.

a) En calculant le produit $\boldsymbol{y^T A x}$ de deux façons différentes, montrer que de tels vecteurs $\boldsymbol{x}$ et $\boldsymbol{y}$ ne peuvent pas exister à la fois.

b) On suppose qu'il n'existe pas de vecteur $\boldsymbol{y}$ tel que $\boldsymbol{y^T A \geq 0}$ et $\boldsymbol{y^T b}<0$.
Considérons le programme linéaire (P) : ***minimiser*** $z = \boldsymbol{b^T y}$ ***sous*** $\boldsymbol{A^T y \geq 0}$.

(1) Donner une solution réalisable de (P).

(2) Montrer que $z_{min} = 0$.

(3) Ecrire le dual (D) de (P).

(4) Déduire de (2) que (D) admet une solution réalisable et en conclure.

EXAMEN du 12 mai 2010. Durée: 3h. 2 pages numérotées.
Documents manuscrits ou polycopiés autorisés. Aucun livre.

EXERCICE 4 :

Des envahisseurs extra-terrestres disposent d'un budget de **367** dogens (c'est le nom de leur monnaie) pour acheter des vaisseaux permettant d'acheminer progressivement leurs combattants grimés en terriens vers les grandes capitales du monde. Ils disposent uniquement de **149** pilotes de l'espace capables de mener les vaisseaux à bon port. Ces vaisseaux, à énergie solaire, peuvent être de trois types : cigare, soucoupe, ou anneau.

- Le cigare ne peut transporter que **10** extra-terrestres, peut parcourir une distance maximale de **70 années-lumière/jour** et coûte **8** dogens. Il peut être conduit par un seul pilote, mais ce dernier aura alors besoin de se reposer en laissant son vaisseau en position stationnaire la moitié du temps. Si on lui affecte un second pilote le vaisseau pourra voler en permanence, mais il ne restera que 8 places pour le transport des combattants.
- La soucoupe, capable de transporter **34** extra-terrestres peut parcourir une distance maximale de **60 années-lumière/jour** et coûte **13** dogens. La piloter se fait par équipe de **3** pilotes. La soucoupe pourra voler la moitié du temps si une seule équipe est présente, et en permanence si deux équipes de pilotes (donc 6 personnes) se relaient.
- L'anneau peut transporter **32** extra-terrestres, vole à la même vitesse que la soucoupe et coûte **15** dogens. Le pilotage est plus automatisé : la manœuvre ne peut pas être faite par une seule personne, mais trois pilotes se relayant sont capables de faire voler un anneau en permanence, tandis que deux pilotes ne peuvent le faire voler que la moitié du temps.

Quel est l'achat optimal qui permet aux envahisseurs de maximiser leur capacité de transport de troupes, exprimée en nombre de combattants $\times$ distance (année-lumière) par jour ? Modéliser ce problème par un programme linéaire. La solution n'est pas demandée.

EXERCICE 5 :

Une entreprise fabrique les produits A, B et C à partir de ressources. Le tableau ci-dessous donne les consommations par unité de produit fabriquée ainsi que les disponibilités des ressources.

	A	B	C	disponibilité
heure machine	2	3	1	10
matière première	1	4	3	15

Les profits par unité de produit fabriqué sont respectivement de 6, 5 et 4 k€ (1k€=1000 €).

a) Déterminer un plan de production qui maximise la recette, c'est-à-dire écrire et résoudre le programme linéaire correspondant.

b) Ce plan de production qui maximise la recette est-il unique? Justifier votre réponse.

c) Interpréter les variables duales du problème. Peut-on trouver une solution optimale du dual dans le tableau optimal du simplexe ? Cette solution optimale du dual est-elle unique ? Justifier votre réponse.

d) Retrouvez cette solution optimale du dual en utilisant le théorème des écarts complémentaires.

e) L'entreprise pourrait disposer d'une heure-machine supplémentaire au prix de 3 k€. A-t-elle intérêt à accepter cette offre ? Et si on lui propose plutôt une unité de matière première au prix de ½ k€ ? Bien justifier vos réponses.

Ensimag 1ière année Le 17 mai 2010

Théorie des Langages 2

Durée : 3 heures.
Documents : tout document autorisé.

Pensez au lecteur. Commentez, utilisez des noms explicites et ne renommez pas les noms employés dans l'énoncé.

Exercice (6 points)

Soient $f : \mathbb{N} \to \mathbb{N}$ et $g : \mathbb{N} \to \mathbb{N}$ deux fonctions calculables quelconque. On se propose dans cet exercice de montrer que le problème "$\forall x \in \mathbb{N}, f(x) = g(x)$" n'est pas décidable.

Considérons les fonctions $f_i : \mathbb{N} \to \mathbb{N}$ et $g_{i,e} : \mathbb{N} \to \mathbb{N}$ définies ci-dessous, avec i l'indice d'une machine de Turing et e un entier naturel quelconque.

$$f_i(x) = \begin{cases} 1 & \text{si } MTU(i,x) \text{ s'arrête} \\ \text{indéfinie} & \text{sinon} \end{cases}$$

$$g_{i,e}(x) = \begin{cases} f_i(x) & \text{si } x \neq e \\ 1 & \text{si } x = e \end{cases}$$

▷ **Question 1 (2 points)**

Montrer que "$MTU(i,e)$ s'arrête" si et seulement si "$\forall x \in \mathbb{N}, f_i(x) = g_{i,e}(x)$".

▷ **Question 2 (2 points)**

Justifier brièvement que les fonctions f_i et $g_{i,e}$ sont calculables.

▷ **Question 3 (2 points)**

Déduire l'indécidabilité du problème "$\forall x \in \mathbb{N}, f(x) = g(x)$", pour f et g calculables.

PROBLEME (14 points)

On s'intéresse à un langage de commandes, défini par la grammaire G_1 suivante (l'axiome est le non-terminal programme) :

programme	$\rightarrow$	**begin** commande **end**		
commande	$\rightarrow$	sequencement	| affectation	
sequencement	$\rightarrow$	commande **;** affectation		
affection	$\rightarrow$	**idf :=** exp		
exp	$\rightarrow$	(exp + exp)	| **idf**	| **num**

Le vocabulaire terminal est VT={ **begin, end, idf, num, ;, :=, (,), +** }.

▷ **Question 4 (1 point)**
Justifier en quoi cette grammaire n'est pas LL(1).

▷ **Question 5 (3 points)**
Donner une grammaire LL(1) pour le langage L(G_1). On fera bien attention à préserver le langage. On prouvera le caractère LL(1) de la grammaire proposée (on pourra numéroter les règles pour les calculs de directeur).

▷ **Question 6 (3 points)**
On veut étendre la notation pour permettre l'affectation simultanée de plusieurs variables. Exemples :

1. x, y, z := a, (a+1), 0
2. x, y := y, x

Proposer une nouvelle définition du non-terminal **affectation**, sous forme LL(1), prenant en compte cette extension. La grammaire proposée devra garantir qu'il y a autant d'éléments de chaque côté du signe :=. On étend le vocabulaire terminal à l'aide du symbole **,**.

▷ **Question 7 (2 points)**
On ajoute la contrainte suivante pour l'affectation simultanée : **les identificateurs apparaissant en partie gauche du signe := doivent être distincts deux à deux.**

Ajouter à la grammaire précédente des attributs permettant de vérifier cette contrainte. Les attributs manipulés représenteront des ensembles de noms et on pourra utiliser les opérations classiques sur les ensembles ($\cup, \cap, \in, \notin, \ldots$).

▷ **Question 8 (3 points)**
Ecrire un analyseur LL(1) qui reconnaît les affectations simultanées et vérifie la contrainte de la question précédente. On utilisera la fonction **lire_mot return token** avec **token** = $VT \cup \{\$\}$, où *\$* représente le marqueur de fin de texte, ainsi que la fonction **nom_idf return string** qui renvoie le nom d'un identificateur lorsque le mot reconnu est de catégorie **idf**. On n'écrira pas la procédure **exp**, qui analyse les expressions.

On supposera aussi disposer d'un type abstrait **Ens** qui décrit les ensembles de string ainsi que les opérations classiques sur ces ensembles.

▷ **Question 9 (2 points)**

Soit `x1, ..., xn := e1, ..., en` une affectation simultanée. Cette commande peut être traduite en une séquence d'affectation simple de la manière suivante :

```
a1 := x1 ; ... an := xn ; x1 := e1'; ...  xn := en'
```

avec `ai` des identificateurs n'apparaissant pas dans le programme. L'expression `ei'` désigne l'expression obtenue à partir de `ei` en remplaçant les occurrences des identificateurs `xi` par `ai` (`ei' = ei [x1← a1] ... [xn← an]`).

Appliquer cette traduction aux 2 exemples de la question 6.

On s'intéresse au cas où les identificateurs `x1, ..., xn` n'apparaissent pas dans les expressions `e1, ..., en`. Expliquer en quoi cette hypothèse permet de simplifier la traduction ci-dessus. Donner un calcul d'attributs sur la grammaire de la question 6 permettant de déterminer si la condition ci-dessus est vérifiée.

Examen de Traitement du Signal - Ensimag - 1A - Mai 2010
DOCUMENTS ET CALCULATRICES AUTORISES

1 Propriétés à l'ordre 2 d'un signal déterministe

On considère le signal causal (fonction réelle de la variable réelle)

$$x(t) = \frac{1}{T} \exp\left[-\frac{t}{T}\right] \quad \text{avec } T > 0, t \in \mathbb{R}^+$$

1. Montrer que ce signal est d'énergie finie. Quelle est son énergie ?
2. Calculer sa densité spectrale d'énergie $S_{xx}(\nu)$.
3. Est-il possible d'échantillonner $x(t)$ en respectant les conditions du théorème d'échantillonnage et si oui, quelle est la fréquence d'échantillonnage minimale ?
4. Un signal peut-il être à la fois causal et à bande limitée ?

2 Propriétés élémentaires du filtrage numérique

1. On considère le signal $\{x_k\}_{k\in\mathbb{Z}}$ valant 1 pour $k \geq 0$ et 0 sinon. Quelle est la transformée en z, $X(z)$, de ce signal ? Préciser son domaine de convergence.
2. On considère le filtre dont la réponse impulsionnelle $\{h_k\}_{k\in\mathbb{Z}}$ vaut $\exp[-k]$ pour $k \geq 0$ et 0 sinon.
   (a) Quelle est la fonction de transfert en z, $H(z)$, de ce filtre ? Préciser son domaine de convergence.
   (b) Comment peut-on vérifier la causalité du filtre dans le domaine z ?
   (c) Comment peut-on vérifier la stabilité du filtre dans le domaine z ?
   (d) Quelle est la réponse fréquentielle de ce filtre ? Par quel terme peut-on le qualifier (passe-haut, passe-bas, passe-bande, ...) ?
3. On filtre le signal $\{x_k\}_{k\in\mathbb{Z}}$ par le filtre $\{h_k\}_{k\in\mathbb{Z}}$, calculer le signal en sortie du filtre ?
4. On applique maintenant à l'entrée du filtre le même signal retardé de n échantillons : $\{x_{k-n}\}_{k\in\mathbb{Z}}$ avec n entier positif. Quelle est la nouvelle sortie du filtre ?
5. On considère maintenant une «porte» discrète de durée n, c'est-à-dire un signal numérique $\{\pi_k\}_{k\in\mathbb{Z}}$ valant 1 pour $0 \leq k < n$ et 0 sinon. Déduire des questions précédentes la sortie du filtre $\{h_k\}_{k\in\mathbb{Z}}$ lorsque son entrée est $\{\pi_k\}_{k\in\mathbb{Z}}$.
6. Quelle est la transformée de Fourier de $\{\pi_k\}_{k\in\mathbb{Z}}$? Quel lien pourrait-on voir entre cette transformée de Fourier et celle d'une porte analogique (fonction valant 1 sur un intervalle $[0, T]$ et 0 en dehors) ? Dans quelle limite la transformée de Fourier de la porte discrète est-elle proche d'un sinus cardinal ?

3 Fréquences plus ou moins pures

On considère le signal $s(t) = \cos[2\pi\nu_0 t] + 2\sin[6\pi\nu_0 t]$, t réel.

1. $s(t)$ est-il d'énergie finie ? Si oui, quelle est son énergie ?
2. Quelle est la transformée de Fourier de $s(t)$?
3. A quelle fréquence doit-on échantillonner $s(t)$ pour respecter le théorème d'échantillonnage ?
4. $s(t)$ est maintenant observé sur l'intervalle $[0, T]$ seulement ($T > 0$).
   (a) Comment peut-on modéliser cette restriction de l'intervalle d'observation ?
   (b) Quelle est son impact sur le contenu fréquentiel du signal ?
   (c) Calculer la transformée de Fourier du signal observé.
   (d) A quelle fréquence doit-on échantillonner ce signal à support limité pour respecter les conditions du théorème d'échantillonnage ?
5. On filtre $s(t)$ par un filtre de gain complexe (fonction de transfert en Fourier)

   $$H(\nu) = \exp\left[i\frac{\pi}{2}\right] \text{ avec } i^2 = -1 \text{ pour } \nu \in [-2\nu_0, -2\nu_0] \text{ et } 0 \text{ sinon.}$$

   Calculer la sortie du filtre.
6. La sortie du filtre est seuillée à $\pm\frac{1}{2}$, c'est-à-dire que les valeurs supérieures à $\frac{1}{2}$ sont transformées en la valeur $\frac{1}{2}$, les valeurs inférieures à $-\frac{1}{2}$ en $-\frac{1}{2}$ tandis que les valeurs comprises entre $-\frac{1}{2}$ et $+\frac{1}{2}$ restent inchangées.
   (a) Quelle sont les fréquences contenues dans ce signal seuillé ?
   (b) Quelle est la fonction de transfert (en Fourier) de ce système de seuillage ?

4 Modulation d'amplitude

Soit le signal réel à bande limitée $s(t) = \frac{2\sin(\pi t)}{\pi t}$ ($t \in \mathbb{R}$).

1. Donner la transformée de Fourier de $s(t)$ puis tracer $|\hat{s}(\nu)|^2$.

On transmet ce signal à l'aide d'une modulation d'amplitude. Le signal modulé s'écrit :

$$x(t) = s(t)\cos(2\pi\nu_0 t).$$

2. On suppose la fréquence ν_0 plus grande que la bande passante de $s(t)$. Calculer la transformée de Fourier du signal modulé $x(t)$ puis tracer $|\hat{x}(\nu)|^2$.
3. En considérant le canal de transmission comme un filtre linéaire invariant, quelle condition doit-il remplir pour ne pas affecter le signal modulé transmis ? (Autrement dit, donner son gabarit en fréquence.)

On démodule le signal à l'aide d'un démodulateur cohérent. Le signal en sortie de celui-ci s'écrit :

$$x_d(t) = x(t)\cos(2\pi\nu_0 t + \varphi) \text{ avec } \varphi \in [0, 2\pi).$$

4. Calculer la transformée de Fourier de $x_d(t)$ puis tracer $|\hat{s}(\nu)|^2$.
5. Imaginer un scénario de filtrage/amplification permettant de retrouver $s(t)$ à partir de $x_d(t)$.

Mercredi 19 mai 2010
Durée : 2 heures

Cours et documents autorisés
Calculatrice autorisée

Examen : Transmettre l'information

1ère année Ensimag

Exercice 1 : Code en ligne

On étudie ici une liaison réalisée par un câble Ethernet. Le codage en ligne utilisé est donc du type Manchester.

1) Afin de montrer la règle du codage de Manchester, donnez l'évolution temporelle des tensions relevées sur le câble dans le cas de la transmission des bits suivants : 01001101.

2) Concernant la répartition fréquentielle de la puissance, expliquez la différence principale avec un code simple de type NRZ. Expliquez alors dans quel cas le codage de Manchester présente un intérêt et pourquoi il a été choisi ici.

3) Quelle est la technique d'accès utilisée avec le protocole Ethernet ? Expliquez son principe de fonctionnement. Précisez notamment comment une machine peut détecter si le support de transmission est libre ou non et pourquoi cela est possible avec un code de Manchester alors que cela ne le serait pas dans tous les cas avec, par exemple, un code NRZ.

Exercice 2 : Transmission radio

L'objectif de cet exercice est d'étudier une transmission de signaux numériques par un système de faisceaux hertziens. Ce système utilise des canaux fréquentiels de largeur égale à 40 MHz autour de 6 GHz et une modulation QAM64.

1) **Donnez le diagramme de constellation de cette modulation en expliquant ce qu'il représente.**

Le système de transmission est utilisé pour réaliser une transmission à 155 Mbit/s entre deux sites d'une entreprise situés à 48 km à vol d'oiseau. La zone entre les deux sites est plate et essentiellement constituée d'une forêt avec des arbres culminant à 15 m.

2) **A quelle hauteur faut-il positionner les antennes d'émission et de réception ?**

A l'entrée du récepteur se trouve un filtre.

3) **Quelles sont les fonctions du filtre de réception ? Quelles propriétés doit-il vérifier pour correspondre à un filtre 'optimal' ?**

Par la suite, on supposera que le filtrage global sur la chaîne de transmission est un filtrage de Nyquist avec un coefficient d'arrondi α de 0,5.

4) **Montrez que la modulation QAM64 convient pour transmettre le débit binaire dans la bande du canal.**

On suppose que la puissance d'émission est choisie pour qu'en régime normal de transmission (beau temps) le taux d'erreurs binaires soit au maximum de 10^{-6}.

5) **La liaison fonctionne-t-elle encore en cas d'averse ? Quelle est en effet l'atténuation possible par rapport au beau temps ? Quelle est la dégradation du taux d'erreurs binaires que l'on peut attendre ? Que se passe-t-il en cas d'orage ?**

On utilise des antennes cornet à l'émission et à la réception de gain égal à 12 dB.

6) **Rappelez la définition du gain d'une antenne.**

7) **Si on utilise des antennes paraboliques de gain égal à 45 dB, quelle marge gagne-t-on sur le bilan de liaison total ? Est-ce que cela permet de faire fonctionner plus facilement la liaison en cas d'orage ?**

8) **Pourquoi l'antenne de type parabolique a-t-elle un plus fort gain ?**

L'élément rayonnant présente à 6 GHz une impédance d'entrée égale à : (23+j 357) Ω.

9) **Est-il adapté à un câble coaxial d'impédance 50 ohms ?**

10) **Placez l'impédance de l'élément rayonnant sur l'abaque de Smith.**

11) **Que vaut son coefficient de réflexion ? Si on ne fait pas d'adaptation, quelles pertes de puissance aura-t-on ?**

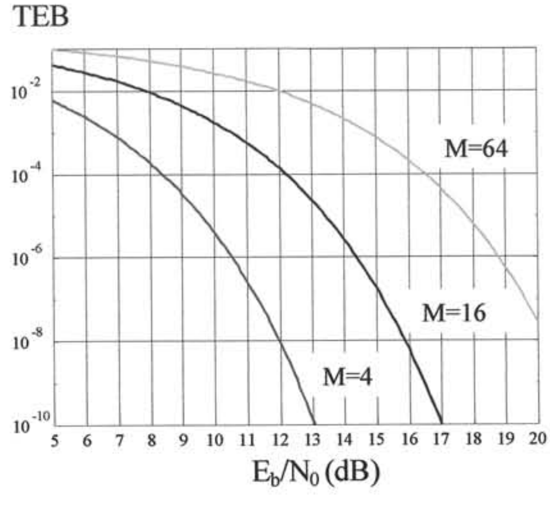
12) **Les spécifications du système précisent que les protection des données sont faites par des codes CRC/FEC et par acquittements/reprises ARQ.**

13) **Que font les codes CRC ? Quel est le principe de la technique ARQ ?**

14) **Les systèmes radio tels que le Wi-Fi utilisent la modulation OFDM. Cette modulation n'est pas spécialement intéressante pour les systèmes de faisceaux hertziens. Pourquoi ?**

15) **Imaginons qu'on veuille utiliser plutôt une liaison filaire entre les deux sites. Que proposeriez-vous alors d'utiliser (type de support de transmission, émetteurs, récepteurs ...) ?**

<u>Annexe : Taux d'erreurs binaires dans le cas des modulations QAM-M</u>



The Complete Smith Chart

Black Magic Design