Grenoble INP
ENSIMAG

Annales d'examens 1ère année

Année universitaire 2012/2013

PARTIEL DU MOIS DE NOVEMBRE 2012

ANALYSE POUR L'INGENIEUR -	1 PAGE
PROBABILITES APPLIQUEES -	2 PAGES
THEORIE DES LANGAGES 1 -	3 PAGES

Examen à mi-parcours d'Analyse pour l'ingénieur

Mardi 6 novembre 2012 - 1h30

Documents de cours manuscrits et polycopié autorisés.

La rédaction sera prise en compte dans la notation. Toute affirmation doit être justifiée.

Exercice 1 *Soit E un ensemble et $f : E \to \mathbb{R}$ une application. On pose $d(x, y) = |f(x) - f(y)|$.*

1. *Trouver une condition nécessaire et suffisante sur f pour que d soit une distance.*
2. *On pose $E = \mathbb{R}$ et $f : x \in \mathbb{R} \to e^x \in \mathbb{R}$. Montrer que d est une distance et expliciter les boules ouvertes.*

Exercice 2 *Soit (E, d) un espace métrique et A une partie non vide de E. Pour $x \in E$, on note $d_A(x) = \inf_{y \in A} d(x, y)$ la distance de x à A. Nous avons vu en TD que*

$$x \in \overline{A} \Longleftrightarrow d_A(x) = 0.$$

1. *Pour $\varepsilon > 0$, on pose :*
$$V_\varepsilon(A) = \{x \in E,\ d_A(x) < \varepsilon\}.$$
Montrer que
$$\overline{A} = \bigcap_{\varepsilon > 0} V_\varepsilon(A).$$
2. *Montrer que*
$$d_A = d_B \Longleftrightarrow \overline{A} = \overline{B}.$$

Exercice 3 *Soit E l'espace vectoriel des fonctions continues de $[-1; 1]$ dans $\mathbb{R}$, muni de la norme :*

$$\|f\|_\infty = \sup_{t \in [-1,1]} |f(t)|$$

Soit F l'espace vectoriel des fonctions 2π-périodiques et continues de $\mathbb{R}$ dans $\mathbb{R}$, que l'on munit soit de la norme N_2 :

$$N_2(f) = \left(\frac{1}{2\pi} \int_0^{2\pi} |f(t)|^2 dt \right)^{1/2}$$

soit de la norme N_∞ :

$$N_\infty(f) = \sup_{t \in \mathbb{R}} |f(t)|$$

Soit $L : E \to F$ l'application définie par $L(f)(t) = f(cos(t))$, $\forall t \in \mathbb{R}$.

1. *Montrer que L est bien définie, est linéaire et injective.*
2. *Montrer que L est continue pour chacune des normes N_2 et N_∞ de F, et calculer pour chacune de ces normes la norme de l'opérateur L, que l'on notera $|||L|||_2$ et $|||L|||_\infty$.*

Exercice 4 *Soient (E, d) et (F, δ) des espaces métriques et $f : E \to F$ une application. On appelle graphe de f :*

$$G_f = \{(x, f(x)),\ x \in E\} \subset E \times F.$$

1. *Montrer que si f est continue alors le graphe G_f est fermé.*
2. *Montrer que si F est compact, la réciproque est vraie.*

2012-2013
Ensimag 1A

Examen de Probabilités appliquées
Mardi 6 novembre 2012

Durée : 2h00.
Document autorisé : 1 feuille A4 manuscrite recto.
Calculatrices et téléphones portables interdits.
Merci d'inscrire votre numéro de groupe sur la copie.
Le sujet comporte 2 pages. Le barème est indicatif. Les 4 exercices sont indépendants et peuvent être traités par le candidat dans l'ordre qu'il préfère.

Exercice 1. [5pts]
Les deux questions sont indépendantes.

1. (a) On lance simultanément deux dés à 6 faces équilibrés de façon indépendante, jusqu'à ce que l'un des deux dés au moins produise un six. Soit N le nombre de lancers. Calculer, pour $n \in \mathbb{N}^*$, $P(N = n)$.
   (b) Quelle est l'espérance de N ?
2. Une usine possède deux machines M_1 et M_2 qui fabriquent respectivement 40 et 60% de la production totale par jour. Les pièces fabriquées par la machine M_1 sont défectueuses dans 1% des cas et celles fabriquées par M_2 dans 6% des cas.
   (a) On prend une pièce au hasard dans la fabrication d'un jour. Quelle est la probabilité qu'elle soit défectueuse ?
   (b) On prend une pièce au hasard dans la fabrication d'un jour. Elle n'est pas défectueuse. Quelle est la probabilité qu'elle ait été fabriquée par la machine M_1 ? On donnera le résultat sous forme d'une fraction irréductible.
   (c) Le vendredi 2 novembre, l'usine a fabriqué 200 pièces. Soit Y la variable aléatoire donnant le nombre de pièces défectueuses. Calculer, pour $i \in \{1, \ldots, 200\}$, $P(Y = i)$. Quelle est l'espérance de Y ?

Exercice 2. [4pts]
Au football, on peut marquer des buts de la tête ou du pied. On suppose que le nombre de buts marqués lors d'un match est un nombre aléatoire K suivant une loi de Poisson de paramètre $\lambda > 0$.
C'est à dire, pout tout $n \in \mathbb{N}$, $P(K = n) = \dfrac{\lambda^n}{n!}e^{-\lambda}$.

La probabilité pour qu'un point soit marqué de la tête est p $(0 < p < 1)$. On suppose que les points sont marqués indépendamment les uns des autres.

1. Soit L la variable aléatoire qui représente le nombre de buts marqués de la tête. Sachant que $K = k$ buts ont été marqués lors d'un match, montrer que la probabilité conditionnelle pour que $L = l$ buts soient marqués de la tête est
$$P(L = l|K = k) = \frac{k!}{l!(k-l)!}p^l(1-p)^{k-l}, l = 0, \ldots, k.$$
2. En déduire la probabilité d'observer $L = l$ points marqués de la tête lors d'un match, pour $l \in \mathbb{N}$.

Exercice 3. [4pts]
On tire au hasard un nombre U uniformément dans l'intervalle $[0, 1]$ puis on lance une pièce équilibrée dont le résultat est indépendant de U. Si le résultat est pile, on définit $X = U$. Si le résultat est face, on définit $X = 2U$.

1. Calculer, pour tout $t \in \mathbb{R}$, la fonction de répartition $F_X(t)$ de la variable aléatoire X.
2. Représenter graphiquement cette fonction. Par une vérification très simple, contrôler la cohérence du résultat obtenu à la question précédente.
3. Calculer l'espérance de la variable aléatoire X. Par une vérification très simple, contrôler la cohérence de votre résultat.

Exercice 4. [7pts]
Une variable aléatoire X suit une loi exponentielle de paramètre $\lambda > 0$ si sa fonction de répartition F_X est donnée par, pour tout $t \in \mathbb{R}$,
$$F_X(t) = \begin{cases} 1 - e^{-\lambda t} & \text{si } t \geqslant 0 \\ 0 & \text{si } t < 0. \end{cases}$$

1. Montrer qu'une variable aléatoire X qui suit une loi exponentielle de paramètre λ est sans mémoire, c'est à dire que pour tout $s \geqslant 0$ et pour tout $t \geqslant 0$,
$$P(X \geqslant t + s|X \geqslant t) = P(X \geqslant s).$$
2. Le temps d'attente à un serveur téléphonique suit une loi exponentielle de paramètre $\lambda = 0,2$ (en minutes). Marie a appellé à ce serveur et attend une réponse depuis 5 minutes. Quelle est la probabilité qu'elle attende encore pendant une durée supérieure ou égale à 5 minutes ?
3. Soient $X_1, \ldots, X_n$, n variables aléatoires indépendantes de même loi exponentielle de paramètre λ. Soit $W = \min(X_i, i = 1, \ldots, n)$.
Calculer, pour $t \in \mathbb{R}$, $P(W > t)$. En déduire la fonction de répartition F_W de la variable aléatoire W. Quelle est la loi de la variable aléatoire W ?
4. Soit N une variable aléatoire de loi de Poisson de paramètre $\lambda > 0$. Soit $M = N+1$. Soit $W_M = \min_{i=1,\ldots,M} X_i$. Calculer, pour $t \in \mathbb{R}$, la probabilité pour que W_M soit supérieur à t.

Grenoble INP - Ensimag 1ère année 5 Novembre 2012

Théorie des langages 1

Durée : 2h.
Documents : tous les documents sont autorisés.

Nb : le barème est donné à titre indicatif; la rigueur des preuves et la clarté de la rédaction seront prises en compte dans la notation. Cet examen comporte deux exercices indépendants.

Exercice 1 - Transformations d'automates (11 points)

Soit $Q = \{p_0, \ldots, p_4\}$. On considère l'automate $A = (Q, \{a, b\}, \delta, \{p_0\}, \{p_4\})$, où la relation de transition δ est définie par :

δ	a	b	ε
p_0	p_1	-	p_2
p_1	-	-	p_3
p_2	p_1	p_0, p_2	-
p_3	p_4	-	p_2
p_4	-	-	-

On note L le langage reconnu par A. Dans tout cet exercice, on prendra soin de **détailler les étapes de calcul** lors des transformations d'automates.

▷ **Question 1** *(3 points) Construire un automate déterministe équivalent à A.* ***On ne demande pas de minimiser l'automate déterministe obtenu.***

Soit V un vocabulaire et soit une fonction $\pi : V \to V$ qui est bijective. La fonction π est étendue aux mots de la façon suivante :

- $\pi(\varepsilon) = \varepsilon$,
- $\forall a \in V$, $\forall w \in V^*$, $\pi(aw) = \pi(a)\pi(w)$.

La fonction π est également étendue aux langages en posant, pour tout $L \subseteq V^*$, $\pi(L) = \{\pi(w) \mid w \in L\}$.

Pour tout automate $A = (Q, V, \delta, I, F)$ sans epsilon-transition, on définit l'automate $A^\pi = (Q, V, \delta^\pi, I, F)$, où

$$\delta^\pi = \{(p, \pi(a), q) \mid (p, a, q) \in \delta\}.$$

▷ **Question 2** *(2 points) On pose $V = \{a, b, c\}$ et on considère le langage $L \subseteq V^*$ constitué de tous les mots contenant* ***au moins*** *un 'a', et* ***au plus*** *un 'b'. Construire un automate A reconnaissant L.*

▷ **Question 3** *(2 points) On pose $\pi : a \mapsto b, b \mapsto c, c \mapsto a$. Décrire en français le langage reconnu par A^π.*

▷ **Question 4** *(2 points) Montrer par induction sur les chemins que pour tout $p, q \in Q$ et pour tout $w \in V^*$, les deux propriétés suivantes sont équivalentes :*

1. *il existe un chemin de p à q de trace w dans A ;*
2. *il existe un chemin de p à q de trace $\pi(w)$ dans A^π.*

▷ **Question 5** *(2 points) Notons L le langage reconnu par A et M le langage reconnu par A^π. Montrer que $M = \pi(L)$ et en déduire que pour tout langage régulier L, le langage $\pi(L)$ est également régulier.*

Exercice 2 - Induction : expressions postfixées (9 points)

Soit V un vocabulaire, et considérons un sous-ensemble $A \subseteq V$. Nous cherchons à définir la fonction $|.|_A : V^* \to \mathbb{N}$ qui à tout mot $w \in V^*$ associe le nombre d'éléments de A présents dans w. Par exemple, si $V = \{a, b, c, d\}$ et $A = \{a, b\}$, alors :

- $|cabdbacc|_A = 4$,
- $|bbaba|_A = 5$,
- $|ccdc|_A = 0$.

▷ **Question 1** *(2 points) Pour des vocabulaires V et $A \subseteq V$ donnés, définir la fonction $|.|_A$ par induction sur V^*.*

On considère les vocabulaires suivants :

$$\begin{aligned} \text{Num} &= \{0, 1, \ldots, 9\}, \\ \text{Op} &= \{+, -, \times\}, \\ V_{\text{pf}} &= \text{Num} \cup \text{Op}, \end{aligned}$$

et on définit le langage $L \subseteq V_{\text{pf}}^*$ par induction de la façon suivante :

Cas de base : si $x \in \text{Num}$, alors $x \in L$.

Règles : si $u \in L$ et $v \in L$, alors :

- $u\ v\ + \in L$,
- $u\ v\ - \in L$,
- $u\ v\ \times \in L$.

Le langage L est appelé le langage des expressions postfixées portant sur des chiffres (et non pas sur des nombres).

▷ **Question 2** *(2 points) Montrer que les éléments suivants sont dans L :*

1. $w_1 = 1\ 3\ +$
2. $w_2 = 2\ 4\ *\ 5\ -$
3. $w_3 = 3\ 4\ +\ 5\ 2\ -\ \times$
4. $w_4 = 3\ 4\ 5\ 6\ \times\ -\ +$

Etant donnés un vocabulaire V et un élément $w \in V^*$, on définit l'ensemble $\text{Pre}(w)$ des préfixes de w et l'ensemble $\text{PP}(w)$ des préfixes propres de w de la façon suivante :

$$\begin{aligned} \text{Pre}(w) &= \{u \in V^* \mid \exists v \in V^*,\ uv = w\}, \\ \text{PP}(w) &= \{u \in V^* \mid u \in \text{Pre}(w) \text{ et } u \neq w\}. \end{aligned}$$

Par exemple, si $V = \{a, b\}$ et $w = bbaab$, alors :

$$\begin{aligned} \mathrm{Pre}(w) &= \{\varepsilon, b, bb, bba, bbaa, bbaab\}, \\ \mathrm{PP}(w) &= \{\varepsilon, b, bb, bba, bbaa\}. \end{aligned}$$

On définit $M \subseteq V_{\mathrm{pf}}^*$ comme étant l'ensemble des mots $w \in V_{\mathrm{pf}}^*$ tels que :
- $|w|_{\mathrm{Num}} = 1 + |w|_{\mathrm{Op}}$, et
- $\forall w' \in \mathrm{PP}(w) \setminus \{\varepsilon\}$, $|w'|_{\mathrm{Num}} > |w'|_{\mathrm{Op}}$.

On veut prouver que $L = M$.

▷ **Question 3** *(2 points) Montrer que $L \subseteq M$.*

▷ **Question 4** *(1 point) Montrer que si $w \in M$ est de la forme $w'x$ où $x \in Num$, alors $w = x$.*

On considère maintenant un mot $w \in M$ de la forme $w'x$, où $x \in \mathrm{Op}$. On pose $w' = w_1w_2$, où $w_1, w_2 \in V_{\mathrm{pf}}^*$ et :
- $|w_1|_{\mathrm{Num}} = |w_1|_{\mathrm{Op}} + 1$,
- pour tout $v \in \mathrm{PP}(w_2) \setminus \{\varepsilon\}$, $|w_1v|_{\mathrm{Num}} > |w_1v|_{\mathrm{Op}} + 1$.

▷ **Question 5** *(2 points) Montrer que w_1 et w_2 sont tous deux dans M et en déduire que $M \subseteq L$.*

PARTIEL DU MOIS DE JANVIER 2013

Algorithmique 1

Durée : 3h
Machines électroniques interdites **Tous documents papier autorisés ; hors livres**

Les deux 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. Le barème est donné à titre indicatif.

I - Le problème du sac à dos. (8 points)

On s'intéresse ici à un problème classique en informatique : le problème du sac à dos.

De voyage dans un pays lointain et fabuleux, un touriste décide de faire un maximum d'achats avant de rentrer chez lui. Malheureusement, la compagnie aérienne impose une limite de poids P sur ses bagages et il lui faut donc effectuer un choix complexe sur les objets à rapporter. Notre touriste considère donc les n types d'objets qu'il aime et associe à chacun d'eux :
- un poids p_i ;
- une qualité q_i.

Chaque type d'objet peut être acheté zéro, une ou plusieurs fois. L'objectif du touriste est d'effectuer le choix d'objets qui maximise la qualité totale de ce qu'il rapporte tout en respectant les contraintes de poids imposées. Nous désignons par $x_i \in \mathbb{N}$ le nombre de fois où l'objet de type i est acheté. Plus formellement, on cherche à maximiser $\sum_{i=1}^{n} q_i \times x_i$ sous la contrainte $\sum_{i=1}^{n} p_i \times x_i \leq P$.

Lors d'une exécution, lorsque certains x_i ne sont pas encore affectés, on dit que l'on dispose d'une *solution partielle.* Lorsque tous les x_i sont affectés on dispose alors d'une *solution complète.* Tout solution complète validant la contrainte de poids est appellée *solution valide* et tout solution valide de qualité maximale est appellée *solution optimale.*

Les p_i et q_i sont stockés dans deux tableaux de n entiers strictement positifs.

Poids	2	3	1
Qualité	11	15	2

$$P = 7$$

$x_1 = 2, x_2 = 1, x_3 = 0$: solution optimale (qualité : 37)

$x_1 = 0, x_2 = 2, x_3 = 1$: solution valide (qualité : 32)

$x_1 = 1, x_2 = 1, x_3 = 1$: solution valide (qualité : 28)

$x_1 = 2, x_2 = 1, x_3 = 1$: solution invalide (poids : 8)

$x_1 = 1, x_2 = 1$: solution partielle (x_3 n'est pas encore affecté)

FIGURE 1 – Exemple de sac à dos ; différentes solutions

Dans le reste de l'exercice, on utilise les déclarations suivantes :

```
-- type des poids et qualités
type Tab is array (Positive range <>) of Positive;
type Choix is record
   X : Natural;      -- un x_i
   A : Boolean;      -- vrai <=> x_i a été affecté
end record;
type Sol is array (Positive range <>) of Choix;
```

▷ **Question 1 (0,5 points) :**

Écrire une fonction :

```
function Max (Poids : Tab; I : Positive; P : Natural) return Natural
```

prenant en entrée le tableau contenant les poids, le numéro `I` d'un type d'objets ainsi que la limite autorisée `P` et renvoyant le nombre maximal d'objets de type `I` que l'on peut choisir en respectant la limite de poids.

On va chercher désormais à écrire une procédure récursive

```
procedure Aff_Sols_Rec (S               : in out Sol;
                        Poids, Qualité  : in     Tab;
                        P               : in     Natural)
```

réalisant l'affichage de toutes les solutions valides.

▷ **Question 2 (0,5 points) :**

Écrire dans un premier temps une procédure :

```
procedure Aff_Sols (Poids, Qualité : in Tab;
                    P              : in Natural)
```

réalisant l'affichage de toutes les solutions valides à l'aide d'un appel à `Aff_Sols_Rec`.

▷ **Question 3 (4 points) :**

Écrire le code de `Aff_Sols_Rec`. À chaque appel récursif, la procédure choisit un élément de S non encore affecté et essaie de lui affecter les différentes valeurs possibles. Elle continue alors récursivement pour les choix des éléments restants. On fera attention à vérifier les contraintes de poids le plus tôt possible.

▷ **Question 4 (3 points) :**

Proposez une procédure `Affiche_Solution_Optimale` récursive ou utilisant une procédure récursive affichant *une* solution optimale.

II - Immersion dans une usine de production. (12 points)

Cet exercice comprend deux parties. La première manipule des caisses et leur ordre d'arrivée, la deuxième s'occupe de la gestion des lots.

Situons le contexte : vous êtes ingénieur dans une société de services et venez d'être envoyé en mission au sein d'une usine de fabrication de boissons sucrées.

Gestion de la file de caisses (6 points)

Dans un premier lieu, on vous envoie passer l'entretien dans le service d'empaquetage des caisses.

Une caisse a une taille (nombre maximal de bouteilles qu'elle peut contenir), un numéro de lot représenté par une chaîne de 10 caractères, un nombre de bouteilles sodas et un nombre de jus de fruits.

Introduisons donc la structure de données suivante :

```
subtype NumLot is String (1 .. 10);
type Caisse is record
   Taille : Natural;
   NL : NumLot;
   NbSodas : Natural;
   NbJus : Natural;
   EstPleine : Boolean;
end record;
```

On souhaite définir une priorité sur les caisses de la manière suivante :

1. Si une caisse est pleine par rapport à une autre, elle est prioritaire.
2. Dans le cas où on ne peut pas déterminer, la caisse ayant le plus de sodas est plus prioritaire.
3. Si elles ont le même nombre de sodas, celle qui a le plus de jus de fruits est prioritaire.

▷ **Question 5 (1 points) :**

Redéfinissez les fonctions < et <= spécifiées ci-dessous

```
function "<" (C1, C2 : Caisse) return Boolean
function "<=" (C1, C2 : Caisse) return Boolean
```

À l'heure actuelle, la file des caisses en ligne sur le tapis est modélisée par un tableau `T` de taille `N`, de type `File` :

```
type File is array (1 .. N) of Caisse;
T : File;
```

tel que `T(1)` représente la première caisse et `T(N)` la dernière par ordre d'arrivée dans la chaîne de production.

Vous devez trier cette file (tableau) pour que la caisse la moins prioritaire soit au début (dans la case `T(1)`) et la caisse la plus prioritaire soit à la fin (dans `T(N)`).

Le client, après discussions en interne, refuse d'effectuer un tri de tableau car une autre entité de l'usine utilise ce tableau tel quel. Il n'est donc pas possible de modifier l'ordre des éléments dans `T`. Il faut donc trouver un moyen de « trier » `T` sans déplacer ses éléments.

On propose la solution suivante : on crée une « surcouche » qui permet d'accéder à `T` à l'aide d'une liste simplement chaînée :

```
type PointeurCaisse is access Caisse;

type Cellule;
type Liste is access Cellule;

type Cellule is record
   Val : PointeurCaisse;
   Suiv : Liste;
end record;
```

Chaque cellule comporte un champ qui pointe vers une caisse (case du tableau).

Pour récupérer un pointeur vers un élément, on utilise l'attribut **Access** comme sur l'exemple suivant :

```
Ptr : PointeurCaisse;
A : Array (1..10) of Caisse;
begin
    Ptr := A(5)'Access; -- Ptr pointe vers la caisse de la case 5
end;
```

▷ Question 6 (1 points) :

Ecrire une fonction qui construit la liste chaînée à partir du tableau (sans tri).

```
function ContruireListe return Liste
-- utilise le tableau T (variable globale) pour la construction
```

▷ Question 7 (2,5 points) :

Ecrire une procédure qui trie la liste chaînée sans modifier `T`, sans affectation des champs `Val`, et avec un nombre constant (0, 1, 2... indépendant de la taille du problème) de cellules supplémentaires (sentinelles éventuelles...)

```
procedure TrierListe (L : in out Liste)
```

▷ Question 8 (1,5 points) :

Proposez une autre solution pour remédier au problème du tableau en place.
Énoncez les avantages et inconvénients de la solution (complexité algorithmique, temps, clarté, ...).

Gestion des numéros de lot (6 points)

Vous avez acquis de la bouteille, et on vous envoie cette fois dans la branche de gestion de lots.

Un numéro de lot est une chaîne de caractères associée à une ou plusieurs caisses, qui sert à tracer la production de celles-ci. On peut à tout moment déclencher un nouveau numéro de lot (changement d'un paramètre machine, un certain nombre de caisses créés, une anomalie détectée, un différent type de boisson proposée, ...).

En cas de souci sur une bière, ce numéro de lot permet de rappeler toutes les caisses ayant le même numéro de lot et étant potentiellement concernées par le même souci.

Le client a un souci sur l'ensemble de ses numéros de lots (Batch Number). Ces numéros sont actuellement formattés sur le sous-type `NumLot` vu précédemment de la manière suivante :

BYYYYWWDMC

Avec

- **B** la lettre B, comme préfixe (pour Batch - Lot)
- **YYYY** l'année, sur quatre caractères
- **WW** la semaine de l'année, de 01 à 52, sur deux caractères
- **D** le jour de la semaine, de 1 à 7, sur un caractère
- **M** une lettre de marquage machine, sur un caractère
- **C** un compteur incrémental entre 0 et 9, sur un caractère

Par exemple, « B2012422Q2 » représente le 2e lot au jour numéro 2 de la semaine 42 de l'année 2012 (pour la machine de marquage Q). Le premier lot de la journée a toujours pour compteur 0, le suivant 1, etc... le lendemain, le premier lot a pour compteur 0, et ainsi de suite.

Le nombre de caractères est primordial et doit impérativement être de 10 au total. En effet, ce numéro est imprimé sur des étiquettes de taille fixe lues par des scannettes. Ces numéros de lots sont censés être uniques.

De même, cette fonction est située à des endroits critiques et ne doit jamais bloquer l'execution du programme, quitte à renvoyer un faux numéro de lot.

On fournit les fonctions suivantes, qui retournent les valeurs *à l'instant t*

```
function AnneeCourante return Integer
-- retourne l'année en cours, soit 2013 aujourd'hui

function SemaineCourante return Integer
-- retourne la semaine en cours dans l'année
-- soit 4 aujourd'hui (semaine du 21 au 27 Janvier)

function JourCourant return Integer
-- retourne le jour de la semaine, soit 4 aujourd'hui (jeudi)

function Marquage return String
-- retourne le marquage de la machine utilisée
```

```
function EstLettre (C : Character) return Boolean
-- retourne vrai si le caractère est une lettre

function Entier (Str : String; D, F : Positive) return Integer
-- retourne Integer'Value (Str (Str'First - 1 + D .. Str'First - 1 + F))
```

Exemple pour la dernière fonction : `Entier("B2012422Q2", 4, 6)` retourne l'entier 124

▷ **Question 9 (2 points) :**

Ecrire une fonction qui retourne la chaîne du lot suivant à partir d'un numéro de lot d'origine. En cas d'ambiguïtés, faites des choix et justifiez-les.

```
function NouvNumLot (NL : NumLot) return NumLot
-- retourne un numéro de lot de forme BYYYYWWDMC avec les
-- valeurs du jour et le compteur générés à partir de l'ancien
-- numéro de lot NL
```

Le client n'est pas satisfait de sa gestion de lots, principalement car il compte installer des machines avec des marquages compris entre une et trois lettres.

On impose donc le marquage suivant

BYYWWDxxxx

Avec

- **B** la lettre B, comme préfixe (pour Batch - Lot)
- **YY** l'année, sur deux caractères (les deux derniers, 13 pour 2013)
- **WW** la semaine de l'année, de 01 à 52, sur deux caractères
- **D** le jour de la semaine, de 1 à 7, sur un caractère
- **xxxx** quatre caractères pour le marquage ET le compteur

Exemples pour les quatre derniers caractères : pour une lettre, B009, pour deux lettres, VK08.

Par exemple, « B13012AA08 » représente le huitième lot au deuxième jour de la première semaine de l'année 2013 (pour la machine de marquage AA).

Nous allons créer une procédure calculant le nouveau numéro de lot à partir de l'ancien, indifféremment de son format. Une telle opération est complexe, c'est pourquoi votre responsable technique vous introduit les notions de tests unitaires et de TDD.

« En programmation informatique, le test unitaire est un procédé permettant de s'assurer du fonctionnement correct d'une sous-partie déterminée d'un logiciel. »

« Le Test Driven Development (TDD) est une technique de développement de logiciel qui préconise d'écrire les tests unitaires avant d'écrire le code source d'un logiciel. »

Ici on considère un test unitaire très simple, qui prend en entrée un ancien numéro de lot et un nouveau numéro de lot définis par l'utilisateur, ainsi que les valeurs que vont retourner les fonctions données plus haut. Ce test applique la procédure que nous allons créer, et est enregistré comme positif si le résultat concorde (i.e. comme négatif sinon).

Pour simplifier, on suppose que le test unitaire est écrit, qu'il va chercher ses entrées dans un fichier texte avec plusieurs lignes de la forme :

```
{NLA, NLN, A, S, J, M}
```

où les valeurs sont, dans l'ordre :
- l'ancien numéro de lot
- le nouveau numéro de lot attendu
- la valeur retournée par AnneeCourante
- la valeur retournée par SemaineCourante
- la valeur retournée par JourCourant
- la valeur retournée par Marquage

Par exemple, ce fichier texte :

```
{B2012422Q2, B2012422Q3, 2012, 42, 2, Q}
{B2012422Q3, B2012422Q4, 2012, 42, 2, Q}
```

Contient deux tests unitaires avec les valeurs mentionnées, et donne deux comptes rendus positifs si la fonction testée retourne bien le nouveau numéro de lot

▷ **Question 10 (1 points) :**

Écrire plusieurs cas de tests pertinents pour valider la fonction dans son comportement normal et son comportement aux limites. Envisagez d'abord les cas classiques, puis passez aux cas limites. Vous ne serez pas noté sur la quantité, mais sur la pertinence.

▷ **Question 11 (0,5 points) :**

Ecrire la fonction suivante

```
function TailleMarquage (NL : NumLot) return Positive
-- requiert : NL est un numéro sous le nouveau format
```

▷ **Question 12 (2,5 points) :**

Ecrire une fonction qui retourne la chaîne du lot suivant à partir d'un numéro de lot d'origine *quel que soit son format* (ancien / nouveau).
Attention au(x) zéro(s).

```
function NouveauNumLot (NL : NumLot) return NumLot
-- retourne un numéro de lot de forme BYYYYWWDMC avec les
-- valeurs du jour et le compteur générés à partir
-- de l'ancien numéro de lot NL
```

Examen final

Vendredi 25 janvier 2013 - 2h

Documents manuscrits et polycopié du cours autorisés. Tout autre document interdit.

Exercice 1

Soient A une application linéaire de $\mathbb{R}^n$ dans $\mathbb{R}^n$, b un vecteur de $\mathbb{R}^n$, $c \in \mathbb{R}$. Le produit scalaire dans $\mathbb{R}^n$ est noté $\langle , \rangle$. Calculer la différentielle en 0 de l'application F de $\mathbb{R}^n$ dans $\mathbb{R}$ définie par

$$F(x) = \exp[\frac{1}{2} \langle Ax, x\rangle + \langle b, x\rangle + c].$$

Exercice 2

Pour $n \in \mathbb{N}$ et $x \in]0,1]$ on pose

$$f_n(x) = \exp(-n \sin^2(\frac{1}{x}))\,.$$

Montrer que $f_n \in L^1([0,1])$ et étudier la limite de la suite :

$$I_n = \int_0^1 f_n(x)\,dx\,.$$

quand $n \to \infty$.

Exercice 3

L'objectif de cet exercice est de résoudre, pour $g \in L^1(\mathbb{R})$ donnée, l'équation

$$(E) \qquad f(x) - \frac{1}{2}\int_x^{x+1} f(s)ds = g(x)$$

1. *Soit χ la fonction caractéristique de $[-1,0]$ et K l'application qui à une fonction h associe $\chi * h$.*
   (a) *Montrer que pour $h \in L^1(\mathbb{R})$, $\|Kh\|_{L^1} \leq \|h\|_{L^1}$.*
   (b) *En déduire que la série $\sum_{n=0}^{+\infty} 2^{-n}K^n h$ converge dans $L^1(\mathbb{R})$.*
2. *Exprimer l'équation (E) au moyen de l'opérateur K.*
3. *Donner une solution de (E) dans $L^1(\mathbb{R})$. Que peut-on dire de l'unicité de la solution de (E)?*

Exercice 4

Calculez les intégrales suivantes en vous aidant des transformées de Fourier usuelles ($a > 0$, $w \in \mathbb{R}$) :

$$I = \int_0^{+\infty} e^{-ax}\cos(\omega x)\,dx$$

$$J = \int_0^{+\infty} e^{-ax} \sin(\omega x)\, dx$$

$$K = \int_{-1}^{1} (1 - |x|) \cos(\omega x)\, dx$$

Indications :

- Pour calculer I et J, commencer par calculer $I - iJ$.

- Pour calculer K, commencer par calculer la transformée de Fourier de la fonction "triangle' (fonction qui vaut $1 - |x|$ sur $[-1, 1]$ et 0 sinon).

Architecture 1 : Circuits Numériques et Éléments d'Architecture
Examen

ENSIMAG 1A

Année scolaire 2011–2012

- Durée : 3h. Tous documents et calculatrices autorisés ;
- Le barème est donné à titre indicatif ;
- Les exercices sont indépendants et peuvent être traités dans le désordre, et certaines questions au sein du même exercice peuvent aussi être traitées indépendamment ;
- Pensez à indiquer votre numéro de groupe sur chacune de vos copies.

Ex. 1 : Questions de cours (2 pts)

Question 1 Rappellez, en une phrase pour chacun d'elles, les deux conditions qui garantissent le déterminisme d'un automate d'états.

Question 2 Le processeur Core i7 d'Intel possède un cache de niveau 1 de 32 KOctets, organisé sous la forme de 512 lignes constituée chacune de 8 mots de 64 bits. Le bus d'adresse du processeur est sur 48 bits, ce qui est étrange, mais c'est ainsi. Les données manipulées par le processeur sont de 64 bits, et non 32, à la différence de l'exemple du cours.

Donnez la position dans les 48 bits d'adresse et le nombre de bits de chacun des champs *offset*, *index*, et *tag*. On pourra dessiner les 48 bits d'adresse et noter les champs sur le dessin, comme fait en cours.

Ex. 2 : Nombre de bits à 1 dans un vecteur de bits (4 pts)

Dans cet exercice, on souhaite concevoir un circuit combinatoire qui compte le nombre de bits à 1 dans un vecteur de k bits (v_{k-1} à v_0). Ce nombre, noté n, est codé en binaire sur i bits (n_{i-1} à n_0).

Question 1 On considère le cas $k = 4$. Combien vaut i dans ce cas ? Donnez les équations minimisées pour les i bits de n. On utilisera des tables de Karnaugh pour minimiser les équations.

Question 2 On considère un vecteur de 8 bits. De plus, on considère que le circuit de la question précédente est fournit : on le nommera "Count4". Expliquez comment calculer le nombre de bits à 1 d'un vecteur 8 bits en utilisant deux composants Count4. Donnez le schéma du circuit.

Ex. 3 : Conception Partie-Contrôle/Partie Opérative d'un calculateur de racine cubique (8 pts)

Comme toutes les opérations d'exponentiation, la racine cubique est une opération gourmande en ressources et en temps. Nous vous proposons d'étudier l'algorithme suivant :

```
   // x est le nombre entier positif dont on cherche la racine cubique.
1  s ← 30;
2  y ← 0;
3  while s ≥ 0 do
4  |  a ← 2 * y + 4 * y;
5  |  b ← 2 * y + 1;
6  |  c ← a * b;
7  |  d ← c + 1;
8  |  e ← d << s;
9  |  s ← s − 3;
10 |  if x ≥ e then
11 |  |  x ← x − e;
12 |  |  y ← b;
13 |  else
14 |  |  y ← 2 * y;
15 |  end if
16 end while
   // y contient alors la racine cubique de x.
```

On va utiliser la méthode présentée en cours et mise en œuvre en TD pour proposer une implantation de cet algorithme sous la forme PC/PO. On suppose que toutes les variables sont sur 32 bits.

Question 1 Définissez l'ensemble des opérations à réaliser ; déduisez-en le type des unités fonctionnelles (registres ou opérateurs). On cherchera à utiliser les opérateurs *les plus simples* (c'est-à-dire qui implantent une opération arithmétique ou logique et une seule) pour chaque opération. On rappelle que les soustracteurs peuvent fournir un indicateur : n qui vaut 1 si le résultat est strictement négatif, 0 sinon.

Question 2 Proposez une partie opérative n'utilisant qu'un additionneur/soustracteur, un multiplieur, un décaleur à barillet (ou décaleur programmable ou barrel shifter) et autant de décaleurs par puissance de 2 que nécessaire (simple problème de connexions).

Question 3 Revenons à l'algorithme. Quelles opérations peut-on réaliser en parallèle ?
Réécrivez le programme en utilisant la notion d'affectation concurrente présentée en cours.
Rappel. : $(\alpha, \beta) \leftarrow (3, 1)$ affecte 3 dans α et 1 dans β.
En déduire le nombre de cycles d'horloge qu'il faudrait à la partie contrôle pilotant cette PO pour exécuter les lignes 4 à 15 de cet algorithme.

Question 4 On s'autorise l'ajout d'un incrémenteur. Proposez une nouvelle PO en indiquant clairement les signaux de commandes (sélecteurs de mux, signaux de chargement de registres, etc) et les comptes-rendus.

Question 5 Proposez un graphe d'automate d'état décrivant la partie contrôle qui pilote cette partie opérative. Spécifiez la valeur des différents signaux de commande pour chaque état (vous pouvez donner des valeurs par défaut pour simplifier l'écriture).

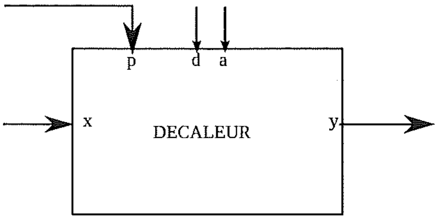
Question 6 Modifiez l'algorithme en utilisant la notion d'affectation conditionnelle pour faire disparaître le **if**.
Pour mémoire : $\nu \leftarrow \gamma ? \alpha : \beta$ affecte α dans ν si $\gamma = 1$, β sinon.

Question 7 En supposant qu'il est possible de réaliser 3 opérations consécutives de type addition, soustraction, décalage dans un même cycle d'horloge[1], montrez qu'il est possible de réaliser le corps de boucle (lignes 4 à 15) en 3 cycles d'horloge en réécrivant l'algorithme en conséquence.
Combien d'opérateurs sont maintenant nécessaires dans la PO ?

Ex. 4 : Conception de processeur (6 pts)

On cherche à ajouter au processeur vu durant les TD 9, 10 et 11 une instruction de décalage programmable. Pour mémoire, le jeu d'instructions, la partie opérative ainsi que la partie contrôle (sans signaux et conditions) sont rappelées en annexe.

On s'intéresse d'abord au décalage à gauche. On utilisera un décaleur programmable de type décaleur à barillet, vu en TD.



d	a	y
0	0	x << p décalage à gauche
1	0	x >> p décalage arithmétique à droite
1	1	x >>> p décalage logique à droite

L'instruction de décalage permet le décalage à gauche d'un registre rd d'après la valeur contenue dans un autre registre rs . Elle se note `shl rs, rd` et effectue $rd \leftarrow rd << rs$.

Question 1 Proposez un encodage de la nouvelle instruction `shl` (qui remplace l'ancienne instruction `shl`).

Question 2 Insérez le décaleur à barillet dans la partie opérative du processeur (vous pouvez compléter le schéma fourni, mais n'oubliez pas d'indiquer votre nom et de le rendre ! ! !). Est-il nécessaire de modifier la partie contrôle ?

Question 3 A l'aide de cette instruction, programmez la multiplication de la variable A, qui est à l'adresse 0x1000 en mémoire, par la valeur décimale constante 28 (= 16 + 8 + 4).

Pour simplifier ce type de programme (opérations avec des constantes), on rajoute une instruction qui permet de charger un registre (8 bits) avec une constante (8 bits également). Elle se note `li IMM, rd` (*load immediat*), et effectue $rd \leftarrow IMM$.

Question 4 Indiquez où doit se trouver la constante à charger dans le registre et proposez un encodage de l'instruction `li` qui s'insère simplement dans l'encodage existant.

Question 5 Est-il nécessaire de modifier la partie opérative ? Si oui, effectuez les modifications. Si non, justifier votre réponse.

Question 6 Donnez la partie contrôle qui commande la partie opérative pour cette nouvelle instruction. Vous pouvez simplement ajouter la partie manquante sur l'annexe et lister les états, conditions et valeurs des signaux qui sont modifiés et/ou ajoutés.

1. Réaliste car la multiplication prend plus de temps que ces opérateurs

Question 7 A l'aide de cette nouvelle instruction, programmez à nouveau la multiplication de A par 28. Ce nouveau programme est-il plus rapide ?

On s'intéresse maintenant à exploiter tous les possibilités du décaleur à barillet : décalage à gauche, à droite, arithmétique ou non.

Question 8 Faut-il une ou plusieurs instructions de décalage ? proposez un codage.

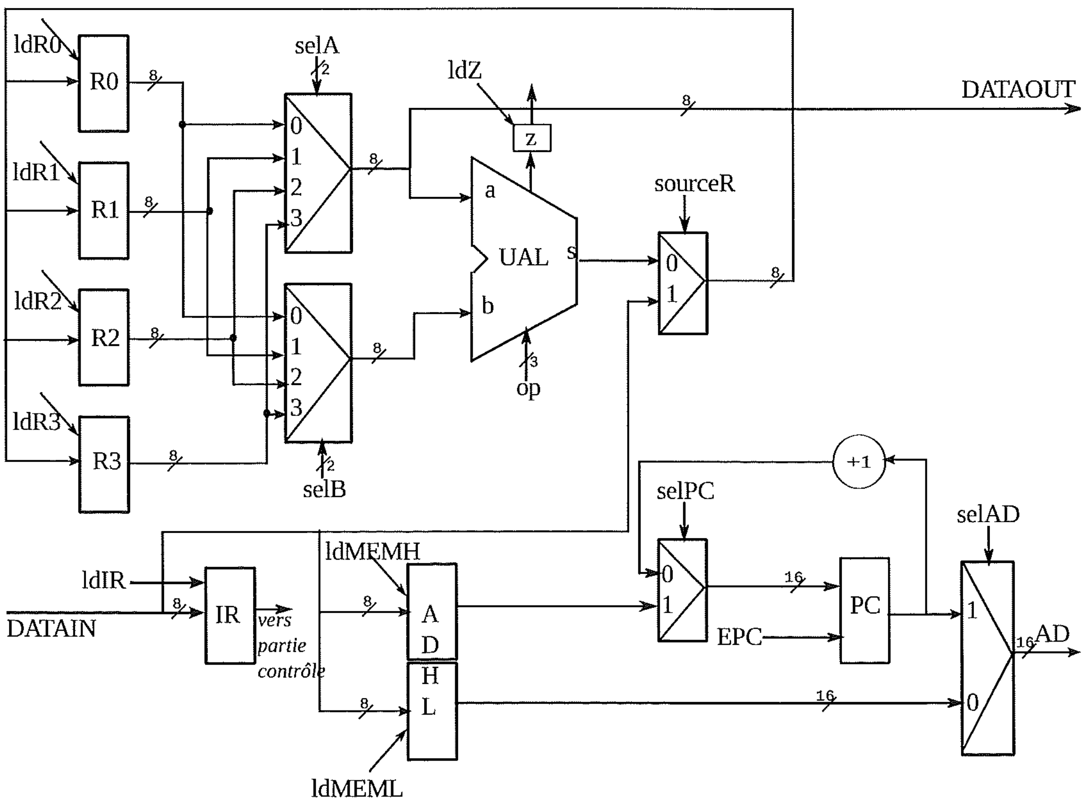
Question 9 Modifiez la partie opérative et la partie contrôle pour pouvoir exécuter ces décalages.

Annexe

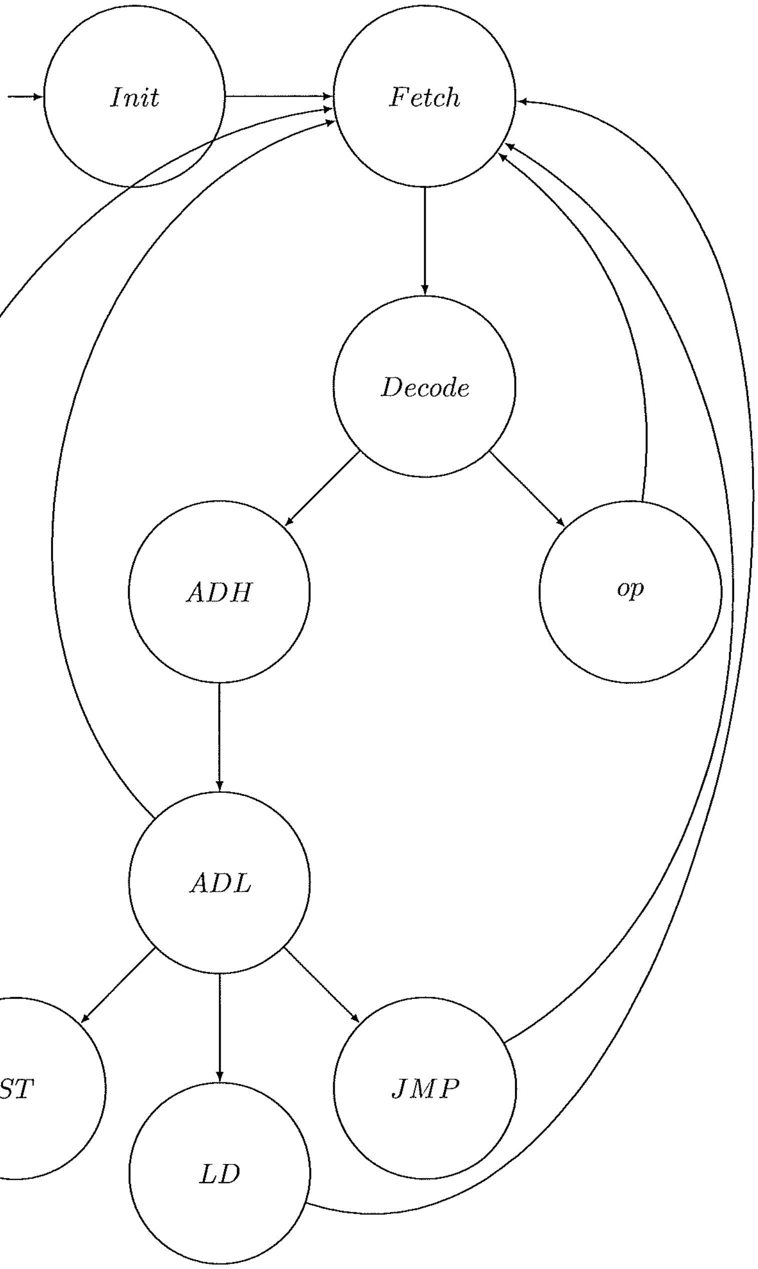
Jeu d'instructions et encodage

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							
Branchements conditionnels : si Z=1 alors $PC = AD$								
`jz AD`	1	0	0	0	0	1	0	1
	ADH							
	ADL							

Partie opérative



Partie Contrôle



ENSIMAG 1ère année | Année 2012/2013

ECONOMIE

DS du 25 janvier 2013

Durée : 2 heures
Documents autorisés

Travail demandé :

Choisir **un** des deux sujets proposés.
Quel que soit le sujet retenu, il vous est demandé de rédiger un commentaire libre du texte associé au sujet **ou** de répondre aux questions posées (Spécifiez vos choix en début de copie).
Veillez dans tous les cas à structurer votre propos, à soigner la rédaction et à mobiliser au maximum vos connaissances personnelles.
Au total votre travail ne devra pas dépasser **4 pages**.

SUJET n°1

La BCE laisse ses taux d'intérêt inchangés

Source : Reuters
08/11/2012

A l'issue de la réunion de son conseil des gouverneurs, la BCE a, comme attendu, laissé son taux de refinancement - principal instrument de sa politique monétaire - fixé à 0,75%, son plus bas niveau historique auquel l'institution l'a ramené début juillet.

Cette décision sans surprise intervient après la publication d'une série d'indicateurs reflétant la fragilité économique des Etats de la zone euro et qui plaidaient plutôt en faveur d'un assouplissement monétaire. Mais une telle décision aurait été susceptible d'alimenter les critiques, en Allemagne notamment, à l'adresse du président de l'institution, Mario Draghi. Pour calmer les esprits, la BCE a donc privilégié le statu quo ce jeudi, en dépit d'une baisse de la confiance des entreprises, du recul du secteur manufacturier de la zone euro pour le quinzième mois consécutif.

« *L'activité économique dans la zone euro devrait rester faible même si elle est toujours soutenue par notre politique monétaire et la confiance des marchés financiers s'est visiblement accrue en réaction à nos décisions* », a déclaré Mario Draghi.

La BCE abaissera d'ailleurs ses prévisions de croissance le mois prochain, a-t-il précisé au lendemain de la révision à la baisse de celles de la Commission européenne.

Il a également précisé que l'inflation dans la zone euro devrait rester au-dessus de l'objectif de la BCE jusqu'à la fin de l'année et tomber en-dessous de 2% en 2013.

La plupart des 73 analystes interrogés par Reuters la semaine dernière ont dit s'attendre à ce que le taux de refinancement soit abaissé à 0,5% dans les prochains mois en raison de la dégradation des perspectives de croissance en zone euro. « *L'économie européenne a besoin d'être relancée. Si ce n'est pas la BCE qui le fait, qui le fera ?* », a commenté Joseph Trevisani, responsable de la stratégie de Worldwide Markets.

Questions (si vous n'avez pas choisi le commentaire libre)

1) Rappelez comment la décision d'une Banque Centrale relative au niveau du taux directeur affecte l'ensemble d'une économie nationale ou internationale (dans le cas d'une monnaie unique telle l'Euro).

2) Qu'attend-on d'une baisse du taux de l'intérêt ?

3) Quelles sont les limites d'une politique de relance monétaire ?

SUJET n°2

Le bonheur est dans la croissance

Claudia Senik, Ecole d'économie de Paris, Université Paris-Sorbonne

L'Express, le 19/06/2008

Comment mesurer le bien-être des individus ? Le niveau des revenus n'est plus un critère suffisant. Car l'homme espère toujours davantage et ne peut s'empêcher de comparer sa situation à celle de son entourage. En réalité, son sort dépend surtout de l'évolution économique. Le chef de l'Etat a réuni un collège d'économistes prestigieux, présidé par un Prix Nobel, Joseph Stiglitz, afin de réfléchir aux indicateurs les mieux à même de mesurer le bonheur. Le célèbre PIB (produit intérieur brut) laissera-t-il place, un jour, au BIB - bonheur intérieur brut ? Pourquoi pas, mais sur quoi se fonderait ce nouvel indice ?

De leur côté, les Nations unies, sous l'autorité d'Amartya Sen, autre Prix Nobel membre de la commission Stiglitz, ont proposé un nouvel indicateur de « développement humain » (l'IDH), qui donne à la santé et à l'éducation un poids égal au PIB. A cette aune, le classement des Nations unies place l'Islande au sommet de la hiérarchie. Pourtant, dans ce pays, certes magnifique, il fait nuit la moitié du temps, le tourisme et la pêche sont à peu près les seules activités et l'on compte moins de 3 habitants au kilomètre carré...

Une autre méthode, plus directe, consiste à analyser le bien-être subjectif déclaré par les individus lors d'enquêtes nationales. Exemple : « *Sur une échelle de 1 à 5, sur quel échelon vous situez-vous, en matière de bonheur ?* » De telles études existent déjà. Ainsi, l'économiste Richard Easterlin a montré que, depuis l'après-guerre, le score moyen de bonheur subjectif déclaré par la population était resté à peu près constant, et cela malgré l'augmentation spectaculaire du revenu moyen dans les pays développés. Comment est-ce possible ?

On note tout d'abord - rien d'étonnant - que, au sein d'un pays, les riches sont plus heureux que les pauvres. Les habitants des pays riches sont également plus heureux que ceux des pays pauvres. L'argent semble faire le bonheur, finalement... Cependant, deux facteurs viennent gâcher l'effet du revenu sur le bien-être subjectif : l'accoutumance et les comparaisons. Au lieu de jouir de son niveau de vie, l'homme s'y habitue et espère davantage. Il se compare aussi à ses collègues, à ses amis, à ses voisins, et évalue son bien-être à l'aune de celui de ses pairs. Finalement, la course au revenu est une illusion, car la satisfaction que l'on en retire est relative à une échelle qui elle-même évolue (Ces idées sont développées dans « *La croissance rend-elle heureux ?* » de Claudia Senik et Andrew Clark, dans 27 Questions d'économie contemporaine, sous la direction de Philippe Askenazy et Daniel Cohen, publié chez Albin Michel).

Et pourquoi pas, si la course est en soi une source de bonheur ? Psychologues, économistes et même neurologues ont mis en évidence le goût des individus pour la progression en tant que telle. Ceux qui ont été le plus heureux (qui connaissent le score de bien-être cumulé le plus élevé) sont ceux qui ont le plus souvent vécu ou même simplement anticipé une progression de leur niveau de vie. Dans ce cas, pourquoi changer d'indicateur ? La croissance reste bonne à prendre. C'est son absence qui rend malheureux.

Questions (si vous n'avez pas choisi le commentaire libre)

1) Pourquoi dit-on généralement que le PIB n'est pas, à l'échelle d'une société, un indicateur de « bien-être » voire de bonheur ?

2) Cette critique du PIB implique-t-elle forcément une remise en cause de la croissance économique et donc l'adoption par nos sociétés d'un objectif de « décroissance » ?

Grenoble INP – ENSIMAG
Année 2012-2013

N° d'inscription (carte étudiant) :

NOM :

Prénom :

Né(e) le : **à (ville + dépt ou pays) :**

N° du groupe de TD:

N° de la place :

Introduction aux réseaux

Examen du 23/01/2013. Durée 2h30
Tous documents (sauf livres) et calculettes autorisés

NOTE IMPORTANTE : vous devez répondre sur les feuilles de ce sujet d'examen (pages 1 à 5) pour la partie QCM. Vous traiterez le problème du hotspot sur une copie d'examen séparée.
La durée de chaque partie devrait être approximativement de 1 h pour le QCM et 1h30 pour le problème.

Partie 1. Questions à Choix Multiples

Modalités des QCM
Pour chaque QCM, une seule des solutions proposées est correcte, c'est-à-dire une réponse adéquate à la question posée (certaines « solutions » peuvent être des affirmations vraies sans répondre à la question). Entourez le numéro (alphabétique) de la réponse exacte.
Barème :
Réponse exacte : 1 point / Réponse fausse : -0,5 point / Omission : 0 point.

Question 1. *Lorsque l'on chiffre un message avec une clef publique (du récepteur), que garantit-on?*
A. On ne garantit rien, c'est insuffisant.
B. On en garantit l'authenticité.
C. On garantit la confidentialité du contenu.
D. On ne peut rien garantir puisque cette clef est publique.

Question 2. *On mesure pendant 10 minutes : 12 communications de 30 secondes et 4 communications d'une minute sur un commutateur téléphonique comportant un total de 1024 lignes. Quel est le trafic enregistré au sein du commutateur ?*
A. 10 Erlang
B. 0,666 Erlang
C. 1 Erlang
D. 0,65 mErlang
E. 9,8 mErlang

Question 3. *Soit un réseau de téléphonie cellulaire utilisant une méthode d'accès basée sur la famille de codes d'étalement orthogonaux utilisée en UMTS et sur un duplexage fréquentiel. La transmission s'effectue avec un débit chip fixe Dc=1 Mchips/s de la station de base vers les mobiles. L'utilisateur A reçoit ses données en utilisant le code $C_{ch,4,2}$ et l'utilisateur B reçoit ses données en utilisant le code $C_{ch,8,3}$. Quel sont les débits descendants bruts de A et B ?*

A. D_A = 2 Mbit/s et D_B = 3 Mbit/s
B. D_A = 500 kbit/s et D_B = 333 kbit/s
C. D_A = 250 kbit/s et D_B = 125 kbit/s
D. D_A = 4 Mbit/s et D_B = 8 Mbit/s
E. Cela dépend aussi des codes affectés pour la voie montante.

Question 4. *On considère un système de transmission émettant un débit symbole de 20 Mbauds. Il sert à transmettre un débit binaire de 60 Mbit/s. Cela signifie qu'on utilise une modulation à combien d'états ?*

A. 3 états
B. 6 états
C. 8 états

Question 5. *Un système radio permet initialement de transmettre un signal à 2 Mbit/s avec une modulation QPSK. Le taux d'erreurs garanti est de 10^{-3}. Le système évolue ensuite pour transmettre un débit de 3 Mbit/s. On utilise alors une modulation de type PSK8. Pour garantir le même taux d'erreurs binaires, il faut modifier la puissance d'émission. De quelle valeur ? (On donne la valeur numérique approchée de $10\log(2/3) \approx -1,8$)*

A. 2,2 dB
B. 3 dB
C. 4 dB
D. 4,8 dB

Question 6. *Quel protocole utilise-t-on pour trouver une adresse de niveau 4 ?*

A. Aucun protocole.
B. DHCP
C. ARP
D. DNS
E. Une combinaison d'ARP et de DNS

Question 7. *Considérons un réseau cellulaire utilisant le FDMA (on se limite à l'étude de la voie descendante). L'opérateur dispose de 20 canaux fréquentiels et utilise des motifs de 4 cellules. Il constate une saturation fréquente de son réseau. Que peut-il faire ?*

A. Augmenter la puissance d'émission de ses stations de bases.
B. Choisir un motif à 7 cellules pour augmenter la réutilisation des fréquences.
C. Augmenter la taille des intervalles de gardes entre canaux fréquentiels de manière à limiter l'interférence entre utilisateurs.
D. Diminuer la taille des cellules.

Question 8. *Une modulation numérique de type QAM est :*

1) une modulation de fréquence d'une porteuse sinusoïdale
2) une modulation hybride d'amplitude et de phase d'une porteuse sinusoïdale
3) une modulation d'amplitude d'une porteuse sinusoïdale additionnée à une modulation d'amplitude de la même porteuse déphasée de $\pi/2$
4) une modulation d'amplitude d'une porteuse sinusoïdale additionnée à une modulation d'amplitude de la même porteuse déphasée de π

Parmi ces propositions, laquelle (ou lesquelles si elles sont plusieurs) est (ou sont) correcte(s) ?

A. 1
B. 2
C. 3
D. 4
E. 2 et 3
F. 2 et 4
G. 1 et 2

Répondez par Vrai ou Faux aux 4 propositions données dans les questions suivantes (barème pour ces 4 dernières questions : réponse correcte = 0,5 point / réponse incorrecte =-0,25 point)

Question 9. *Les diagrammes de l'œil 1 et 2 ont été obtenus lors de la simulation d'un système de transmission binaire en bande de base utilisant un code NRZ et un filtrage de Nyquist, avec dans un cas un coefficient d'arrondi égal à 0,2 et dans l'autre un coefficient d'arrondi égal à 0,8.*

Proposition : « Le diagramme de l'œil 1 correspond au filtrage avec le coefficient d'arrondi égal à 0,2. »

A. Vrai
B. Faux

Question 10. *Les diagrammes de l'œil 3 et 4 ont été obtenus lors de la simulation d'un système de transmission binaire en bande de base utilisant un code NRZ, avec un canal ajoutant du bruit. Dans un cas, le rapport E_b/N_o valait 6 dB et dans l'autre, il valait 10 dB.*

Proposition : « Le diagramme de l'œil 4 correspond à E_b/N_o égal à 10 dB. »

A. Vrai
B. Faux

Question 11. *Le diagramme de l'œil 5 a été obtenu lors de la simulation d'un système de transmission binaire en bande de base utilisant un code NRZ, avec un canal qui n'ajoute pas de bruit.*

Proposition : « Le critère de Nyquist est respecté dans ce système. »

A. Vrai
B. Faux

Question 12. *Le diagramme de l'œil 6 a été obtenu sur les voies en phase et en quadrature en sortie du démodulateur, lors de la simulation d'un système de transmission utilisant une modulation de type QAM.*

Proposition : « La modulation utilisée est de type QAM16. »

A. Vrai
B. Faux

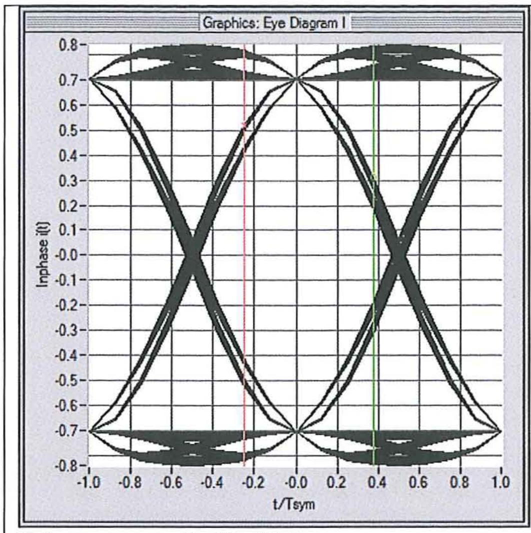


Diagramme de l'œil 1

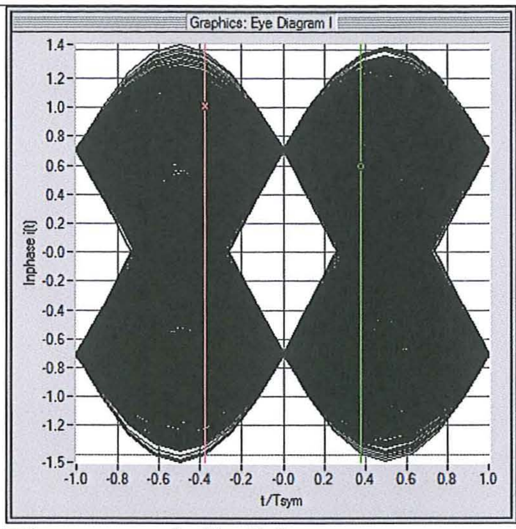


Diagramme de l'œil 2

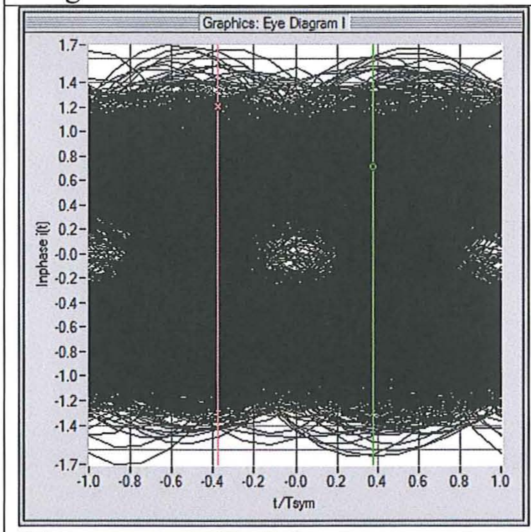


Diagramme de l'œil 3

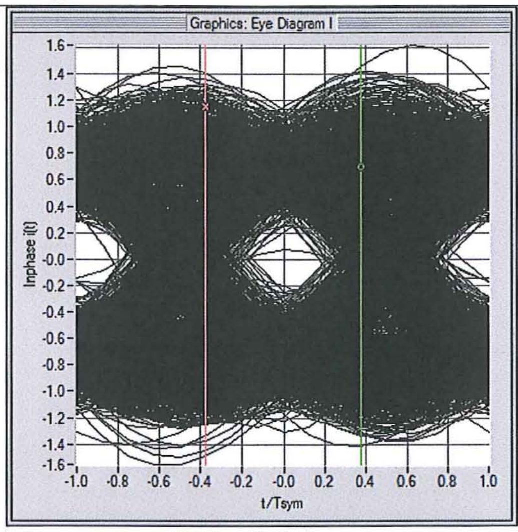
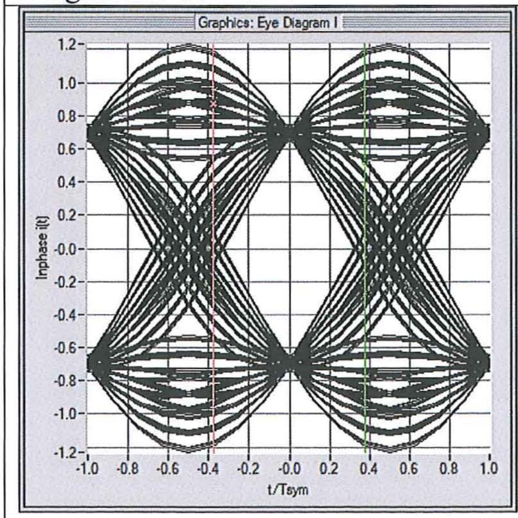
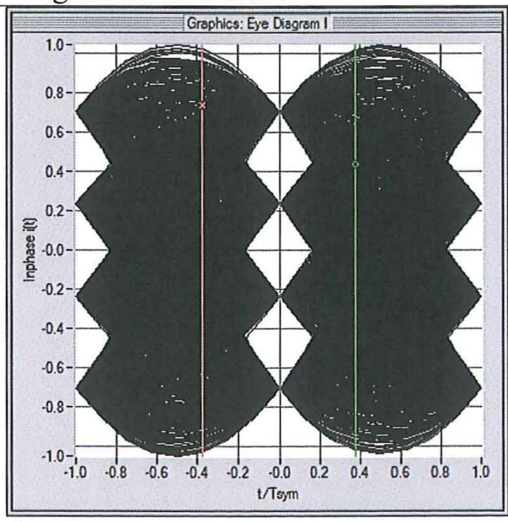


Diagramme de l'œil 4



Cas 5



Cas 6

Partie 2 : Hotspot Wifi à péage

Présentation du service

Dans de très nombreux lieux publics, tels que les hôtels, gares et aéroports, des opérateurs ont installés des services d'accès sans fil par Wifi. On appelle communément hotspot un tel service correspondant à un réseau local sans fil dans une zone restreinte à un bâtiment.

On s'intéresse ici à un service payant mis en place par un opérateur appelé « Citron ».

Le service vu de l'utilisateur doté d'un ordinateur portable est le suivant. Il détecte un réseau sans fil annoncé sous le nom : « Citron_wifi ». Il indique à son ordinateur d'utiliser cette connexion de réseau sans fil. Il ne peut pas accéder à des services sur Internet tant qu'il n'a pas lancé son navigateur. Ce dernier affiche alors non pas la page d'accueil habituelle mais la page d'accueil d'un portail « Wifi payant » de Citron. L'utilisateur doit alors acheter un temps de communication (typiquement une ou plusieurs heures) sur ce site. À partir du moment où il a payé, le réseau Citron_wifi le connecte alors à l'Internet et il peut librement accéder à tous les services d'Internet.

Fonctionnement simplifié pour l'examen

Votre professeur de réseau, Roland Groz, se connecte depuis un ordinateur ordi-groz dans une gare. L'adresse IP de ordi-groz obtenue au démarrage de la connexion Wifi est 172.18.249.232. Il obtient aussi l'adresse d'un serveur DNS 172.18.249.193 (nom canonique DNS : hotspot_Citron_LaGare, alias hsp_lg, d'adresse IP publique 80.10.45.12 sur le réseau Internet). Cette même machine route également les communications entre la zone Wifi et l'Internet.

En fait, ordi-groz communique avec une borne Wifi qui est un pont ayant une interface radio Wifi et une interface Ethernet filaire pour la raccorder à hsp_lg.

Le routage assuré par hsp_lg est filtrant : au départ, tant qu'un ordinateur n'a pas payé, hsp_lg ne fait sortir aucune communication sur l'Internet, et redirige tout vers ses propres serveurs, en fonction du port. Hsp_lg n'écoute en fait que sur 2 ports : 80 (HTTP) et 53 (DNS).

Lorsque ordi-groz lance son serveur Web Firefox dont la page d'accueil est voila.fr (193.252.148.80), c'est donc le serveur httpd de hsp_lg qui, jouant le rôle d'un « proxy » récupère cette demande. Il répond en fournissant une réponse encapsulée dans un paquet IP paraissant venir de voila.fr (l'adresse IP source est bien 193.252.148.80) mais contenant un retour HTML :

```
302 HTTP/1.1 Moved Temporarily
Location :
http://hotspots.Citron.fr/home?CPURL=http%3A%2F%2Fvoila.fr
```

La machine hotspots.Citron.fr est le portail « Wifi payant » de Citron, d'adresse IP 80.10.46.232. La connexion TCP depuis le hotspot vers la machine hotspots.Citron.fr n'est pas filtrée, donc ordi-groz peut se connecter à cette page. Là il peut remplir un formulaire avec les éléments lui permettant de payer, et il achète un crédit de connexion de 1h. Lorsqu'il valide sur le formulaire son crédit, hotspots.Citron.fr informe (par un protocole applicatif de type RADIUS dont on ne détaillera pas les PDU) hsp_lg de ne plus filtrer les communications de ordi-groz, et renvoie vers ordi-groz la page de voila.fr qu'il avait initialement demandée.

Architecture

Dans cette partie, vous devez représenter l'architecture statique en couches de toutes les machines mentionnées dans cet exemple (quelles que soient les couches qu'elles implémentent). On représentera les différentes couches, et on détaillera la couche application. Dans la couche application, lorsqu'une application utilise plusieurs protocoles, on représentera dans des rectangles superposés : le nom de l'application (par ex Firefox, Webmail etc) et en dessous les protocoles qu'elle utilise, tels que HTTP, DNS... Par exemple :

Couche Application	Firefox	
	HTTP	DNS

Précisez bien les adresses IP, les ports, et par des liens traversant les couches les connexions établies au cours de ce scénario.

Question 1. *Faites un diagramme d'architecture sur lequel figurent toutes les machines mentionnées ci-dessus, avec les couches, les applications et les protocoles.*

Borne Wifi

La borne Wifi installée par l'opérateur Citron met en oeuvre les standards IEEE 802.11a/g/n. Les spécifications plus détaillées concernant IEEE802.11g sont données ci-dessous.

Bande de fréquences	• 2,412 à 2,462 GHz (FCC) • 2,412 à 2,472 GHz (ETSI) • 2,412 à 2,472 GHz (TELEC)

Modulation sans fil	**802.11g** OFDM (Orthogonal Frequency Divisional Multiplexing) : • BPSK à 6 et 9 Mbits/s • QPSK à 12 et 18 Mbits/s • 16 QAM (Quadrature Amplitude Modulation) à 24 et 36 Mbits/s • 64 QAM à 48 et 54 Mbits/s
Protocole d'accès au support	CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)
Canaux utilisés	802.11b/g : • ETSI : 13 • Amériques : 11 • TELEC (Japon) : 13
Canaux sans chevauchement	3
Sensibilité en réception	6 Mbits/s : -90 dBm 9 Mbits/s : -89 dBm 12 Mbits/s : -86 dBm 18 Mbits/s : -84 dBm 24 Mbits/s : -81 dBm 36 Mbits/s : -77 dBm 48 Mbits/s : -73 dBm 54 Mbits/s : -72 dBm

Question 2. *Expliquez pourquoi il y a seulement 3 canaux fréquentiels qui ne se chevauchent pas.*

Question 3. *Un autre utilisateur qui se connecterait en même temps que Roland Groz à la même borne WiFi pourra-t-il utiliser le même canal fréquentiel que lui ? Si oui, comment les communications de ces deux utilisateurs sont-elles séparées ? Donnez le nom de la technique qui permettrait cela et expliquez brièvement son principe de fonctionnement.*

Question 4. *La modulation utilisée est de type OFDM. Quel est le principe de l'OFDM ? Cette modulation est-elle facilement réalisable dans les émetteurs et récepteurs radio et si oui, pourquoi ?*

Question 5. *Expliquez comment et pourquoi les modulations utilisées varient en fonction des débits désirés.*

Question 6. *A votre avis, comment est définie la sensibilité en réception ? Et expliquez quelle raison fait que la sensibilité se dégrade lorsque le débit augmente.*

Déroulement des protocoles applicatifs pour l'accès

Pour les questions demandant à illustrer le déroulement des protocoles applicatifs, on utilisera des diagrammes temporels (ou « diagrammes de séquence ») du type de ceux utilisés en cours pour illustrer le protocole du bit alternant ou la connexion TCP. On fera figurer un axe temporel (flèches verticales dans le cours) par machine impliquée. Sur les flèches en biais figurant les messages, on portera le type du PDU et une information minimale synthétique sur les paramètres essentiels par rapport au déroulement expliqué (par exemple pour le DNS : A ? suivi du nom d'une machine, A ? faisant référence à une « query » de type « Address »).

Question 7. *Représentez sur un diagramme temporel l'enchaînement des PDUs des protocoles de la couche application (et uniquement eux) entre le moment où Roland Groz lance son navigateur et le moment où il voit apparaître la page d'accueil de Citron l'invitant à acheter du temps de connexion.*

Question 8. *Sur un diagramme temporel qui s'enchaîne avec le précédent, représentez les PDUs de couche application entre le moment où Roland Groz a confirmé son achat (démarrant son heure de crédit) et l'apparition de la page voila.fr sur son navigateur.*

Voila.fr était la page d'accueil, mais Roland Groz choisit maintenant dans ses signets (favoris, « bookmarks ») d'accéder à google.fr, d'adresse IP 66.249.93.104.

Question 9. *Représentez les PDUs de couche application jusqu'à l'affichage de la page de Google.*

Question de réflexion

Question 10. *Pourquoi laisse-t-on dans un hotspot un accès au service DNS ? Il y a plusieurs éléments de réponse.*

2012-2013

Ensimag 1A

Examen de Probabilités appliquées
Mercredi 23 janvier 2013

Durée : 2h00.
Document autorisé : 1 feuille A4 manuscrite recto-verso.
Calculatrices et téléphones portables interdits.
Merci d'inscrire votre numéro de groupe sur la copie.
Le sujet comporte 2 pages. Le barème est indicatif. Les 3 exercices sont indépendants et peuvent être traités par le candidat dans l'ordre qu'il préfère. Tout élément de vérification des résultats sera très apprécié.

Exercice 1. [5pts]
Soient N et M deux variables aléatoires indépendantes suivant une loi de Poisson de paramètres respectifs $\lambda > 0$ et $\mu > 0$.
C'est à dire, pout tout $n \in \mathbb{N}$, $P(N = n) = \frac{\lambda^n}{n!} e^{-\lambda}$.

1. Soit $S = M + N$. Calculer, pour $k \in \mathbb{N}$, $P(S = k)$. Quelle est la loi de la variable aléatoire S ?
2. Pour tout $n \in \mathbb{N}$, montrer que les lois conditionnelles de M sachant $S = n$ et de N sachant $N = n$ sont des lois binomiales dont on précisera les paramètres.
3. Dans le rapport de projet de fin d'études d'un étudiant de l'ENSIMAG, on peut trouver deux types d'erreurs : les fautes de frappes et les fautes de grammaire. On suppose que le nombre N de fautes de frappes suit une loi de Poisson de paramètre λ et que le nombre M de fautes de grammaire suit une loi de Poisson, indépendante de celle de N, de paramètre μ.
   (a) Combien d'erreurs trouve-t-on en moyenne dans un rapport ?
   (b) Un correcteur a relevé k erreurs dans le rapport d'un étudiant ($k \in \mathbb{N}^*$). Quelle est la probabilité que ces k erreurs soient des fautes de grammaire ? Quelle est la probabilité qu'aucune de ces k erreurs ne soit une faute de grammaire ?
   (c) Calculer la valeur numérique de la moyenne de la question (a) lorsque $\lambda = 4$, $\mu = 6$.

Exercice 2. [8pts]
Soit U une variable aléatoire suivant une loi uniforme sur $[0, 1]$.

1. On définit la variable aléatoire X par $X = \frac{1 - U}{1 + U}$.
   (a) Quelles sont les valeurs prises par la variable aléatoire X ?

(b) Calculer la fonction de répartition F_X de la variable aléatoire X.

(c) Vérifier que la fonction F_X vérifie bien les propriétés d'une fonction de répartition.

(d) Calculer l'espérance de la variable aléatoire X.
Indication : $\dfrac{1-t}{t+1} = -1 + \dfrac{2}{t+1}$.
Vérifier la cohérence de votre résultat. *Rappel :* $\ln 2 \sim 0,69$.

2. On suppose qu'on a un algorithme de simulation pour X. On réalise l'expérience suivante : on répète de manière indépendante la simulation de X jusqu'à ce que la condition $\left(X < \dfrac{1}{2}\right)$ se réalise. On appelle T la variable aléatoire alors obtenue.

   (a) A quel type d'algorithme de simulation la construction de T fait-elle appel ? Déterminer la fonction de répartition de la variable T.

   (b) Soit N le nombre de simulations de X nécessaires pour que la condition $\left(X < \dfrac{1}{2}\right)$ se réalise dans l'agorithme précédent. Quelle est la loi de N ? Quelle est l'espérance de N ?

   (c) Calculer l'espérance de la variable aléatoire T.
   Indication : $1 - \dfrac{3t}{t+1} = -2 + \dfrac{3}{t+1}$.

Exercice 3. [7pts]
Soit (X, Y) un couple de variables aléatoires admettant la densité f définie sur $\mathbb{R}^2$ par $f(x, y) = c(y-x)\mathbb{1}_{0<x<y<1}$

1. Calculer c.
2. Caculer la densité f_X de la loi marginale de X.
3. En déduire que l'espérance de la variable aléatoire X vaut $\dfrac{1}{4}$.
4. Calculer, pour $x \in]0, 1[$ la densité $f_Y^{X=x}$ de la loi conditionnelle de Y sachant $X = x$.
5. Montrer que, pour $x \in]0, 1[$, l'espérance conditionnelle $E(Y|X = x)$ vaut $\dfrac{x+2}{3}$. En déduire $E(Y|X)$.
6. Quelle est la valeur de $E(Y)$? Effectuer le calcul par deux méthodes différentes.

Théorie des langages 1

Durée : 2h.
Documents : tous documents autorisés.

Nb : le barême est donné à titre indicatif; la notation prendra en compte la clarté de la rédaction.

Problème - Grammaires et expressivité (12 points)

On s'intéresse à la déclaration de types dans un langage inspiré d'ADA. On considère la grammaire G suivante :

`decl_type`	$\rightarrow$	`type_simple` | `type_tab`
`type_tab`	$\rightarrow$	**array**[`bornes`] **of** `type_simple`
`type_simple`	$\rightarrow$	**integer** | `bornes`
`bornes`	$\rightarrow$	**cst .. cst**

avec $V_T = \{\textbf{array}, [,], \textbf{of}, \textbf{range}, \textbf{integer}, \textbf{cst}, \textbf{..}\}$ et $V_N = \{\texttt{decl\_type}, \texttt{type\_tab}, \texttt{type\_simple}, \texttt{bornes}\}$, `decl_type` étant l'axiome.

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

▷ **Question 2** (2 points)
Montrer que le langage $\mathcal{L}(G)$ est régulier en donnant une expression régulière qui le décrit.

▷ **Question 3** (2 points)
On veut maintenant pouvoir déclarer des tableaux dont les éléments peuvent également être des tableaux, comme par exemple : **array[cst..cst] of array[cst..cst] of integer**.

Modifier la grammaire G afin de prendre en compte cette extension. De quel type est le langage $\mathcal{L}(G)$? Justifier votre réponse.

▷ **Question 4** (2 points)
On considère maintenant la grammaire G', étendant G en permettant de déclarer des enregistrements. Cette grammaire a comme vocabulaires $V'_T = V_T \cup \{\textbf{record}, \textbf{end}, \textbf{idf}\}$ et $V'_N = V_N \cup \{\texttt{type\_enr}, \texttt{liste\_champ}, \texttt{champ}\}$ et contient les règles supplémentaires suivantes :

`type_enr`	$\rightarrow$	**record** `liste_champ` **end**
`liste_champ`	$\rightarrow$	`champ` `liste_champ` | `champ`
`champ`	$\rightarrow$	**idf :** `decl_type` ;
`decl_type`	$\rightarrow$	`type_enr`

Modifier la grammaire ci-dessus pour prendre en compte la possibilité de déclarer des listes d'identificateurs dans les champs d'enregistrement, comme par exemple **record x, y, z : integer ; end**. On fera attention au positionnement des virgules.

▷ **Question 5** (3 points)
Montrer que le langage engendré par la grammaire G' n'est pas régulier. On admettra que le langage $\{a^n b^n \mid n \geq 1\}$ n'est pas régulier.

▷ **Question 6** (2 points)

On s'intéresse maintenant aux expressions associées aux enregistrements. Soit G_2 la grammaire suivante :

$$\begin{array}{rcl} \texttt{exp} & \rightarrow & \texttt{place} \\ \texttt{place} & \rightarrow & \texttt{place}.\textbf{idf} \mid \textbf{idf} \end{array}$$

avec $V_T = \{\textbf{idf}\ ,.\}$ et $V_N = \{\texttt{exp}, \texttt{place}\}$, `exp` étant l'axiome.

On désire étendre le langage des expressions pour prendre en compte l'opérateur $+$, celui-ci étant associatif à gauche. La grammaire proposée devra être non ambiguë et exhiber la bonne structure des expressions. Donner l'arbre de dérivation associé à l'expression `x+y.z`. **On ne demande pas de prouver la non-ambiguïté de la grammaire proposée.**

Exercice 1 - Raisonnement sur les grammaires hors-contexte (8 points)

Soit $V_T = \{a, b\}$ et L le langage des mots sur V_T contenant autant de a que de b ($L = \{w : w \in V_T^* \wedge\ |w|_a = |w|_b\}$). Soit G_1 et G_2 deux grammaires telles que $L(G_1) = L$ et $L(G_2) = L$ (résultat admis).On s'intéresse ici à établir l'ambiguïté ou la non-ambiguïté de ces grammaires.

Grammaire G_1 :

$\texttt{S} \rightarrow$ **aSb** | **bSa** | S S | ϵ

Grammaire G_2 :

$\texttt{S} \rightarrow$ **aAbS** | **bBaS** | ϵ $\qquad$ $\texttt{A} \rightarrow$ **aAbA** | ϵ $\qquad$ $\texttt{B} \rightarrow$ **bBaB** | ϵ

Rappel : soit u une chaîne sur $(V_T \cup V_N)^*$. On rappelle que $L(u)$ est le langage défini par $L(u) = \{w : w \in V_T^* \wedge\ u \Longrightarrow^* w\}$ et que $L(G) = L(S)$, S étant l'axiome.

▷ **Question 1** (1 point)

Montrer que la grammaire G_1 est ambiguë en utilisant une chaîne de longueur 4.

▷ **Question 2** (2 points)

Donner un arbre de dérivation de la chaîne $aababb$ par la grammaire G_2.

▷ **Question 3** (2 points)

On pose :

$$L_A = \{w : w \in V_T^* \wedge\ |w|_a = |w|_b \wedge\ \forall u \in prefixe(w)\ .\ |u|_a \geq |u|_b\}$$

Et on admet que $L_A = L(A)$ avec A le non-terminal de la grammaire G_2. Montrer la propriété suivante :

$$(w = au_1bu_2 \wedge u_1 \in L_A \wedge u_2 \in L_A \wedge\ w = av_1bv_2 \wedge v_1 \in L_A \wedge v_2 \in L_A) \Rightarrow (u_1 = v_1 \wedge\ u_2 = v_2)$$

Cette propriété garantit que la règle $\texttt{A} \rightarrow$ **aAbA** décompose tout mot $w \in L(aAbA)$ de manière unique : c'est-à-dire $(aAbA \Longrightarrow^* w) \Rightarrow (\exists\ !\ u_1, u_2\ (w = au_1bu_2 \wedge\ A \Longrightarrow^* u_1 \wedge A \Longrightarrow^* u_2))$. A l'opposé la règle $\texttt{S} \rightarrow \textbf{SS}$ de la grammaire G_1 n'a pas cette propriété. Par exemple la chaîne $abbaab$ peut se décomposer en $S \Longrightarrow^* ab$ et $S \Longrightarrow^* baab$ ou bien en $S \Longrightarrow^* abba$ et $S \Longrightarrow^* ab$.

▷ **Question 4** (3 points)

On veut maintenant montrer que la grammaire G_2 n'est pas ambiguë, c'est-à-dire que chaque mot du langage ne peut être décrit que par un unique arbre de dérivation. Une preuve possible revient à montrer que (on admettra la correction de cette preuve pour prouver la non ambiguïé) :

1. pour chaque non-terminal les langages associés aux différentes parties droites de règles sont disjoints deux à deux
2. toute règle de grammaire décompose un mot de $L(\alpha)$ de manière unique, α désignant la partie droite de la règle (voir question précédente)

Appliquer cette démarche pour établir le caractère non ambiguë de la grammaire G_2. On décrira précisément les conditions à vérifier et leur justification (propriétés à prouver).

PARTIEL DU MOIS DE MAI 2013 – SESSION 1

ALGORITHMIQUE 2 -	6 PAGES
METHODES NUMERIQUES -	3 PAGES
PLITIQUES ECONOMIQUES -	3 PAGES
PRINCIPES et METHODES STATS. -	3 PAGES
RECHERCHE OPERATIONNELLE -	2 PAGES
THEORIE DE L'INFORMATION -	2 PAGES
THEORIE DE LANGAGES 2 -	2 PAGES

Algorithmique et structures de données : examen de première session

Ensimag 1A

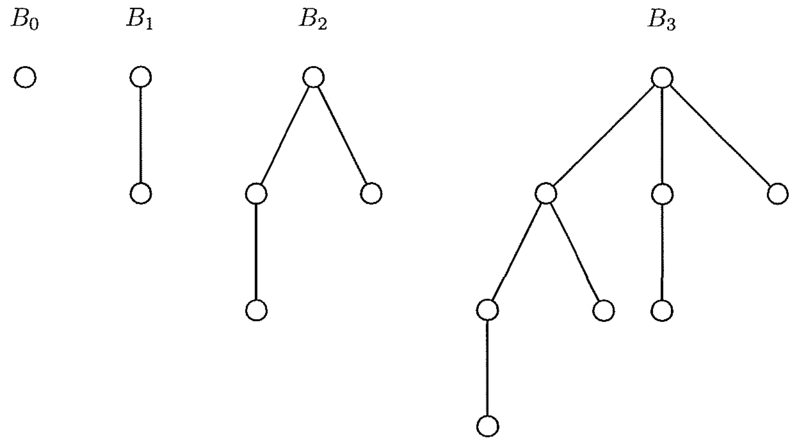
Année scolaire 2012–2013

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 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.**

1 Tas binomiaux (12 points)

Rappelons qu'un arbre au sens informatique du terme possède une *racine* r, pour un sommet u de l'arbre, tous les sommets w tels que u est le dernier sommet avant w sur l'unique chemin de r à w s'appellent les *fils* de u. L'arbre est dessiné par niveaux, tous les sommets à la même distance de la racine sont sur le même niveau et les fils de chaque sommet sont ordonnés de gauche à droite.
Un arbre binômial B_k est un arbre défini récursivement comme suit :
- B_0 est l'arbre formé d'une racine et pas d'autres sommets.
- B_k est composé de deux arbres B_{k-1} qui sont liés de la façon suivante : la racine de l'un est le fils le plus à gauche de la racine de l'autre.

La figure suivante donne B_0, B_1, B_2 et B_3.



Question 1 : Dessiner B_4. **(0.5 point)**
Question 2 : (On pourra considérer comme acquises les propriétés suivantes pour la suite du problème si on n'a pas réussi à faire les démonstrations qui sont toutes très simples) Montrer que pour B_k :

1. Il y a 2^k sommets. **(0.5 point)**
2. La hauteur (ou profondeur) est de k. **(0.5 point)**
3. Il y a exactement C_k^i sommets au niveau i. (Les niveaux sont comptés de haut en bas, le niveau 0 contient la racine et le niveau k est celui le plus bas.) **(0.5 point)**
4. La racine possède k fils. En numérotant les fils de la droite vers la gauche en commençant par 0 (donc le fils le plus à droite est numéroté 0 et le fils le plus à gauche est numéroté $k-1$), le fils numéro i est la racine d'un B_i. **(0.5 point)**

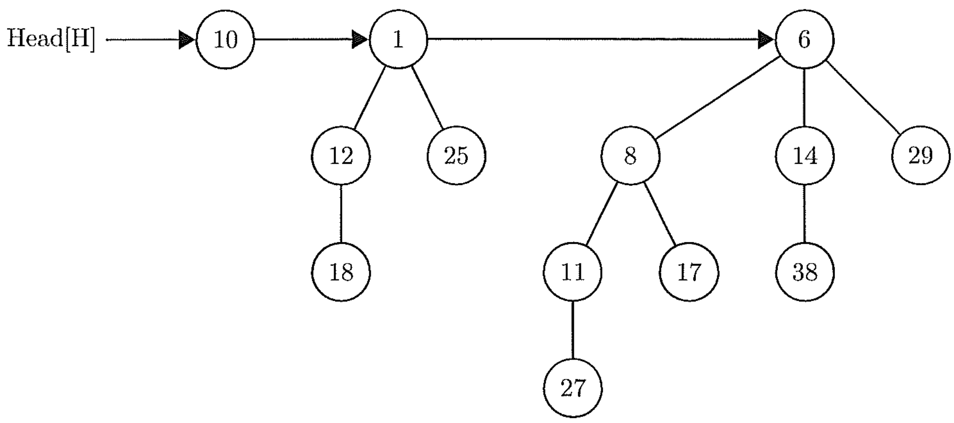
Question 3 : Déduire des propriétés précédentes que si n est le nombre de sommets d'un arbre binomial alors aucun sommet n'a plus de $\log n$ fils. **(1 point)**

Un *tas binômial H* (*binomial heap* en anglais) est un ensemble d'arbres binômiaux tels que les propriétés suivantes sont satisfaites :

1. *Min-ordonné* : Dans chaque arbre binômial de H, la valeur attachée à chaque sommet (appelée dans la suite clé) est inférieure ou égale à celle attachée à ses fils (on aurait pu définir max-ordonné)

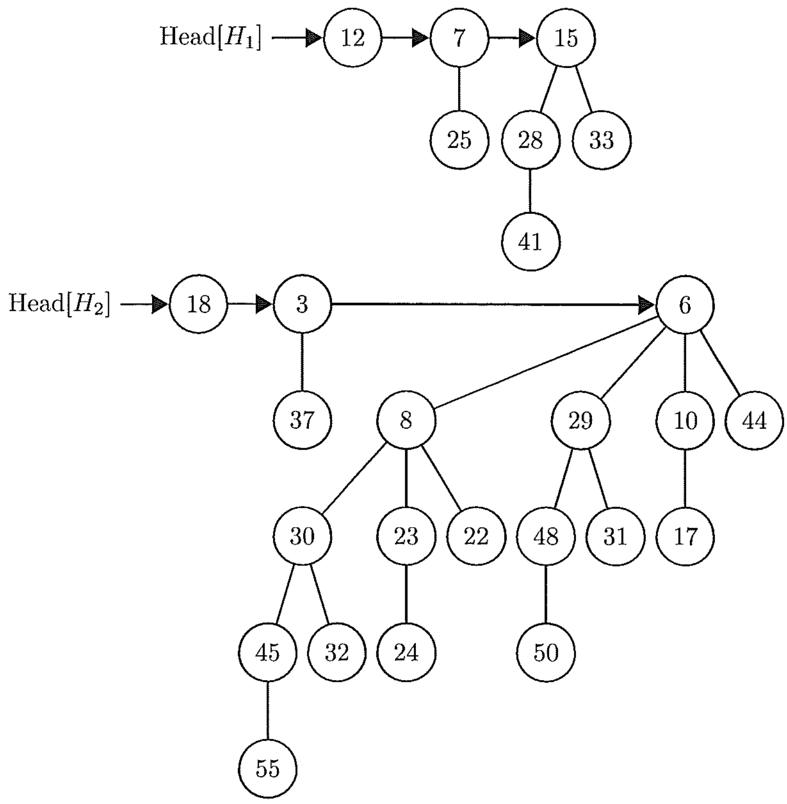
2. Pour tout entier k, il y a **au plus un** arbre binômial dans H de racine de degré k. (Au plus un veut dire au maximum un).

On utilisera la propriété suivante, qui est une conséquence de la deuxième condition ci-dessus, sans la démontrer : Dans un tas binômial H contenant n sommets en tout, il y a au plus $\lfloor \log n \rfloor + 1$ arbres. (Pour voir pourquoi, écrivons n sous forme binaire $n = \sum_{i=0}^{\lfloor \log n \rfloor} b_i 2^i$, on sait qu'un arbre binomial B_i possède 2^i sommets, donc l'arbre binomial B_i apparaîtra dans le tas H si et seulement si $b_i = 1$)
Dans un tas binômial H chaque nœud est une structure avec un lien vers son fils aîné, un lien vers son frère cadet et un autre vers son père. Dans le cas d'absence de tels parents le lien est $NULL$. De plus les racines des arbres sont dans une liste chaînée de gauche à droite (on peut utiliser pour cela le lien frère cadet des racines). La première racine est donnée par $head[H]$. Dans ce qui suit on supposera que les racines sont ordonnées de gauche à droite selon leur degré (nombre de fils), c'est ce qui est fait dans la figure suivante.



Question 4 : Ecrire l'algorithme de recherche de la clé minimum et déduire de ce qui précède que trouver la clé minimum se fait en $O(\log n)$. **(1 point)**
Question 5 : La base de toutes les opérations sur les tas binomiaux est la fusion de deux tas binomiaux. La figure ci-dessous donne deux tas binomiaux H_1 et H_2
Pour fusionner deux tas binômiaux on met les arbres des deux tas dans une même liste ordonnée par degré croissant des racines.

1. Dire pourquoi en général cela ne donne pas un tas binômial. **(0.5 point)**
2. Dans l'exemple qui vous est proposé, par des opérations que vous décrirez soigneusement vous ramener à un tas binômial. Vous pouvez simplifier les figures dans la mesure où on comprend bien ce que vous faites. Les figures ne sont qu'un support des explications mais ne les remplacent pas. **(1 point)** (Aide : essayez de maintenir la propriété que les arbres de la liste sont en ordre croissant des degrés des racines)
3. Donner les grandes lignes d'un algorithme de fusion de deux tas binômiaux. **(2 points)**
4. Quelle est la complexité de l'algorithme précédent en fonction du nombre de sommets final $n = n_1 + n_2$. **(1 point)**



Question 6 : En utilisant l'opération de fusion de deux tas comme une fonction, écrire l'algorithme d'insertion d'un nouvel élément dans un tas binômial. Complexité ? **(1.5 point)**
Question 7 : Ecrire l'algorithme d'extraction (suppression) de l'élément de clé minimum dans un tas binômial. (Cela utilise à nouveau la fusion de deux tas, mais aussi la propriété 4 de la question 2). Complexité ? **(1.5 point)**

2 Algorithme des *4 russes* (8 points)

L'idée de cet exercice est d'analyser l'algorithme dit des *4 russians* qui utilise une technique de décomposition pour multiplier deux matrices carrées booléennes. Cette opération est en particulier utile pour le calcul de la fermeture transitive d'un graphe représenté par sa matrice d'adjacence.
On cherche à concevoir un algorithme efficace pour multiplier deux matrices booléennes carrées A et B de dimension n ($C = A \times B$). Les opérations de base s'entendent ici au sens booléen, c'est-à-dire, la multiplication élémentaire comme un $\wedge$ logique et l'addition comme un $\vee$: $C_{i,j} = \vee_{k=1,\ldots,n} A_{i,k} \wedge B_{k,j}$
Question 8 : On considère le graphe à 4 sommets de la figure 1. Construisez sa matrice d'adjacence A et calculez A^2 ($A \times A$). **(0.5 point)**
Question 9 : L'algorithme usuel de produit de matrices correspond à écrire les n^2 opérations de produits des lignes de A par les colonnes de B.
Écrivez (avec un pseudo-code de bas niveau) la version de l'algorithme qui parcourt la

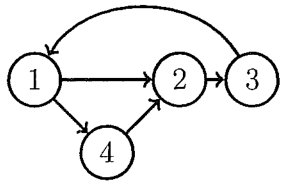


FIGURE 1 – Graphe exemple

matrice A par colonnes et B par lignes. Plus précisemment, ceci signifie que dès que l'on accède pour la première fois à une cellule de la $i^{\text{ème}}$ colonne de A, on s'interdit les accès aux colonnes d'indices inférieurs à i. Ce principe s'applique également aux lignes de B. **(2 points)**

Question 10 : L'idée de base de l'algorithme des 4 russes est de partitionner la matrice A en groupes de tailles égales de λ colonnes consécutives et B en groupes de λ lignes consécutives. Pour simplifier, on supposera ici que λ divise n. On notera A_i la matrice n par λ correspondante et B_i la matrice λ par n. La figure 2 illustre la décomposition de A et B.

Donner l'expression du produit $A \times B$ en utilisant A_i et B_i. **(0.5 point)**

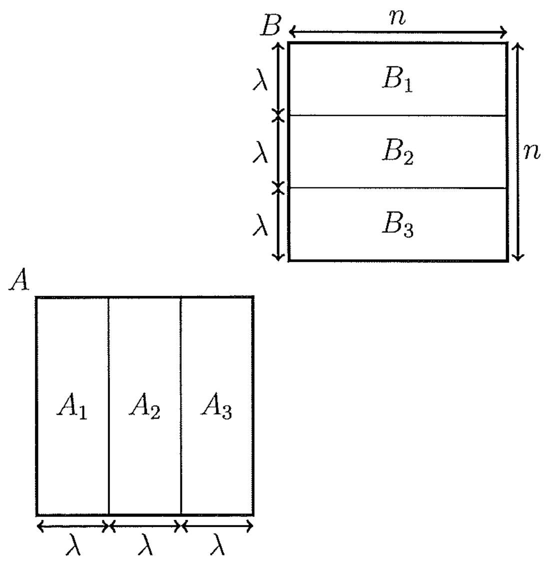


FIGURE 2 – Décomposition des matrices A et B

Question 11 : On se propose d'utiliser une fonction `Produit_Ligne_Matrice` prenant en entrée une ligne d'une matrice A_i ainsi qu'une matrice B_i (plutôt que de donner toute la matrice B_i en paramètre, il est possible de donner uniquement l'indice i et de lire ensuite les données dans un tableau global à 3 dimensions par exemple). Dans cette question on cherche à calculer $C_i = A_i \times B_i$ **pour un i fixé.** La fonction `Produit_Ligne_Matrice` est appellée n fois pour construire chacune des n lignes de C_i. En réalité de nombreux appels sont redondants, c'est à dire que la fonction est appellée plusieurs fois avec des paramètres

exactement identiques. Proposez une borne supérieure sur le nombre maximal d'appels différents possibles pour réaliser le calcul de C_i. **(1 point)**
Question 12 : On se propose de réduire le coût de l'algorithme en évitant de réaliser deux fois le même calcul à l'aide d'une technique appellée mémoïsation. Pour ce faire, on se propose de stocker tous les résultats des appels à la fonction `Produit_Ligne_Matrice` dans un dictionnaire D. Chaque entrée possible (une ligne + une matrice) correspond à une clef à laquelle se trouve associée le résultat de la fonction. On ajoute en début de la fonction `Produit_Ligne_Matrice` une conditionelle vérifiant si une valeur apparait dans D pour la clef correspondant aux arguments et retournant immédiatement le résultat s'il est déjà calculé. Dans le cas contraire, la fonction s'exécute normalement puis sauve le nouveau résultat dans D avant de le renvoyer.
Expliquez le principe d'un algorithme efficace (grâce à la mémoïsation) calculant $A \times B$. Précisez le type de dictionnaire que vous utilisez. **(2 points)**
Question 13 : Evaluez son coût au pire cas en fonction de n et λ. (détaillez le calcul) **(1 point)**
Question 14 : Proposez une valeur de λ permettant d'obtenir un temps de calcul bien meilleur que le produit classique (justifiez). **(0.5 point)**
Question 15 : La fermeture transitive d'un graphe G peut être stockée par une matrice booléenne F contenant sur la case $F(i,j)$ un booléen vrai s'il est possible d'aller de i à j et faux sinon. Si A est la matrice d'adjacence du graphe G, alors $F = (A+I)^n$ où I désigne la matrice identité et n le nombre de sommets de G. Proposez un algorithme rapide (donnez l'idée) pour calculer la fermeture transitive d'un graphe. Quel est son coût au pire cas ? **(0.5 point)**

Examen du 23 Mai 2013

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.

I. Généralisation de l'algorithme du gradient conjugué

Soit A une matrice (n, n) symétrique définie positive, b un vecteur de $\mathbb{R}^n$ non nul. On cherche à résoudre le système linéaire $Ax = b$ dont on note x^* l'unique solution. Pour résoudre ce système, on va chercher à mininiser la fonctionnelle usuelle: $J(x) =< Ax, x > -2 < b, x >$, avec $< \cdot, \cdot >$ qui désigne le produit scalaire euclidien, $(x, y) = y^T x$.
On suppose **connu** n vecteurs de $\mathbb{R}^n$, non nuls, notés $d_0, d_1, \ldots, d_{n-1}$, deux-à-deux A-conjugués i.e.

$$< Ad_i, d_j >= 0 \qquad \text{pour} \qquad i \neq j, \qquad 0 \leq i, j \leq (n-1)$$

et on écrit **l'algorithme de descente à pas optimal** selon les directions $d_0, d_1, \ldots, d_{n-1}$: étant donné un point initial quelconque $x_0 \in \mathbb{R}^n$, on effectue les iterations suivantes:

$$x_{k+1} = x_k + \alpha_k d_k \ , \qquad k = 0, 1, \ldots$$

où α_k est donc le pas de descente optimal.
On notera le résidu $r_k = Ax_k - b$.

1. Rappeler l'expression de α_k et la récurrence sur r_k.

On note $\mathcal{E}(d_0, \ldots, d_{k-1})$ le sous-espace (de dimension k) engendré par $d_0, \ldots, d_{k-1}$ et $\mathcal{E}(Ad_0, \ldots, Ad_{k-1})$ le sous-espace engendr par $Ad_0, \ldots, Ad_{k-1}$. On note également P_k l'opérateur de projection orthogonale de $\mathbb{R}^n$ dans $\mathcal{E}(d_0, \ldots, d_{k-1})$ **au sens du produit scalaire défini par** A. C'est à dire, pour tout $u \in \mathbb{R}^n$, $P_k u$ est l'unique élément de $\mathcal{E}(d_0, \ldots, d_{k-1})$ tel que:

$$< u - P_k u, \ Ad_i >= 0 \qquad i = 0, 1, \ldots, (k-1).$$

2. Vérifier que $(x_k - x_0) \in \mathcal{E}(d_0, \ldots, d_{k-1})$ et $(r_k - r_0) \in \mathcal{E}(Ad_0, \ldots, Ad_{k-1})$.

3. Montrer que: $\quad < r_k, d_k > = < r_0, d_k >, \qquad k = 0, 1, \ldots (n-1).$

4. En déduire qu'à partir d'un même point initial x_0, on atteint le même point x_k en descendant selon les directions $d_0, d_1, \ldots, d_{k-1}$ pris **dans un ordre quelconque**.

5. Montrer que: $\quad x_k - x_0 = P_k\,(x^* - x_0),$

6. a. Que se passe t'il si $(x^* - x_0) \in \mathcal{E}(d_0, \ldots, d_{k-1})$ avec $k < n$?
b. En déduire la convergence de la méthode en au plus n pas.

Nous allons montrer à présent quelques propriétés supplémentaires qui vont nous permettre de retrouver la convergence de la méthode.

7. Monter que: $\quad < r_k, d_i > = 0 \quad i = 0, 1, \ldots, k-1.$

8. Montrer que: $\quad \forall x \in \mathbb{R}^n,$

$$J(x) = J(x_k) + 2 < r_k, x - x_k > + < A(x - x_k), (x - x_k) > .$$

9. En déduire que x_k minimise la fonctionnelle $J(x)$ dans le sous-espace affine $x_0 + \mathcal{E}(d_0, \ldots, d_{k-1})$. Retrouver ainsi la convergence en au plus n pas.

Remarque: Cette méthode, dite des directions conjuguées, généralise celle du gradient conjugué. En effet, le choix des d_{k-1} peut être différent de celui de l'algorithme du G.C.

II. Méthode à pas séparés

On désire résoudre le problème de Cauchy $\begin{cases} y(a)=y_0 \\ y'(t)=f(t,y(t)) \end{cases}$ pour $t \in [a, b]$ par une méthode à pas séparés (on dit aussi méthode à un pas).
On note α, β, γ trois paramètres réels à déterminer. On pose $R_0(h) = y_k$ et $R_1(h) = y_k + \alpha f(t_x, y_k)$ et on considère la méthode à pas séparés suivante:

$$y_{k+1} = y_k + h\,(\beta f(t_k, y_k) + \gamma f(t_k + \alpha_k,\ R_1(h)))\,.$$

On supposera f Lipschitz par rapport à la deuxième variable et suffisamment dérivable par rapport à ces deux variables.

1. Donner une interprétation géométrique de la méthode entre t_k et t_{k+1}.

2. Etudier la convergence et l'ordre maximum de la méthode en fonction de α, β, γ.

3. Déterminer la méthode correspondant à $\beta = 0$. Que retrouvez-vous ?

III. Exercice

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 du nombre d'opérations arithmétiques réalisées avec cet algorithme 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 par rapport à la méthode précédente lorsque n est grand.

ENSIMAG 1ère année 2012/2013

Politique économique

DS du lundi 27 mai 2013

Durée : 2 heures
Documents autorisés

Choisir un des deux sujets suivants.

Sujet 1 : Commentaire libre du texte joint.

Sujet 2 : Les dirigeants de certains pays souhaitent doter l'ensemble de leurs pays d'une monnaie unique. Ils vous demandent ce que vous pensez d'un tel projet. Compte tenu de l'expérience européenne en la matière, qu'auriez-vous à leur dire ?

Texte associé au sujet 1

Faut-il lâcher la pression sur l'inflation ?

L'Expansion, Emilie Lévêque - le 10/03/2010

Un peu plus d'inflation serait bénéfique. Cette suggestion a été avancée récemment par les économistes du FMI. Un pavé dans la mare des banques centrales qui la rejette. Dans le contexte actuel de sortie de crise et d'envolée de la dette des Etats, ce débat très théorique sur l'inflation prend une nouvelle dimension.

Le débat est déjà virulent quoiqu'encore limité aux cercles spécialisés. Mais il pourrait être bientôt relayé auprès du grand public. A l'origine, une note publiée mi février par l'économiste en chef du Fonds Monétaire International, Olivier Blanchard. Il y évoque la possibilité pour les banques centrales de viser une inflation plus élevée en temps normal, de 4% au lieu de 2% aujourd'hui. Cette suggestion a de suite soulevé une levée de boucliers parmi les garants de l'orthodoxie monétaire. « *Le FMI joue avec le feu* », a mis en garde le président de la Bundesbank Axel Weber, membre du conseil des gouverneurs de la BCE. « *4%... comment savons-nous qu'ensuite nous n'irons pas vers 5 ou 6 ou 7%* ? », a ironisé le président de la Fed Ben Bernanke. Aussitôt posé, aussitôt rejeté. Ce débat sur la « bonne inflation » mérite toutefois d'être examiné de plus près.

Quels seraient les avantages d'une inflation plus élevée?

Dans l'esprit du néo-keynésien Olivier Blanchard, se fixer un objectif d'inflation plus lâche permettrait de faire jouer pleinement les mécanismes de politique monétaire. En clair : en période de crise, les banques centrales auraient plus de marges de manœuvre pour baisser leurs taux d'intérêt directeur, donc injecter plus rapidement des liquidités dans l'économie ou encore lui offrir une plus grande bouffée d'oxygène. Cela permettrait aussi de limiter un recours trop violent aux politiques budgétaires expansives en cas de choc sur l'économie.

Un autre argument en faveur de l'inflation, défendu par les économistes de tendance néo-keynésienne, est qu'elle permet de résorber le chômage. Ce rapport est illustré par la courbe dite de Phillips (économiste Néo-zélandais des années 1950)[1]. Schématiquement : l'inflation entraîne une hausse des salaires nominaux, donc du pouvoir d'achat des ménages alors incités à plus consommer; les entreprises augmentent en conséquence leur production et sont susceptibles de créer des emplois. (...)

Qui sont les gagnants et les perdants de l'inflation?

Contrairement à l'idée reçu, les salariés ne perdent pas en pouvoir d'achat puisque leur salaire nominal augmente. Le premier effet de l'inflation est d'éroder la valeur réelle des créances. Exemple : une inflation de 5% annule la charge des intérêts d'une dette si le taux d'intérêt est lui aussi de 5%. Si l'inflation est supérieure au taux d'intérêt de la dette, c'est la valeur même

[1] La courbe de Phillips exprime une relation inverse entre les évolutions respectives du taux de chômage et du taux d'inflation.

de celle-ci qui diminue. Un tel scénario bénéficie donc aux ménages endettés, mais aussi aux Etats.

L'inflation pénalise en revanche les créanciers « *l'inflation est l'euthanasie des rentier* »" selon Keynes - que ce soit des particuliers, des banques, des investisseurs ou des marchés financiers. Ces derniers n'ont justement qu'une seule crainte, en cette période d'explosion de la dette publique : que les Etats ne laissent volontairement filer les prix, afin de réduire le poids de leur dette.

Quels sont les dangers de l'inflation?

« *L'inflation, c'est comme la pâte dentifrice. Quand elle est sortie du tube, impossible de l'y faire rentre* »", avait coutume de dire l'Allemand Karl Otto Pöhl, président de la Bundesbank dans les années 1980. Cette phrase résume la principale crainte des banquiers centraux classiques : l'impossibilité d'enrayer un emballement des prix. L'idée est que l'inflation s'auto-alimente par des mécanismes d'anticipation et de hausse soutenue des prix. Ce phénomène irrationnel, non lié à l'activité réelle de l'économie, est appelé hyperinflation. L'épisode le plus célèbre de notre histoire moderne restera l'hyperinflation qui a frappé l'Allemagne de 1922 à 1924, ruinant des couches entières de la population. Mais soyons honnêtes, ce danger relève plus de l'ordre du fantasme de nos jours que de celui du possible pour les économies développées.

ENSIMAG 1$^{\text{ère}}$ année 23 mai 2013

Principes et Méthodes Statistiques

Durée : 3 heures. Tous documents autorisés.

Les deux parties sont indépendantes.

Les résultats vus en cours ou en TD peuvent être utilisés sans être redémontrés.

Il sera grandement tenu compte de la qualité de la rédaction (présentation et justification des réponses) dans la notation.

Barème indicatif - Partie 1 : 6 pts, Partie 2 : 14 pts.

Première partie

A l'automne 2012, une équipe de biologistes français a publié une étude visant à étudier l'impact de la nourriture à base d'Organismes Génétiquement Modifiés (OGM) sur le développement de tumeurs chez les rats. Pour cela, un échantillon de 140 rats, 70 mâles et 70 femelles, a été divisé en 4 groupes. Pour chaque sexe, un groupe témoin de 10 rats a été nourri sans OGM et un groupe de 60 rats a été nourri avec du maïs génétiquement modifié à différentes doses.

Le tableau suivant donne le nombre de rats ayant développé des tumeurs avant la fin de l'étude :

	sans OGM	avec OGM	total
mâles	2	30	32
femelles	5	44	49
total	7	74	81

1. Peut-on en déduire que la prise d'OGM augmente le risque de développer des tumeurs ?

2. Le risque de développer une tumeur est-il plus fort pour les femelles que pour les mâles ?

Les réponses à ces questions doivent être données à l'aide de tests d'hypothèses : décrire les tests utilisés en donnant les hypothèses de modélisation, donner une approximation des p-valeurs, donner les commandes `R` permettant de mettre en œuvre les tests et répondre aux questions posées. Pour la question 1, interpréter les conséquences des erreurs de première et deuxième espèce.

Deuxième partie

Une variable aléatoire X positive est dite de loi de Pareto $\mathcal{P}a(a,b)$, avec $a \geq 2$ et $b > 0$, si et seulement sa densité est :

$$f(x) = \frac{a\,b^a}{(b+x)^{a+1}},\ \forall x \geq 0.$$

1. Montrer que $E(X) = \dfrac{b}{a-1}$ et $Var(X) = \dfrac{a\,b^2}{(a-1)^2\,(a-2)}$. Calculer la fonction de répartition de X.

2. Soient $x_1, \ldots, x_n$ n réalisations de variables aléatoires $X_1, \ldots, X_n$ indépendantes et de même loi $\mathcal{P}a(a,b)$. Calculer les estimateurs de a et b par la méthode des moments.

3. Ecrire les équations donnant les estimateurs de a et b par la méthode du maximum de vraisemblance. Exprimer l'estimateur de a en fonction de celui de b et montrer que l'estimateur de b est solution d'une équation implicite.

   Dans toute la suite, on supposera que b est connu, égal à 1.

4. Calculer l'estimateur des moments $\tilde{a}_n$ de a.

5. Calculer l'estimateur de maximum de vraisemblance $\hat{a}_n$ de a.

6. Donner la loi de probabilité de $Y = \ln(1+X)$.

7. Montrer que $\hat{a}_n$ est biaisé. En déduire un estimateur sans biais $\hat{a}'_n$. Montrer que cet estimateur est convergent.

8. Donner l'expression du graphe de probabilités pour la loi $\mathcal{P}a(a,1)$. Expliquer comment calculer un estimateur graphique a_g de a.

9. On a relevé les durées en milliers d'heures entre les défaillances successives d'un système informatique, supposées indépendantes et de même loi :

   0.252 1.017 0.094 0.980 0.046 0.449

   On donne les résultats suivants en `R` :

```
> x<-c(0.252,1.017,0.094,0.980,0.046,0.449)
> y<-log(1+x)

> mean(x)
[1] 0.473
> mean(y)
[1] 0.352
```

   La régression linéaire sur le graphe de probabilités de la question 8 donne :

```
> summary(reg)

Residuals:
        1         2         3         4         5
 0.066891 -0.045836 -0.001531 -0.047373  0.027848

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.13854    0.04005  -3.459 0.040670 *
sort(y)[1:5] -2.46096    0.10984 -22.405 0.000195 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05656 on 3 degrees of freedom
Multiple R-Squared: 0.9941,     Adjusted R-squared: 0.9921
F-statistic:   502 on 1 and 3 DF,  p-value: 0.0001947
```

Justifier le fait que l'on admette que cet échantillon est issu de la loi $\mathcal{P}a(a, 1)$. Donner les valeurs des estimations $\hat{a}'_n$, $\tilde{a}_n$ et a_g. Laquelle doit-on retenir ?

10. Donner une fonction pivotale pour a. En déduire l'expression d'un intervalle de confiance bilatéral de seuil α pour a. Pour les données de l'exemple, donner cet intervalle au seuil 5%.

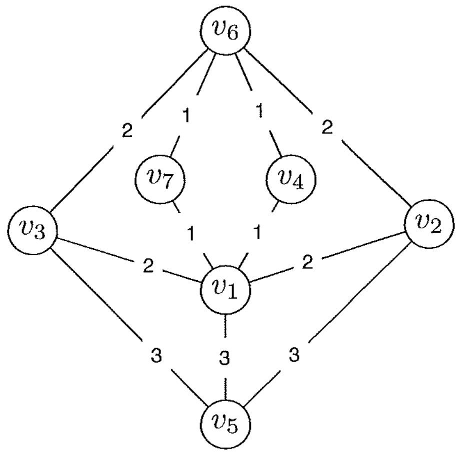
EXAMEN du 22 mai 2013. Durée: 3h. 2 pages numérotées.
Documents manuscrits ou polycopiés autorisés. Aucun livre.

Il est important de bien expliquer ce que vous faites. 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 STAUFFER ou SZIGETI

EXERCICE 1:

Déterminer un arbre de coût minimum dans le graphe ci-dessous :



NB : les données sur les arêtes représentent les coûts.

a) Appliquer soit l'algorithme de Kruskal soit l'algorithme de Prim en partant du sommet v_1. Préciser la structure courante pour chaque itération de l'algorithme.

b) Combien y-a-t'il d'arbres couvrants de coût minimum.

EXERCICE 2 :

Considérons le projet simple ci-contre:

a) Quelle est la durée minimum du projet ? Donner la date de début au plus tôt et au plus tard pour chaque tâche qui permet de terminer dans cette durée minimum. Donner la liste des tâches critiques et la marge des autres.

b) Supposons que l'on peut diminuer la durée d'une seule tâche. Quelle tâche doit-on choisir pour diminuer le plus fortement la durée minimum du projet? (Justifier la réponse.) Donner la nouvelle durée de la tâche choisie et la nouvelle durée minimum du projet.

tâche	durée	tâches précédentes
A	3	–
B	6	–
C	6	
D	4	A
E	4	A, C
F	2	B, C, D

On utilisera au choix une des méthodes de chemin critique à savoir la méthode *potentiels-tâches* (*graphe des tâches*) ou *PERT* (*graphe des événements*).

EXAMEN du 22 mai 2013. Durée: 3h. 2 pages numérotées.
Documents manuscrits ou polycopiés autorisés. Aucun livre.

EXERCICE 3:

Une entreprise disposant de 10 000 m^2 de carton en réserve fabrique et commercialise 2 types de boîtes en carton. La fabrication d'une boîte en carton de type 1 ou 2 requiert, respectivement, 1 et 2 m^2 de carton ainsi que 2 et 3 minutes de temps d'assemblage. Seules 200 heures de travail sont disponibles pendant la semaine à venir. Les boîtes sont agrafées et il faut quatre fois plus d'agrafes pour une boîte du second type que pour une du premier. Le stock d'agrafes disponible permet d'assembler au maximum 15 000 boîtes du premier type. Les boîtes sont vendues, respectivement, 3 et 5 €.

a) Formuler le problème de la recherche d'un plan de production maximisant le chiffre d'affaires de l'entreprise sous forme d'un programme linéaire canonique.

b) Déterminer un plan de production optimal à l'aide de l'algorithme du **simplexe**.

c) Ecrire le programme **dual**. Donner la solution optimale du dual. Confirmer l'optimalité des deux solutions en utilisant le théorème des écarts complémentaires.

d) Un étudiant se propose de venir assembler des boîtes pendant quelques heures (à la même cadence que les employés réguliers de l'entreprise) mais demande à être payé 60 € de l'heure. Que conseillez-vous au chef du personnel et pourquoi?

e) En téléphonant à son fournisseur, l'entreprise apprend qu'il lui est possible de se faire livrer immédiatement du carton au prix de 2 centimes le m^2. Que conseillez-vous au responsable des réapprovisionnements et pourquoi?

EXERCICE 4 :

Trouver par une méthode vue en cours, que vous expliquerez soigneusement, une base réalisable du programme linéaire suivant (on ne demande pas de trouver la solution optimale du programme linéaire) :

$$\begin{array}{ll} \textit{maximiser :} & z = -x_1 + 2x_2 + 4x_3 + x_4 + x_5 \\ \textit{sous :} & x_1 + x_3 + x_4 = 3 \\ & x_2 - x_3 + x_4 = 1 \\ & x_1 + x_2 + x_3 + x_4 + x_5 = 6 \\ & x_1,\ x_2,\ x_3,\ x_4,\ x_5 \geq 0 \end{array}$$

EXERCICE 5 :

Soit G un graphe biparti. Soit M un couplage de G tel qu'on ne peut pas ajouter d'arête de G à M pour obtenir un couplage. Montrer qu'un couplage de cardinalité maximum de G contient au plus 2|M| arêtes.

Indication : On montrera que l'ensemble des sommets de M est un ensemble transversal de G.

Examen de théorie de l'information - Ensimag - 1A

21 mai 2013

POLYCOPIE, NOTES DE COURS ET CALCULATRICES AUTORISES

Le sujet est composé de 3 problèmes indépendants. Le barème est donné à titre indicatif.

1 CODAGE SOURCE (9 points)

On considère une source simple S à 3 états notés a, b, c de probabilités $\begin{array}{rcl} p(a) & = & 1/4 \\ p(b) & = & 1/4 \\ p(c) & = & 1/2 \end{array}$.

On code d'abord cette source à l'aide du code binaire C_1 suivant : $\begin{array}{ccc} \text{a} & \rightarrow & 11 \\ \text{b} & \rightarrow & 10 \\ \text{c} & \rightarrow & 00 \end{array}$

1. Quelle est l'entropie de la source S ?
2. Quelle est la redondance de cette source ?
3. Pourquoi est-il utile d'utiliser un codage source pour compresser S ?
4. Dire pourquoi le code C_1 est instantané ? C_1 est-il déchiffrable ?
5. Quelle est la longueur moyenne (compacité) du code C_1 ?
6. Démontrer que C_1 n'est pas 100 % efficace.
7. Existe-t-il un autre jeu de probabilités (pour les états a,b,c) pour lequel C_1 serait 100% efficace ? Pourquoi ?
8. Construire C_2 un code de Fano-Shannon pour cette source.
9. Quelle est l'efficacité de C_2 ?
10. Justifier pourquoi le jeu de longueurs du code C_2 permet de coder la souce S avec une telle efficacité ?
11. Quel peut -être l'intérêt de coder les extensions d'ordre 2 de S ?
12. On s'intéresse désormais à un codage de Huffman ternaire (alphabet de codage avec 3 caractères $\{0, 1, 2\}$) de la source S ou de son extension d'ordre k.
    - (a) Donner un encadrement *a priori* de la longueur moyenne ν du code ternaire (en nombre de caractères ternaires par état de la source) en fonction de k.
    - (b) Donner un code de Huffman et sa longueur moyenne ν pour le cas $k = 1$. Commenter.
    - (c) Même question pour $k = 2$. Commenter.

2 TRANSFORMATIONS DE L'ENTROPIE (7 points)

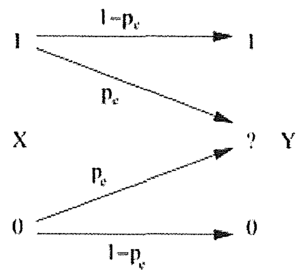
Soient une constante $\theta \in]-1, +1[$ et une variable aléatoire U uniforme sur l'intervalle $[-1, +1]$.

On considère la variable aléatoire $X = \text{sign}(U - \theta)$ à valeurs dans $\{-1, +1\}$ ($\text{sign}(x)$ vaut -1 pour $x < 0$ et $+1$ sinon.).

1. Quelle est l'entropie de X ?
2. Soit une constante $\beta \in [1, +\infty[$ et $Y = X + \beta$. Quelle est l'entropie de Y ?
3. Soit $\alpha \in [1, +\infty[$ et $Z = \alpha X$. Quelle est l'entropie de Z ?
4. Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction C^1 bijective. Quelle est l'entropie de $f(X)$?
5. Trouver une fonction $g : \mathbb{R} \to \mathbb{R}$ non constante telle que l'entropie $H(g(X)) = 0$.
6. Montrer que pour toute variable aléatoire discrète V à $N < \infty$ états et toute fonction (C^1) $f : \mathbb{R} \to \mathbb{R}$ l'entropie $H(f(V))$ de $f(V)$ est telle que $H(f(V)) \leq H(V)$.

3 CANAL À EFFACEMENT (4 points)

Le canal à effacement est à entrée X binaire $\{0; 1\}$ et sa sortie Y peut prendre 3 états $\{0; 1; ?\}$. Lorsque '0' est observé en sortie, il est certain que l'entrée est '0'. Lorsque '1' est observé en sortie, il est certain que l'entrée est '1'. Lorsque '?' est observé en sortie, le symbole d'entrée est effacé et l'on ne sait rien de l'entrée. La probabilité d'effacement est notée p_e.



1. Quelle est la matrice de transition de ce canal ?
2. Est-il uniforme en entrée ? Est-il uniforme en sortie ?
3. Calculer la capacité du canal à effacement en fonction de p_e en précisant la loi d'entrée qui maximise l'information mutuelle $I(X; Y)$.
4. On considère un code de parité qui, à partir de k valeurs binaires, construit un mot de longueur $n = k + 1$ par ajout d'une $(k + 1)^{\text{ème}}$ valeur binaire égale à la somme modulo 2 des k valeurs initiales.
   (a) Dans cette question, on suppose qu'au plus un effacement affecte un mot ; donner une méthode pratique de correction de cet effacement (c'est-à-dire donner le décodeur).
   (b) Quel que soit k, le code corrige un effacement mais plus k est grand, meilleur est le rendement. Quelle valeur maximale de k impose le second théorème de Shannon ?

Grenoble INP - Ensimag 1$^{\text{ière}}$ année mai 2013

Théorie des Langages 2

Durée : 3 heures.
Documents : notes de cours et poly autorisés, livres interdits.

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

Exercice (6 points)

Soit $<\cdot,\cdot>$ une fonction totale calculable bijective de $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$

Soit U une variante de machine de Turing universelle telle que $\mathsf{U}(<m,n>)$ calcule le résultat éventuel (dans $\mathbb{N}$) de la machine de Turing de code m sur l'entrée de code n

▷ Question 1 (3 points)

Chacune des 2 fonctions totales suivantes est-elle calculable ? Justifier la réponse.

Pour m, n fixés : $f_{<m,n>}(i) = 0$ si $\mathsf{U}(<m,n>)$ termine en au plus i pas d'exécution
$= 1$ sinon

$h(<m,n>) = 0$ si $\mathsf{U}(<m,n>)$ termine
$= 1$ sinon

▷ Question 2 (3 points)

Soit g la fonction partielle de $\mathbb{N} \to \mathbb{N}$ telle que

$\mathrm{Dom}(g) = \{m \mid \exists n : \mathsf{U}(<m,n>) \text{ termine}\}$

$\forall m \in \mathrm{Dom}(g) : g(m) = \mathrm{Min}\{\mathsf{U}(<m,n>) \mid \mathsf{U}(<m,n>)\text{termine}\}$

– Justifier que cette définition a un sens (i.e. que g est bien une fonction...)
– Soit le polynôme $P(X) = X^2 - 6X + 16$. Montrer que $\forall n \in \mathbb{N}, P(n) \in \mathbb{N}$
– Soit m le code d'une MT calculant P ; que vaut $g(m)$?
– En utilisant les réponses de la question 1, montrer que g n'est pas calculable.

Problème (14 points)

On s'intéresse à un langage d'instructions (non-terminal *inst*) constituées d'une suite d'affectations (non-terminaux *seq* et *aff*), défini par la grammaire G_1 suivante (l'axiome est le non-terminal *prog*) :

prog	$\to$	**begin** *inst* **end**	
inst	$\to$	*seq*	| *aff*
seq	$\to$	*inst* ; *aff*	
aff	$\to$	**idf** := *exp*	
exp	$\to$	(*exp* + *exp*)	| **idf** | **num**

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

▷ Question 3 (1 point)

Justifier en quoi cette grammaire n'est pas LL(1).

▷ **Question 4 (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 directeurs).

▷ **Question 5 (2 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 *aff*, 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 6 (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.** Par exemple on voudra refuser l'affectation multiple x, y, x := 1, 2, 3.

Ajouter à la grammaire précédente un calcul d'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 7 (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** $= V_T \cup \{\$\}$, 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 8 (3 points)**

On s'intéresse ici à «concaténer» des grammaires LL(1) définies sur le même vocabulaire terminal V_T. Soit $G_1 = (V_T, V_{N_1}, S_1, R_1)$ et $G_2 = (V_T, V_{N_2}, S_2, R_2)$ deux grammaires LL(1) avec $V_{N_1} \cap V_{N_2} = \emptyset$. On construit la grammaire G suivante : $(V_T, \{S\} \cup V_{N_1} \cup V_{N_2}, S, \{S \to S_1S_2\} \cup R_1 \cup R_2)$, avec S un nouveau symbole ($S \notin (V_{N_1} \cup V_{N_2})$). On rappelle qu'on définit l'ensemble des directeurs d'une règle par (S étant l'axiome) :

$$Dir(A \to w) = \{\mathbf{x} \in (V_T \cup \{\$\}) \mid \exists w_1, w_2, w_3 \ : \ S\$ \Rightarrow^* w_1Aw_2 \Rightarrow w_1ww_2 \Rightarrow^* w_1\mathbf{x}w_3\}$$

- Avec $R_1 = \{S_1 \to aS_1b \mid \varepsilon\}$ et $R_2 = \{S_2 \to cS_2 \mid b\}$ montrer que la «concaténation» de G_1 et G_2 est LL(1).
- Donner un contre-exemple montrant que G n'est pas toujours LL(1).
- Donner une condition suffisante, la moins restrictive possible, sur les ensembles *Dir*, *Prem* et *Suiv* déjà calculés pour G_1 et G_2 permettant de garantir que G est LL(1).
- (*Bonus*) vos conditions sont-elles nécessaires ?

SESSION de RATTRAPAGE JUIN 2013

Algorithmique 1

Ensimag - 1A

Juillet 2013

Durée : 2h

Machines électroniques interdites **Tous documents papier autorisés ; hors livres**

Les deux 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. Le barème est donné à titre indicatif.
Attention, votre code doit fonctionner pour toutes les entrées possibles respectant les préconditions et **obligatoirement** lever une exception en cas de problème.

---

1 Cercles circassiens (14 points)

Un programmeur cherche à réaliser une simulation d'un groupe de danseurs folkloriques. Pour ce faire, il commence par choisir une structure de données lui permettant de stocker les danseurs. Les danseurs dansent toujours en cercle, avec une alternance d'hommes et de femmes et la structure la plus raisonnable semble être une liste circulaire doublement chainée. Il y a autant d'hommes que de femmes et le nombre de danseurs ne peut être nul. La figure 1 illustre un exemple d'une telle structure. Nous supposerons également qu'aucun cercle ne peut contenir plus de 1000 participants.

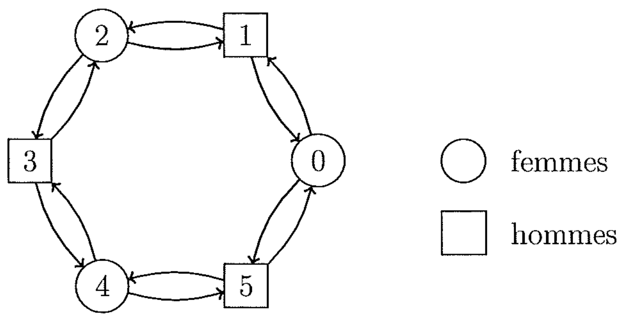


Figure 1: Cercle à 6 danseurs

On définit un danseur par la structure suivante :

```
type Genre is (homme, femme);
type Danseur;
type Pointeur is Access Danseur;
type Danseur is record
        Id : Natural; --Identifiant du danseur;
        G : Genre; --Genre du danseur;
        Suivant : Pointeur; --Pointeur vers le danseur suivant
        Precedent : Pointeur; --Pointeur vers le danseur precedent
end record;
```

Tout danseur peut avoir un identifiant quelconque qui sera toujours supposé unique. **Attention**, dans les exemples les identifiants sont souvent consécutifs mais ce n'est pas le cas dans le cas général.

Enfin, pour que ce que l'on attend de votre part soit très clair, nous soulignons encore une fois que votre code doit fonctionner pour toutes les entrées possibles respectant les préconditions et **obligatoirement** lever une exception en cas de problème.

1. (1 point) Écrire une fonction **function** `Compte(C:Pointeur)` **return** `Positive` qui retourne le nombre de danseurs dans un cercle.
   Précondition : `C` est un cercle valide.

2. (3 points) Lors d'une danse, il est possible que le cercle se scinde en deux cercles plus petits comme illustré figure 2 (seules deux poignées de main se défont). Écrire une procédure
   procedure `Decoupe(C1 :` **in out** `Pointeur ; C2 :` **out** `Pointeur ; Taille : Positive)` qui découpe un cercle initialement donné par `C1` en deux cercles `C1` et `C2`. `Taille` désigne ici la taille que doit avoir `C1` après la découpe. Votre algorithme doit fonctionner sans allouer de nouvelles structures de danseurs.
   Précondition : `C1` est un cercle valide.

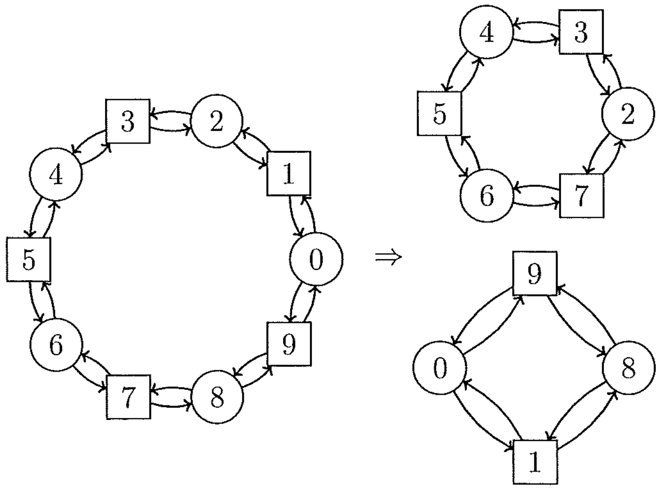


Figure 2: Découpe: `C1` pointe avant l'appel sur le danseur 2 et `Taille` vaut 6

3. (2 points) Un autre mouvement de danse consiste à séparer un cercle en deux cercles : un contenant toutes les filles et un autre tous les garçons (on perd alors la

contrainte de mixité). Écrire une procédure :
procedure `Separation(C:` **in out** `Pointeur ; Filles :` **out** `Pointeur)` qui prend en entrée en cercle `C` et met en sortie un cercle contenant tous les garçons dans `C` et toutes les filles dans `Filles`. Votre algorithme doit fonctionner sans allouer de nouvelles structures de danseurs.
Préconditions : `C` est un cercle valide.

4. (2 points) Enfin, il est possible de décaler l'ensemble des filles d'une place. Proposez une procédure **procedure** `Decalage(C:`**in out** `Pointeur)` réalisant cette opération comme illustré figure 3 (toujours sans allocations de danseurs). Préconditions : `C` est un cercle valide.

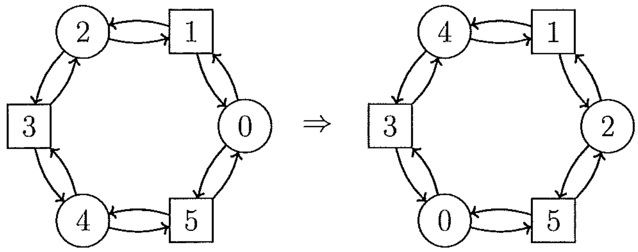


Figure 3: Décalage

5. (4 points) Écrire une fonction **function** `Verification(C:Pointeur)` **return** `Boolean` qui effectue *le plus possible de vérifications* sur un cercle de danseurs mixte et retourne vrai s'il est correct et faux sinon.
Précondition : Tous les identifiants des danseurs sont uniques.

6. (2 points) On cherche maintenant à améliorer le code existant. Une possibilité envisagée par votre chef de projet consiste à définir une structure de cercle comme un record contenant un pointeur vers un danseur ainsi qu'un compteur entier comptant le nombre total de participants au cercle. Cette structure remplacerait alors tous les `Pointeurs` dans les prototypes des différentes procédures et fonctions. On vous demande d'évaluer si cela peut valoir le coup de réaliser une telle modification. Quels algorithmes se trouvent simplifiés par cette modification (le code serait plus clair) ? Quel est le coût en temps d'une telle modification ? Les algorithmes se trouveraient-ils ralentis, accélérés ? Est-il possible de se servir de ce compteur pour réaliser des optimisations ? Justifiez fortement vos réponses.

2 Tic Tac Toe (6 points)

On considère dans cet exercice un jeu de Tic Tac Toe. Dans ce jeu, on dispose d'un plateau de jeu de 9 cases (3 par 3). Un joueur joue les "ronds" et l'autre les "croix". Le premier joueur à aligner trois de ses symboles gagne la partie. Les alignements peuvent être en ligne, colonne, ou diagonale. Le jeu est très simple : chacun son tour pose un de ses pions dans une case vide jusqu'à victoire d'un joueur ou match nul. On ne peut pas passer son tour. Si le plateau se retrouve rempli sans que quelqu'un n'arrive à aligner trois pions, personne ne gagne.

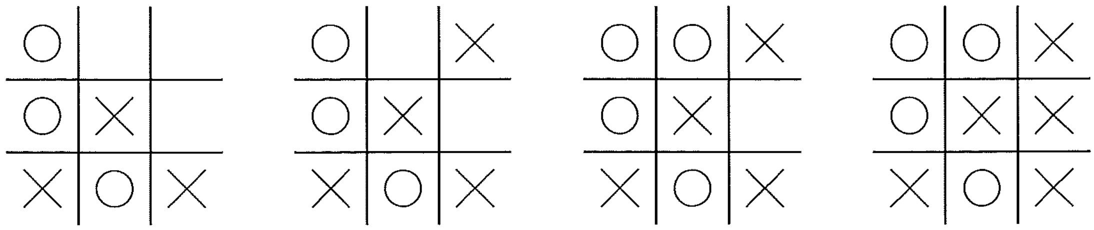


Figure 4: Tic Tac Toe : 3 derniers coups d'une partie exemple

1. (2 points) Proposez une structure de données `configuration` permettant de stocker une configuration du jeu, c'est à dire l'état courant de la partie. Vous donnerez les définitions de tous les types que vous utilisez et chercherez à avoir une structure claire et efficace.

2. (4 points) On se propose de déterminer parmi tous les coups possibles, celui qui semble le meilleur. On définit le *potentiel* $p(c)$ d'une configuration c de la manière suivante:

   - 1 si le joueur courant a gagné ;
   - -1 si le joueur courant a perdu ;
   - 0 si personne n'a gagné mais que toutes les cases sont remplies ;
   - sinon, on regarde l'ensemble $S(c)$ de toutes les configurations atteignables par le joueur courant en jouant un coup à partir de c :

   $$p(c) = \sum_{s \in S(c)} -p(s)$$

   Dans l'exemple on a ainsi un potentiel au dernier coup de 1 car le joueur courant (croix) gagne. A l'avant dernier coup, le joueur rond a le choix entre deux emplacements possibles : un menant à un potentiel du joueur croix de 0 et l'autre menant au potentiel du joueur croix calculé précedemment de 1 ce qui donne au final un potentiel pour le joueur rond de $-1 * 0 + -1 * 1 = -1$, etc...

   Écrivez le code d'une fonction récursive :
   function `Potentiel(C: Configuration)` **return** `Integer` calculant le potentiel d'une configuration. Vous ferez attention aux allocations et libération de mémoires éventuelles. Aucune contrainte n'est posée sur le temps de calcul.
   Précondition : `C` est une configuration valide.

Algorithmique 2

Ensimag - 1A

Juillet 2013

Durée : 2h
Machines électroniques interdites **Tous documents papier autorisés ; hors livres**

Les deux 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. Le barème est donné à titre indicatif.
Toutes les analyses de coût doivent être justifiées.

---

1 Médianes (7 points)

On cherche dans cet exercice à calculer la médiane d'un tableau T de n entiers. On rappelle que la médiane m est l'entier du tableau tel que $\lceil (n-1)/2 \rceil$ éléments soient inférieurs ou égaux à m et $\lfloor (n-1)/2 \rfloor$ éléments soient supérieurs ou égaux à m.

Pour simplifier nous considérons dans cet exercice que tous les éléments sont garantis uniques.

1.1 QuickSelect

On commence dans un premier temps par écrire un algorithme de sélection du $k^{\text{ème}}$ élément retournant le $k^{\text{ème}}$ plus petit élément du tableau T. Cet algorithme permet de trouver la médiane en prenant $k = n/2$. On se base ici sur un principe similaire au *quicksort* et on suppose donc disposer d'une procédure :
`segmentation(T : in out Tableau ; P : Indice_Pivot)` réorganisant les éléments tel que T contienne au sortir de la procédure tous les éléments inférieurs au pivot puis le pivot, puis tous les éléments supérieurs au pivot comme illustré figure 1. La procédure segmentation est linéaire en la taille du tableau. Dans tous l'exercice, la procédure *segmentation* est fournie et on ne vous demande donc pas de l'écrire.

Étant donné un tableau T, l'algorithme *Quickselect* fonctionne en tirant aléatoirement un pivot puis en effectuant une segmentation. Il réalise alors une recherche récursivement dans un sous-tableau du tableau segmenté.

1. (1 point) Soit f la position finale du pivot et k le numéro de l'élément recherché par l'algorithme de sélection. Dans quel sous-tableau du tableau T segmenté peut-on

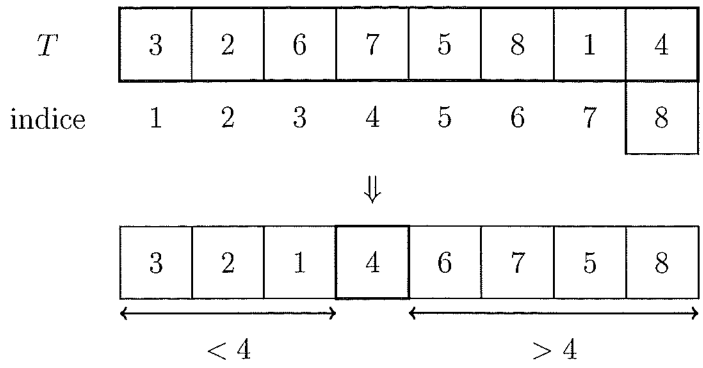


Figure 1: segmentation de T avec un pivot en place 8

restreindre la recherche du $k^{\text{ème}}$ élément ? Quel est alors le numéro *dans ce sous-tableau* de l'élément recherché ?

2. (1 point) Écrire en pseudo-code l'algorithme récursif du *Quickselect.*

3. (1 point) Quel est son coût au pire cas ? À votre avis, le coût en moyenne est-il identique ? (Justifiez vos réponses)

1.2 Algorithme de Blum, Floyd, Pratt, Rivest et Tarjan

On cherche à améliorer l'algorithme de la question précédente. Pour ce faire, nous changeons uniquement la manière de sélectionner le pivot. Dans le *Quickselect*, le pivot est tiré aléatoirement, ce qui laisse la possibilité de prendre un mauvais pivot. Nous allons remplacer ce tirage aléatoire par un algorithme réalisant ce choix de manière déterministe. On obtient donc l'algorithme *select2* suivant :

> **Entrées** : T, k
> **Sorties** : le $k^{\text{ème}}$ plus petit élément de T
> Couper le tableau T en $n/5$ tableaux $C_1, C_2, \ldots, C_{n/5}$ de taille 5;
> Calculer les $n/5$ médianes $(m_i)_{1 \leq i \leq n/5}$ de chacun des tableaux C_i;
> Créer un tableau S de taille $n/5$ et contenant les (m_i);
> $m_s = \text{select2}(S, n/10)$;
> $T = \text{segmenter}(T, m_s)$;
> Effectuer une sélection récursive sur un sous-tableau de T comme pour l'algorithme précédent;
> **retourner** *l'élément obtenu*

Algorithme 1 : Algorithme *select2*

Le pivot est donc la médiane des médianes.

1. (1 point) Prouvez que le nombre d'éléments de T inférieurs à m_s est toujours supérieur à $0,3 \times n$.

2. (1 point) Donnez une formule récursive calculant le coût au pire cas de l'algorithme *select2.*

3. (2 points) Montrez que le coût au pire cas de *select2* est linéaire.

2 Tableaux de Young (13 points)

Un tableau $m \times n$ est dit "de Young" si tous les éléments d'une même ligne sont triés de gauche à droite et tous ceux d'une même colonne de haut en bas. Les cases ne contenant pas d'éléments sont supposées contenir $+\infty$. Le tableau Y de la figure 2 illustre un tableau 4×4 contenant les éléments $\{9, 16, 3, 2, 4, 8, 5, 14, 12\}$. Précisons que les lignes sont indicées de 1 à m et les colonnes de 1 à n.

2	8	12	14
3	9	16	$+\infty$
4	$+\infty$	$+\infty$	$+\infty$
5	$+\infty$	$+\infty$	$+\infty$

Figure 2: Exemple de tableau de Young

1. (1 point) Où se trouve le plus petit élément d'un tableau non vide ?
2. (1 point) On insère le nombre 6 dans la case $(3, 3)$ du tableau de la figure 2. Le "remonter" à sa place en ne faisant que des échanges avec une case voisine. (dessiner toutes les étapes)
3. (2 points) Supposons que l'on change la valeur d'une case (i, j). Écrire une fonction récursive qui "descend" ou "remonte" l'élément modifié de (i, j) à sa nouvelle place (selon que la valeur soit augmentée ou diminuée).
4. (1 point) Donnez un algorithme simple et rapide qui supprime le plus petit élément du tableau.
5. (1 point) Quel est le coût au pire cas d'une telle suppression ?
6. (2 points) Donnez un algorithme qui insère un nouvel élément dans un tableau de Young. Quel est son coût au pire cas ?
7. (3 points) On se propose d'écrire maintenant une fonction `Conversion` qui prend en entrée un tableau carré non trié et renvoie le tableau de Young correspondant (en rajoutant toutes les contraintes de tri). On cherche à travailler sur le principe du diviser avec une découpe en sous problèmes similaire à celle de l'algorithme de multiplication de matrices de Strassen. On voudrait donc compléter l'algorithme 2 pour obtenir un algorithme de coût au pire cas de $O(n^2)$. Vous pourrez supposer disposer d'une procédure `Fusion` réalisant la fusion de 2 tableaux triés unidimensionnels en temps linéaire similaire à celle utilisée dans le tri fusion.

   Sans écrire de code, expliquez le principe de l'algorithme.
8. (1 point) Prouvez que votre algorithme de conversion a bien un coût au pire cas de $O(n^2)$.
9. (1 point) En utilisant un tableau de Young de taille $n \times n$ montrez comment on peut trier n^2 éléments en $O(n^3)$.

Entrées : T : tableau de taille $n \times n$
Sorties : Y : un tableau de Young de taille $n \times n$ contenant tous les éléments de T
Couper le tableau T en 4 parties T_1, T_2, T_3, T_4 de taille $n/2 \times n/2$;
$Y_1 =$ Conversion(T_1);
$Y_2 =$ Conversion(T_2);
$Y_3 =$ Conversion(T_3);
$Y_4 =$ Conversion(T_4);
"algorithme à terminer…";
retourner Y

Algorithme 2 : Conversion

Examen Session 2
Mardi 3 juillet 2012 - 2h

Documents manuscrits et polycopié de cours autorisés. Tout autre document et calculatrices interdits.
N.B. : *La rédaction sera prise en compte dans la notation. Toute affirmation devra être justifiée.*

Exercice 1
Soient α, $\beta \in]0,1[$. Pour $t \in]0,1[$ et $n \in \mathbb{N}$ on pose

$$u_n(t) = t^{-\alpha}(1-t)^{-\beta}(\sin\frac{1}{t})^n\,.$$

1. *Montrer que $u_n \in L^1(]0,1[)$ pout tout $n \in \mathbb{N}$.*
2. *Etudier la limite de la suite*

$$I_n = \int_0^1 u_n(t)\,dt\,.$$

Exercice 2
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 de l'équation différentielle linéaire du premier ordre :*

$$y'(x) - y(x) = -\frac{A}{\sqrt{x}}$$

   avec $A = \int_0^{+\infty} e^{-t^2}dt$.

4. *Intégrer cette équation différentielle et en déduire la valeur de $\int_0^{+\infty} e^{-t^2}dt$.*

Exercice 3

On rappelle que la transformée de Fourier de la fonction $G(x) = e^{-\pi x^2}$ *est égale à* $\hat{G}(\nu) = e^{-\pi \nu^2}$.

1. *Pour* $a > 0$, *on pose :*
$$G_a = \frac{1}{\sqrt{a}}\, e^{-\frac{x^2}{2a}}$$
Calculer la transformée de Fourier $\hat{G}_a$ *de* G_a.

2. *Soit* $a, b > 0$ *deux réels.*
   (a) *Calculer la transformée de Fourier du produit de convolution* $G_a * G_b$.
   (b) *En déduire que* $G_a * G_b = \alpha G_c$, *où* α *et* c *sont des constantes à préciser.*

Exercice 4

Soit $(X; d)$ *un espace métrique et* a *un point de* X. *Pour tout* $x, y \in X$ *on pose* $d'(x, y) = 0$ *si* $x = y$ *et* $d(x, y) = \max(d(a, x), d(a, y))$ *si* $x \neq y$.

1. *Montrer que* d' *est une distance sur* X.

2. *Soit* $(x_n)_{n\in\mathbb{N}}$ *une suite de points de* X.

   (a) *Montrer que* (x_n) *converge vers* a *relativement à* d *si et seulement si* (x_n) *converge vers* a *relativement à* d'.

   (b) *On suppose que* (x_n) *converge au sens de* d' *vers un point* $\ell \in X$ *différent de* a. *Montrer qu'on a* $x_n = \ell$ *à partir d'un certain rang.*

3. *Soit* $f : X \to \mathbb{R}$ *une application quelconque. Montrer que* f *est continue relativement à* d' *en tout point* $x \in X$ *différent de* a.

4. *On suppose* $X = \mathbb{R}$, *muni de la distance usuelle* d. *La distance* d' *est-elle équivalente à* d*?*

Examen d'Architecture
Durée : 2 heures

ENSIMAG 1A

Année scolaire 2012–2013
2ème session

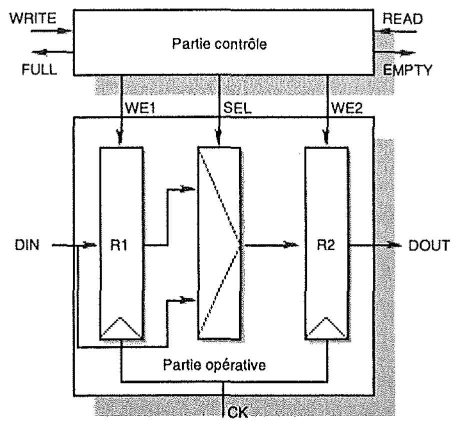
Ex. 1 : PC/PO : Implantation matérielle d'une file (9 pts)

On cherche à réaliser matériellement une file (structure Fisrt In First Out : FIFO) permettant une communication entre deux processus, selon un protocole producteur/consommateur très simple :

- le processus producteur peut écrire un mot de 32 bits dans la FIFO en présentant le mot sur l'entrée DIN et en activant simultanément le signal WRITE. Si la FIFO est pleine – FULL vaut 1 – alors l'écriture n'est pas prise en compte, sinon la donnée est enregistrée dans la FIFO sur le front montant de l'horloge ;

- le processus consommateur peut lire un mot de 32 bits en activant le signal READ, et en échantillonnant la valeur présente sur DOUT. Si la FIFO est vide – EMPTY vaut 1 – alors la donnée lue n'est pas valide, sinon elle est retirée de la FIFO sur le front montant de l'horloge.

Nous nous intéressons à une FIFO de profondeur 2. La FIFO est constituée d'une partie opérative et d'une partie contrôle. La partie opérative est constituée de 2 registres et d'un multiplexeur, comme illustré ci-dessous. Elle est simple, mais il faut en comprendre le fonctionnement.

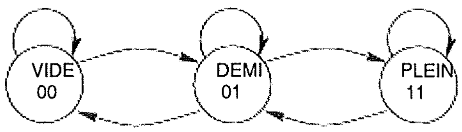


La partie contrôle est un automate de Mealy qui reçoit en entrée les signaux READ et WRITE. Elle fournit FULL et EMPTY, ainsi que SEL, WE1 et WE2.
Si SEL='1', alors DIN apparaît en sortie du multiplexeur. Sinon, c'est la sortie de R1 – notée Q1 – qui y apparaît.
WE1 et WE2 sont les signaux d'autorisation d'écriture dans les registres R1 et R2.

La FIFO peut être dans les trois états suivants :
- vide : R1 et R2 ne contiennent pas de données significatives ;
- demi : R1 est non significatif, R2 est valide ;
- plein : R1 et R2 contiennent des données valides.

Dans chacun de ces trois états, la FIFO peut recevoir :
- aucune requête : READ = '0' et WRITE = '0' ;
- une requête de lecture : READ = '1' et WRITE = '0' ;
- une requête d'écriture : READ = '0' et WRITE = '1' ;
- les deux requêtes simultanément : READ = '1' et WRITE = '1'.

Soit l'automate de contrôle suivant :



Question 1 Exprimez la fonction de transition, dépendante des entrées READ et WRITE, de chaque arc de l'automate. On prendra soin d'étudier toutes les valeurs possibles des entrées pour chaque état.

Le code de chacun des états est donné sous son nom. Les entrées du registre d'état sont notés D1 D0, et ses sorties sont notés Q1 Q0. Les sorties FULL et EMPTY sont des signaux de Moore, alors que les sorties SEL, WE1 et WE2 sont des signaux de Mealy.

Question 2 Écrivez la table de transitions de l'automate, qui donne la valeur de l'état futur en fonction de l'état courant et des entrées.

Question 3 Donnez les expressions booléennes des signaux D1 et D0.

Question 4 Donnez les expressions booléennes des signaux FULL et EMPTY.

Question 5 Écrivez la table de vérité des signaux SEL, WE1 et WE2.

Question 6 Donnez les expressions simplifiées de ces signaux.

Ex. 2 : Conception Partie-Contrôle/Partie Opérative (11 pts)

Tracer des lignes et des cercles sur un écran (qui est une matrice discrète) ne 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 tracé de lignes et de cercles. Nous nous intéresserons ici à la méthode d'Andres dont l'algorithme pour tracer le 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 opérations à réaliser ; déduisez-en le type des unités fonctionnelles (registres ou opérateurs) et le nombre minimal de chacune d'elles, 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 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). Vous pourrez supposer l'existence de sorties z (valant 1 quand le résultat est nul) et sign (valant 1 quand le résultat est négatif) pour les opérateurs utilisés.

Question 2 Proposez une partie opérative interconnectant ces unités fonctionnelles et pensez à faire ressortir les signaux 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 le résultat du programme.

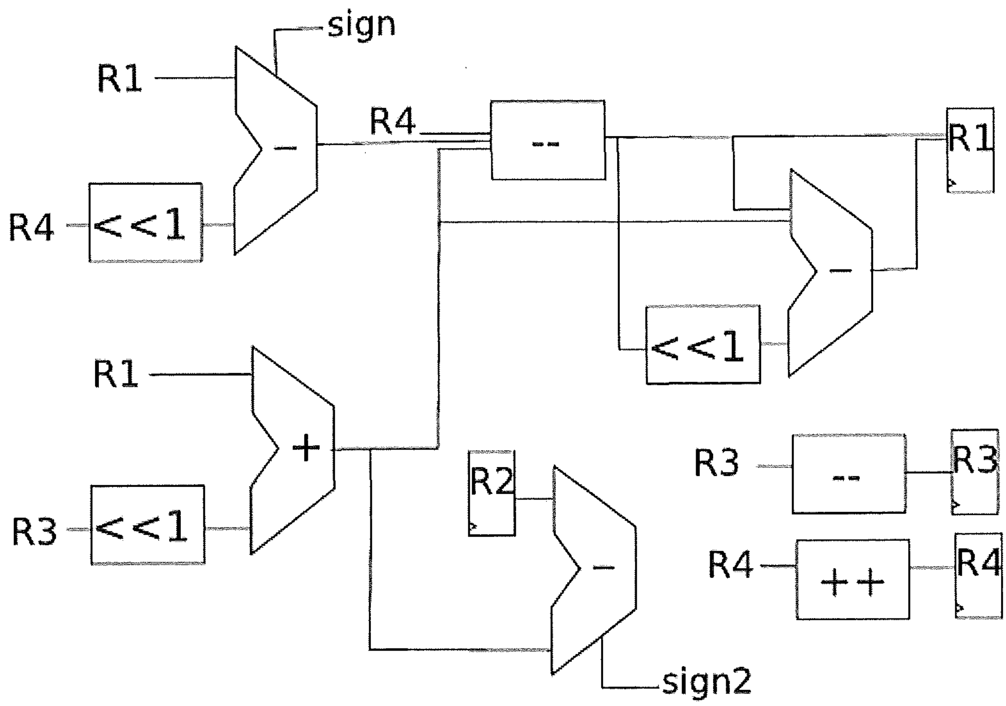
Question 4 Afin de mettre en évidence les opérations que l'on désire faire en parallèle, réécrivez le programme en utilisant la notion d'affectation concurrente présentée en cours, par exemple (a, b) ← (3, 1) affecte 3 dans a et 1 dans b.

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 5 Reformulez 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.


sign
R1
-
R4
<<1
R4
--
R1
-
<<1
R1
+
R3
<<1
R2
-
R3
--
R3
R4
++
R4
sign2

ENSIMAG 1ère année 2012 /2013

ECONOMIE

Examen session II

Durée : 2 heures
Documents autorisés

Travail demandé : Commentaire libre du texte ci-joint.

La flexi-sécurité à la française reste à inventer

Jean-Marc Le Gall, Professeur associé au CELSA (Centre d'Etudes Littéraires et scientifiques Appliquées)
Les Echos, 29 mai 2013

Nombreux sont les économistes pour qui la forte protection de l'emploi en France entraîne des effets secondaires négatifs, pour l'économie, mais aussi, paradoxalement, pour nombre de salariés. Une telle protection, soulignent-ils, ralentit les adaptations nécessaires à la compétitivité et à la croissance, en particulier quand elle se combine à une faible mobilité des salariés. En réaction, les entreprises recourent massivement à des formes de travail temporaire, aux dépens des jeunes, des femmes et des travailleurs peu qualifiés. En conséquence, si la probabilité de perdre son emploi en CDI est objectivement faible, celle d'en trouver un (pour les jeunes), ou d'en retrouver un (pour les chômeurs) l'est tout autant. Diagnostic confirmé par toutes les études, qui soulignent que les embauches et les pertes d'emploi concernent essentiellement des travailleurs au statut précaire.

Le caractère inégalitaire de cette flexibilité a conduit de nombreux acteurs à s'intéresser au supposé « modèle » scandinave. Alternative a priori idéale, il est présenté depuis les années 1990 comme associant une protection de l'emploi faible à une protection sociale développée. Cette dernière, faisant du retour à l'emploi sa priorité, met l'accompagnement des chômeurs au premier plan : considéré plus efficace économiquement, ce modèle est aussi crédité de plus de justice. La flexi-sécurité est ainsi devenue la référence obligée de tous les discours réformistes. Telle est en particulier la logique de l'accord sur la sécurisation de l'emploi, confirmé par la loi, le 14 mai dernier. Diverses mesures accroissent en effet la flexibilité, tant interne (modulation des rémunérations ou du temps de travail « en cas de graves difficultés conjoncturelles », etc.) qu'externe (assouplissement des procédures de licenciement économique...).

Des réserves se font toutefois entendre, qui alertent sur le risque de voir dans la conjoncture actuelle s'accroître fortement la flexibilité (du fait de la multiplication des licenciements), sans que la sécurité progresse significativement, en particulier pour les personnes précaires ou peu qualifiées. L'exemple des 782 salariés du pôle frais du groupe Doux licenciés en septembre 2012 confirme ces craintes : seulement 116 ont retrouvé un emploi dont 18 un CDI (« Les Echos » du 3 mai). Cet échec s'explique largement par l'absence de formation, initiale et continue, de ces salariés. Ce constat accablant rappelle que la vraie sécurité pour les salariés réside dans leur capacité à retrouver si nécessaire un emploi. Or cette employabilité engage la responsabilité conjointe du système public de formation et des entreprises, au moins autant qu'elle concerne le fonctionnement du marché du travail. Quelle peut être la portée de cette flexi-sécurité pour les plus de 150.000 jeunes qui quittent chaque année le système éducatif sans diplôme ni compétences professionnelles ? Ou pour les salariés ne bénéficiant ni de formation ni de parcours qualifiants ?

Une efficace « flexibilité à la française » exige donc, outre un accompagnement très renforcé des chômeurs, une évolution radicale et conjointe du système éducatif et des politiques d'entreprise. S'agissant du premier, il s'agit de réduire l'échec scolaire, d'une part, en expérimentant des modalités innovantes de discrimination positive, comme l'étaient les internats d'excellence, et, d'autre part, en valorisant l'apprentissage en entreprise. Quant à la gestion des ressources humaines, elle doit concerner tous les salariés de l'entreprise et non ses seuls « talents ». Les nombreux « oubliés » de la formation et de la carrière doivent bénéficier de bilans de compétences systématiques, d'une mobilité professionnelle diversifiée et de formations régulières, dispositifs jusqu'alors largement réservés aux plus qualifiés et diplômés. Réforme de la formation, généralisation de la GRH, personnalisation de l'appui aux chômeurs, il y a encore loin de l'accord négocié à l'objectif de sécurité proclamé.

Grenoble INP – ENSIMAG
Année 2011-2012
N° d'inscription (carte étudiant) :
NOM :
Prénom :
Né(e) le : **à (ville + dépt ou pays) :**
N° de la place :

Introduction aux réseaux

Examen du 26/06/2013. Durée 1h00
Livres interdits et calculettes autorisées

Exercice 1 : Communications numériques

On désire transmettre des données à un débit de 100 Mbit/s entre deux bâtiments d'une entreprise, espacés de 500 m.

Une première solution consiste à utiliser une liaison radio autour d'une fréquence porteuse de 7 GHz. L'atténuation est alors de 103 dB sur 500 m. La documentation technique du système radio précise les points suivants :

- La modulation peut être soit une modulation QPSK à 4 états, soit une modulation QAM16 à 16 états.
- La largeur spectrale du canal disponible est de 28 MHz.
- La puissance émise est de 29 dBm pour la modulation QPSK, de 23 dBm pour la modulation QAM16.
- Dans le cas d'une transmission à 100 Mbit/s, le seuil minimal de puissance reçue pour obtenir un taux d'erreurs binaires de 10^{-3} est de -77 dBm pour la modulation QAM16.
- Dans le cas d'une transmission à 50 Mbit/s, le seuil minimal de puissance reçue pour obtenir un taux d'erreurs binaires de 10^{-3} est de -84 dBm pour la modulation QPSK, de -80 dBm pour la modulation QAM16.

1) Calculez le débit symboles pour réaliser la transmission numérique à 100 Mbit/s, avec les deux modulations QPSK et QAM16.
2) Pour un débit de 100 Mbit/s, les spécifications techniques donnent la valeur du seuil minimal de puissance reçue seulement pour la modulation QAM16. Cela sous-entend-il qu'on ne peut pas utiliser la modulation QPSK à ce débit ? Justifiez la réponse.
3) Dans le cas présent d'une transmission sur 500 m à 100 Mbit/s, peut-on avoir une idée du taux d'erreurs binaires obtenu en réception ? Si oui, donnez sa valeur.
4) Pour une transmission à 50 Mbit/s, les spécifications techniques donnent des valeurs de seuils minimaux de puissance reçue pour les deux types de modulation. De plus, ces seuils sont tous les deux plus faibles que celui pour une transmission à 100 Mbit/s. Est-ce normal ? Leurs valeurs précises sont-elles conformes à la théorie ?

Une deuxième solution consiste à utiliser une liaison par fibre optique basée sur le protocole Fast-Ethernet utilisant un code de type NRZ.

5) Que signifie le sigle NRZ ? Expliquez le principe de ce codage et donnez la forme de la densité spectrale de puissance du signal émis avec un code NRZ.

<u>Exercice 2</u> : Adressage

Un utilisateur sur telesun.imag.fr interroge le service DNS à l'aide de nslookup. On suppose qu'au départ tous les caches des serveurs DNS concernés sont vides. Voici les observations de sa session.

```
telesun ~: nslookup
Default Server:  dns.ensimag.fr
Address:  195.221.228.2

> dns.imag.fr
Server:  dns.ensimag.fr
Address:  195.221.228.2

Name:    dns.imag.fr
Address:  129.88.38.2
> exit
```

1) Expliquez les commandes utilisées et commentez la réponse obtenue.

2) Décrivez la suite d'opérations que le système DNS effectue pour donner cette réponse.

<u>Exercice 3</u> : Trafic engendré par des applications

Un utilisateur travaille à l'IPB (Institut Polytechnique de Bordeaux) sur un terminal : tx01.ipb.fr, 128.178.156.3. Il a une session de travail sur la machine in1sun5.ipb.fr, 128.178.164.5 qui utilise le serveur DNS stisun1.ipb.fr, 128.178.15.8.

1) Quel est le résultat de l'exécution de la commande, vu de l'utilisateur: `[in1sun5 1] ping -s tx01` ?

2) Quel trafic au niveau IP peut-on observer ? Indiquez les types de messages qui circulent ainsi que les adresses IP source et destination de ce trafic.

<u>Exercice 4</u> : Déductions sur un trafic observé

La séquence suivante a ´été captée :

```
fidji -> telesun TCP D=22 S=1021 Ack=1187 Seq=3760 Len=0 Win=8760
telesun -> fidji ICMP Echo request
fidji -> telesun ICMP Echo reply
telesun -> fidji TCP D=1021 S=22 Ack=3760 Seq=1187 Len=116 Win=24820
fidji -> telesun TCP D=22 S=1021 Ack=1303 Seq=3760 Len=0 Win=8760
telesun -> fidji ICMP Echo request
fidji -> telesun ICMP Echo reply
telesun -> fidji TCP D=1021 S=22 Ack=3760 Seq=1303 Len=84 Win=24820
fidji -> telesun TCP D=22 S=1021 Ack=1387 Seq=3760 Len=0 Win=8760
telesun -> fidji ICMP Echo request
fidji -> telesun ICMP Echo reply
telesun -> fidji TCP D=1021 S=22 Ack=3760 Seq=1387 Len=84 Win=24820
fidji -> telesun TCP D=22 S=1021 Ack=1471 Seq=3760 Len=0 Win=8760
```

1) Quelles applications sur fidji et telesun engendrent ce trafic ? Expliquez ce qui se passe, les commandes effectuées qui ont créé ce trafic.

ENSIMAG 1ère année (2012-13) Méthodes Numériques

Examen du 2 Juillet 2013 — Session 2

Durée : 2h.

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

On considère une matrice réelle $A = (a_{ij})_{1 \le i,j \le n}$ telle que pour tout $j = 1, \dots, n$

$$|a_{jj}| > \sum_{\substack{i=1 \\ i \neq j}}^{n} a_{ij}. \qquad \text{Attention aux indices !!}$$

1. Montrer que A est inversible.

2. Montrer qu'il existe $L \in M_n(\mathbb{R})$ triangulaire inférieure de diagonale unité et $U \in M_n(\mathbb{R})$ triangulaire supérieure telles que $A = LU$.

3. La matrice A suivante admet-elle une factorisation $A = LU$? Calculer cette factorisation s'il y a lieu.

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 0 & -3 & -1 \\ -1 & 1 & -2 \end{pmatrix}.$$

Exercice 2

Soit $A \in M_n(\mathbb{R})$ une matrice inversible. On considère la suite de matrices $(B_k)_{k \ge 0}$ définie par

$$B_{k+1} = B_k(2I - AB_k), \quad k \ge 0,$$

la matrice $B_0 \in M_n(\mathbb{R})$ étant fixée.

1. Montrer que si la suite (B_k) converge et que sa limite est une matrice inversible, alors cette limite vaut A^{-1}.

2. Vérifier que $I - AB_{k+1} = (I - AB_k)^2$.

3. Montrer que si $\rho(I - AB_0) < 1$ alors $\lim_{k \to +\infty} B_k = A^{-1}$. A quelle vitesse s'effectue cette convergence?

ENSIMAG 1$^{\text{ère}}$ année juin 2013

Principes et Méthodes Statistiques

Durée : 2 heures. Tous documents autorisés.

Les deux parties sont indépendantes.

Les résultats vus en cours ou en TD peuvent être utilisés sans être redémontrés.

Il sera grandement tenu compte de la qualité de la rédaction (présentation et justification des réponses) dans la notation.

Barème indicatif - Partie 1 : 9 pts, Partie 2 : 11 pts.

Première partie

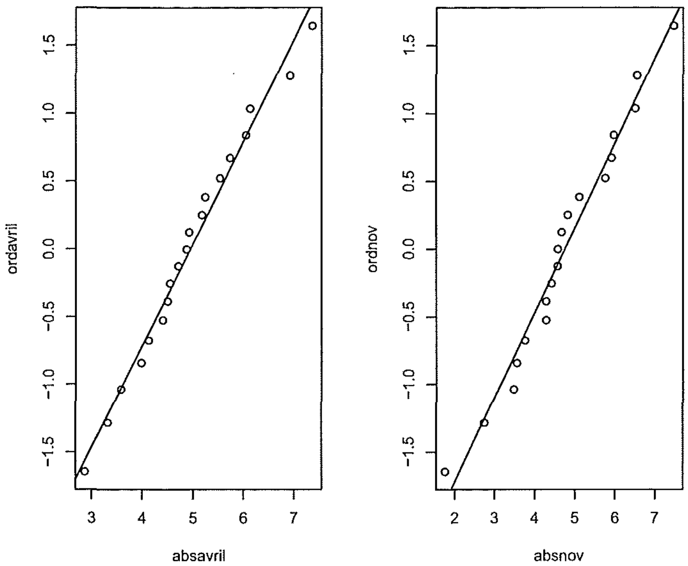
Après des années de forte hausse, les prix de l'immobilier se sont stabilisés. Certains spécialistes craignent même que les prix baissent. Afin de le vérifier, une enquête a été effectuée auprès d'une agence immobilière parisienne. Les prix au mètre carré (en milliers d'euros) ont été relevés sur des biens comparables pour des ventes effectuées en avril 2012 et novembre 2012. Les résultats sont les suivants.

```
> avril
 [1] 6.056 5.531 3.983 4.402 4.926 7.364 5.735 4.712 4.876 2.867
[11] 5.173 4.122 6.928 5.239 3.579 3.320 6.135 4.549 4.495 5.374
> novembre
 [1] 4.825 5.759 1.756 6.513 5.916 5.975 4.675 5.109 3.754 3.547
[11] 6.566 7.457 2.748 3.469 4.415 4.578 4.567 4.284 4.282 3.057
```

Une analyse statistique en R de ces données est présentée ci-dessous.

```
> summary(avril)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.867   4.332   4.901   4.968   5.582   7.364
> var(avril)
[1] 1.308125
> summary(novembre)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.756   3.702   4.573   4.663   5.798   7.457
> var(novembre)
[1] 2.006813

> absavril<-sort(avril[1:19])
> ordavril<-qnorm(seq(1:19)/20)
> regavril<-lm(ordavril~absavril)
```

```
> summary(regavril)

Call:
lm(formula = ordavril ~ absavril)

Residuals:
      Min        1Q    Median        3Q       Max
-0.208682 -0.068960  0.007344  0.082492  0.165626

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.72131    0.11393  -32.66   <2e-16 ***
absavril     0.75224    0.02244   33.52   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1115 on 17 degrees of freedom
Multiple R-squared: 0.9851,     Adjusted R-squared: 0.9842
F-statistic:  1123 on 1 and 17 DF,  p-value: < 2.2e-16

> absnov<-sort(novembre[1:19])
> ordnov<-qnorm(seq(1:19)/20)
> regnov<-lm(ordnov~absnov)
> summary(regnov)

Call:
lm(formula = ordnov ~ absnov)

Residuals:
     Min       1Q   Median       3Q      Max
-0.23755 -0.07942 -0.04576  0.12517  0.22475

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.9672     0.1176  -25.24 6.47e-15 ***
absnov        0.6250     0.0238   26.26 3.35e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1416 on 17 degrees of freedom
Multiple R-squared: 0.9759,     Adjusted R-squared: 0.9745
F-statistic: 689.8 on 1 and 17 DF,  p-value: 3.348e-15

> par(mfcol=c(1,2))
> plot(absavril, ordavril)
> abline(regavril)
> plot(absnov, ordnov)
> abline(regnov)
```




```
> var.test(avril, novembre, alternative="two.sided")

        F test to compare two variances

data:  avril and novembre
F = 0.6518, num df = 19, denom df = 19, p-value = 0.3591
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.2580071 1.6468475
sample estimates:
ratio of variances
         0.6518423

> t.test(avril, novembre, alternative="greater")

        Welch Two Sample t-test

data:  avril and novembre
t = 0.7509, df = 36.384, p-value = 0.2288
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -0.3814489        Inf
sample estimates:
mean of x mean of y
   4.9683    4.6626
```

En vous appuyant sur les résultats de cette analyse, répondez aux questions suivantes. Dans chaque cas, expliquer quelle est la méthode utilisée et comment trouver les valeurs numériques appropriées dans les résultats fournis par R.

1. Peut-on admettre que ces deux échantillons sont de loi normale ?
2. Donner les estimations sans biais et de variance minimale, et les estimations graphiques, des moyennes et variances des deux échantillons.
3. Donner des intervalles de confiance de seuil 5% pour les prix moyens au mètre carré en avril et en novembre. Donner les commandes de R permettant d'obtenir ces résultats.
4. Peut-on conclure qu'il y a eu une baisse significative du prix moyen au mètre carré entre avril et novembre ?

Deuxième partie

Des faussaires fabriquent des contrefaçons de produits d'une grande marque de luxe. La police a trouvé un moyen de distinguer les produits contrefaits des produits originaux en mesurant une certaine caractéristique. En effet, cette caractéristique est de loi normale de moyenne 50 et de variance 16 pour les produits originaux, tandis qu'elle est de loi normale de moyenne 44 et de variance 16 pour les produits contrefaits.

La douane a saisi un lot de 4 produits. On admettra que la loi de la caractéristique mesurée est normale et de variance 16. La moyenne empirique sur les 4 produits saisis est de 46. La douane pense que les produits sont contrefaits et décide de les détruire. On souhaite construire un test d'hypothèses permettant de se prononcer sur l'avis de la douane et de limiter le risque de détruire des produits originaux.

1. Expliquer quelles doivent être les hypothèses H_0 et H_1 de ce test.
2. Montrer que, pour un seuil α, la région critique de ce test s'écrit :
$$W = \{\bar{x}_4 < 50 - 2\, u_{2\alpha}\}.$$
3. Donner l'expression de la puissance du test β en fonction de α. Expliquer pourquoi diminuer α conduit à diminuer β.
4. Appliquer le test au seuil 5%, calculer β et conclure.
5. Calculer la p-valeur du test et interpréter le résultat.
6. En admettant que la moyenne empirique reste à 46, combien faudrait-il saisir de produits pour pouvoir conclure que les produits sont contrefaits avec une confiance supérieure à 99% ?

2012-2013

Ensimag 1A

Examen de Probabilités appliquées - 2ème session
Lundi 1er juillet 2013

Durée : 1h30.
Document autorisé : 1 feuille A4 manuscrite recto-verso.
Calculatrices et téléphones portables interdits.
Merci d'inscrire votre numéro de groupe sur la copie.
Le sujet comporte 2 pages. Le barème est indicatif. Les 3 exercices sont indépendants et peuvent être traités par le candidat dans l'ordre qu'il préfère. Tout élément de vérification des résultats sera très apprécié.

Exercice 1. [4pts]
Le petit chaperon rouge s'apprête à rendre visite à sa mère grand. Trois chemins mènent jusqu'à la maison de sa mère-grand : le chemin de la forêt (1), le chemin de la plage (2) et le chemin des collines (3). Il choisit un chemin au hasard de telle façon qu'il passe par la forêt avec probabilité $1/5$, par la plage avec probabilité $2/5$ et et par les collines avec la probabilité $2/5$. Il se fait dévorer par le loup avec une probabilité de $1/4$ sur le chemin de la forêt, $1/3$ sur le chemin de la plage et $1/6$ sur le chemin des collines.

1. Calculer la probabilité de l'événement *Le petit chaperon rouge se fait dévorer par le loup.*
2. La mère-grand, qui attend le petit chaperon rouge chez elle, ne le voit pas arriver : il s'est donc fait dévorer par le loup. Sachant cela, quelle est la probabilité que le petit chaperon rouge ait emprunté le chemin de la forêt ? Le chemin de la plage ? Le chemin de la colline ? Sur quel chemin la mère-grand a-t-elle le plus de chances de trouver le panier abandonné par le petit chaperon rouge lorsqu'il s'est fait dévorer par le loup ?

Exercice 2. [7pts]
Soient $(X_i)_{1\leq i\leq n}$ des variables aléatoires de Bernoulli indépendantes et de même loi :

$$P(X_i = 0) = 1 - p, \text{ et } P(X_i = 1) = p.$$

1. Soit $S_n = \sum_{i=1}^n X_i$. Quelle est la loi de S_n ?
2. Quelle est l'espérance de S_n ?
3. Calculer, pour $k = 0, \dots, n$, la loi conditionnelle de X_1 sachant $S_n = k$. Montrer que c'est une loi de Bernoulli dont on précisera le paramètre.
4. Quelle est la loi conditionnelle de S_n sachant $X_1 = 0$?

5. Un rapport scientifique comporte exactement 50 feuilles. La probabilité de trouver au moins une faute sur chaque feuille est $p = 0,1$. Les évènements "Trouver au moins une faute sur la page i" sont indépendants, pour $i = 1, \ldots, 50$.
   (a) Chaque feuille qui comporte au moins une faute doit être réimprimée. Combien de feuilles doit-on réimprimer en moyenne ?
   (b) Un correcteur a lu une première page, elle ne comporte pas de faute. Quelle est la probabilité qu'il n'y ait aucune faute dans le rapport ?
   (c) On sait qu'un rapport contient exactement une faute. Quelle est la probabilité que cette faute se trouve sur la première page ?

Exercice 3. [9pts]
Soit λ un réel strictement positif. Soit (X, Y) un couple de variables aléatoires admettant la densité f définie sur $\mathbb{R}^2$ par

$$f(x, y) = c\lambda x^{-1} e^{-\lambda x} \mathbb{1}_{0<y<x}.$$

On rappelle que, pour tout entier $n > 1$, $\int_0^\infty \lambda^n x^{n-1} e^{-\lambda x} dx = \Gamma(n-1) = n!$

1. Montrer que $c = 1$.
2. Caculer la densité f_X de la loi marginale de X. Quelle est la loi marginale de X ?
3. En déduire la valeur de l'espérance de la variable aléatoire X.
4. Montrer que, pour $x > 0$, la densité $f_Y^{X=x}$ de la loi conditionnelle de Y sachant $X = x$, vaut $\frac{1}{x}\mathbb{1}_{0<y<x}$.
5. En déduire que, pour $x > 0$, l'espérance conditionnelle $E(Y|X = x)$ vaut $\frac{x}{2}$. En déduire $E(Y|X)$.
6. Quelle est la valeur de $E(Y)$? Effectuer le calcul par deux méthodes différentes.

Examen de RECHERCHE OPERATIONNELLE

Ensimag 1ère année

1^{er} juillet 2013

Durée : 2 heures
Polycopié et tous documents manuscrits autorisés. Calculatrices interdites.
Il sera tenu le plus grand compte de la rédaction. Vous devez expliquer tout ce que vous faites.
Veuillez noter sur votre copie le nom de votre enseignant : BIENIA ou STAUFFER ou SZIGETI

Exercice 1 : Un boulanger a en réserve 300 kilogrammes de céréales de type 1 et 500 kilogrammes de céréales de type 2. En mélangeant ces céréales, il produit 3 types de farine. Les proportions respectives, en %, des 2 types de céréales nécessaires à la fabrication des différentes farines sont données dans la table suivante.

céréales \ farines	1	2	3
1	75	50	25
2	25	50	75

Le boulanger produit 4 types de pains à partir des différentes farines. Les quantités nécessaires, en kg, de chaque sorte de farine pour produire un pain des différents types sont données dans la table suivante.

farines \ pains	1	2	3	4
1	0.2	0.25	0.1	-
2	0.3	-	0.1	0.1
3	-	0.25	0.3	0.2

Les pains sont vendus à l'unité aux prix respectifs de 1, 2.3, 1.5 et 1.6 €.

Modéliser le problème de la recherche d'un plan de production de pains maximisant le chiffre d'affaires du boulanger (on suppose qu'il vend toute sa production quelle qu'elle soit) sous forme d'un programme linéaire. Préciser clairement les variables de décision, les contraintes et la fonction objectif.

Exercice 2 : On considère un problème d'ordonnancement comportant 5 tâches A, B, C, D, E. Pour chacune des tâches on donne sa durée et la liste des tâches qui doivent la précéder.

Tâches	A	B	C	D	E
Durée	2	5	3	4	1
Prédécesseurs	$-$	$-$	A	A, B, C	B

On utilisera au choix une des méthodes de chemin critique à savoir la méthode potentiels-tâches (graphe des tâches) ou PERT (graphe des événements).

(a) Quelle est la durée minimale de la réalisation de ce projet ? Donner les détails de l'algorithme.
(b) Donner la liste des tâches critiques et des chemins critiques.
(c) Supposons que l'on peut diminuer la durée d'une seule tâche. Quelle tâche doit-on choisir pour diminuer le plus fortement la durée minimum du projet ? *(Justifier la réponse.)*
Donner la nouvelle durée de la tâche choisie et la nouvelle durée minimum du projet.

Exercice 3 : Considérons le programme linéaire suivant :

$$\begin{array}{l} \min \quad 8x_1 + 7x_2 + 3x_3 \\ \text{sous les contraintes} \\ 2x_1 + 1x_2 \qquad\qquad \geq 4 \\ 1x_1 + 2x_2 + 1x_3 \geq 5 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{array}$$

(a) Ecrire le dual de ce programme linéaire.

(b) Appliquer la phase I de l'algorithme du simplexe pour trouver une base réalisable du programme linéaire donné. *Utiliser la règle de Bland (choisir le plus petit indice en cas de choix multiple).*

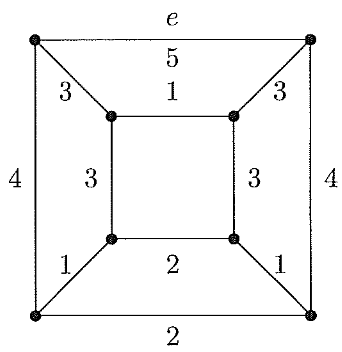
(c) Montrer que la base réalisable trouvée en (b) est optimale en utilisant les écarts complémentaires.

Exercice 4 : On considère le graphe G avec fonction de coût sur les arêtes indiqué sur la figure. *Justifier bien toutes les réponses.*

(a) Exécuter l'algorithme de Kruskal pour trouver un arbre couvrant de G de coût minimum.

(b) Montrer qu'aucun arbre couvrant de G de coût minimum ne contient l'arête e indiquée sur la figure.

(c) De quelle valeur minimum doit-on diminuer le coût de l'arête e pour qu'il existe un arbre couvrant de G de coût minimum qui contienne e?



Examen de théorie de l'information - Ensimag - 1A - Session 2

1 juillet 2013 — 10h15-12h15 — H 105

POLYCOPIE, NOTES DE COURS ET CALCULATRICES AUTORISES

Le sujet est composé de 3 problèmes indépendants. Le barème est donné à titre indicatif.

1 CODAGE SOURCE (8 points)

On considère une source simple S à 5 états notés 'a', 'b', 'c', 'd', 'e' de probabilités
$$\begin{array}{rcl} p(a) & = & 0,04 \\ p(b) & = & 0,06 \\ p(c) & = & 0,4 \\ p(d) & = & 0,2 \\ p(e) & = & 0,3 \end{array}.$$

1. En général, quelle est l'entropie maximale d'une source à 5 états ?
2. Quelle est l'entropie de la source S considérée ici ?
3. Quelle est la redondance de cette source S ?
4. Pourquoi est-il utile d'utiliser un codage source pour compresser S ?

On code d'abord la source S à l'aide du code binaire C_1 suivant :
$$\begin{array}{lcl} \text{a} & \to & 1110 \\ \text{b} & \to & 1111 \\ \text{c} & \to & 00 \\ \text{d} & \to & 100 \\ \text{e} & \to & 10 \end{array}$$

5. Dire pourquoi le code C_1 est instantané ?
6. Dire pourquoi C_1 ne peut pas être un code optimal ?
7. Construire C_2 un code de Fano-Shannon pour cette source.
8. Quelle la longueur moyenne des mots de C_2 ?
9. Quelle est l'efficacité de C_2 ?
10. Est-ce que le code C_2 sature l'inégalité de Kraft ?
11. Construire C_3 un code de Huffman pour cette source.
12. Quelle la longueur moyenne des mots de C_3 ?
13. Quelle est l'efficacité de C_3 ?
14. Le code C_2 est-il optimal ?

2 TRANSFORMATIONS DE L'ENTROPIE (4 points)

Dans cet exercice $N > 2$.

1. L'entropie d'une variable aléatoire (v.a.) à $N-1$ états peut-elle être supérieure à celle d'une v.a. à N états ?

2. On considère une v.a. X à N états de probabilités $p_i > 0$, $i = 1 \cdots N$ et la v.a. Y à $N-1$ états obtenus en groupant deux des états de X. Situer l'entropie de Y par rapport à celle de X.

Dans la suite, B est une v.a. binaire à valeurs dans $\{-1, +1\}$ avec $p(B = +1) = p$.

3. Soit une constante $\beta \in [1, +\infty[$ et $Y = B + \beta$. Quelle est l'entropie de Y ?
4. Soit $\alpha \in [1, +\infty[$ et $Z = \alpha B$. Quelle est l'entropie de Z ?

3 Codage canal (4 points)

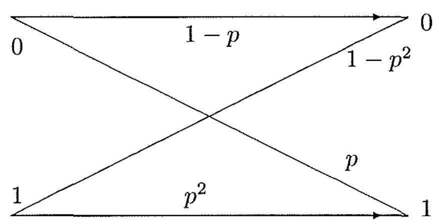
On considère un code canal dont les mots sont :

m_1	000
m_2	110

1. Quel est le rendement de ce code ?
2. Quelle est sa capacité de détection ?
3. Quelle est sa capacité de correction ?
4. En conservant des mots de même longueur et le même rendement :
   (a) proposer un meilleur code,
   (b) donner sa capacité de détection,
   (c) donner sa capacité de correction.

4 Canal binaire asymétrique (4 points)

Soit $0 < p < 1$, on considère le canal binaire asymétrique suivant :



Ce canal est à entrée X binaire $\{0; 1\}$ et à sortie Y binaire $\{0; 1\}$.

1. Quelle est sa matrice de transition ?
2. Est-il uniforme en entrée ?
3. Est-il uniforme en sortie ?
4. En supposant la loi de l'entrée X uniforme, calculer l'information mutuelle $I(X; Y)$.
5. Pensez-vous que cette information mutuelle soit égale à la capacité. Argumenter.

Théorie des langages 1 (session 2)

Durée : 1h30.
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 - Transformation d'automates (4 points)

▷ **Question 1** (3 points) Construire l'automate minimal reconnaissant le langage représenté par l'expression régulière suivante (vocabulaire $V = \{a, b\}$) :

$$b^*(a+\varepsilon)(a+ba+bba)^*[\varepsilon+b(b+\varepsilon)].$$

On prendra soin de détailler les étapes de la construction.

▷ **Question 2** (1 points) Caractériser en français le langage reconnu par cet automate.

Exercice 2 - Langages réguliers (5 points)

Etant donnés un vocabulaire V et un langage $L \subseteq V^*$, on définit le *langage des préfixes de* L par :

$$\mathrm{Pre}(L) \quad = \quad \{u \in V^* \,|\, \exists v \in V^* \ : \ uv \in L\}.$$

Autrement dit, le langage des préfixes de L est constitué des mots qui sont des préfixes de mots de L.

▷ **Question 1** (1 point) On pose $V = \{a, b\}$. A quoi correspond $\mathrm{Pre}(L)$ pour les langages L suivants :

- $L = \{$mots de V^* contenant deux $'a'$ consécutifs$\}$
- $L = \{$mots qui commencent par $'ab'\}$

▷ **Question 2** (2 points) Soit $A = (Q, V, \delta, I, F)$ un automate *quelconque* reconnaissant un langage L. Construire à partir de A un automate qui reconnaît $\mathrm{Pre}(L)$.

▷ **Question 3** (2 points) Démontrer que la construction est correcte, autrement dit, que si A' est l'automate obtenu à partir de A, alors on a bien $L(A') = \mathrm{Pre}(L)$.

Exercice 3 - Grammaires (11 points)

On considère les vocabulaires $V_T = \{\#, 0, 1\}$ et $V_N = \{S, T\}$, ainsi que la grammaire $G = (V_T, V_N, S, R)$, où R contient les règles suivantes :

$$\begin{aligned} S &\rightarrow 0S0 \mid 1S1 \mid 0T1 \\ T &\rightarrow 1T0 \mid \# \end{aligned}$$

▷ **Question 1** (2 points) Les mots suivants sont-ils dans $\mathcal{L}(G)$, le langage engendré par G ? On produira une dérivation des mots dans $\mathcal{L}(G)$, et on justifiera en quelques mots pourquoi les autres ne sont pas dans $\mathcal{L}(G)$.
- 11#00
- 0111#0001
- 11#100
- 1011#0011

Soit $V = \{0,1\}$. On définit les fonctions $\texttt{val} : V^* \to \mathbb{N}$ et $\texttt{lav} : V^* \to \mathbb{N}$ de la façon suivante[1] : si $w = w_0 \cdots w_n$, alors

$$\texttt{val}(w) = \sum_{i=0}^{n} 2^i w_i \quad \text{et} \quad \texttt{lav}(w) = \sum_{i=0}^{n} 2^{n-i} w_i.$$

Par exemple, $\texttt{val}(1111) = \texttt{val}(01111) = 15$, $\texttt{lav}(1111) = 15$ et $\texttt{lav}(01111) = 30$.

Notons que pour tout mot $w \in V^*$, on a $\texttt{val}(w) = \texttt{lav}(\tilde{w})$, où $\sim$ est la fonction miroir : si $w = w_0 \cdots w_n$, alors $\tilde{w} = w_n \cdots w_0$.

▷ **Question 2** (2 points) Définir la fonction $\texttt{lav}$ par induction structurelle sur V^*.

On peut remarquer que tout mot de $\mathcal{L}(G)$ est de la forme $u\#v$, où $u, v \in \{0,1\}^*$.

▷ **Question 3** (2 points) Prouver par induction que tout mot de $\mathcal{L}(G)$ est de la forme $u\#v$, où :
- $|u| = |v|$,
- $\texttt{val}(u) = \texttt{lav}(v) + 1$.

▷ **Question 4** (2 points) Soit $u\#v \in \mathcal{L}(G)$. On note w le plus grand préfixe commun à u et à $\tilde{v}$; il existe donc u' et v' tels que
- $u = wu'$ et $\tilde{v} = wv'$,
- u' et v' n'ont aucun préfixe commun.

De quelles formes sont u' et v' ?

▷ **Question 5** (3 points) On note $L = \{u\#v \mid u, v \in \{0,1\}^* \wedge |u| = |v| \wedge \texttt{val}(u) = \texttt{lav}(v) + 1\}$. Démontrer que $\mathcal{L}(G) = L$.

1. Notez qu'on confond dans cette définition les lettres 0 et 1, avec les valeurs correspondantes.

Grenoble INP - Ensimag $1^{\text{ière}}$ année juillet 2013

Théorie des Langages 2 (rattrapage)

Durée : 2 heures.
Documents : notes de cours et poly autorisés, livres interdits.

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

Exercice (6 points)

▷ Question 1 (3 points)

Reprenons l'exercice de l'examen de première session (les questions sont à la fin)...

Pour m, n fixés : $f_{\lhd m,n \rhd}(i) = 0$ si $\mathsf{U}(<m,n>)$ termine en au plus i pas d'exécution
$= 1$ sinon

... $f_{\lhd m,n \rhd}$ est calculable : il suffit de simuler U en comptant ses pas d'exécution (sur un ruban auxiliaire) et de retourner 1 si on dépasse i (retourner 0 si on s'arrête avant $i + 1$ pas)

$h(<m,n>) = 0$ si $\mathsf{U}(<m,n>)$ termine
$= 1$ sinon

... h n'est pas calculable : sinon $u = 1 - h$, fonction caractéristique de L_u le serait aussi

Soit g la fonction partielle de $\mathbb{N} \to \mathbb{N}$ telle que

$\mathrm{Dom}(g) = \{m \mid \exists n : \mathsf{U}(<m,n>) \text{ termine}\}$

$\forall m \in \mathrm{Dom}(g) : g(m) = \mathrm{Min}\{\mathsf{U}(<m,n>) \mid \mathsf{U}(<m,n>) \text{termine}\}$

... $g(m)$ est le Min *(dans $\mathbb{N}$) des valeurs de la fonction réalisée par m*

Il s'agit ici de montrer que g n'est pas calculable.

Soit m et n quelconques. $f_{\lhd m,n \rhd}$ est (totale) calculable. Il existe donc une MT m' réalisant $f_{\lhd m,n \rhd}$

1. Que peut valoir $g(m')$?
2. Que peut-on déduire de $g(m') = 0$?
3. Conclure...

▷ Question 2 (3 points)

Soit un ruban infini contenant uniquement des « blancs » B sauf une unique cellule contenant un a (cette cellule est n'importe où sur le ruban).

Définir une machine de Turing *déterministe*, à une seule tête, travaillant sur cet unique ruban, qui change le a en b puis s'arrête.

Indication : il faut chercher le a, mais on ne peut pas se contenter de partir dans un sens... il faut donc « faire l'essuie-glaces »...

Problème (14 points) Langages hors-contexte et reconnaisseur

On considère le langage **LB** décrit par la grammaire G suivante :

1	program → bloc	2	bloc → decl **begin** suite-inst **end**
3	suite-inst → suite-inst inst **;**	4	suite-inst → inst **;**
5	inst → bloc	6	inst → **idf :=** exp
7	exp → **idf**	8	exp → **num**
9	decl → ϵ	10	decl → **declare** suite-idf **: integer ;**
11	suite-idf → **idf ,** suite-idf	12	suite-idf → **idf**

Les éléments de vocabulaire terminal sont notés en gras. L'axiome est le non-terminal program.

▷ **Question 1 (3 points)**

Donner une grammaire LL(1) pour ce langage. On justifiera la réponse en calculant les directeurs de chaque règle. On pourra utiliser des numéros de règles, à condition que la numérotation utilisée soit claire.

▷ **Question 2 (3 points)**

Ecrire les procédures d'analyse permettant de reconnaître les blocs (non-terminal bloc) et les suite d'instructions (non-terminal suite-inst). On utilisera les conventions du TP (variable mot-cour contenant le mot courant et fonction lire-mot retournant le prochain mot). **On supposera que les procédures permettant de reconnaître les instructions (non-terminaux inst et exp) et les déclarations (non terminaux decl et suite-idf) sont déjà écrites.**

▷ **Question 3 (3 points)**

Le langage **LB** permet d'imbriquer des blocs qui délimitent la durée de vie des variables déclarées dans ces blocs. Soit le programme suivant :

```
1.    declare x : integer ;
2.    begin    x:= 1 ;
3.             declare y : integer ; begin y:=x ; end ;
4.             declare z, u : integer ;  begin u:=x ; z:=u ; end  ;
5.    end
```

Dans l'exemple précédent la variable `y` ne peut être utilisée que dans le bloc qui la déclare, de même que `z` et `u`. En terme de place mémoire on peut donc réutiliser le même emplacement mémoire pour stocker les variables `y` et `z` par exemple. On veut calculer la place mémoire maximum nécessaire à un programme. On supposera que les entiers sont codés sur 1 mot. Donner un calcul d'attributs, **sur la grammaire initiale** G, permettant d'évaluer le nombre de mots nécessaires (ici 3 : soit 1 mot pour `x`, 1 pour `y` ou `z` et 1 pour `u`).

▷ **Question 4 (2 points)**

Expliquer comment modifier les procédures de la question 2 pour inclure le calcul d'attributs précédent. On illustrera la réponse en donnant le code de la procédure bloc uniquement.

▷ **Question 5 (3 points)**

On s'intéresse ici à «fusionner» des grammaires LL(1) définies sur le même vocabulaire terminal V_T. Soit $G_1 = (V_T, V_{N_1}, S_1, R_1)$ et $G_2 = (V_T, V_{N_2}, S_2, R_2)$ deux grammaires LL(1) avec $V_{N_1} \cap V_{N_2} = \emptyset$. On construit la grammaire G suivante : $(V_T, \{S\} \cup V_{N_1} \cup V_{N_2}, S, \{S \to S_1, S \to S_2\} \cup R_1 \cup R_2)$, avec S un nouveau symbole ($S \notin (V_{N_1} \cup V_{N_2})$). On rappelle qu'on définit l'ensemble des directeurs d'une règle par (S étant l'axiome) :

$$Dir(A \to w) = \{\mathtt{x} \in (V_T \cup \{\$\}) \mid \exists w_1, w_2, w_3 \ : \ S\$ \Rightarrow^* w_1 A w_2 \Rightarrow w_1 w w_2 \Rightarrow^* w_1 \mathtt{x} w_3\}$$

- Donner un contre-exemple montrant que G n'est pas toujours LL(1).
- Donner une condition suffisante, la moins restrictive possible, sur les ensembles *Dir*, *Prem* et *Suiv* déjà calculés pour G_1 et G_2 permettant de garantir que G est LL(1).
- Vos conditions sont-elles nécessaires ?