Grenoble INP
ENSIMAG

Annales d'examens 1ère année

Année universitaire 2013/2014

PARTIELS DU MOIS DE NOVEMBRE 2013

ANALYSE POUR L'INGENIEUR - 2 PAGES

PROBABILITES APPLIQUEES - 2 PAGES

Examen Session 1
Lundi 4 novembre - 1h30

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
Soit (E, d) un espace métrique.

1. *Montrer que*
$$d'(x,y) = \frac{d(x,y)}{1+d(x,y)}$$
est une distance sur E.

2. *Monter que tout sous-ensemble $A \subset E$ est borné dans (E, d') et de diamètre inférieur ou égal à 1.*

3. *Soit $E = \mathbb{R}^2$ et d la distance euclidienne. Expliciter, dans (E, d'), les boules ouvertes $B(0_E; 1/2)$, $B(0_E; 1)$ et l'ensemble*
$$S(0_E; 1) = \{x \in E : \ d'(x, 0_E) = 1\}.$$

Exercice 2
Soit $E = C^0([0,1])$ l'espace des fonctions $f : [0,1] \to \mathbb{R}$ continues sur $[0,1]$.

1. *Montrer que*
$$\|f\|_1 = \int_0^1 |f(x)|\, dx$$
est une norme sur E. Quelle distance est associée à cette norme?

2. *Soit la suite $(f_n)_{n\geq 1}$ des fonctions définies par*
$$f_n(x) = \begin{cases} (2x)^n, & x \in [0, \frac{1}{2}], \\ 1, & x \in [\frac{1}{2}, 1]. \end{cases}$$
Montrer que la suite $(f_n)_{n\geq 1}$ est une suite de Cauchy de $(E, \|\cdot\|_1)$.

3. *L'espace E muni de la norme $\|\cdot\|_1$ est-il complet?*

Exercice 3
Soit (E, d) un espace métrique complet, et $(x_n)_{n\in\mathbb{N}}$ une suite d'éléments de E.

1. *On suppose que $\sum_{n\in\mathbb{N}} d(x_n, x_{n+1})$ converge. Montrer que la suite $(x_n)_{n\in\mathbb{N}}$ est convergente.*

2. *Montrer que $\sum_{n\in\mathbb{N}} d(x_n, x_{n+1})^2$ peut être convergente, sans que $(x_n)_{n\in\mathbb{N}}$ le soit. On pourra considérer $E = \mathbb{R}$ et $x_n = \ln n$ pour $n \in \mathbb{N}^*$.*

Exercice 4

Soit E un espace vectoriel normé et C un sous-ensemble non vide, convexe[1] *et compact de E. Soit $f \in \mathcal{L}(E)$ une application linéaire continue de E dans E, telle que $f(C) \subset C$. Soit enfin $a \in C$ fixé. On définit l'itération suivante :*

$$a_0 = a, \quad a_n = \frac{1}{n+1}\left(a + f(a) + f(f(a)) + \cdots + f^{(n)}(a)\right), \; n \in \mathbb{N}.$$

1. *Montrer que (a_n) est une suite d'éléments de C.*
2. *Montrer que $f(a_n) - a_n$ converge vers 0 lorsque $n \to +\infty$.*
3. *En déduire que f admet au moins un point fixe dans C.*
4. *Montrer que l'ensemble des points fixes de f est convexe et compact.*
5. *On se donne maintenant une famille finie d'applications linéaires continues $(f_i)_{i\in I}$ qui commutent, telles que $f_i(C) \subset C$. Montrer qu'elles admettent un point fixe commun. On pourra commencer par deux applications f_1 et f_2, en remarquant que l'ensemble des points fixes de f_1, soit F_1, est stable par l'autre (c'est à dire $f_2(F_1) \subset F_1$).*

[1] On dit que $C \subset E$ est convexe si $\forall x, y \in C$, $\forall \alpha \in [0,1]$, $\alpha x + (1-\alpha)y \in C$

Ensimag 1re année **2013-2014**

Probabilités Appliquées
Novembre 2013

Durée 2h00 - Une feuille de format A4 manuscrite autorisée. Pas de calculatrices. Veuillez inscrire votre numéro de groupe sur la copie.

Exercice I (5 pts) – On dispose d'un dé équilibré à 6 faces.

1) Décrire un algorithme de rejet utilisant des lancers d'un dé à 6 faces pour simuler un dé à 5 faces (justifier la réponse) ? Quelle est la loi du nombre de lancers du dé à 6 faces dans cet algorithme ? Combien de lancers effectue-t-on en moyenne pour obtenir une simulation d'un dé à 5 faces ?

2) Décrire un algorithme utilisant un seul lancer du dé à 6 faces pour simuler le lancer d'un dé à 3 faces (justifier la réponse) ?

3) Quelle est la probabilité que le numéro résultant du lancer d'un dé à 5 faces soit inférieur ou égal au numéro résultant du lancer d'un dé à 3 faces (lancers supposés indépendants) ?

Exercice II (4 pts) – À Monte-Carlo, lorsque l'on joue à *pile* ou *face* contre une personne, il y a une chance sur deux pour qu'il s'agisse d'un tricheur. Lorsque l'on joue contre un tricheur, il est impossible de gagner. Dans le cas contraire, on gagne avec la probabilité $1/2$.

1) On joue une partie de *pile* ou *face* contre une personne à Monte-Carlo et on perd cette partie. Quelle est la probabilité d'avoir joué contre un tricheur ?

2) On joue et on perd n ($n \geq 2$) parties consécutives contre la même personne. Quelle est la probabilité d'avoir joué contre un tricheur ?

3) Après combien de parties consécutives perdues contre la même personne peut-on estimer avoir joué contre un tricheur avec une probabilité supérieure à $1 - \epsilon$ $(0 < \epsilon < 1)$?

Exercice III (4 pts) – On lance un dé équilibré à 6 faces. Soit U une variable de loi uniforme sur $(0, 1)$, dont le tirage est indépendant du lancer précédent. On considère la variable Z égale à $-U$ si le résultat du dé est inférieur ou égal à 2, et égale à $2U$ sinon.

1) Déterminer la fonction de répartition de la variable aléatoire Z. Donner une représentation graphique de cette fonction.

2) Calculer la valeur médiane, t_{med}, de la variable Z, définie par l'équation suivante

$$\mathrm{P}(Z \leq t_{\text{med}}) = \frac{1}{2}.$$

Problème (7 pts) – Gogoland est un lieu avec des machines à sous. En jouant une partie sur une machine, on peut y gagner un jackpot avec la probabilité p, et ne rien gagner avec la probabilité $q = 1-p$. Les machines sont indépendantes les unes des autres. Le patron du lieu reçoit un visiteur. Il ne dit pas au visiteur combien il possède de machines à sous, mais il lui dit que **chacune de ses machines a joué plus de k parties sans qu'aucun jackpot n'ait encore été gagné**. Pour le visiteur, le nombre inconnu de machines à sous est modélisé comme la réalisation d'une variable aléatoire notée N.

1) Sous quelles hypothèses la loi de probabilité décrivant le nombre de parties jouées sur une machine à sous avant un jackpot est la loi géométrique de paramètre p ? On supposera ces hypothèses vérifiées.

2) Soit T le plus petit nombre de parties jouées quelle que soit la machine à sous. Montrer que

$$\mathrm{P}(T > k | N = n) = q^{nk}, \quad n \geq 1.$$

3) On suppose que N suit la loi géométrique de paramètre $(1-\alpha)$, où α vérifie $0 < \alpha < 1$. Calculer la probabilité de l'événement $(T > k)$.

4) À l'aide des questions précédentes, montrer que la probabilité conditionnelle de l'événement $(N = n)$ sachant $(T > k)$ est égale à

$$\mathrm{P}(N = n | T > k) = (1 - \alpha q^k)(\alpha q^k)^{n-1}, \quad n \geq 1.$$

Identifier la loi de probabilité conditionnelle définie ci-dessus.

5) Vérifier que l'expression précédente admet une limite pour tout $n \geq 1$ lorsque α tend vers 1. Vérifier que le passage à limite définit une loi de probabilité que l'on peut reconnaître.

6) Calculer l'espérance de la loi identifiée à la question 5). On suppose que α tend vers 1, que $p = 10^{-6}$ et que $k = 50000$. En déduire que l'espérance du nombre de machines à sous tend vers une valeur proche de $1/(1-e^{-0.05}) \approx 21$.

PARTIELS DU MOIS DE JANVIER 2014

ALGORITHMIQUE 1 -	6 PAGES
ANALYSE POUR L'INGENIEUR -	2 PAGES
ARCHITECTURE 1 (Circuit Numériques et Eléments d'Architecture) -	6 PAGES
ECONOMIE -	3 PAGES
INTRODUCTION AUX RESEAUX -	4 PAGES
PROBABILITES APPLIQUEES -	2 PAGES
THEORIE DES LANGAGES 1 -	3 PAGES

Algorithmique 1

Ensimag - 1A

Janvier 2014

Durée : 3h
Machines électroniques interdites **document autorisé : une feuille manuscripte recto-verso**

Les trois parties du sujet sont indépendantes.

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

---

1 Mots palindromes (4 points)

Un mot palindrome, ou *mot miroir*, est un mot dont l'ordre des lettres reste le même qu'on le lise de droite à gauche ou de gauche à droite. Par exemple, les suites de caractères suivantes :

```
ABBA
LAVAL
```

sont des mots palindromes.

On s'intéresse ici à l'implantation d'une fonction ADA `EstPalindrome` qui renvoie vrai si le mot passé en paramètre est un palindrome, et faux sinon.

Mot vu comme un tableau de caractères

Dans un premier temps, on considère qu'un mot est stocké dans un objet ADA `String` qui représente une chaine de caractères sous la forme d'un tableau d'objets `Character`.

1. (½ point) Ecrire la procédure **procedure** `AfficherMot(Mot:` **in** `String)` qui affiche le mot en paramètre sur la sortie standard.
2. (½ point) Ecrire la fonction **function** `EstPalindrome(Mot: String)` **return** `Boolean` qui renvoie `True` si le mot passé en paramètre est un mot palindrome, et `False` sinon.

Mot vu comme une liste de caractères

Dans cette partie, on considère les mots comme une liste doublement chainée de lettres :
- Une lettre est caractérisée par le caractère qui lui est associé, un pointeur vers la lettre suivante du mot auquel elle appartient ainsi qu'un pointeur vers la lettre précédente.
- Un mot est défini comme un objet ADA contenant deux pointeurs : un pointeur vers la première lettre du mot et un pointeur vers la dernière.

L'interface du paquetage `Mot` associé s'écrit donc:

```
type Lettre;

type Pointeur is access Lettre;

type Lettre is record
   C: Character;
   Prec: Pointeur;
   Suiv: Pointeur;
end record;

type Mot is record
   Tete: Pointeur;
   Queue: Pointeur;
end record;
```

3. ($\frac{1}{2}$ point) Ecrire la fonction **function** `LongueurMot(M: Mot)` **return** `Integer` qui retourne le nombre de lettres qui composent le mot passé en paramètre.
4. (1 point) Ecrire la procédure **procedure** `CreerMot(S:` **in** `String, M:` **out** `Mot)` qui crée la liste de caractères M associée à la chaine S.
5. ($1\frac{1}{2}$ points) Ecrire la fonction **function** `EstPalindrome(M: Mot)` **return** `Boolean` qui renvoie `True` si le mot passé en paramètre est un mot palindrome, et `False` sinon.

2 Gestion d'un annuaire

Cet exercice s'intéresse à l'implémentation d'un paquetage Ada fournissant un objet représentant un annuaire de contacts et des primitives permettant de manipuler cet objet.

Dans cette partie, un contact est vu comme un couple *(Nom, Numero)* où `Nom` désigne le nom de famille de la personne et `Numero` son numéro de téléphone portable à 10 chiffres.

L'implémentation Ada correspondante s'écrit:

```
type Contact is record
   Nom: String(1..32) := (others => ' ');
   Numero: Integer := 0;
end record;
```

Un annuaire est un ensemble de contacts. On l'implémente ici sous la forme d'un tableau de taille fixe dont chacune des cases peut contenir un élément de type `Contact`.

On peut donc définir le type `Annuaire` de cette façon:

```
type Annuaire is array(Integer range <>) of Contact;
```

et déclarer un nouvel annuaire de taille 16 de cette façon:

```
A: Annuaire(1..16);
```

On supposera ici que les cellules d'un tableau d'annuaire sont **triées par leurs numéros de téléphone** (le numéro le plus petit en première case, le plus grand en dernière case).

Recherche dans un annuaire

On suppose l'existence d'une fonction:

```
function SontEgaux(Nom1: in String; Nom2: in String) return Boolean;
```

qui renvoie `True` si les noms `Nom1` et `Nom2` sont les mêmes.

1. (½ point) Ecrire une procedure **itérative**

   ```
   RechercherNumero(A: in Annuaire; Nom: in String; Numero: out Integer);
   ```

   qui recherche dans l'annuaire le numéro de téléphone du contact dont le nom est `Nom`.

2. (1½ points) Ecrire une procedure

   ```
   RechercherNom(A: in Annuaire; Numero: in Integer; Nom: out String);
   ```

   qui recherche dans l'annuaire le nom du contact dont le numéro de téléphone est `Numero`. Afin de limiter le nombre de contacts à tester, cette procédure devra faire appel à une procédure **récursive** de prototype

   ```
   RechercherNomRec(A: in Annuaire; Deb, Fin: in Positive;
                    Numero: in Integer; Nom: out String);
   ```

Gestion dynamique d'un annuaire

On souhaite dans cette partie être capable d'ajouter et d'enlever des contacts dynamiquement tout en respectant la contrainte de représentation en mémoire d'un annuaire sous la forme d'un tableau de `Contact` de taille fixe.
En particulier, votre nouveau paquetage devra au minimum fournir une nouvelle procédure `AjouterContact` pour gérer l'ajout d'un contact de l'annuaire.
Cette partie est plus libre que les précédentes: c'est à vous de définir les prototypes des procédures à rajouter, d'étendre les types introduits en section précédente comme bon vous semble, voire d'en introduire de nouveaux.

3. (2 points) Donnez l'interface et l'implémentation de ce nouveau paquetage de gestion des annuaires, en prenant bien soin de **justifier vos choix**.

3 Master Mind (12 points)

Introduction

Vous avez certainement joué au Mastermind dans votre enfance. Souvenez-vous...

Il se présente généralement sous la forme d'un plateau perforé de 10 rangées de 4 trous pouvant accueillir des pions de couleurs. En général, les couleurs sont rouge, jaune, vert, bleu, orange, blanc, violet et fuchsia. Il y a également des pions blancs et rouges pour donner des indications à chaque étape du jeu.

Le jeu se joue à 2 joueurs. Le joueur 1 commence par définir une énigme, c'est-à-dire une combinaison de 4 pions que l'autre joueur devra deviner. Il place, pour ce faire, sans être vu de l'autre joueur, sa combinaison de pions à l'arrière d'un cache qui les masquera à la vue de celui-ci jusqu'à la fin de la manche.

Le joueur 2, c'est-à-dire celui qui n'a pas défini l'énigme, devra trouver la combinaison cachée (mêmes couleurs dans le même ordre). Pour cela, à chaque tour, le joueur 2 propose une combinaison en plaçant des pions sur une rangée libre du plateau. Une fois les pions placés, le joueur 1 indique à quel point la combinaison proposée se rapproche de la combinaison cachée de la façon suivante :

- il indique le nombre de pions de la bonne couleur bien placés en posant autant de pions rouges ;
- il indique le nombre de pions de la bonne couleur, mais mal placés, en posant autant de pions blancs.

Le joueur 2 gagne s'il réussit à trouver exactement la combinaison cachée (obtenant donc une réponse de quatre pions rouges) avant qu'il n'y ait plus de rangées libres sur le plateau, c'est-à-dire en faisant au plus 10 propositions. Il perd s'il fait 10 propositions incorrectes.

Implémentation du Mastermind

Dans cet exercice, pour simplifier, on ajoutera à la règle la restriction que les combinaisons sont toujours constituées de quatre pions **de couleurs différentes** (que ce soit la combinaison définie par le joueur 1 ou les combinaisons proposées par le joueur 2).

Les différentes fonctions et procédures demandées ne sont pas indépendantes, mais les questions le sont, c'est-à-dire que vous pouvez répondre à une question en supposant que vous disposez des fonctions et procédures demandées dans les questions précédentes, même si vous ne les avez pas faites.

La spécification du paquetage Mastermind est réduite à une unique procédure, `JouerPartie`, qui permettra de mener une partie.

```
package Mastermind is
   procedure JouerPartie;
end Mastermind;
```

Dans le corps du paquetage, pour simplifier, les couleurs des pions sont représentées par un sous-type `Couleur` du type `Character`. Les combinaisons de pions sont simplement des tableaux de `Couleurs`.

```
subtype Couleur is Character range 'a'..'h';
   -- les 8 couleurs des pions constituant les combinaisons
type Combinaison is array (1..4) of Couleur;
```

1. ($2\frac{1}{2}$ points) Écrire la procédure : **procedure** `EntrerCombinaison(C: `**out**` Combinaison)` qui permet la saisie au clavier d'une combinaison. Les couleurs entrées doivent être toutes distinctes ; si ce n'est pas le cas on lèvera l'exception `Couleur_En_Double` lorsqu'une couleur déjà utilisée l'est de nouveau.

   Une couleur saisie peut être incorrecte soit parce qu'elle est en double, soit parce que le caractère saisi ne représente pas une couleur. On rappelle que, dans ce dernier cas, l'exception `Constraint_Error` est levée par le système lors de l'affectation. Cette procédure gère les saisies incorrectes de la façon suivante :
   — inviter l'utilisateur à saisir de nouveau la couleur.
   — au-delà de 3 erreurs pour cette couleur, lever l'exception `Partie_Annulee` (cette exception sera traitée dans la procédure principale).

2. (2 points) Ecrire la procédure :

```
procedure EvaluerCombinaison(Enigme, Proposition:in Combinaison;
                             NbBienPlaces, NbMalPlaces:out Natural)
```

   qui détermine le nombre de pions bien placés et le nombre de pions mal placés dans une proposition par rapport à l'énigme. On rappelle que **ni la proposition ni l'énigme ne peuvent contenir plusieurs fois la même couleur.**

3. ($\frac{1}{2}$ point) Ecrire la fonction :
   function `CombinaisonGagnante(Enigme, Proposition:`**in**` Combinaison)` **return** `Boolean`
   qui détermine si la proposition a découvert l'énigme.

4. (2 points) Ecrire la procédure **procedure** `JouerPartie` qui effectue une partie entière :
   1. L'ordinateur demande au joueur 1 d'entrer l'énigme.
   2. L'ordinateur joue ensuite le rôle du joueur 1 : il demande dix fois au joueur 2 d'entrer une combinaison, en lui indiquant à chaque fois le nombre de pions bien et mal placés.
   3. Si le joueur 2 n'a pas trouvé la bonne combinaison après 10 essais, on lui indique qu'il a perdu. On lui demande alors s'il souhaite voir l'ordinateur tenter sa chance ; si oui, on exécute la procédure `JouerAutomatique(enigme)` (qui sera implémentée plus loin dans cet exercice).

   La condition de victoire sera gérée par une exception `Victoire`. Dès que l'utilisateur a trouvé la bonne combinaison, cette exception est déclenchée. La procédure affiche alors un message indiquant que le joueur a gagné et se termine.

   La procédure doit également traiter l'exception `Partie_Annulee` : lorsque cette exception est déclenchée, un message est affiché indiquant que la partie a été annulée en raison d'un trop grand nombre de saisies incorrectes, puis la procédure se termine.

5. (1 point) Dans un objectif de polyvalence du jeu, imaginons que nous souhaitions rendre paramétrables le nombre de couleurs ainsi que le nombre de rangées. Que faudrait-il modifier et comment (on ne demande pas ici un code complet, mais des explications claires) ?

Stratégie pour le joueur 2

Afin d'implémenter la procédure `JouerAutomatique`, qui joue le rôle du joueur 2 (sans tricher), il est nécessaire de définir une stratégie déterminant une combinaison à proposer en fonction des informations obtenues jusqu'à présent.

La stratégie la plus simple (pour un ordinateur) consiste à énumérer toutes les combinaisons possibles dans un ordre donné et à proposer la première qui est compatible avec les informations que l'on a. La compatibilité signifie qu'il est possible d'après ce qu'on sait que la nouvelle combinaison soit la bonne réponse, autrement dit, qu'en comparant chacune des propositions faites précédemment avec la nouvelle combinaison on obtient à chaque fois exactement le même nombre de pions bien et mal placés que ce que le joueur 1 avait répondu.

On appelle combinaison sans doublon une combinaison de 4 couleurs différentes. Seules ces combinaisons sont autorisées avec notre version de la règle.

Pour énumérer les combinaisons, on propose l'algorithme simple suivant :
— on part d'une combinaison arbitraire sans doublon
— pour passer d'une combinaison à la suivante :

1. Si c'est possible, on remplace la couleur située en dernière position par son successeur dans l'ordre des couleurs. Sinon on la remplace par la première couleur dans l'ordre des couleurs et on modifie de la même façon la couleur située juste avant dans la combinaison, et ainsi de suite.

   L'idée est la même que sur un compteur heures, minutes, secondes : quand les secondes reviennent à zéro on augmente le compteur des minutes, etc.

2. Si la nouvelle combinaison ainsi obtenue comporte un doublon, on recommence l'étape 1. Sinon on s'arrête.

6. (2 points) Écrire une procédure `Combinaison_Suivante(C : `**in out**` Combinaison)` qui transforme C en la combinaison sans doublon suivante, en suivant l'algorithme ci-dessus. On pourra utiliser une fonction auxiliaire
   function `Sans_Doublon(C : Combinaison)` **return** `Boolean` (à définir).

7. (2 points) Écrire la procédure **procedure** `JouerAutomatique(Enigme : Combinaison)`. Cette procédure ne doit **pas tricher**, c'est-à-dire que le paramètre Enigme ne peut être utilisé **que** comme paramètre dans un appel à `EvaluerCombinaison`, et ce au plus dix fois au cours de l'exécution.

   Le déroulement est similaire à celui de `JouerPartie`, si ce n'est qu'à chaque étape la combinaison à tester n'est pas demandée au clavier mais déterminée par la stratégie décrite ci-dessus (il est donc nécessaire, afin d'appliquer cette stratégie, de conserver les réponses obtenues à chaque tour).

   En principe cette procédure devrait afficher tout ce qu'elle fait, cependant dans le cadre de l'examen on ne vous demande pas le détail des `Put`. Indiquez seulement si le joueur 2 a gagné ou perdu.

Examen final
Lundi 20 janvier 2014 - 2h

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

Exercice 1
Soit $n \in \mathbb{N}^*$, X un espace de Banach et $f : X \to \mathbb{R}$ de classe $\mathcal{C}^n$ sur X. On suppose que $f(tx) = t^n f(x)$ pour tout $t \in \mathbb{R}$.

1. *On considère dans cette question le cas $n = 1$. Soit $x \in X$ fixé, et $\varphi(t) = f(tx)$.*
   (a) *Exprimer la dérivée de φ en t en fonction de la différentielle de f.*
   (b) *En déduire que $\forall x \in X, \quad f(x) = df_0(x)$.*
2. *Dans le cas général, montrer que*
$$f(x) = \frac{1}{n!} d^n f_0(x^{(n)})$$
où $x^{(n)}$ est le n-uplet $(x, x, \ldots, x)$ et $d^n f_0$ la différentielle n-ième de f au point $0 \in X$.

Exercice 2
Déterminer les limites suivantes :
$$(a)\ \lim_{n\to+\infty} \int_0^1 (1+nx^2)(1+x^2)^{-n} dx \qquad (b)\ \lim_{n\to+\infty} \int_0^\infty \frac{n\sin(\frac{x}{n})}{x(1+x^2)} dx$$

Exercice 3
On définit le produit de convolution de deux fonctions f et g **positives** *sur $\mathbb{R}$ par :*
$$\forall x \in \mathbb{R}, \quad f * g(x) = \int_{\mathbb{R}} f(t) g(x-t) dt$$

1. *Montrer que $f * g$ est une fonction bien définie, à valeurs dans $\overline{\mathbb{R}_+}$, qui vérifie :*
$$\int_{\mathbb{R}} f * g(x) dx = \left(\int_{\mathbb{R}} f(x) dx\right) \left(\int_{\mathbb{R}} g(x) dx\right)$$
En déduire que si f et g sont intégrables, alors $f * g$ est fini presque partout sur $\mathbb{R}$.
2. *On suppose maintenant que $f \in L^1(\mathbb{R})$ et $g \in L^\infty(\mathbb{R})$. Montrer qu'alors $f * g$ est une fonction bornée sur $\mathbb{R}$.*
En admettant la propriété suivante :
$$\lim_{h\to 0} \int_{\mathbb{R}} |h(x+h) - h(x)| dx = 0, \qquad \forall h \in L^1(\mathbb{R}),$$
montrer que $f * g$ est de plus uniformément continue sur $\mathbb{R}$.
3. *Exemple : calculer*
- $f * 0$
- $f * \mathbb{1}$, *pour $f \in L^1(\mathbb{R})$ où $\mathbb{1}$ est la fonction constante égale à 1 sur $\mathbb{R}$.*
- $f * f$, *avec $f = \mathbb{1}_{[0,1]}$, fonction indicatrice de l'intervalle $[0, 1]$.*

Vérifier que la fonction obtenue est bien continue!

Exercice 4

1. *Montrer, en utilisant la définition, que la transformée de Fourier d'une fonction réelle paire est une fonction réelle paire.*

2. *Soit f la fonction définie sur $\mathbb{R}$ par :*

$$f(x) = e^{-2\pi|x|}$$

   Calculer $\hat{f}$, la transformée de Fourier de f.

3. *Montrer que $\hat{f}$ est intégrable sur $\mathbb{R}$. En déduire la transformée de Fourier de la fonction :*

$$g(x) = \frac{1}{1+x^2}$$

Architecture 1 : Circuits Numériques et Éléments d'Architecture
Examen

ENSIMAG 1A

Année scolaire 2013–2014

— Durée : 3h. Tous documents et calculatrices autorisés ;
— Le barème est donné à titre <u>indicatif</u> ;
— Les exercices sont indépendants et peuvent être traités dans le désordre, 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 (1.5 pts)

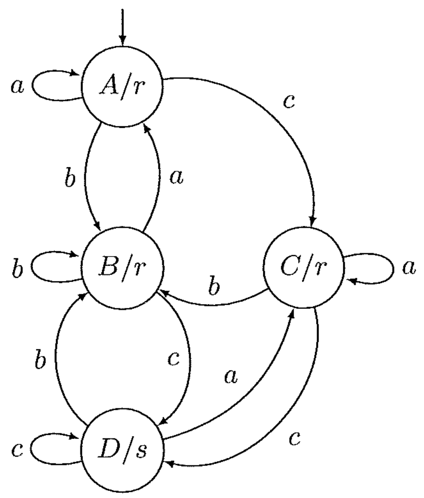
Question 1 Le répertoire d'un cache contient, en plus des bits de *tag* permettant de vérifier si l'adresse que l'on cherche à accéder est bien celle qui correspond aux données présentes dans le cache, un bit supplémentaire que nous avons appelé V dans le cours. Précisez, en une phrase d'une ligne, l'utilité de ce bit, et en une autre phrase d'une ligne, comment il est mis à jour.

Question 2 Le processeur Cortex A15 de ARM possède un cache de niveau 1 de 32 Ko, chaque ligne étant constituée de 16 mots de 32 bits (soit 64 octets). La taille des données manipulées par le processeur est de 32 bits.

Donnez la position dans les 32 bits d'adresse et le nombre de bits de chacun des champs *offset*, *index*, et *tag*. On pourra dessiner les 32 bits d'adresse et noter les champs sur le dessin, comme fait en cours.

Ex. 2 : Synthèse d'automate (3.5 pts)

Soit l'automate défini par le graphe ci-dessous, prenant en entrée des lettres dans l'ensemble $\{a, b, c\}$ et affichant en sortie des lettres dans l'ensemble $\{r, s\}$.



Les états sont codés sur les variables d'état q_1 et q_0, les lettres en entrée sur e_1 et e_0 et les lettres en sortie sur z. Le codage choisi est le suivant :

	VARIABLES D'ÉTAT	
ÉTAT	q_1	q_0
A	0	0
B	0	1
C	1	0
D	1	0

	e_1	e_0
a	0	0
b	0	1
c	1	0

	z
r	0
s	1

Question 1 Donnez les équations minimisées des variables d'états $q'1$ $q'0$ et de la sortie z. On utilisera des tables de Karnaugh pour minimiser les équations.

Question 2 Dessinez le schéma du circuit implémentant cet automate, en utilisant des bascules D et des portes AND, OR, NOT.

Question 3 Quelle séquence reconnait cet automate, si on suppose que la sortie s indique "séquence reconnue" ?

Ex. 3 : Conception Partie-Contrôle/Partie Opérative d'un (7 pts)

Il s'agit de réaliser un circuit de division d'entiers positifs suivant l'algorithme ci-dessous, qui permet de diviser l'entier 64 bits constitué par la concaténation des variables x et y (notée $x||y$, comme en cours) par l'entier 32 bits z :

```
1  i : integer ; c : boolean;
2  Get(x) ; -- x := x0
3  Get(y) ; -- y := y0
4  Get(z) ; -- z := z0
5  c := false;
6  i := 31;
7  while i >= 0 do
8    if c = false then
9      c := MSB(x);
10     -- lignes 11 et 12 réalisent le décalage à gauche de (x||y) de 1 position
11     x := (x * 2)||MSB(y);
12     y := y * 2;
13     c := c xor(x < z);
14     x := x - z;
15   else
16     c := MSB(x);
17     -- lignes 18 et 19 réalisent le décalage à gauche de (x||y) de 1 position
18     x := (x * 2)||MSB(y);
19     y := y * 2;
20     x := x + z;
21     c := c xor(x < z);
22   end if
23   if c = false then
24     y := y + 1;
25   end if
26   i := i - 1;
27 end while
28 Put(y);
```

MSB(α) est le "most significative bit" (bit de poids fort, celui le plus à gauche) de la variable α, $||$ est l'opérateur de concaténation, et le résultat de la multiplication par 2 donne un résultat sur 32 bits, donc le bit de poids fort du résultat est perdu.

Il s'agit de proposer une implantation de cet algorithme sous la forme PC/PO.

Question 1 Définissez l'ensemble des opérations à réaliser ; déduisez-en le type des unités fonctionnelles (registres ou opérateurs). En particulier, examinez la séquence suivante :

$c :=$ MSB(x);
$--$ décalage à gauche de $(x||y)$ de 1 position
$x := (x * 2)||$MSB(y);
$y := y * 2$;

— Quel type de registres peut-on utiliser pour l'implémenter ?

Une autre portion de code à examiner :

$y := y * 2$;
......
if $c = false$ **then**
| $y := y + 1$;
end if

— Pour la séquence ci-dessus, est-il nécessaire d'utiliser un incrémenteur ?

Question 2 Proposez une partie opérative utilisant un additionneur/soustracteur et un comparateur (avec les sorties LT : *less than* et GT : *greater than*), en faisant bien apparaître les signaux de commandes et les compte-rendus de/vers la partie contrôle.

Question 3 Revenons à l'algorithme. Quelles opérations peut-on réaliser en parallèle avec la partie opérative que vous avez proposée ?
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.

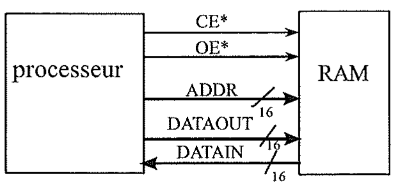
Question 4 Proposez un graphe d'automate d'état décrivant la partie contrôle qui pilote cette partie opérative. Spécifiez la valeur des différents signaux de commande pour chaque état (vous pouvez donner des valeurs par défaut pour simplifier l'écriture).

Question 5 Peut-on modifier 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.

Que faut-il ajouter dans la PO pour cela ?

Ex. 4 : Conception de processeur (8 pts)

L'objectif de cet exercice est de concevoir un processeur 16 bits suivant la méthode vue durant les TD 9, 10 et 11 pour un processeur 8 bits.



Ce processeur 16 bits traite des données 16 bits, des adresses 16 bits et a 16 registres généraux, R0 à R15. La notion de *mot* fait donc toujours référence à un mot de 16 bits dans la suite du texte.

On ne considère d'abord que le jeu d'instructions suivant :

— `st ri, @mem` stocke le contenu du registre `ri` dans le mot de 16 bits situé à l'adresse `@mem` ;

— `ld @mem, ri` stocke le contenu du mot de 16 bits situé à l'adresse `@mem` dans le registre `ri` ;

— `op rs, rd` stocke dans le registre `rd` le résultat de l'opération `rd op rs`.

Les instructions `st ri, @mem` et `ld @mem, ri` sont codées sur 2 mots de 16 bits consécutifs, les instructions `op rs, rd` sont codées sur 1 seul mot.

Les opérations à deux opérandes supportées par ce processeur sont l'addition (instruction `add`), la soustraction (instruction `sub`), la conjonction (instruction `and`), la disjonction (instruction `or`) et la disjonction exclusive (instruction `xor`).

Les opérations unaires supportées (`rd := op rd`) sont la négation (instruction `not`), le décalage d'un bit vers la gauche (instruction `shl`) et le décalage arithmétique d'un bit vers la droite (instruction `shr`).

Une partie opérative est proposée en dernière page. On vous demandera de compléter ce schéma : n'oubliez pas de le joindre à votre copie !

Cette partie opérative comporte les 16 registres R0 à R15, le registre d'instruction IR, le compteur programme PC ainsi que 2 registres T et U nécessaires à l'exécution des instructions.

Pour les registres R0, .., R15, PC et T, l'entrée est représentée à gauche, la sortie à droite. Tous ces registres sont connectés en sortie aux entrées de l'unité arithmétique et logique ALU via des multiplexeurs, vers les 2 entrées pour R0, .., R15, vers une des entrées pour PC et T.

Chacun des registres reçoit un signal de chargement $ld < nomduregistre >$ connecté à l'entrée *ClockEnable*, un signal d'horloge *clk* et le signal de reset, *rst*, qui les initialise à 0, sauf PC qui est initialisé à 0x4000.

L'ALU à utiliser est décrite dans le tableau ci-dessous :

Sel	S
0000	A or B
0001	A xor B
0010	A and B
0011	not A
0100	A + B
0101	A - B
0110	A << 1
0111	A >> 1
1000	A
1001	B
1010	A + 1
1100	A + B

Les sorties Z et N indiquent respectivement que la sortie S est nulle et que la sortie S est négative.

Question 1 Ajoutez sur le schéma de la partie opérative donnée en annexe les signaux de commande nécessaires.

Question 2 Pour le jeu d'instructions donné ci-dessus, proposez un graphe d'états de l'automate de la partie contrôle, en associant aux états les opérations de transfert de registre à registre et aux transitions les fonctions booléennes *AccesMem(IR)*[1] et *AccesEcr(IR)*[2]

Question 3 Complétez la spécification de la partie contrôle en indiquant la valeur des signaux de commande pour chaque état, en utilisant les fonctions booléennes *selRs(IR)*[3], *selRd(IR)*, *selRi(IR)*, *selUAL(IR)*[4].

Question 4 Proposer un codage astucieux de chaque instruction en essayant de minimiser les fonctions utilisées précédemment.

Dans les questions suivantes, nous allons enrichir le jeu d'instructions en ajoutant des modes d'adressage et des instructions.

Question 5 On propose un mode d'adressage `val(rj)`, la valeur `val` étant codée dans le 2ème mot de l'instruction et l'adresse calculée étant `val + rj`. Proposez un codage pour les instructions `st ri, val(rj)` et `ld val(rj), ri`. Est-il nécessaire de modifier la partie opérative ? Quelles modifications doivent être apportées à la partie contrôle ?

Question 6 On ajoute une instruction `li val, ri`, qui charge une valeur immédiate de 16 bits dans un registre. Proposez un codage de l'instruction et modifiez le graphe d'états.

Question 7 On s'intéresse maintenant aux instructions de saut :
- `j @mem` fait sauter l'exécution à l'adresse `@mem` ;
- `jz ri,@mem` fait sauter l'exécution à l'adresse `@mem` si le registre `ri` est égal à 0 ;
- `jnz ri,@mem` fait sauter l'exécution à l'adresse `@mem` si le registre `ri` n'est pas égal à 0 ;
- `jlt ri,@mem` fait sauter l'exécution à l'adresse `@mem` si le registre `ri` est inférieur à 0 ;
- `jgt ri,@mem` fait sauter l'exécution à l'adresse `@mem` si le registre `ri` est supérieur à 0.

Proposez un codage de ces instructions. Faut-il modifier la partie opérative ? Modifiez le graphe d'états.

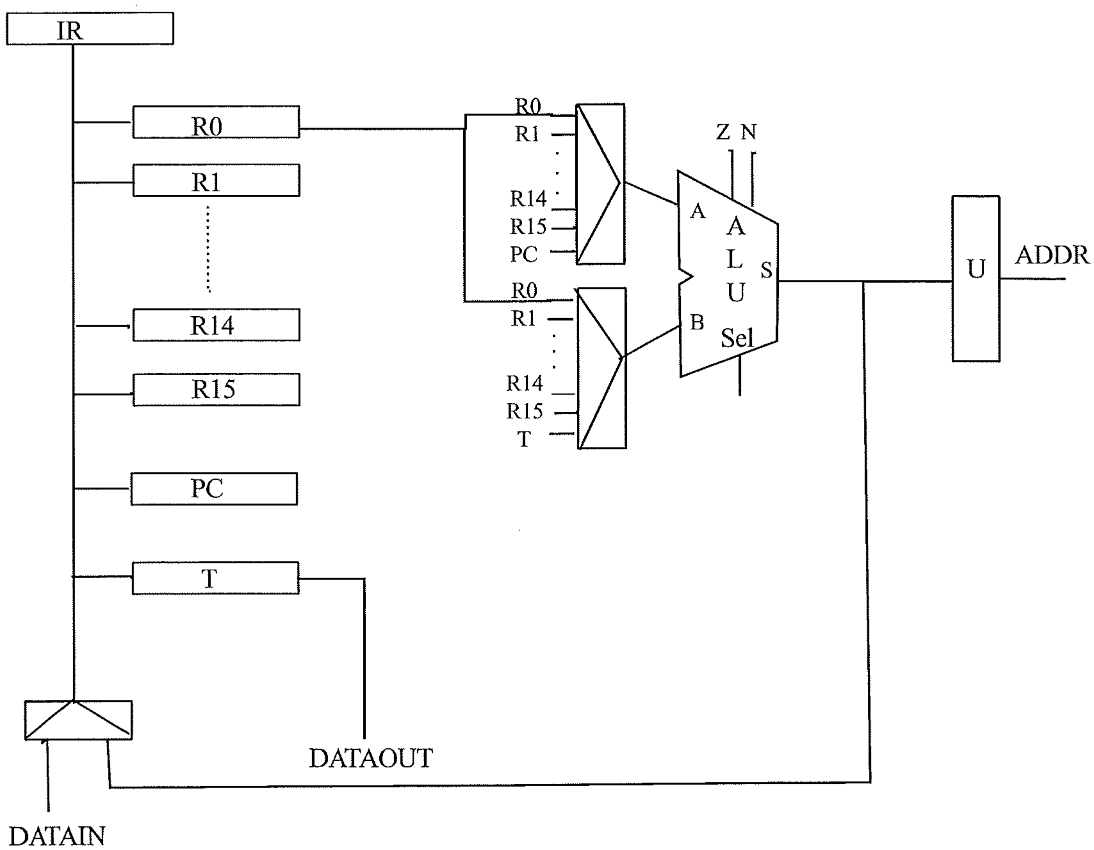
Question 8 On ajoute un mode d'adressage relatif au PC pour les instructions de saut :
- `b val(PC)` fait sauter l'exécution à l'adresse `Valeur courante de PC + val` ;
- `bz ri,val(PC)` fait sauter l'exécution si le registre `ri` est égal à 0 ;
- `bnz ri,val(PC)` fait sauter l'exécution si le registre `ri` n'est pas égal à 0 ;
- `blt ri,val(PC)` fait sauter l'exécution si le registre `ri` est inférieur à 0 ;
- `bgt ri,val(PC)` fait sauter l'exécution si le registre `ri` est supérieur à 0.

Proposez un codage de ces instructions. Faut-il modifier la partie opérative ? Modifiez le graphe d'états.

---

1. Fonction exprimant si le codop de l'instruction stockée dans IR est un accès à la mémoire.
2. Fonction exprimant si le codop de l'instruction stockée dans IR est un accès à la mémoire en écriture.
3. Fonction prenant les 16 bits de IR en entrée et retournant les 4 bits déterminant le numéro de registre Rs de l'instruction décodée.
4. Fonction retournant les 3 bits poids faible de l'entrée Sel de l'ALU correspondants à l'opération que doit réaliser une instruction de type `op rs, rd`

NOM :
PRENOM :
GROUPE :



ENSIMAG 1ère année 2013/2014

ECONOMIE

DS du mercredi 22 janvier 2014

Durée : 2 heures
Documents autorisés

Traiter **l'un des deux** sujets proposés.
Quel que soit votre choix, votre rédaction ne devra pas dépasser **4 pages**)

Sujet 1 : Commentaire libre de l'information suivante :

« **La BCE maintient son principal taux directeur.** Sans surprise, la Banque centrale européenne (BCE) a maintenu son principal taux d'intérêt directeur inchangé jeudi 9 janvier. Ce dernier reste donc à un niveau historiquement bas de 0,25 %, a annoncé l'institution lors de sa réunion mensuelle de politique monétaire. »

Source **:** Le Monde, 9 janvier 2014.

NB : Veillez à structurer votre travail (introduction, développements, conclusion), à soigner la rédaction et à mobiliser au maximum vos connaissances personnelles.

Sujet 2 : Commentaire du texte joint relatif

Vous avez le **choix** entre rédiger un commentaire libre du texte **ou** répondre aux questions posées (précisez votre choix sur la copie).

Questions :

1) Que répondrait encore aujourd'hui un économiste « standard » à la thèse centrale du texte pour la nuancer voire la réfuter et soutenir que la croissance peut encore se poursuivre très longtemps si ce n'est indéfiniment ?

2) A quels problèmes majeurs la société sera-t-elle confrontée dans le cas où effectivement le taux de croissance futur devait être proche de zéro ?

NB : Si vous optez pour le commentaire libre, veillez à structurer votre travail (introduction, développements, conclusion), à soigner la rédaction et à mobiliser au maximum vos connaissances personnelles.
Si vous choisissez la formule « Réponses aux questions », répondez précisément aux questions posées et mobilisez au maximum vos connaissances personnelles.

Texte

Les limites de la croissance économique: la revanche de Meadows

Bernard Maris (Professeur d'Economie à Paris VIII)
Chronique France-Inter, Jeudi 22 Janvier 2009

Lors de la parution de son rapport sur les limites de la croissance, en 1972, Meadows était la risée de ses confrères économistes. Aujourd'hui, il est considéré comme un visionnaire et vient même de recevoir un prix. La crise a fait au moins un heureux.
Le professeur Meadows vient de remporter le « Japan Prize », l'un des prix scientifiques les plus prestigieux, pour un ouvrage qu'il a dirigé en 1972. Soit... 37 ans plus tard. Le rapport du professeur Meadows s'appelait « Les limites à la croissance » et il a été traduit en France sous le titre « Halte à la croissance ! » En effet le professeur Meadows, et les autres, proposaient, tout simplement, une croissance zéro.

Meadows n'est pas si vieux que ça, il a 66 ans. Il a été prof au prestigieux MIT, Massachussets Institute of Technologie. Moi qui ai fréquenté les milieux économiques, je n'ai jamais vu autant de ricanements, je veux dire autour de Meadows et de son rapport. Meadows était un gauchiste au pire, un idéaliste au mieux, un ennemi du progrès, un amateur des cavernes et de la viande boucanée, un ennemi du Sud et des pauvres qui ne demandaient qu'à se développer ou à s'enrichir etc. etc.

Son rapport reposait sur une hypothèse infiniment simple. Les ressources sont limitées : le pétrole, le charbon, l'eau, l'uranium, les forêts, n'existent pas à profusion. Or la population humaine semble devoir croître indéfiniment. Et les besoins de la population humaine croissent encore plus vite que la population elle-même. Rien de commun entre les besoins de l'Américain moyen et les besoins du Bushiman ou du Pygmée lambda. Donc l'humanité va se heurter de façon dramatique au mur de la rareté. Cqfd. Stop à la croissance, vive la croissance zéro, arrêtons tout.

Ce qui faisait ricaner les économistes. Ils s'esclaffaient, même. Ah ! Ah ! Tout ce que dit Meadows, Malthus le disait déjà, Malthus le pasteur de l'apocalypse économique[1]. Il disait ça en 1800, dans son « Essai sur le principe de population » et nous sommes en 1972, Ah ! Ah ! Car Malthus, pas plus que ce pauvre Meadows, n'avait prévu une chose : la hausse des rendements ! L'incroyable progrès de la productivité qui fait que le quintal de blé produit par cent paysans en 1800 est produit par un seul paysan aujourd'hui. Et les économistes de rire.
En réalité, il n'y a pas de quoi rire. Malthus et Meadows ont simplement eu raison trop tôt. Le mur de la rareté approche, et nous allons nous heurter contre lui. Nous avons eu quelques prémisses de la crise des matières premières, la crise de l'eau est proche, les déchets commencent à envahir le monde, et la technique, hélas, n'a pas que des effets positifs. La technique peut être maléfique. A suivre.

La devinette du jour. Un nénuphar double de surface tous les jours et occupe la surface du lac au bout de cent jours. A quel jour était-il à la moitié ? (Réponse : la veille).

[1] NB : Au début du XIXème T. Malthus expliquait déjà que l'enrichissement perpétuel (on dirait aujourd'hui la « croissance perpétuelle ») n'est pas possible au niveau collectif car il observait que le taux de croissance de la population avait systématiquement tendance à être supérieur au taux de croissance de la production de biens disponibles à la consommation. Les sociétés buteront donc toujours sur une contrainte « naturelle » limitant leur possibilité d'enrichissement : l'impossibilité de tirer toujours plus de la Nature et de ses ressources en raison de leur rareté.

Grenoble INP – ENSIMAG
Année 2013-2014

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 20/01/2014. Durée 2h30
Tous documents (sauf livres) et calculettes autorisés

Les exercices sont indépendants. Vous pouvez les traiter dans l'ordre que vous souhaitez. Le barème est indicatif.

Exercice 1 : Communications numériques (7 points)

Remarque : Pour cet exercice, les réponses doivent être courtes, mais bien justifiées.

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

Question 1. *Donnez la forme du diagramme de constellation des deux modulations utilisées dans le système radio proposé. Sur une droite de l'échelle des fréquences, représentez la partie de spectre qu'occupe le signal émis par le système radio.*

Question 2. *Calculez le débit symboles pour réaliser la transmission numérique à 100 Mbit/s, avec les deux modulations QPSK et QAM16.*

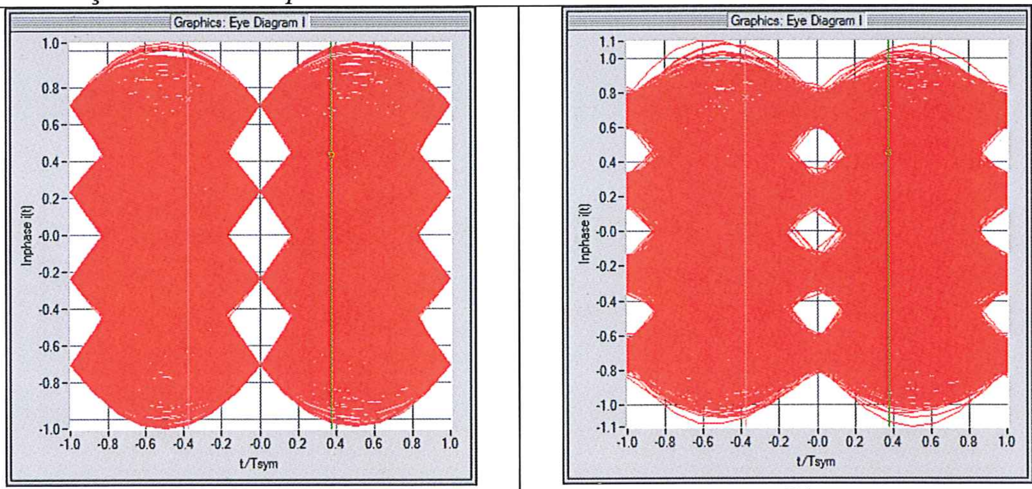
Question 3. *Pour un débit de 100 Mbit/s, les spécifications techniques donnent la valeur du seuil minimal de puissance reçue seulement pour la modulation QAM16. Cela signifie-t-il qu'on ne peut pas utiliser la modulation QPSK à ce débit ?*

Question 4. *Dans le cas présent d'une transmission sur 500 m à 100 Mbit/s, avec une modulation QAM16, quelle est la puissance reçue ? Peut-on alors avoir une idée du taux d'erreurs binaires obtenu en réception ? Si oui, donnez sa valeur.*

Question 5. *Si la distance entre les bâtiments n'était que de 300 m, quel taux d'erreurs binaires obtiendrait-on pour une transmission à 100 Mbit/s ?*

Des mesures préliminaires sont réalisées à un débit de 50 Mbit/s. La figure suivante montre les diagrammes de l'œil mesurés en mettant le récepteur juste en sortie du récepteur, puis dans le bâtiment situé à 500 m.

Question 6. *Quel est le type de modulation qui a été utilisé dans ces mesures ? D'où provient la différence essentielle entre les deux diagrammes de l'œil ? D'après le premier diagramme de l'œil, peut-on dire si le système de transmission a été conçu de manière optimale ?*



(a) (b)

Figure 1 : Diagrammes de l'œil mesurés à la sortie de l'émetteur (a) et dans le bâtiment situé à 500 m (b) (échelles verticales normalisées)

Question 7. *Pour une transmission à 50 Mbit/s, les spécifications techniques donnent des niveaux de seuils minimaux de puissance reçue plus faibles que pour la transmission à 100 Mbit/s : de 7 dB plus bas si on utilise la modulation QPSK et 3 dB plus bas si on utilise la modulation QAM16. Ces valeurs précises de diminutions sont-elles conformes à la théorie ?*

Exercice 2 : Étude d'un protocole de déconnexion (7 points)

On s'intéresse à un protocole de transport en mode connecté. Le service rendu par ce protocole est d'assurer le transfert de données fiable sur une connexion bidirectionnelle simultanée. On ne s'intéresse ici ni à la phase de connexion (qui amène dans l'état connecté), ni à la phase de transfert de données, mais uniquement à la phase de déconnexion.

1. Chaque entité, dans l'état "connecté" peut demander une déconnexion en envoyant un PDU de requête de déconnexion DR. Elle passe alors dans l'état "attente de déconnexion".
2. Une entité dans l'état "connecté" recevant un PDU de déconnexion DR envoie un PDU de confirmation de déconnexion DC et retourne à l'état "repos".

3. Une entité dans l'état "attente de déconnexion" recevant un PDU de confirmation de déconnexion DC retourne à l'état de repos.

Dans les états autres que "connecté" une entité ne peut émettre ni recevoir des données. On suppose que les paquets sont transmis par la couche sous-jacente sans perte, sans duplication et sont délivrés dans l'ordre d'émission.

Question 8. *Décrire sous forme d'un diagramme temporel un scénario simple de la phase de déconnexion du protocole de transport, et écrivez l'automate du protocole.*

Question 9. *Peut-il se produire une collision de déconnexion ? Justifier la réponse en s'aidant éventuellement d'exemple et en décrivant un scénario par un diagramme temporel.*

Question 10. *Dans le cas où vous pensez qu'une collision de déconnexion est possible le protocole décrit ci-dessus peut-il conduire à un interblocage ? Justifiez votre réponse. Si elle est positive, proposez une modification très simple du protocole de déconnexion décrit ci-dessus pour pallier ce dysfonctionnement.*

Question 11. *. Le protocole de déconnexion assure-t-il qu'aucun PDU de données ne peut être perdu? Justifiez votre réponse en vous aidant éventuellement d'un diagramme temporel.*

Question 12. *Si l'ordre n'est pas préservé, est-ce que cela change votre réponse à la question précédente ?*

Question 13. *Si le réseau sous-jacent peut perdre des messages, tout en préservant l'ordre, comment proposez-vous de modifier le protocole ?*

Exercice 3 : Transmission sécurisée et authentifiée (4 points)

Alice veut envoyer un fichier à Bob de manière sécurisée et authentifiée.

Les données vont être envoyées sous forme de paquets sur une connexion ayant un débit D_{cnx}. Chaque paquet contient un entête de E bits, une signature électronique de F bits, et G bits de données chiffrées.

Les données sont chiffrées avec une fonction de chiffrement symétrique qui possède un débit D_{sym} (on suppose qu'une clef secrète est déjà partagée par Alice et Bob).

Les données sont signées à l'aide de la fonction de chiffrement asymétrique (débit D_{asym} et de la fonction de hachage (débit D_{hash} . Cette signature est effectuée en deux étapes:

1. premièrement la fonction de hachage prend en entrée un ensemble de données de longueur quelconque, et calcule un résumé de F bits;
2. dans un second temps ce résumé est chiffré avec la clef privée d'Alice afin de produire la signature électronique.

Question 14. *Exprimez le temps de transfert d'un fichier de S bits en fonction du nombre de paquets N et des débits des différentes fonctions.*

Question 15. *En déduire la valeur de N qui minimise le temps de transfert.*

La connexion utilisée n'est pas fiable et une partie des paquets sont perdus avec une probabilité qui dépend de leur longueur L. Grâce à un système d'acquittement et de retransmission les paquets finissent tous par être reçus correctement, mais le débit effectif de la connexion s'en trouve dégradé et est égal à $D'_{cnx} = D_{cnx} /L$.

Question 16. *Donnez la nouvelle expression du temps de transfert. En déduire la valeur de N qui minimise le temps de transfert.*

Exercice 4 : Protocoles d'accès multiple (2 points)

Un réseau Ethernet à 10 Mbit/s utilisant le protocole d'accès multiple CSMA/CD est composé de 3 stations A, B et C. A et C sont placées aux deux extrémités du réseau (qui a la longueur maximale possible) et B est au milieu de A et C.

Question 17. *A l'instant t_o, la station A émet vers B et à l'instant $t_o+t_p/3$, la station C veut émettre vers B. Pourquoi la station C va-t-elle commencer à émettre ? A quel instant la collision se produit-elle ?*

Question 18. *A quel instant la collision est-elle détectée par C et A ? Que font alors C et A ?*

Question 19. *Pourquoi le protocole d'accès CSMA/CD n'est-il pas utilisé dans le cas d'un réseau sans fil de type Wi-Fi ? Donnez le protocole d'accès utilisé en Wi-Fi et expliquez rapidement ses principes de fonctionnement.*

Question 20. *Pourquoi le protocole d'accès CSMA/CD n'est-il pas utilisé dans le cas des réseaux mobiles cellulaires ? Citez les protocoles d'accès utilisés dans ces réseaux et expliquez rapidement leurs principes de fonctionnement.*

Ensimag 1re année **2013-2014**

Probabilités Appliquées
Janvier 2014

Durée 2h00 - Une feuille de format A4 manuscrite autorisée. Pas de calculatrices. Veuillez inscrire votre numéro de groupe sur la copie.

Exercice I (6 pts) – Soit U une variable aléatoire de loi uniforme sur $(0,1)$. On considère une variable aléatoire X de loi de densité

$$\forall x \in \mathbb{R}, \qquad f(x) = \frac{1/3 + x^{5/2}}{\sqrt{x}} \mathbb{1}_{(0,1)}(x)$$

1) Calculer la fonction de répartition, $F_1(t)$ de la loi de U^2. Calculer la fonction de répartition, $F_2(t)$, de la loi de $U^{1/3}$.

2) Calculer la fonction de répartition, $F(t)$, de la loi de X.

3) Montrer qu'il existe p, $0 < p < 1$, tel que

$$\forall t \in \mathbb{R}, \qquad F(t) = pF_1(t) + (1-p)F_2(t)$$

4) Soit p la valeur trouvée précédemment. On considère la variable Y définie par:

$$Y = VU^2 + (1-V)U^{1/3}$$

où V est une variable aléatoire de Bernoulli de paramètre p, indépendante de U. Calculer l'espérance et la variance de la variable Y.

5) Montrer que la variable X admet la même loi que Y et en déduire $\mathrm{E}[X]$ et $\mathrm{Var}(X)$.

6) Déduire des questions précédentes un algorithme de simulation de la loi de X utilisant un générateur aléatoire de loi uniforme sur $(0,1)$.

Exercice II (7 pts) – Soit U et V deux variables aléatoires indépendantes de loi uniforme sur $(0,1)$. L'objectif de cet exercice est de montrer que la loi du minimum $\min(U,V)$ est identique à la loi de $U\sqrt{V}$. Dans un premier temps, on cherche la loi de la variable $Y = U\sqrt{V}$.

1) On pose $X = U$. Soit $x \in (0,1)$. Montrer que

$$\forall t \in (0,1), \quad \mathrm{P}(Y \leq t | X = x) = \begin{cases} t^2/x^2 & \text{si } t \leq x \\ 1 & \text{si } t \geq x \end{cases}$$

2) Déduire de la question précédente que la densité du couple (X, Y) est donnée par

$$\forall x \in \mathbb{R}^2, \quad f(x,y) = \frac{2y}{x^2}\mathbb{1}_\Delta(x,y)\,,$$

où $\Delta = \{(x,y) \in \mathbb{R}^2,\, 0 < y < x < 1\}$.

3) Retrouver la formule de la question 2) à l'aide d'un changement de variables approprié.

4) Déterminer la densité de la loi de la variable Y.

5) Calculer l'espérance de Y, ainsi que la covariance $\mathrm{Cov}(X, Y)$.

6) On souhaite retrouver le résultat de la question 4) en utilisant une méthode de conditionnement. Montrer en utilisant une telle méthode que nous avons

$$\forall t \in (0,1), \quad \mathrm{P}(Y \leq t) = t + \int_t^1 \frac{t^2}{x^2}dx\,.$$

En déduire la fonction de répartition de la loi de Y. Retrouver la densité de cette loi.

7) Montrer que la loi de Y est identique à celle de la variable $\min(U, V)$.

Exercice III (7 pts) – Soit $n \geq 2$ et $(U_i)_{i=1,\dots,n}$, une suite de n variables indépendantes de loi uniforme sur $(0,1)$. Pour tout $i = 1, \dots, n-1$, on pose

$$X_i = \begin{cases} 1 & \text{si } U_i < U_{i+1} \\ 0 & \text{sinon.} \end{cases} \qquad \text{et} \quad X = \sum_{i=1}^{n-1} X_i\,.$$

1) Démontrer que $\mathrm{P}(U_1 < U_2) = 1/2$ et que $\mathrm{P}(U_1 < U_2 < U_3) = 1/6$.

2) Calculer $\mathrm{E}[X]$.

3) Calculer $\mathrm{Var}(X_1)$ et $\mathrm{Cov}(X_1, X_2)$. En déduire la valeur de $\mathrm{Var}(X)$ lorsque $n = 3$.

4) Montrer que $\mathrm{Var}(X) = \sum_{i=1}^{n-1} \mathrm{Var}(X_i) + 2\sum_{i=1}^{n-2} \mathrm{Cov}(X_i, X_{i+1})$

5) Calculer $\mathrm{Var}(X)$ pour tout $n \geq 2$.

6) On note $A_{n-1} = \frac{X}{n-1}$ le nombre moyen d'accroissements de la suite. Calculer l'espérance et la variance de A_{n-1}.

7) Soit $\epsilon > 0$. À l'aide de l'inégalité de Chebyshev, montrer que

$$\lim_{n\to\infty} \mathrm{P}\left(|A_{n-1} - \frac{1}{2}| > \epsilon\right) = 0.$$

Grenoble INP - Ensimag 1ère année 21 Janvier 2014

Théorie des langages 1

Durée : 3h.
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 trois exercices. L'automate construit dans l'exercice 1 sera réutilisé dans l'exercice 2.

Exercice 1 - Automates minimaux (4 points)

Soit L le langage sur $\{0, 1\}$ constitué des mots tels que tout 0 est suivi d'un 1, et soit M le langage représenté par l'expression régulière $10^*1 + \varepsilon$. Construire un automate déterministe minimal reconnaissant le langage $L \cup M$.

Exercice 2 - Transformations d'automates (10 points)

Soient V et W des vocabulaires, et $h : V \to W^*$ une fonction. Cette fonction est étendue à une fonction sur V^* par induction en posant[1] :

- $h(\varepsilon) = \varepsilon$,
- pour tout $a \in V$ et pour tout $w \in V^*$, $h(a.w) = h(a).h(w)$.

Pour $w \in W^*$, on note $h^{-1}(w)$ l'image réciproque de w. Formellement :

$$h^{-1}(w) = \{v \in V^* \mid h(v) = w\}.$$

Images réciproques

On pose $V = \{a, b, c\}$, $W = \{0, 1\}$ et on considère la fonction h_1 définie par :

$$h_1 : \begin{array}{ccc} a & \to & 01 \\ b & \to & 10 \\ c & \to & 0110 \end{array}$$

Ainsi, $h_1(bac) = 10010110$, $h_1(cca) = 0110011001$ et $h_1^{-1}(0110) = \{c, ab\}$.

▷ **Question 1** *(2 points) Calculer les images réciproques par h_1 des mots suivants :*

1. $w_1 = \varepsilon$
2. $w_2 = 0101$
3. $w_3 = 01011010$
4. $w_4 = 0101101$

Les images réciproques sont étendues aux langages de la façon suivante :

$$\forall L \subseteq W^*,\ h^{-1}(L) = \bigcup_{w \in L} h^{-1}(w).$$

[1] On rappelle que l'opérateur de concaténation peut être omis : on peut aussi bien écrire $w.w'$ que ww'.

▷ **Question 2** *(2 points) Calculer les images réciproques par h_1 des langages suivants :*

1. $L_1 = 0^*1^* + 1^*0^*$
2. $L_2 = (0110)^*$
3. $L_3 = \{(01)^n(10)^n \mid n \geq 1\}$
4. $L_4 = (0 + \varepsilon)(10)^*(1 + \varepsilon)$

Soit $A = (Q, W, \delta, \{q_0\}, F)$ un automate fini **déterministe** et considérons une fonction quelconque $h : V \to W^*$. On définit l'automate $h^{-1}(A) \stackrel{\text{def}}{=} (Q, V, \mu, \{q_0\}, F)$, où la fonction de transition μ est définie pour tout $q \in Q$ et pour tout $a \in V$ par[2] $\mu(q, a) = \delta^*(q, h(a))$.

▷ **Question 3** *(1,5 points) Soit $W = \{0, 1\}$, on définit la fonction h_2 par :*

$$h_2 : \begin{array}{ccc} a & \to & 00 \\ b & \to & 001 \\ c & \to & 100. \end{array}$$

On note B l'automate minimal construit dans l'Exercice 1. Construire l'automate $h_2^{-1}(B)$. Quel langage est reconnu par cet automate ?

On prend maintenant un automate déterministe A et une fonction h **quelconques**, on souhaite prouver que si A reconnaît un langage L, alors $h^{-1}(A)$ reconnaît le langage $h^{-1}(L)$.

▷ **Question 4** *(1,5 points) Montrer par induction sur w que pour tout $w \in V^*$ et pour tout $q \in Q$, $\mu^*(q, w) = \delta^*(q, h(w))$.*
Remarque : On pourra admettre que $h(w.w') = h(w).h(w')$ et que pour tout $q \in Q$, $\delta^*(\delta^*(q, w), w') = \delta^*(q, ww')$.

▷ **Question 5** *(1 point) En déduire que $h^{-1}(A)$ reconnaît $w \in V^*$ si et seulement si A reconaît $h(w)$. Si L est régulier, que peut-on dire du langage $h^{-1}(L)$?*

▷ **Question 6** *(2 points) En admettant que $\{0^n1^n \mid n \geq 0\}$ n'est pas régulier et* ***sans se servir du lemme de l'étoile****, montrer que $\{a^nb^{2n} \mid n \geq 0\}$ n'est pas régulier.*

Exercice 3 - Modélisation de langages (6 points)

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

$$\begin{array}{ll} (1) & S \to S\ I \\ (2) & S \to I \\ (3) & I \to \mathbf{i}\,; \\ (4) & I \to \mathbf{if\ c\ then}\ S\ \mathbf{end}\,; \\ (5) & I \to \mathbf{if\ c\ then}\ S\ \mathbf{else}\ S\ \mathbf{end}\,; \end{array}$$

avec $V_N = \{S, I\}$, S étant l'axiome, et $V_T = \{;, \mathbf{i}, \mathbf{c}, \mathbf{if}, \mathbf{then}, \mathbf{else}, \mathbf{end}\}$. Le non-terminal S dénote une suite d'instructions et I une instruction. Le terminal **i** représente une instruction quelconque autre que la conditionnelle, et le terminal **c** représente une condition quelconque.

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

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

▷ **Question 2** *(1 point)*
Modifier la grammaire G afin de décrire les instructions conditionnelles générales du langage Ada, c'est-à-dire contenant une suite éventuellement vide de elsif. Pour cela on étend le vocabulaire terminal par l'élément **elsif***. On pourra introduire de nouveaux non-terminaux, à condition de les définir.*

[2] On rappelle que $\delta^*(q, \varepsilon) = q$ et $\delta^*(q, a.w) = \delta^*(\delta(q, a), w)$.

▷ **Question 3** *(2 points)*

Dans certains langages de programmation il n'y pas de marqueur de fin d'instruction conditionnelle. On considère la grammaire G' obtenue à partir de G en remplaçant les règles (4) et (5) par les règles (4') et (5') ci-dessous et en ajoutant la règle (6) pour autoriser les suites d'instructions. On étend le vocabulaire terminal avec le symbole **begin**.

$$
\begin{array}{ll}
(4') & I \rightarrow \textbf{if c then } I \\
(5') & I \rightarrow \textbf{if c then } I \textbf{ else } I \\
(6) & I \rightarrow \textbf{begin } S \textbf{ end}
\end{array}
$$

Montrer que la grammaire G' est ambiguë. Quel problème peut poser l'ambiguïté de cette grammaire ?

▷ **Question 4** *(2 points)*

On considère maintenant le sous-langage suivant :

$$
\begin{array}{ll}
(1) & I \rightarrow \textbf{i ;} \\
(2) & I \rightarrow \textbf{if c then } I \textbf{ end ;} \\
(3) & I \rightarrow \textbf{if c then } I \textbf{ else } I \textbf{ end ;}
\end{array}
$$

On désire montrer que tout mot w de L(I) vérifie la propriété suivante : $|w|_{then} \geq |w|_{else}$. *On rappelle que la notation* $|w|_x$ *désigne le nombre d'occurrence du symbole x dans w. Démontrer formellement ce résultat. On prendra soin d'expliquer précisément le principe de preuve utilisé.*

PARTIELS DU MOIS DE MAI 2014 – SESSION 1

ALGORITHMIQUE 2 -	**5 PAGES**
METHODES NUMERIQUES -	**3 PAGES**
PLITIQUES ECONOMIQUES -	**5 PAGES**
PRINCIPES et METHODES STATS. -	**3 PAGES**
RECHERCHE OPERATIONNELLE -	**2 PAGES**
THEORIE DE L'INFORMATION -	**2 PAGES**
THEORIE DE LANGAGES 2 -	**3 PAGES**

Algorithmique 2

Ensimag - 1A

Mai 2014

Durée : 3h
Machines électroniques interdites **documents autorisés : notes de cours**

Les différentes parties du sujet sont indépendantes.

Le barème est donné à titre indicatif.

---

1 Intersections d'ensembles de segments (13 points)

On se place pour cet exercice dans le plan. Chaque point P est ici défini par 2 coordonnées flottantes $(P_x, P_y) \in \mathbb{R}^2$. Un segment de droite s est défini comme un couple de points p_1, p_2. Nous considérons ici le problème consistant à calculer à partir d'un ensemble S de n segments différents l'ensemble V de tous les points d'intersection des segments de S.

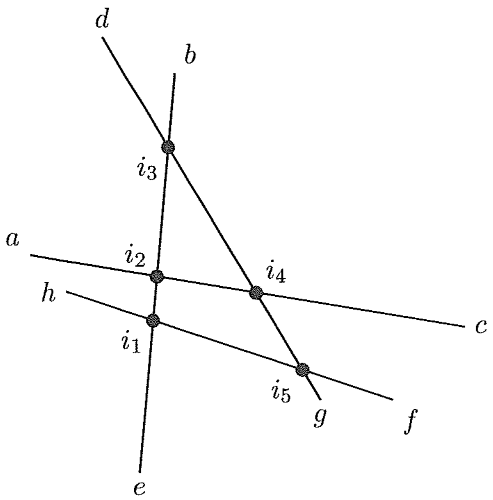


FIGURE 1 – Exemple de calcul d'intersections

La figure 1 illustre ainsi notre problème. Nous avons en entrée les segments (h, f), (e, b), (a, c), (d, g) sous forme d'un *vecteur* et en sortie, un *vecteur* contenant les points i_1, i_2, i_3, i_4, i_5.

Dans un souci de simplification, nous supposerons par la suite que les opérations usuelles sur les vecteurs sont réalisées en temps $O(1)$ en pire cas (au lieu de $O(1)$ en coût amorti).

Nous supposerons également qu'il n'est *pas possible* que plus de deux segments s'intersectent en un même point, que tous les points (d'entrée ou d'intersection) ont une abscisse différente, et enfin qu'aucune extrémité d'un segment ne se trouve sur un autre segment.

Enfin, nous disposons d'une fonction *intersection* renvoyant l'intersection de *deux* segments, si elle existe, en temps $O(1)$.

1.1 Algorithme naïf

1.1.1. (1 point) Donnez en fonction de n un ordre de grandeur asymptotique (Θ) du nombre de points d'intersection au pire cas. Justifiez.

1.1.2. (1 point) Écrire en pseudo-code un algorithme très simple de coût $O(n^2)$ de calcul du vecteur V des points d'intersection.

1.2 Algorithme de Bentley-Ottmann

On se propose d'améliorer l'algorithme précédent. Pour ce faire, nous introduisons une notation supplémentaire : k, le nombre de points d'intersection renvoyés.

L'algorithme de Bentley-Ottmann fonctionne en simulant un balayage du plan par une droite verticale Δ se déplaçant de gauche à droite. L'espace ainsi balayé est continu mais en fait seul un nombre fini d'abscisses nous intéresse : là où "il se passe quelque-chose d'intéressant". On ne considère donc que les "événements" intéressants : début d'un segment, fin d'un segment, intersection.

L'algorithme utilise deux structures de données différentes :

— une file de priorité F contenant les événements à venir (avec comme priorités l'opposé de leurs abscisses) ;

— un dictionnaire D contenant les segments actuellements coupés par la droite Δ (classés par ordonnée croissante d'intersection avec Δ pour son abscisse courante).

Nous avons une propriété importante : si deux segments s_1 et s_2 s'intersectent, alors il existe une abscisse pour Δ pour laquelle s_1 et s_2 sont côte à côte dans D. Nous en déduisons l'algorithme 1 (avec l'aimable participation de Wikipedia).

```
1  intialiser F, D et V (vecteur résultat) à des structures vides;
2  pour chaque segment s ∈ S faire
3  |  ajouter les deux événements "début de s" et "fin de s" à F;
4  fin
5  tant que F n'est pas vide faire
6  |  récupérer e, événement le plus prioritaire de F, et l'enlever de F;
7  |  si e n'a pas encore été vu alors
8  |  |  si e correspond au début d'un segment s alors
9  |  |  |  ajouter s à D;
10 |  |  |  soient r et t les segments immédiatement au-dessous et au-dessus de s dans D;
11 |  |  |  si r existe et s intersecte r alors
12 |  |  |  |  ajouter l'événement "intersection(s,r)" dans F;
13 |  |  |  fin
14 |  |  |  si t existe et s intersecte t alors
15 |  |  |  |  ajouter l'événement "intersection(s,t)" dans F;
16 |  |  |  fin
17 |  |  sinon si e correspond à la fin d'un segment s alors
18 |  |  |  soient r et t les segments immédiatement au-dessous et au-dessus de s dans D;
19 |  |  |  enlever s de D;
20 |  |  |  si r et t existent et r intersecte t alors
21 |  |  |  |  ajouter l'événement "intersection(r,t)" dans F;
22 |  |  |  fin
23 |  |  sinon
24 |  |  |  nous sommes sur une intersection, l'ajouter à V;
25 |  |  |  on récupère s1 et s2 les deux segments intersectés;
26 |  |  |  échanger les positions de s1 et s2 dans D;
27 |  |  |  trouver le segment r juste au dessous de s1 dans D;
28 |  |  |  trouver le segment t juste au dessus de s2 dans D;
29 |  |  |  si r existe et r intersecte s2 alors
30 |  |  |  |  ajouter l'événement "intersection(r,s2)" dans F;
31 |  |  |  fin
32 |  |  |  si t existe et t intersecte s1 alors
33 |  |  |  |  ajouter l'événement "intersection(t,s1)" dans F;
34 |  |  |  fin
35 |  |  fin
36 |  fin
37 fin
38 retourner V;
```

Algorithme 1 : Bentley-Ottmann

1.2.1. (0.5 points) Exécutez l'algorithme de Bentley-Ottmann sur l'entrée de la figure 1. Vous afficherez l'ensemble des événements rencontrés (dans l'ordre où ils sont rencontrés) ainsi que le changement du contenu de D lors de l'événement "intersection de (d, g) et (b, e)".

1.2.2. (1 point) Expliquez comment réaliser l'opération de la ligne 7 vérifiant si un évènement a déjà été traité en temps $O(1)$. Vous préciserez les structures de données utilisées.

1.2.3. (1 point) Donnez une borne supérieure sur le nombre d'éléments de F. (Justifiez)

1.2.4. (1 point) Proposez une structure de donnée efficace pour stocker F. Quel est le coût total de toutes les insertions et toutes les suppressions dans F ?

1.2.5. (1 point) On s'intéresse maintenant au choix de la structure utilisée pour le dictionnaire contenant les segments. On propose d'utiliser un arbre binaire de recherche. Nous supposerons ici que les choses se passent bien et que la hauteur de l'arbre est logarithmique. Nous supposons également disposer dans chaque nœud d'un pointeur vers le nœud père (en plus des pointeurs vers les fils). Chaque clef est un segment. Les arbres binaires de recherche fonctionnent à l'aide d'une fonction de comparaison. Écrivez en pseudo-Ada le code de la fonction de comparaison *(Indication: vous utiliserez une variable globale X contenant l'abscisse courante de la droite de balayage)*. L'objectif est qu'un parcours infixe en profondeur voit tous les segments de bas en haut pour l'abscisse X.

1.2.6. (2 points) Étant donné un pointeur sur un nœud de l'arbre (contenant un segment donc), comment retrouver le segment situé immédiatement au-dessous et celui situé immédiatement au-dessus ? Quel est le coût au pire cas d'une telle opération ?

1.2.7. (1 point) Écrivez en pseudo-Ada une procédure réalisant l'échange de deux segments dans l'arbre. Vous ferez très attention à bien expliquer le prototype (le profil, l'en-tête, la signature) de votre procédure et à en préciser les éventuelles pré-conditions.

1.2.8. (1 point) Quel est le coût total (au pire cas), en fonction de n et k de tous les accès à D ?

1.2.9. (1.5 points) Donnez le coût total de l'algorithme (au pire cas).

1.3 Application-extension

On dispose de deux ensembles de segments, R contenant une abstraction du réseau routier et S contenant une abstraction de l'ensemble des sentiers de randonnée sur le département de l'Isère. Un sentier ou une route est dans la réalité courbe, mais cette courbe est approchée par un ensemble de segments.

1.3.1. (1 point) On cherche à trouver les croisements entre les routes et les sentiers. Quel est le meilleur choix d'algorithme ? Justifiez-vous en analysant les caractéristiques des données en entrée.

2 Algorithmes de sélection

Dans ce second exercice, on cherche à résoudre le problème de la *sélection* du k$^{\text{ème}}$ élément. On dispose en entrée d'un tableau T de n éléments $T(1), T(2), \ldots, T(n)$. T n'est pas nécessairement trié. On dispose également d'un entier k tel que $1 \leq k \leq n$. On cherche à déterminer en sortie la valeur du k$^{\text{ème}}$ plus petit élément de T. Par exemple pour $k = 1$ on doit renvoyer la valeur du plus petit élément de T. Nous supposerons pour faire simple que tous les éléments de T sont différents.

Une solution naïve au problème consiste à trier T par ordre croissant puis à renvoyer $T(k)$. Malheureusement cette solution coûte $O(n \log(n))$ ce qui est un coût élevé pour un problème aussi simple.

2.1 Quickselect (7 points)

On propose un premier algorithme utilisant une procédure de *Segmentation* similaire à celle du *Quicksort*. On tire *au hasard* un indice pivot p du tableau puis la segmentation place en temps linéaire (et en n comparaisons) tous les éléments inférieurs à p devant p et tous les éléments supérieurs à p derrière p. La procédure retourne également la nouvelle place du pivot. Nous supposerons disposer d'une procédure de segmentation et son code n'est donc pas à écrire.

2.1.1. (1 point) Utilisez la procédure de segmentation pour écrire un algorithme récursif (en pseudo-Ada) résolvant *sélection*.

2.1.2. (1 point) L'algorithme étant probabiliste, on cherche à calculer l'espérance (sur les tirages réalisés) du coût. En considérant un tirage uniforme du pivot, proposez une formule récursive calculant $C(n)$, l'espérance du coût pour la sélection d'un élément dans un tableau de taille n. On ne comptera dans ce coût que les comparaisons d'éléments du tableau.

2.1.3. (2 points) Montrez par récurrence que $C(n) = r \times n$ où r est une constante. On s'autorisera à borner en remplaçant k par $n/2$ dans nos formules (ce qui correspond au pire cas).

2.2 Soft Heaps

On cherche maintenant à obtenir un algorithme *déterministe* de performance équivalente. Pour ce faire, nous gardons le même principe de récursion, mais modifions la manière de choisir le pivot.

Nous utilisons une variante des tas appellée *soft heap* et introduite par Chazelle en 2000. Il s'agit comme un tas d'une file de priorité. On insère donc des éléments avec une certaine priorité puis on dispose d'une opération permettant d'extraire l'élément le plus prioritaire du tas.

Le soft heap à la particularité de *corrompre* certaines priorités ce qui signifie qu'elles peuvent alors prendre une valeur *erronée*, inférieure à la valeur réelle. On dispose d'un paramètre ϵ, compris entre 0 et 1 que l'utilisateur choisit pour obtenir les garanties suivantes :

— si n éléments ont été insérés dans le tas, au plus $n \times \epsilon$ priorités ont été corrompues ;

— les insertions se font en temps $O(1)$;

— les suppressions se font en temps $O(\log(1/\epsilon))$.

Nous prenons ici $\epsilon = 1/3$.

2.2.1. (1.5 points) Proposez un algorithme de choix de pivot garantissant que le choix n'est jamais trop mauvais. *(Indication: essayez d'abord avec un tas normal qui ne corrompt pas les clefs pour utiliser comme pivot la médiane)*

2.2.2. (1.5 points) Montrez à l'aide du *master theorem* que le coût au pire cas de l'algorithme est $O(n)$.

Examen du 28 Mai 2014

Durée : 3h.

Les seuls documents autorisés sont les notes du cours et des travaux dirigés de méthodes numériques et du soutien. Tout matérel électronique est interdit.

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

I. Normes et Conditionnement

$\mathcal{M}_n(\mathbb{C})$ désigne l'espace vectoriel des matrices (n, n) à coefficients dans $\mathbb{C}$.
$\|.\|$ est une norme sur $\mathcal{M}_n(\mathbb{C})$, supposée de plus *sous-multiplicative*, c'est-à-dire:

$$\forall A, B \in \mathcal{M}_n(\mathbb{C}), \qquad \|AB\| \leq \|A\| \cdot \|B\|.$$

Soit $A \in \mathcal{M}_n(\mathbb{C})$ une matrice inversible.
On perturbe la matrice A en $(A + \Delta A)$. On désire alors savoir si la matrice perturbée $(A + \Delta A)$ est encore inversible et si oui, si $(A + \Delta A)^{-1}$ diffère beaucoup de A^{-1}.

1. Montrer que si $S \in \mathcal{M}_n(\mathbb{C})$ vérifie $\|S\| < 1$, alors la matrice $(I + S)$ est inversible et on a

$$(I + S)^{-1} = \sum_{k=0}^{+\infty} (-1)^k S^k = I - S + S^2 \dots$$

2. En déduire que si $\|\Delta A\| < \frac{1}{\|A^{-1}\|}$ alors $(A + \Delta A)$ est inversible.

3. Sous l'hypothèse de la question précédente $\left(\|\Delta A\| < \frac{1}{\|A^{-1}\|}\right)$ montrer que:

$$\frac{\|(A + \Delta A)^{-1} - A^{-1}\|}{\|(A + \Delta A)^{-1}\|} \leq \kappa(A) \cdot \frac{\|\Delta A\|}{\|A\|}$$

avec $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ le conditionnement de A.

4. Qu'en déduisez-vous quant au lien entre le conditionnement de A et le fait que $(A + \Delta A)^{-1}$ diffère peu ou beaucoup de A^{-1} ?

II. Méthode de Newton pour un système partiellement linéaire

Soit deux entiers k et n avec $0 < k < n$. On se donne A une matrice (k, k) inversible, un vecteur b de $\mathbb{R}^k$ et une application H de $\mathbb{R}^n$ dans $\mathbb{R}^{n-k}$ non-linéaire.

Tout vecteur x de $\mathbb{R}^n$ sera décomposé en:

$$x = \begin{pmatrix} u \\ v \end{pmatrix} \qquad \text{avec } u \in \mathbb{R}^k \text{ et } v \in \mathbb{R}^{n-k}.$$

On considère alors l'application F de $\mathbb{R}^n$ dans $\mathbb{R}^n$ définie par :

$$F(x) = F(u, v) = \begin{pmatrix} Au - b \\ H(u, v) \end{pmatrix}$$

et on cherche à résoudre le système partiellement linéaire $F(x) = 0$.

1. Montrer que la matrice jacobienne $F'(x)$ de F s'écrit :

$$F'(x) = \begin{pmatrix} B & 0 \\ H_1(x) & H_2(x) \end{pmatrix}.$$

On explicitera les matrice $B, H_1(x)$ et $H_2(x)$.

2. Expliciter l'itération de Newton dans $\mathbb{R}^n$ pour résoudre $F(x) = 0$:
pour $x^r = \begin{pmatrix} u^r \\ v^r \end{pmatrix}$ on explicitera le système linéaire à résoudre à chaque pas, ainsi que $x^{r+1} = \begin{pmatrix} u^{r+1} \\ v^{r+1} \end{pmatrix}$.

On envisage une autre façon de résoudre le système : dans un premier temps, on résout le système linéaire $Au = b$; on note u^* la solution. On réinjecte la solution dans le système et on obtient un nouveau système non-linéaire à résoudre :

$$G(v) = 0 \text{ où } G : \mathbb{R}^{n-k} \to \mathbb{R}^{n-k}.$$

3. Calculer la matrice jacobienne de G.

4. Expliciter l'itération de Newton dans $\mathbb{R}^{n-k}$ pour résoudre ce nouveau système, et montrer que l'on obtient le même algorithme que celui de la question 2.

III. Factorisation de Cholesky

1. Effectuer la factorisation de Cholesky LL^T de la matrice A suivante:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 1 & 10 \\ 3 & 1 & 35 & 5 \\ 4 & 10 & 5 & 45 \end{pmatrix}.$$

2. Montrer que l'algorithme de factorisation de Cholesky préserve la structure bande des matrices, i.e.:

$$\text{Si } a_{ij} = 0 \text{ pour } |i-j| \geq p \text{ alors } l_{ij} = 0 \text{ pour } |i-j| \geq p.$$

IV. Itérations linéaires

Soient L_1 et L_2 deux matrices (n,n) à éléments réels, et z un vecteur de $\mathbb{R}^n$.

On cherche à résoudre le système suivant: $\quad (\Sigma) \quad \begin{cases} x_1 = L_1(z - x_2) \\ x_2 = L_2(z - x_1) \end{cases}$

où x_1 et x_2 sont deux vecteurs inconnus de $\mathbb{R}^n$, et L_1 et L_2 deux matrices telles que le rayon spectral de L_1L_2 est égal au rayon spectral de L_2L_1 (i.e. $\rho(L_1L_2) = \rho(L_2L_1)$).

On considère les deux itérations suivantes:

$$(J) \quad \begin{cases} x_1^{(0)} \text{ et } x_2^{(0)} \text{ donnés dans } \mathbb{R}^n \\ x_1^{(r+1)} = L_1(z - x_2^{(r)}) \\ x_2^{(r+1)} = L_2(z - x_1^{(r)}) \end{cases} \qquad (G) \quad \begin{cases} y_1^{(0)} \text{ et } y_2^{(0)} \text{ donnés dans } \mathbb{R}^n \\ y_1^{(r+1)} = L_1(z - y_2^{(r)}) \\ y_2^{(r+1)} = L_2(z - y_1^{(r+1)}) \end{cases}$$

1. Mettre respectivement (J) et (G) sous forme d'itérations linéaires dans $\mathbb{R}^{2n}$

$$\begin{cases} u^{(r+1)} = T_J\, u^{(r)} + h_J & \text{pour } (J) \\ v^{(r+1)} = T_G\, v^{(r)} + h_G & \text{pour } (G) \end{cases}$$

On explicitera soigneusement les matrices $(2n, 2n)$ T_J et T_G, ainsi que les vecteurs h_J et h_G.

2. Montrer que (J) et (G) convergent ou divergent simultanément, selon la valeur de $\rho(L_1L_2)$.

3. Montrer que si (J) et (G) convergent, alors (G) converge deux fois plus vite que (J) vers l'unique solution du problème (Σ).

ENSIMAG 1ère année 2013/2014

POLITIQUE ECONOMIQUE

DS du jeudi 22 mai 2014

Durée : 2 heures
Documents autorisés

Il vous est demandé de **choisir**, sur la base du texte ci-joint, entre rédiger un commentaire libre ou répondre aux questions posées.

Dans les deux cas, veillez à soigner la rédaction et à mobiliser au mieux vos connaissances personnelles.

Questions :

1) Synthèse du texte :

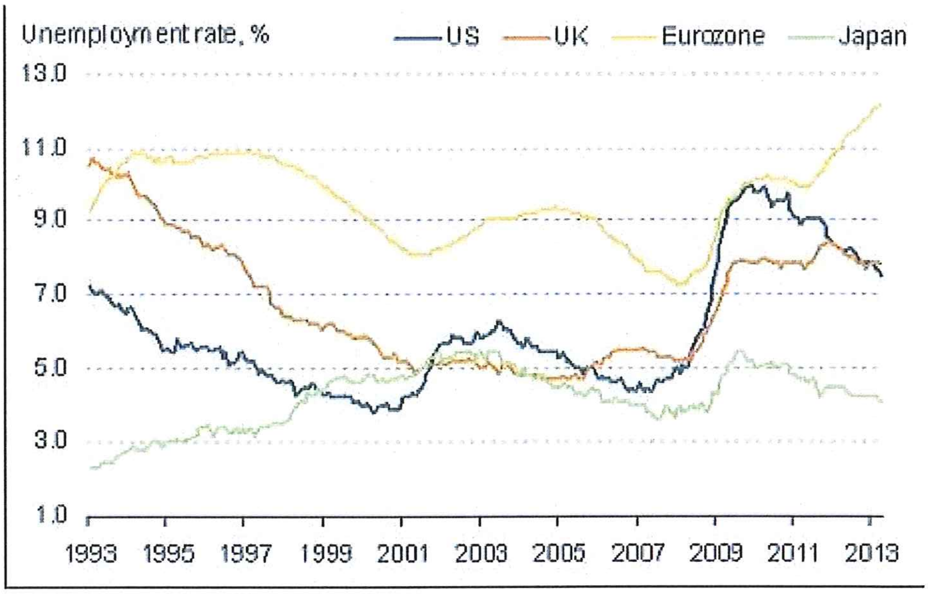
Quelle est la question posée par A. Delaigue ?
Quelle réponse lui donne-t-il ?
Résumez en une phrase courte l'essentiel de son argumentation.

2) Au-delà de ce qu'écrit A. Delaigue quand il évoque « *le processus politique qui a présidé à la construction européenne* », rappelez l'essentiel de ce qu'il faut savoir concernant la manière avec laquelle l'Europe s'est construite. Vous insisterez particulièrement sur l'articulation entre l'unification économique d'une part et l'unification politique d'autre part.

3) Si la priorité est de « sauver l'euro », quelles solutions peut-on envisager ? (Veillez ici à justifier vos propositions).

L'euro est un formidable succès

Alexandre Delaigue, 24 avril 2014
http://blog.francetvinfo.fr



Il est difficile d'échapper au contraste entre le sentiment de triomphalisme qui prévaut au sommet de la zone euro, et les réalités sur le terrain: croissance nulle, austérité budgétaire sans fin, chômage de masse. D'un côté, un discours de victoire, claironné au rythme du retour des pays périphériques sur les marchés financiers et de la baisse des taux d'intérêt; de l'autre, la promesse d'un score historiquement élevé pour les partis extrémistes s'appuyant sur un euroscepticisme plus fort que jamais. Entre fédéralisme et euroscepticisme, les partis traditionnels ne savent plus trop ou ils en sont.

Les extrêmes ont un discours clair, à défaut d'être réaliste : il faut selon eux sortir de la zone euro, défaire la construction européenne telle qu'elle est. Ce propos dépasse d'ailleurs largement les extrêmes en France, le nombre de coming-out eurosceptiques y augmente sans cesse.

Une union politique de l'euro?

Des eurosceptiques en force; des partis traditionnels en plein désarroi; dans ce contexte, quelques-uns tentent de trouver une sortie qui maintienne la construction européenne et résolve ses contradictions actuelles. C'est le cas de deux appels pour constituer respectivement une union politique de l'euro, ou une communauté politique de l'euro. Les formes sont différentes, mais l'idée est à chaque fois la même : renforcer la construction institutionnelle en

Europe, faire de la zone euro un ensemble politique spécifique, destiné à aller plus loin dans l'unification.
Un budget, une assemblée, des dépenses spécifiques, sont les points communs des deux appels. L'appel pour une union politique de l'euro préconise aussi une mutualisation d'une partie des dettes publiques nationales et un impôt commun sur les sociétés pour mettre fin au dumping fiscal entre les pays de l'Union.

Néanmoins, la lecture de ces appels laisse un sentiment étrange. Quel que soit le mérite que l'on accorde à toutes ces propositions, elles ont un point commun : rien de ce qu'elles ne proposent aurait, s'il avait été en place, évité la crise actuelle de la zone euro; rien de ce qui est proposé ne serait susceptible d'avoir le moindre effet sur les problèmes concrets, actuels, de la zone euro, qui la décrédibilisent : faible croissance et chômage. Il est assez paradoxal de répéter sans cesse qu'il est possible de réparer la locomotive en ne proposant que de climatiser les wagons. C'est peut-être une bonne idée, mais cela ne résout rien.

La seule logique que l'on puisse donner à ces appels est la suivante : la zone euro souffre d'un déficit démocratique, qui la rend illégitime aux yeux des citoyens et la conduit à des politiques inefficaces. Il est nécessaire de la rendre plus démocratique afin de pouvoir mettre en œuvre des politiques plus efficaces. On peut donc résumer leurs idées en deux postulats :

- Le problème de l'Europe vient de ce que ses institutions ne sont pas assez démocratiques, ce qui rend l'Europe illégitime et peu efficace;
- La crise de l'Europe est de nature technique, mécanique (réparer la locomotive); Une fois celle-ci rendue plus démocratique, il sera possible de mettre en œuvre les politiques plus susceptibles de résoudre la crise que le cocktail d'austérité budgétaire et de vœux pieux actuel.

Le problème de cette approche, c'est que ces postulats vont à l'encontre de tout le processus politique qui a présidé à la construction européenne.

La réaction en chaîne de Monnet

Ce processus a été décrit par Enrico Spolaore dans un récent article du Journal of Economic Perspectives. Il y décrit le problème de la construction européenne sous l'angle de la théorie économique de la taille des Nations. Dans cette théorie, la taille des Nations résulte d'un compromis entre deux forces contraires : d'un côté les rendements croissants qui confèrent un avantage à la grande taille, de l'autre, les préférences hétérogènes des citoyens, qui font que plus la nation grandit, plus le risque de devoir subir des politiques qui nous déplaisent augmentent.

Le projet européen, selon la célèbre méthode Monnet, a consisté à contourner les préférences des citoyens, qui n'ont pas la moindre envie d'être gouvernés par un autre pays, en transférant progressivement des compétences des gouvernements nationaux vers l'Europe, et en espérant que cela entraînerait une réaction en chaîne, consistant à transférer de plus en plus de compétences des états vers l'Europe. Ce processus devait se faire par l'exemple : au fur et à mesure qu'ils constatent que transférer des compétences à l'Europe leur apporte des résultats positifs, les citoyens deviennent moins méfiants envers celle-ci, et de plus en plus disposés à transférer d'autres compétences. C'est l'approche fonctionnaliste.

Mais l'approche fonctionnaliste a un côté obscur : lorsqu'on transfère certaines compétences à l'Europe tout en en laissant d'autres aux gouvernements nationaux, cela provoque rapidement des contradictions et des crises, qui ne laissent aux populations aucun autre choix que d'accepter, sous peine d'effondrement économique, d'autres transferts de compétences vers l'Europe. Comme ces transferts se font dans la douleur et dans l'urgence, ils conduisent à d'autres contradictions, qui ne peuvent à leur tour être résolues qu'avec de nouveaux transferts de compétence. La réaction en chaîne de Monnet devient un processus dans lequel les gouvernements n'ont pas d'autre choix que d'être dépossédés de leurs prérogatives au profit de l'Europe, dans un climat de crise permanente.

Cette approche conduit donc à inverser les deux postulats précédents. En réalité :

- Ce n'est pas parce qu'elle est peu démocratique que l'Europe est impopulaire; en réalité, c'est parce qu'il est impopulaire que le processus de construction européenne ne peut pas être démocratique.
- Les crises ne sont pas des défauts du système qui peuvent être corrigés techniquement; Elles sont au contraire consubstantielles au processus de construction européenne, qui ne saurait avancer sans cela. Sans les crises, jamais les gouvernements ne se résoudront à transférer des compétences vers l'Europe et le processus s'enlisera.

La crise de l'Euro, une chance pour l'Europe

Dès lors qu'on a compris ces deux postulats, la crise européenne devient très claire. Les défauts de conception de la zone euro au départ sont une conséquence logique du mécanisme de construction européenne; ils ne sont pas un bug, mais une contradiction nécessaire au progrès du processus d'unification.

Et que de chemin parcouru depuis le début de la crise de la zone euro! certains pays périphériques voient leurs finances publiques sous tutelle intégrale de l'Europe. La BCE tient en respect les gouvernements européens, éliminant les dirigeants qui ne lui conviennent pas en les soumettant à la pression des marchés. Tous les gouvernements ont cédé un pouvoir considérable, transférant à la commission un droit de regard sur les budgets nationaux qui pourrait rendre jaloux le parlement français. Dans le cadre de l'Union Bancaire, les gouvernements ont mis fin à la relation établie avec leurs banques nationales, conférant à la BCE le soin de les réguler et à l'Europe le pouvoir de les sauver ou de les couler.

Les critiques de l'union bancaire, qui lui reprochent son caractère inachevé et incomplet, sont à côté de la plaque. Ce caractère est parfaitement compréhensible dès lors qu'on saisit que dans le processus de construction européenne, l'essentiel n'est pas de construire des choses qui fonctionnent, mais de transférer autant de pouvoir que possible en dehors des gouvernements nationaux. La prochaine crise qui résultera de ce mécanisme incomplet sera l'occasion de nouveaux transferts de compétences, parce qu'il n'y aura pas de choix.

Aucun de ces transferts n'aurait été possible sans la crise de la zone euro, elle-même née des contradictions inévitables lorsqu'on confère une monnaie unique à un ensemble de pays disparates. Les critiques peuvent déplorer autant qu'ils le veulent ce processus : le niveau de souffrance subi par les pays périphériques de la zone euro montre que les populations nationales, vieillissantes, ne feront jamais le saut dans l'inconnu qu'impliquerait le démantèlement de la construction européenne et de l'euro. Ils préféreront le diable qu'ils connaissent à celui qu'ils ne connaissent pas.

La prochaine étape du processus sera peut-être de voir un gouvernement national extrêmiste arriver au pouvoir sur un programme anti-européen, pour se casser les dents et constater qu'il n'a d'autre choix que d'avaler sa chique et de faire ce qu'on lui dit, comme l'ont fait tant d'autres avant lui.

Comment se créent les nations

Vous pouvez trouver ce processus non démocratique, mais il faut constater que les créations de nations se sont toujours faites dans la douleur. La réaction en chaîne de Monnet est probablement préférable à la conquête violente ou aux constructions de frontières qui ont fait les drames du 20ième siècle. Il vaut certainement mieux subir cela que les sombres manœuvres de Vladimir Poutine; Les ex-pays de l'Est ne s'y trompent pas, qui sont prêts à tout subir de l'Europe plutôt que de courir le risque de retourner dans le giron russe. Ceux qui veulent la construction d'une Europe différente, selon un autre mécanisme, devraient en tous les cas s'appuyer sur une approche réaliste du mécanisme actuel plutôt que de s'imaginer que tout pourrait s'arranger, pourvu qu'on mette en place la bonne rustine. L'Europe se construit dans la douleur et les crises, mais on ne construit pas de nation sans douleur.

ENSIMAG $1^{\text{ère}}$ année — Mai 2014

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 : 11 pts, Partie 2 : 9 pts.

Première partie

En médecine, le risque de thrombose (formation d'un caillot dans le réseau veineux des membres inférieurs) d'un adulte peut être évalué en mesurant son taux de D-dimères par dosage sanguin. La variable d'intérêt, notée X, est le logarithme de ce taux. X est une variable aléatoire de loi normale de moyenne m et de variance connue $\sigma^2 = 0.09$. On considère que m ne peut prendre que deux valeurs : $m = -1$ pour les individus n'ayant pas de risque de thrombose, et $m = 0$ pour les individus ayant un risque de thrombose.

Pour un patient donné, on mesure la réalisation x de X. On souhaite, au vu de x, se prononcer sur les deux hypothèses "le patient a un risque de thrombose" et "le patient n'a pas de risque de thrombose".

1. Le docteur A pense que le plus important est de ne pas inquiéter ses patients à tort.
   (a) Expliquer pourquoi cela le conduit à tester H_0 : "$m = -1$" contre H_1 : "$m = 0$".
   (b) Construire la région critique de ce test.
   (c) A partir de quelle valeur de x peut-on conclure que le patient a un risque de thrombose, au seuil $\alpha = 5\%$ puis $\alpha = 1\%$?
   (d) Calculer la puissance du test. Donner la probabilité que le docteur A détecte correctement un patient à risque pour $\alpha = 5\%$ puis $\alpha = 1\%$.

2. Le docteur B préfère inquiéter un patient à tort plutôt que de ne pas l'avertir d'un risque réel.
   (a) Donner les hypothèses du test mis en place par le docteur B et construire sa région critique.

(b) A partir de quelle valeur de x peut-on conclure que le patient n'a pas de risque de thrombose, au seuil $\alpha = 5\%$ puis $\alpha = 1\%$?

3. Montrer que, selon la valeur de α, il peut ou non exister des valeurs de x pour lesquelles le docteur A conclura que le patient a un risque de thrombose alors que le docteur B conclura qu'il n'en a pas. Etudier les cas $\alpha = 5\%$ et $\alpha = 1\%$.

4. Il y a un mois, $n = 9$ personnes ont été diagnostiquées comme présentant un risque de thrombose. Depuis, elles ont suivi un régime alimentaire particulier. Aujourd'hui, on mesure à nouveau le logarithme de leur taux de D-dimères, noté Y. Y est de loi normale de moyenne μ. Dans un premier temps, on suppose que Y est de variance connue $\sigma^2 = 0.09$. Donc les observations forment un échantillon de n variables aléatoires $Y_1, \ldots, Y_n$ indépendantes et de même loi $\mathcal{N}(\mu, 0.09)$. La moyenne empirique observée est $\bar{y}_n = -0.24$. On souhaite déterminer si le régime alimentaire a permis à ces patients d'éliminer le risque de thrombose.

   Ecrire le problème sous forme de test d'hypothèses simples, construire sa région critique, calculer la p-valeur et conclure.

5. Dans ce groupe de $n = 9$ patients, l'écart-type estimé est $s'_n = 0.37$. Cela remet-il en cause l'hypothèse que $\sigma^2 = 0.09$?

Deuxième partie

On observe des réalisations $x_1, \ldots, x_n$ de variables aléatoires $X_1, \ldots, X_n$ indépendantes et de même loi de probabilité à valeurs dans $\mathbb{R}^+$, définie par la densité :

$$f(x) = \sqrt{\frac{\lambda}{2\pi x^3}}\, e^{-\dfrac{\lambda(x-m)^2}{2m^2 x}}$$

où m et λ sont deux paramètres dans $\mathbb{R}^{+*}$.

1. Calculer les estimateurs de maximum de vraisemblance de m et λ. Montrer qu'on peut les exprimer à l'aide de $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ et de $\overline{1/X}_n = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{X_i}$.

2. On admet que l'espérance et la variance de cette loi de probabilité sont respectivement m et $\frac{m^3}{\lambda}$. En déduire les estimateurs des moments de m et λ. Montrer que l'estimateur de m est sans biais.

On suppose à partir de maintenant que $\lambda = 1$.

3. Calculer la quantité d'information sur m apportée par les observations. En déduire que l'estimateur de m obtenu est sans biais et de variance minimale.

4. On suppose maintenant que la taille de l'échantillon est suffisamment grande pour que l'on puisse approcher la loi de $\bar{X}_n$ par une loi normale, en utilisant le théorème central-limite. Construire un test asymptotique de H_0 : "$m \leq m_0$" contre H_1 : "$m > m_0$".

5. Pour $n = 100$, à quelle condition sur la moyenne empirique des observations conclura-t-on que $m > 2$ au seuil 5% ?

EXAMEN du 27 mai 2014. Durée: 3h. 2 pages numérotées.
Documents manuscrits ou polycopiés autorisés. Aucun livre. Calculatrices interdites.

Il sera tenu le plus grand compte de la rédaction. Il est important de bien expliquer ce que vous faites.
Veuillez noter sur votre copie le nom de votre enseignant : BIENIA ou SZIGETI

EXERCICE 1: [4 pts]

Une société spécialisée en transport routier dispose d'un budget de *3 750 000 €* pour renouveler son équipement. On a le choix entre trois types de véhicules: ***A***, ***B*** et ***C***. Le véhicule ***A***, avec une charge utile de *10* tonnes, peut rouler avec une vitesse moyenne de *70 km/h* et coûte *80 000 €*. Le véhicule ***B***, avec une charge utile de *20* tonnes, peut rouler avec une vitesse moyenne de *60 km/h* et coûte *130 000 €*. Le véhicule ***C***, qui est une version du ***B*** avec compartiment-lit permettant à un chauffeur de dormir (pendant que l'autre conduit), a une charge utile de *18* tonnes et coûte *150 000 €*.

Le véhicule ***A***, conduit par un camionneur, peut rouler jusqu'à *6* heures par jour; si l'on lui affecte deux camionneurs il peut rouler jusqu'à *12* heures et jusqu'à *18* heures par jour si l'on organise un relais de trois camionneurs.

Les véhicules ***B*** et ***C*** nécessitent une équipe de deux camionneurs. Le véhicule ***B*** peut rouler jusqu'à *6* heures, *12* heures ou *18* heures respectivement par jour, en fonction du nombre d'équipes (une, deux ou trois) qui sont lui affectées. Ces limites dans les situations analogues pour le véhicule ***C*** vont jusqu'à *7* heures, *14* heures ou *21* heures par jour.

La société emploie actuellement *149* camionneurs et il est impossible d'envisager des embauches supplémentaires. Les garages de la société ne peuvent garantir la maintenance quotidienne de plus de *30* véhicules.

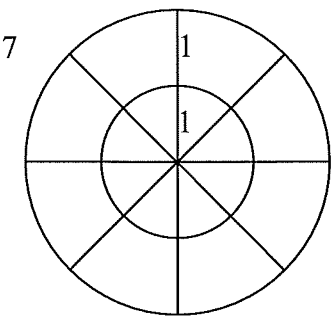
Quel est l'achat optimal qui permet de maximiser la capacité de la société exprimée en *tonnes×km par jour* ? **Modéliser** ce problème par un **programme linéaire**. La solution n'est pas demandée.

EXERCICE 2: [3 pts]

Soit G un graphe 3-régulier (tous les sommets ont même degré $= 3$). Montrer que si G possède un cycle hamiltonien alors l'ensemble d'arêtes de G est la réunion de 3 couplages parfaits.

EXERCICE 3: [3 pts]

Déterminer un arbre de poids minimum dans le graphe ci-contre avec 17 sommets où les poids des arêtes correspondent à leur longueur géométrique. Donner ce poids total minimum.



Combien y-a-t-il d'arbres couvrants de poids minimum (en supposant que les sommets sont étiquetés) ?

EXAMEN du 27 mai 2014. Durée: 3h. 2 pages numérotées.
Documents manuscrits ou polycopiés autorisés. Aucun livre. Calculatrices interdites.

EXERCICE 4: [4 pts]

Considérons le problème d'ordonnancement simple où les tâches sont tous les diviseurs de $4! = 24$. Le début du projet est *1* et la fin *24*. La tâche *a* précède la tâche *b* $\Leftrightarrow$ *a divise b.*

La durée d'une tâche est égale à sa valeur. Quelle est la durée minimum du projet ? Quelles sont les tâches critiques ? Quelles tâches ont la plus grande marge ?

Utiliser la méthode *potentiels-tâches* (*graphe des tâches*). Pour accélérer les calculs il est conseillé d'enlever les arcs de transitivité.

EXERCICE 5: [4 pts]

Considérons le programme linéaire suivant :

$$\begin{array}{lrl} \textit{minimiser:} & z = 2x_1 + x_3 & \\ \textit{sous:} & x_1 + x_2 - x_3 & \geq 5 \\ & x_1 - 2x_2 + 4x_3 & \geq 8 \\ & x_1 \geq 0, x_2 \geq 0, x_3 & \geq 0. \end{array}$$

a) Résoudre ce programme par l'algorithme du simplexe.

b) Pour le moment le coefficient de x_1 dans de la fonction-objectif z est 2. A partir de quelle valeur de ce coefficient verra-t-on apparaître une valeur de x_1 non nulle dans la solution optimale ? Expliquez clairement pourquoi.

EXERCICE 6 : [4 pts]

Deux joueurs ***X*** et ***Y*** doivent choisir simultanément une mise de *1€* ou *5€*. Si les mises des deux joueurs sont identiques alors le joueur ***X*** gagne la mise de son adversaire, sinon c'est le joueur ***Y*** qui gagne la mise de ***X***.

a) Donner la matrice des gains. **[0,5 pt]**

b) Ecrire les deux programmes (duaux) qui correspondent à la recherche des stratégies mixtes optimales des deux joueurs. **[1 pts]**

c) Utiliser la méthode graphique vue en cours pour trouver la stratégie mixte optimale du joueur ***X***. **[1 pts]**

d) En utilisant cette stratégie mixte optimale du joueur X, déterminer la stratégie mixte optimale du joueur ***Y*** par le théorème des écarts complémentaires. **[1 pts]**

e) Quelle est la valeur du jeu? Ce jeu est-il juste (équitable)? **[0,5 pts]**

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

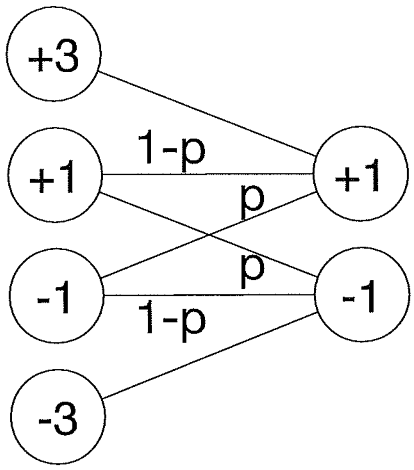
Lundi 26 mai 2013, 14h-16h.

POLYCOPIE, NOTES DE COURS ET CALCULATRICES AUTORISES

Le sujet est composé de 4 problèmes indépendants. Le barème est donné à titre indicatif.

1 CAPACITÉ D'UN CANAL (7 points)

On considère le canal sans mémoire représenté par le diagramme de transition ci-contre.



On note les probabilités en entrée

$P_X(+3), P_X(+1), P_X(-1), P_X(-3)$

et

$P_Y(+1), P_Y(-1)$ celles de sortie.

1. (a) Ecrire la matrice de transition de ce canal (1 point)
   (b) Est-il uniforme par rapport à son entrée ? (1/2 point)
   (c) Est-il uniforme par rapport à sa sortie ? (1/2 point)
2. Lorsque la loi d'entrée est uniforme, quelle est la loi de la sortie ? (1/2 point)
3. Existe-t-il d'autres lois d'entrée pour lesquelles la sortie suit une loi uniforme ? (1 point)
4. Exprimer $H(Y|X)$ (l'entropie conditionnelle de sa sortie Y par rapport à son entrée X) en fonction de p et de $P_X(+1)$ et $P_X(-1)$. (2 points)
5. En déduire la capacité de ce canal. (1 point)
6. Pouvez donner un canal plus simple équivalent à celui traité dans cet exercice ? (1/2 point)

2 CASCADE DE CANAUX BINAIRES (3 points)

On note $C(p)$ la capacité d'un canal binaire symétrique (CBS) de probabilité d'erreur $P_e = p$. On considère la cascade de deux CBS, avec pour le premier $P_e^1 = p$ et pour le second $P_e^2 = q$.

1. Quel est la matrice de transition du canal équivalent à cette cascade ? (2 points)
2. Quelle est la capacité de ce canal équivalent ? (1 point)

3 CODAGE SOURCE (7 points)

On considère une source simple S à 3 états notés a, b, c de probabilités $\begin{array}{rcl} p(a) & = & 1/8 \\ p(b) & = & 1/8 \\ p(c) & = & 3/4 \end{array}$.

1. Quelle est la redondance de la source S ? (1.5 point)

   On code la source S à l'aide du code binaire C suivant : $\begin{array}{ccc} \text{a} & \rightarrow & 1100 \\ \text{b} & \rightarrow & 1001 \\ \text{c} & \rightarrow & 00 \end{array}$

2. Dire pourquoi le code C est instantané ? C est-il déchiffrable ? (1 point)
3. Démontrer que C n'est pas 100 % efficace. (0.5 point)
4. Existe-t-il un autre jeu de probabilités (pour les états a,b,c) pour lequel C serait 100% efficace ? Pourquoi ? (1 point)
5. Codage binaire de la source S :
   (a) Construire un code de Huffman pour la source S, donner son efficacité. (1 point)
   (b) Construire un codage avec une efficacité supérieure, donner son efficacité. (2 point)

4 TRANSFORMATIONS DE L'ENTROPIE (3 points)

Soit X une variable aléatoire discrète à valeurs dans $\{+1; +2; +4; +8\}$ d'entropie $H = 2$ bits.

1. Quelle est la loi de X ? (0.5 point)
2. Calculer l'entropie de la variable aléatoire $Y = \log_2 X$. (0.5 point)
3. Donner une fonction f telle que l'entropie de $f(X)$ soit non nulle et strictement inférieure à H. (0.5 point)
4. Montrer qu'il n'existe pas de fonction g telle que l'entropie de $g(X)$ soit strictement supérieure à H ? (1.5 point)

Ensimag 1ère année — mai 2014

Théorie des langages 2

Durée : 3h.
Documents : tous documents autorisés.

Exercice (6 points) Calculabilité

▷ **Question 1 (2 points)** On considère un vocabulaire Σ ayant au moins 2 éléments ; tous les langages dont on parle sont des sous-ensembles de Σ^*. En particulier, le complémentaire d'un langage L s'entend comme complémentaire dans Σ^*.

Les affirmations suivantes sont-elles vraies ou fausses ? Justifiez... Chaque bonne réponse justifiée vaut $\frac{1}{4}$ point, chaque bonne réponse non (ou mal) justifiée vaut $\frac{1}{8}$ point, chaque mauvaise réponse vaut $-\frac{1}{4}$ point, une absence de réponse vaut $-\frac{1}{8}$ point, le total sur la question ne pouvant pas être négatif.

1.1 si L et M sont non récursifs, alors $L \cap M$ est non récursif

1.2 si L et M sont récursifs, alors $L \cup M$ est récursif

1.3 si L est non récursif, alors son complémentaire est non récursif

1.4 si L est fini, alors son complémentaire est récursif

1.5 si $L \subseteq M$ et M est récursif, alors L est récursif

1.6 une réduction est une application injective

1.7 une réduction est une application surjective

1.8 on sait simuler une MTN par une MTD, avec une perte de coût quadratiaque

▷ **Question 2 (4 points)** Soit L un langage récursivement énumérable.

2.1 (2 points)
Montrer que si, $\forall n \in \mathbb{N}$ L contient *exactement un* mot de longueur n, alors L est récursif.

2.2 (1 point)
Que peut-on dire si en question 2.1 on remplace "*exactement un*" par "*exactement n*" ?

2.3 (0,5 point)
Que peut-on dire si en question 2.1 on remplace "*exactement un*" par "*au plus n*" ?

2.4 (0,5 point)
Que peut-on dire si en question 2.1 on remplace "*exactement un*" par "*au moins n*" ?

Problème (14 points) Langages hors-contexte et reconnaisseurs

On s'intéresse à un sous-langage des expressions entières du langage C. Soit G_1 la grammaire suivante, E_1 étant l'axiome :

1)	E_1	$\rightarrow$	$E_1 + E_2$
2)	E_1	$\rightarrow$	E_2
3)	E_2	$\rightarrow$	(E_1)
4)	E_2	$\rightarrow$	*place*
5)	*place*	$\rightarrow$	**idf**
6)	*place*	$\rightarrow$	**++ idf**
7)	*place*	$\rightarrow$	**idf ++**

Le vocabulaire terminal est VT_1={**idf**, +, (,), ++}. Les notations ont leur sens habituel. On rappelle la sémantique des opérateurs de pré-incrémentation et de post-incrémentation : 1) l'expression **++a** où **a** vaut **n** a pour valeur **n+1** et cette nouvelle valeur est effectée à **a**, 2) l'expression **a++** où **a** vaut **n** a pour valeur **n** et la valeur **n+1** est affectée à **a**. Attention, + et ++ sont deux symboles distincts du vocabulaire terminal.

▷ **Question 3 (2 points)**
Donnez une grammaire LL(1) pour le langage $L(G_1)$. On prouvera le caractère LL(1) de la grammaire proposée.

▷ **Question 4 (2 points)**
Une expression a des effets de bord si l'évaluation de cette expression entraine une modification de la mémoire. Donnez un calcul d'attributs **sur la grammaire d'origine** G_1 permettant de calculer si une expression peut provoquer ou non un effet de bord (on utilisera la valeur **true** pour les effets de bord et **false** pour les expressions sans effet de bord).

▷ **Question 5 (3 points)**
Ecrire en Ada un analyseur LL(1) pour la grammaire de la question 2 donnant en résultat un booléan indiquant si l'expression analysée contient, ou non, des effets de bord. **On utilisera la méthode d'écriture d'analyseurs vue en cours. On écrira toutes les procédures d'analyse sauf la procédure associée au non-terminal E_2 dont on précisera les paramètres, si besoin est.** On suppose donnée la fonction `lire_mot return Token` avec `type Token is (Idf, Plus, Par_Ouv, Par_Ferm, PlusPlus, Dollar);` `Dollar` indiquant la fin de fichier.

▷ **Question 6 (2 points)**
On ajoute dans le langage l'opérateur d'affectation = en étendant la grammaire G_1 par les deux règles suivantes :

a) $E_0 \rightarrow$ *place* $= E_0$
b) $E_0 \rightarrow E_1$

et en remplaçant la règle 3) par $E_2 \rightarrow (E_0)$.

E_0 est maintenant l'axiome et $VT_0 = VT_1 \cup \{=\}$. Soit G_0 cette grammaire. On rappelle qu'une affectation de la forme *place* = *exp*, vue comme une expression, réalise l'affectation et produit la valeur de l'expression.

1. Quelle est la priorité de l'opérateur = par rapport à l'opérateur + ?
2. Donner une grammaire LL(1) pour le langage $L(G_0)$. On prouvera son caractère LL(1).

▷ **Question 7 (2 points)**

Les expressions avec effet de bord peuvent être problématiques car elles sont sensibles à l'ordre d'évaluation. Certaines normes de programmation restreignent leur utilisation : une expression ne peut contenir qu'un effet de bord au plus haut niveau de l'expression.

Exemples admis : **x++** **x=y+7**

Exemples interdits : **x++ + z** **(x=y)+z** **x=y=z+z**

Proposer un calcul d'attributs sur la grammaire G_0 permettant de vérifier cette restriction.

▷ **Question 8 (3 points)**

Soit G une grammaire quelconque LL(1), d'axiome S et telle que $\epsilon \notin L(G)$. Montrer que le langage $(L(G))^+$ est LL(1). Que se passe t-il si $\epsilon \in L(G)$? Peut-on en déduire quelque chose sur le caractère LL(1) du langage $(L(G))^+$?

SESSION de RATTRAPAGE JUIN 2014

ALGORITHMIQUE 1	- 3 PAGES
ALGORITHMIQUE 2	- 3 PAGES
ANALYSE POUR L'INGENIEUR	- 1 PAGE
ARCHITECTURE	- 5 PAGES
INTRODUCTION AUX RESEAUX	- 2 PAGES
METHODES NUMERIQUES	- 2 PAGES
PRINCIPES et METHODES STATS.	- 2 PAGES
PROBABILITES APPLIQUEES	- 1 PAGE
RECHERCHE OPERATIONNELLE	- 2 PAGES
THEORIE DE L'INFORMATION	- 2 PAGES
THEORIE DES LANGUANGES 1	- 3 PAGES
THEORIE DES LANGUANGES 2	- 2 PAGES

Algorithmique 1

Ensimag - 1A

Juillet 2014

Durée : 2h

Machines électroniques interdites **documents autorisés : 1 feuille de notes de cours**

Les différentes parties du sujet sont indépendantes.

Le barème est donné à titre indicatif.
Les algorithmes sont à écrire en Ada et non en pseudo-code.

---

1 Listes triées (15 points)

On s'intéresse dans cet exercice à l'écriture d'un paquetage de listes triées d'entiers. Les listes seront ici simplement chainées et les éléments triés par ordre croissant.

1.1 Listes simples

1.1.1. (½ point) Proposez une structure de *Cellule* ainsi qu'une structure de *Liste.* Le paquetage devra permettre des insertions en tête et en queue ainsi que la possibilité de récupérer la taille d'une liste ; toutes ces opérations se réalisant en temps constant.

1.1.2. (½ point) Proposez une fonction vérifiant que les éléments d'une liste donnée en argument sont bien triés par ordre croissant (renvoyant vrai si c'est OK et faux sinon).

1.1.3. (3 points) Écrivez une procédure `Creation(T:in Tableau ; L:out Liste)` créant une liste triée des éléments. *Attention*, les éléments du tableau ne sont pas forcément triés. De plus, votre code devra être simple, clair et respecter le prototype donné.

1.2 Listes avancées

On se propose maintenant d'améliorer les coûts de stockage en mémoire car nos utilisateurs font une utilisation particulière de la bibliothèque de listes et beaucoup de listes partagent énormément de données. On cherche donc à compresser les listes comme illustré figure 1. *Note: un élément peut être partagé par plus de deux listes ; aucune restriction n'est posée sur la forme et la quantité de données partagées.*

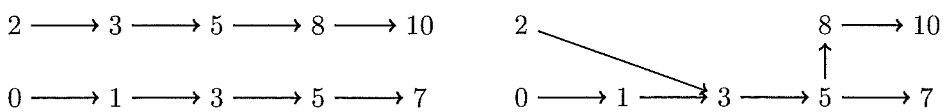


FIGURE 1 – Avant/Après compression

1.2.1. (2 points) Proposez une modification des structures de *Liste* et *Cellule* permettant de stocker ce type de liste compressée. Vous expliquerez clairement l'idée derrière vos structures. Vous pouvez également proposer de nouvelles structures.

1.2.2. (2 points) Écrivez une procédure **procedure** `Affichage(L : in Liste)` réalisant l'affichage d'une liste compressée.

1.2.3. (4 points) Écrivez une procédure **procedure** `Compression(L1, L2 : in out Liste)` réalisant la compression des deux listes compressées données en argument (et libérant la mémoire économisée).

1.2.4. (3 points) Écrivez une procédure **procedure** `Insertion(L : in out Liste ; E : Entier)` qui insère un nouvel élément dans une liste compressée.

2 Arithmétique (5 points)

Le but de cet exercice est de construire les opérations arithmétiques usuelles pour des entiers binaires non bornés. Ces entiers sont représentés sous la forme binaire usuelle, en écrivant les bits sous forme d'une liste de booléens, poids faibles en tête, et sans zéro superflu sur les poids forts. L'entier 0 est donc la liste vide. On fournit le paquetage suivant pour manipuler les entiers :

```
package Entiers is

   -- type des entiers binaires récursifs non bornés.
   type Entier is private ;

   function Zero return Entier ;

   function EstZero(X:Entier) return Boolean ;
   -- retourne True ssi X est Zero.

   function Cons(Bit: Boolean; Suiv: Entier) return Entier ;
   -- retourne si Bit alors 2*Suiv+1
   --                 sinon 2*Suiv

   function PremierBit(E:Entier) return Boolean ;
   -- requiert  not EstZero(E)
   -- retourne  (E mod 2)=1.

   function Suiv(E:Entier) return Entier ;
   -- requiert  not EstZero(E)
   -- retourne  E / 2.

private

   type Cellule ;
   type Entier is access Cellule ;

end;
```

. L'implémentation est également donnée.

Dans les programmes à réaliser ici, on ne se soucie pas de la désallocation des pointeurs (on imagine qu'un glaneur de cellules pourrait le faire automatiquement). On s'autorise à faire du partage sur les listes chaînées. Par contre, le paquetage `Entiers` interdit syntaxiquement de modifier en place une liste existante. Ça donne un style de programmation appelé *programmation fonctionnelle.*

Pour cette structure de donnée, les algorithmes peuvent s'écrire de façon très naturelle en remarquant que le type `Entier` a une structure récursive. Étant donné un élément `N` de type `Entier` :

— Soit `N` est zéro.
— Soit `N` est de la forme 2`M` avec `M` entier différent de zéro.
— Soit `N` est de la forme 2`M`+1 avec `M` entier.

Par exemple, la fonction `EvalNat` suivante prend un `Entier` et retourne l'élément correspondant du type `Natural` (en levant éventuellement `Constraint_Error` si débordement).

```
function EvalNat(X: Entier) return Natural is
begin
   if EstZero(X)
   then
      return 0 ;
   elsif PremierBit(X) then
      return 2*EvalNat(Suiv(X))+1 ;
   else
      return 2*EvalNat(Suiv(X)) ;
   end if ;
end EvalNat ;
```

On demande d'écrire les fonctions et procédures suivantes, en utilisant ce principe récursif. On ne demande pas que ces sous-programmes soient récursifs terminaux, ni que les algos soient optimaux. Par contre, on demande de justifier soigneusement la correction des algorithmes.

2.1. (1 point) Écrire la réciproque de `EvalNat` :

```
function LisNat(N: Natural) return Entier ;
```

2.2. (1 point) Écrire la fonction qui calcule le successeur d'un entier :

```
function Succ(X: Entier) return Entier ;
```

2.3. (1 point) Écrire une fonction qui additionne 2 entiers :

```
function Plus(X,Y: Entier) return Entier ;
```

2.4. (1 point) Écrire la fonction qui calcule la différence de 2 entiers. Lorsque le deuxième entier est strictement supérieur au premier, elle lève l'exception `ErreurArith` :

```
ErreurArith: exception ;
function Moins(X,Y: Entier) return Entier ;
```

2.5. (1 point) Écrire une procédure qui prend un entier `X` et l'affiche en base 10 en terminant par un retour chariot. Attention, ici, il ne faut pas passer par `EvalNat(X)` car on veut pouvoir afficher un nombre `X` arbitrairement grand.

```
procedure AfficheNL(X:Entier) ;
```

Algorithmique 2

Ensimag - 1A

Juillet 2014

Durée : 2h
Machines électroniques interdites **documents autorisés : notes de cours**

Les différentes parties du sujet sont indépendantes.

Le barème est donné à titre indicatif.
Tous les calculs de coût doivent être justifiés.
On peut répondre à chaque question dans un espace court (maximum 1/3 de page).

---

1 Points proches (7 points)

On considère un ensemble P de points du plan. On utilise ici une fonction d de distance utilisant la norme L^1 ($d((x_1, y_1), (x_2, y_2)) = x_2 - x_1 + y_2 - y_1$).

1.1 Algorithme naïf

1.1.1. (1 point) Proposez une fonction naïve renvoyant les deux points de P les plus proches. Quel est son coût au pire cas ?

1.2 Second algorithme

On considère dans un premier temps une grille virtuelle de carreaux de taille 1×1, plaquée sur le plan.

1.2.1. ($^1\!/_2$ point) Donnez une borne supérieure sur la distance entre deux points quelconques localisés dans un même carreau.

1.2.2. (2 points) On cherche à insérer P dans un dictionnaire tel qu'il soit facile et rapide d'itérer sur tous les points d'un même carreau. Quelle structure de dictionnaire vous parait indiquée ? Quel est le coût total de l'insertion de tous les points ? Quel est le coût mémoire (raisonnable) de la structure ?

1.2.3. ($^1\!/_2$ point) Donnez un exemple de deux points très proches n'étant pas situés dans un même carreau.

1.2.4. (2 points) En utilisant plusieurs grilles virtuelles, proposez un algorithme renvoyant, s'il existe, un couple (quelconque) de points d'une distance inférieure ou égale à la borne de la question 1.2.1. Quel est son coût au pire cas ?

1.2.5. (1 point) En déduire à l'aide d'une dichotomie un algorithme rapide trouvant le couple de points le plus proche. Quel est son coût au pire cas ?

2 Algorithmes de sélection (7 points)

Dans ce second exercice, on cherche à résoudre le problème de la *sélection* du $k^{\text{ème}}$ élément. On dispose en entrée d'un tableau T de n éléments $T(1), T(2), \dots, T(n)$. T n'est pas nécessairement trié. On dispose également d'un entier k tel que $1 \leq k \leq n$. On cherche à déterminer en sortie la valeur du $k^{\text{ème}}$ plus petit élément de T. Par exemple pour $k = 1$ on doit renvoyer la valeur du plus petit élément de T. Nous supposerons pour faire simple que tous les éléments de T sont différents.

Une solution naïve au problème consiste à trier T par ordre croissant puis à renvoyer $T(k)$. Malheureusement cette solution coûte $O(n \log(n))$ ce qui est un coût élevé pour un problème aussi simple.

2.1 Quickselect

On propose un premier algorithme utilisant une procédure de *Segmentation* similaire à celle du *Quicksort*. On tire *au hasard* un indice pivot p du tableau puis la segmentation place en temps linéaire (et en n comparaisons) tous les éléments inférieurs à $T(p)$ devant p et tous les éléments supérieurs à $T(p)$ derrière p. La procédure retourne également la nouvelle place du pivot. Nous supposerons disposer d'une procédure de segmentation et son code n'est donc pas à écrire.

2.1.1. (2 points) Utilisez la procédure de segmentation pour écrire un algorithme récursif (en pseudo-Ada) résolvant *sélection.*

2.1.2. (1 point) L'algorithme étant probabiliste, on cherche à calculer l'espérance (sur les tirages réalisés) du coût. En considérant un tirage uniforme du pivot, proposez une formule récursive calculant $C(n, k)$, l'espérance du coût pour la sélection du $k^{\text{ème}}$ élément dans un tableau de taille n. On ne comptera dans ce coût que les comparaisons d'éléments du tableau.

2.2 Soft Heaps

On cherche maintenant à obtenir un algorithme *déterministe* de performance équivalente. Pour ce faire, nous gardons le même principe de récursion, mais modifions la manière de choisir le pivot.

Nous utilisons une variante des tas appellée *soft heap* et introduite par Chazelle en 2000. Il s'agit comme un tas d'une file de priorité. On insère donc des éléments avec une certaine priorité puis on dispose d'une opération permettant d'extraire l'élément le plus prioritaire du tas.

Le soft heap à la particularité de *corrompre* certaines priorités ce qui signifie qu'elles peuvent alors prendre une valeur *erronée*, inférieure à la valeur réelle. On dispose d'un paramètre ϵ, compris entre 0 et 1 que l'utilisateur choisit pour obtenir les garanties suivantes :

- si n éléments ont été insérés dans le tas, au plus $n \times \epsilon$ priorités ont été corrompues ;
- les insertions se font en temps $O(1)$;
- les suppressions se font en temps $O(\log(1/\epsilon))$.

Nous prenons ici $\epsilon = 1/3$.

2.2.1. (1 point) Supposons dans un premier temps que l'on dispose d'un tas classique, mais donc on néglige les coûts (ce qui n'est bien sûr pas réaliste). Comment utiliser un tel tas pour choisir un bon pivot ?

2.2.2. (2 points) En utilisant un *soft heap*, proposez un algorithme de choix de pivot garantissant que le choix n'est jamais trop mauvais.

2.2.3. (1 point) Montrez à l'aide du *master theorem* que le coût au pire cas de l'algorithme est $O(n)$.

3 Graphes (6 points)

On cherche à réaliser la fermeture transitive d'un graphe pondéré. On dispose en entrée d'une matrice d'adjacence A telle que $A(i,j) = -1$ s'il n'existe pas d'arète entre les sommets i et j et la distance (positive) entre i et j s'il existe une arète les reliant. On cherche à calculer la matrice B telle que $B(i,j) = -1$ s'il n'existe aucun chemin de i à j et la longeur du plus court chemin de i à j sinon. Le degré du graphe codé par A est supposé inférieur à 6.

On se propose d'utiliser l'algorithme de *Dijsktra* en utilisant un tas comme file de priorité.

3.1. (2 points) Expliquez comment utiliser Dijkstra pour construire B.

3.2. (1 point) Expliquez comment sont gérés les accès au tas. En particulier comment faire pour éviter de traiter plusieurs fois un même point.

3.3. (3 points) Quelle structure de graphe vaut-il mieux utiliser ? Pourquoi ? Quel est alors le coût total de l'algorithme ?

Examen Session 2

Mercredi 2 juillet 2014 - 2h

Documents manuscrits et polycopié de cours. Tout autre document et calculatrices interdits.

N.B. : *La rédaction sera prise en compte dans la notation. Toute affirmation devra être justifiée.*

Exercice 1

Soit E l'espace vectoriel des fonctions à valeurs dans $\mathbb{R}$ de classe C^1 sur $[0,1]$ et vérifiant $f(0)=0$. On définit sur cet espace les deux normes suivantes : $N_1(f) = \|f\|_\infty$ et $N_2(f) = \|f'\|_\infty$

1. *Montrer que $N_1(f) \leq N_2(f)$. En déduire que l'application identique de (E, N_2) vers (E, N_1) est continue.*
2. *A l'aide de la fonction $f_n = \frac{x^n}{n}$, montrer que l'application identique de (E, N_1) vers (E, N_2) n'est pas continue.*

Exercice 2

On pose

$$f(x) = \int_0^\infty \frac{e^{-\frac{x^2}{2t}-\frac{t}{2}}}{\sqrt{t}}\, dt$$

1. *Montrer que $f \in L^1(\mathbb{R})$ et calculer $\int_\mathbb{R} f$.*
2. *Calculer $\hat{f}$, transformée de Fourier de f.*

Exercice 3

Soit la suite de fonctions définies sur $\mathbb{R}$:

$$f_n(x) = \mathbb{1}_{[0,n[}\,(1-\frac{x}{n})^n \quad , \qquad n \geq 1$$

où $\mathbb{1}$ désigne la fonction indicatrice. Calculer $\lim_{n\to+\infty} \int_\mathbb{R} f_n$.

Exercice 4

Soit $K = [0,1]^d$, $E = C^1(K)$

1. *Soit*

$$\|u\|_E = \max\left(\|u\|_\infty, \max_{x\in K} \|\nabla u(x)\|_2\right)$$

où $\|\cdot\|_2$ est la norme euclidienne dans $\mathbb{R}^d$. Montrer que $\|\cdot\|_E$ est une norme sur E, et que E est complet pour cette norme.

2. *Soit $A : E \to \mathbb{R}$ définie par*

$$A(u) = \int_K \|\nabla u(x)\|_2^2\, dx,$$

Montrer que A est différentiable sur E et calculer sa différentielle.

Architecture 1 :
Circuits Numériques et Éléments d'Architecture
Examen de rattrapage

ENSIMAG 1A

Année scolaire 2013–2014

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

Du tracé de droites en matériel

On se propose de réaliser un circuit permettant de tracer une ligne droite sur une grille en n'utilisant que des entiers et des opérations élémentaires. L'algorithme, développé par J. Bresenham en 1965 [1] et utilisé dans toutes les cartes graphiques peu ou prou, est décrit ci-après. On fait l'hypothèse que les variables $x0$, $y0$, $x1$ et $y1$ sont mémorisées dans des registres et disponibles dès le début de l'exécution de l'algorithme :

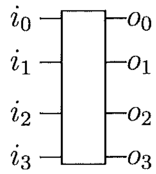
Données: quatre variables $x0$, $y0$, $x1$ et $y1$
Résultat: une suite d'appels à *set_pixel* qui permet de tracer la ligne

```
1  dx ← abs(x1 − x0);
2  dy ← −abs(y1 − y0);
3  if x0 < x1 then sx ← 1 ; else sx ← −1 ; endif;
4  if y0 < y1 then sy ← 1 ; else sy ← −1 ; endif;
5  err ← dx + dy;
6  while true do
7  |   set_pixel(x0, y0);
8  |   if x0 = x1 and y0 = y1 then break; endif;/* sort de la boucle while, va ligne 13 */
9  |   e2 ← 2 × err;
10 |   if e2 ≥ dy then err ← err + dy; x0 ← x0 + sx ; endif;
11 |   if e2 ≤ dx then err ← err + dx; y0 ← y0 + sy ; endif;
12 end
```

Ex. 1 : Circuit combinatoire (4 pts)

On a besoin d'un opérateur permettant de calculer la valeur absolue. On cherche à concevoir un circuit permettant de calculer la valeur absolue d'un nombre sur 4 bits codé en complément à 2. Ce circuit possède donc 4 entrées et 4 sorties, comme illustré sur la figure ci-dessous. Pour l'entrée comme pour la sortie, le bit numéro 0 est le bit de poids faible et le bit numéro 3 est le bit de poids fort. On rappelle que le plus petit nombre négatif, ici -8 codé 1000 en binaire, n'a pas d'opposé, mais qu'interprété en tant que nombre non signé, ce codage vaut 8, donc on adoptera le code 1000 comme valeur absolue de -8.

1. Jack E. Bresenham, Algorithm for Computer Control of a Digital Plotter, IBM Systems Journal, 4(1) :25-30, 1965.



Question 1 [1pt] Donnez la table de vérité des fonctions ainsi définies.

Question 2 [3pt] Simplifiez les fonctions booléennes des sorties en fonction des entrées. On le fera à l'aide de tables de Karnaugh en utilisant la forme normale disjonctive (ou de et) sauf pour les cas triviaux.

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

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. L'exécution de la fonction *set_pixel* consiste simplement à positionner un signal `plot` à 1 durant 1 cycle (allume le pixel de coordonnées (x0,y0) si `plot` est à 1).

Dans un premier temps, on va réaliser une version séquentielle de l'implantation de l'algorithme minimisant le matériel. On dispose pour cela d'**un seul** additionneur/soustracteur (add/-sub), de l'opérateur de calcul de la valeur absolue vue à l'exercice précédent (dont on dispose d'une version sur n bits), d'un comparateur et d'autant de registres et de multiplexeurs que l'on veut.

Question 1 [3 pts] Proposez une partie opérative minimaliste sous forme d'interconnexion d'opérateurs, de registres et de multiplexeurs, et la partie contrôle sous forme d'automate d'états.

On va dans la suite utiliser les transformations syntaxiques vues en cours pour faire apparaître le parallélisme dans l'algorithme. On suppose que l'on possède autant d'opérateurs que l'on veut, et l'on cherche à réduire le nombre de cycles pour réaliser une itération.

Question 2 [2 pts] Réécrivez le programme en utilisant les notions d'affectation concurrente et d'affectation conditionnelle présentées en cours, et en simplifiant les opérateurs si possibles.

Question 3 [3 pts] Proposez une partie opérative et une partie contrôle réalisant le programme tel que vous l'avez réécrit.

De la conception de processeurs

Ces exercices utilisent le processeur vu en TD et TP.

Ex. 3 : Calcul de la valeur absolue (4 pts)

On désire introduire une instruction permettant de calculer la valeur absolue de la différence de deux nombres (`abs rd, rs`) qui met donc dans rd la valeur absolue de $rd - rs$.

On rappelle le codage et la structure des instructions supportées par ce processeur :

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

Question 1 [2 pt] Étant donnée la PO en annexe, précisez les ajouts matériels qui sont nécessaires à l'exécution de cette nouvelle instruction. On ne se permet pas d'ajouter de nouvelles fonctionnalités à l'ALU (par ex. le calcul direct de la valeur absolue) ou d'ajouter un nouvel opérateur. On rappelle que le bit définissant le signe du résultat d'un calcul est le bit 31 de la sortie 's' de l'ALU. Vous pouvez ne dessiner que la partie de la PO qui se trouve modifiée sur la feuille fournie en annexe.

Question 2 [2 pt] Ajoutez à la PC vue en TD (et rappelée ci-dessous) les modifications permettant de supporter l'instruction `abs`. Précisez la valeur des signaux dans les états que vous ajoutez ou que vous seriez amené à modifier. Vous pouvez ajouter les modifications directement à la PC fournie en annexe.

Ex. 4 : Opération atomique *test and set* (4 pts)

On se propose d'ajouter l'instruction « atomique »[2] *test and set* (`tst ri, AD`) dans ce processeur. Le comportement de cette instruction est le suivant : le processeur va lire une donnée à l'adresse AD, la charge dans le registre rd et écrit la valeur 1 à cette même adresse.

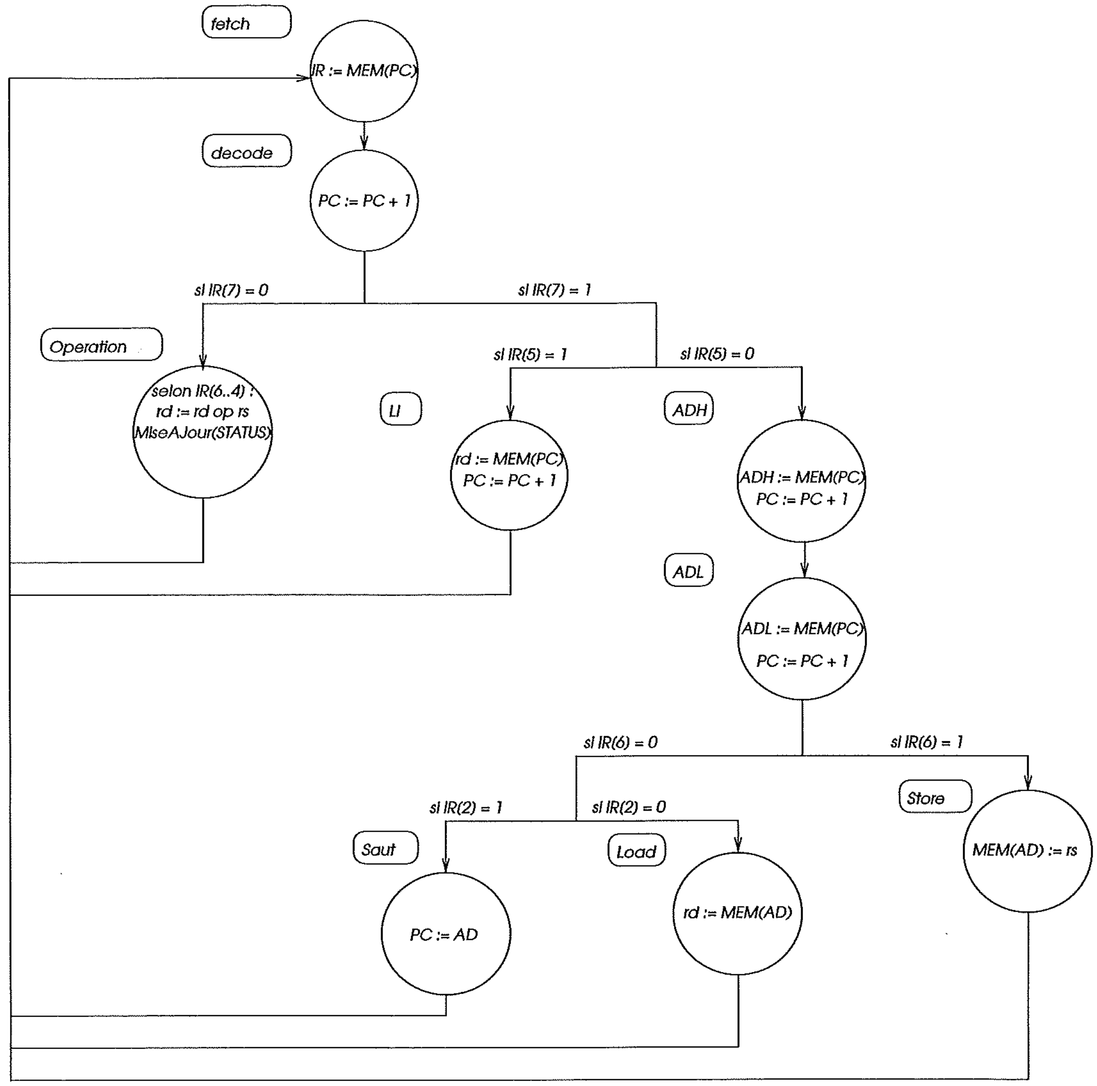
Question 1 [0,5 pt] Précisez un codage possible de l'instruction `tst` ainsi que sa structure en mémoire.

Question 2 [1,5 pt] Étant donnée la PO en annexe, précisez les ajouts matériels qui sont nécessaires à l'exécution de cette nouvelle instruction. Vous pouvez ne dessiner que la partie de la PO qui se trouve modifiée sur la feuille fournie en annexe.

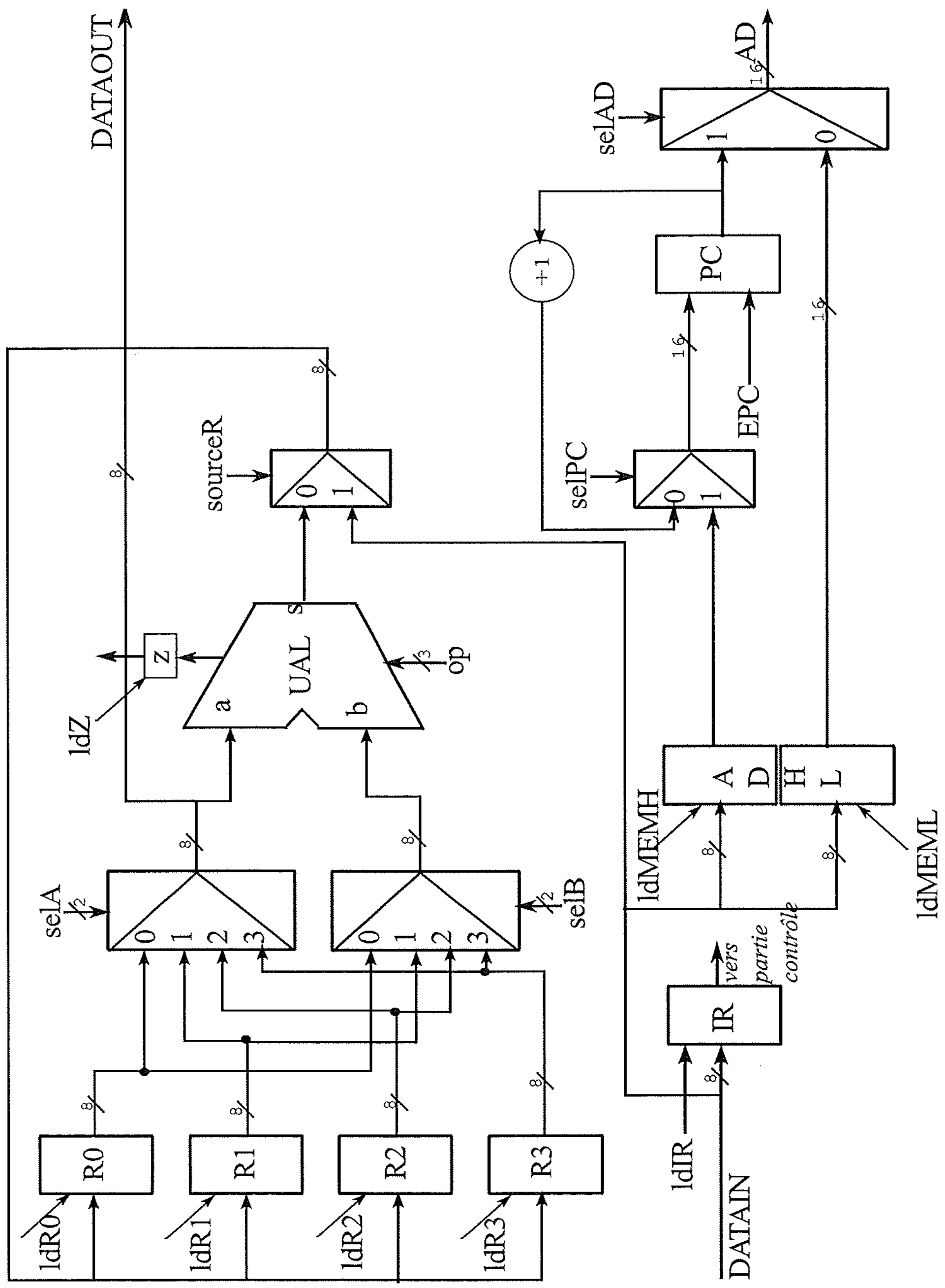
Question 3 [2 pt] Ajoutez à la PC vue en TD (et rappelée ci-dessous) les modifications permettant de supporter l'instruction `tst`. Précisez la valeur des signaux dans les états que vous ajoutez ou que vous seriez amené à modifier. Vous pouvez ajouter les modifications directement à la PC fournie en annexe.

2. L'instruction est dite « atomique » car le processeur ne relâche pas le bus entre la lecture et l'écriture, ce qui empêche un autre processeur d'accéder à cette case mémoire et ainsi permet d'implanter des synchronisations en multi-processeur. Cette information n'est pas nécessaire pour l'exercice.

Nom :
Prénom :


fetch
IR := MEM(PC)
decode
PC := PC + 1
si IR(7) = 0
si IR(7) = 1
Operation
selon IR(6..4) :
rd := rd op rs
MiseAJour(STATUS)
LI
si IR(5) = 1
si IR(5) = 0
rd := MEM(PC)
PC := PC + 1
ADH
ADH := MEM(PC)
PC := PC + 1
ADL
ADL := MEM(PC)
PC := PC + 1
si IR(6) = 0
si IR(6) = 1
si IR(2) = 1
si IR(2) = 0
Saut
PC := AD
Load
rd := MEM(AD)
Store
MEM(AD) := rs

Nom :
Prénom :



DATAOUT
AD
selAD
PC
+1
selPC
EPC
sourceR
UAL
op
ldZ
z
selA
selB
R0
R1
R2
R3
ldR0
ldR1
ldR2
ldR3
IR
vers partie contrôle
ldIR
DATAIN
ldMEMH
ldMEML
AD
HL

Grenoble INP – ENSIMAG
Année 2013-2014
N° d'inscription (carte étudiant) :
NOM :
Prénom :
Né(e) le : **à (ville + dépt ou pays) :**
N° de la place :

Introduction aux réseaux

Examen du 4/7/2014. Durée 1h00
Tous documents et calculatrice interdits

Exercice 1 : 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.

Exercice 2 : 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.

Questions de cours

1) *Soit une transmission réalisée en 'bande de base'. Que cela signifie-t-il ? Donnez un exemple de milieu de transmission adapté à ce type de système.*

2) *Faites la représentation temporelle de l'évolution des signaux pour la suite de bits suivante à transmettre : 1100101101, dans le cas où le codage utilisé est un code NRZ unipolaire (Expliquez la signification de NRZ).*

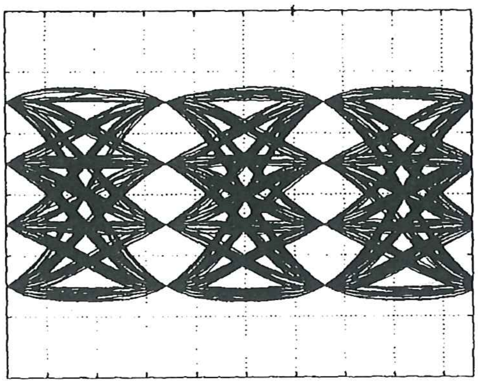
Le réseau sans fil Wi-Fi IEEE 802.11n utilise une modulation OFDM.

3) *Les différentes sous-porteuses radio peuvent être modulées avec des modulations BPSK, QPSK, QAM16 ou QAM 64. Dessinez le diagramme de constellation de ces quatre modulations.*
4) *Expliquez pourquoi quand le débit augmente, la modulation utilisée comporte un nombre plus grand de symboles.*
5) *Expliquez pourquoi quand le débit augmente, la portée diminue.*

6) *Quelle est la bande passante théorique minimale d'un canal laissant passer un signal de débit binaire de 600 Mbit/s transmis avec une modulation PSK8 ?*

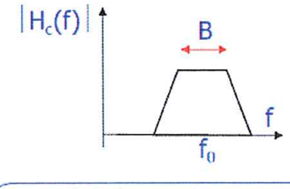
7) *Qu'appelle-t-on une 'technique d'accès' dans un système de télécommunication ? Donnez un exemple de technique d'accès en expliquant brièvement son principe.*

8) Le diagramme de l'œil de la figure suivante correspond au signal simulé au niveau d'un récepteur juste avant l'échantillonnage, sur les voies I et Q. Y a-t-il de l'interférence entre symboles dans ce système ? Justifiez votre réponse.



Annexe : extrait du transparent CN4-25

Transmission avec modulation



Propriété du canal :

Il est possible de ne pas avoir d'IES

si $B \geq B_{occ} = R\,(1+\alpha)$

$B_{min} = R$ ou $R_{max} = B$

Examen du Juillet 2014 — Session 2

Durée : 2h.

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

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

Exercice I

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

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

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.

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

Exercice II

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 lorsque $n \to +\infty$ du nombre d'opérations arithmétiques réalisées avec cet algorithme.

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

Exercice III

Etant donné une matrice symétrique définie positive $A \in M_n(\mathbb{R})$ et $b \in \mathbb{R}^n$, on souhaite résoudre le système linéaire

$$A\,x = b, \quad x \in \mathbb{R}^n, \tag{2}$$

en utilisant une méthode itérative. Pour cela on décompose la matrice A en $A = M - N$ avec M inversible, et on considère la suite $(x_k)_{k\geq 0}$ dans $\mathbb{R}^n$ définie par

$$M\,x_{k+1} = N\,x_k + b, \quad x_0 \in \mathbb{R}^n. \tag{3}$$

Dans la suite de l'exercice on munit $\mathbb{R}^n$ de la norme $\|x\| = \sqrt{x^t A x}$, et $M_n(\mathbb{R})$ de la norme induite associée.

1. Montrer que $M^T + N$ est symétrique.

2. Etant donné $v \in \mathbb{R}^n$ et $w = M^{-1}Av$, montrer les égalités

$$\|M^{-1}Nv\|^2 = \|v - w\|^2 = \|v\|^2 - w^T(M^T + N)w.$$

3. Montrer que si $M^T + N$ est symétrique définie positive alors $\|M^{-1}N\| < 1$.

4. En déduire que si $M^T + N$ est symétrique définie positive alors la suite (x_k) converge vers x.

5. On décompose la matrice A en $A = L + D + U$, où D est la matrice diagonale telle que $d_{ii} = a_{ii}$ et

$$L = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix}.$$

On considère la méthode SOR définie par

$$(D + \omega L)\,x_{k+1} = [(1-\omega)D - \omega U]\,x_k + \omega b, \quad x_0 \in \mathbb{R}^n \tag{4}$$

où $\omega > 0$ est le paramètre de relaxation. En utilisant la question **4**, montrer que si $0 < \omega < 2$ alors la méthode SOR converge.

ENSIMAG 1ère année — Juillet 2014

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 : 10 pts, Partie 2 : 10 pts.

Première partie

Une variété de souris a une probabilité connue de 10% de développer une maladie. On administre un nouveau traitement à 200 souris et on constate que 17 d'entre elles ont développé la maladie. On souhaite déterminer si le résultat de cette expérience permet de conclure que le traitement a modifié la probabilité de développer la maladie.

1. (a) Présenter le problème sous forme d'un test sur une proportion.

   (b) Calculer la p-valeur et conclure.

   (c) Donner la commande de R permettant de faire ce test.

2. (a) Présenter le problème sous forme d'un test du χ^2.

   (b) Donner un encadrement de la p-valeur et conclure.

   (c) Donner la commande de R permettant de faire ce test.

3. Montrer que $\forall \alpha \in [0,1], z_{1,\alpha} = u_\alpha^2$. En déduire que les 2 tests sont équivalents, c'est-à-dire que leurs régions critiques sont identiques.

Deuxième partie

On observe des réalisations $x_1, \dots, x_n$ de variables aléatoires $X_1, \dots, X_n$ indépendantes et de même loi de probabilité $Pu(1,c)$, à valeurs dans $[0,1]$, définie par la densité :

$$f(x;c) = cx^{c-1}, \text{ pour } x \in [0,1]$$

où c est un paramètre réel strictement positif.

1. Montrer que l'estimateur de maximum de vraisemblance de c est $\hat{c}_n = -\dfrac{n}{\sum_{i=1}^{n} \ln X_i}$.

2. Soit X une variable aléatoire de densité f. Calculer $E[X]$. En déduire l'estimateur de c par la méthode des moments.

3. Montrer que $Y = -\ln X$ est une variable aléatoire de loi exponentielle $\exp(c)$.

4. En déduire, en utilisant un résultat vu en TD, que $\hat{c}_n$ est un estimateur biaisé. Donner un estimateur sans biais de c.

5. Expliquer pourquoi $-2c \sum_{i=1}^{n} \ln X_i$ est une fonction pivotale pour c. En déduire un intervalle de confiance de seuil α pour c.

Ensimag 1re année **2013-2014**

Probabilités Appliquées
Session 2

Durée 1h30 - Une feuille manuscrite format A4 autorisée. Calculatrices non autorisées.

Exercice I - On considère une variable aléatoire entière, N, à valeurs dans $\{1, 2, 3\}$, telle que

$$\mathrm{P}(N = 1) = \frac{1}{2}, \quad \mathrm{P}(N = 2) = \frac{1}{3}, \quad \mathrm{P}(N = 3) = \frac{1}{6}.$$

Soit $X = U^{1/N}$, où U est une variable de loi uniforme sur $(0, 1)$ indépendante de N.

1. En utilisant la formule des probabilités totales, déterminer la fonction de répartition de la loi de X.
2. En déduire la densité de la loi de X.
3. Calculer l'espérance de X.

Exercice II - Soit N_1 et N_2 deux variables aléatoires indépendantes de loi géométrique de paramètre p.

1. Rappeler la définition de la loi géométrique. Décrire un exemple d'expérience dans laquelle cette loi apparaît.
2. On considère la somme $S = N_1 + N_2$. En utilisant la formule des probabilités totales, montrer que
$$\mathrm{P}(S = k) = (k-1)p^2(1-p)^{k-2}, \quad k \geq 2.$$
3. Donner, sans calcul, l'espérance et la variance de la variable aléatoire S.
4. On considère n variables aléatoires indépendantes de loi géométrique de paramètre p. En utilisant un raisonnement par récurrence, montrer que la loi de la somme, S_n, est donnée par
$$\mathrm{P}(S_n = k) = \frac{(k-1)!}{(n-1)!(k-n)!} p^n (1-p)^{k-n}, \quad k \geq n.$$

Examen de RECHERCHE OPERATIONNELLE

Ensimag 1ère année

30 juin 2014

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 numéro de votre groupe.

Exercice 1 : Une ferme comporte deux parcelles P_{20} et P_{40} (de 20 et de 40 hectares respectivement) où six sortes de céréales (numérotées de 1 à 6) peuvent être cultivées (simultanément, en occupant une partie d'une parcelle). Pour cultiver chaque céréale on a besoin des surfaces (en hectares) et des quantités d'eau (en m^3). Le volume total d'eau disponible est de 400000 m^3. Le profit de chaque céréale est différent. On cherche à rendre maximum le profit total, tout en respectant les diverses contraintes.

Le tableau suivant contient toutes les données numériques nécessaires pour la formulation :

Céréales	1	2	3	4	5	6
Profit (euro/tonne)	24	31	14	18	61	47
Surface nécessaire dans parcelle P_{20} (hectare/tonne)	0,10	0,15	0,10	0,08	0,25	0,20
Surface nécessaire dans parcelle P_{40} (hectare/tonne)	0,10	0,17	0,12	0,10	0,30	0,17
Eau (m^3/tonne)	60	80	50	70	107	90

Pour faciliter la lecture du tableau, prenons un exemple : pour faire pousser une tonne de céréale 2, qui apporte 31 euros, on a besoin de 80 m^3 d'eau et de 0,15 hectares de parcelle P_{20} ou de 0,17 hectares de parcelle P_{40}.

Formaliser ce problème comme un programme linéaire. La solution n'est pas demandée. Précisez votre choix de variables et expliquez les contraintes.

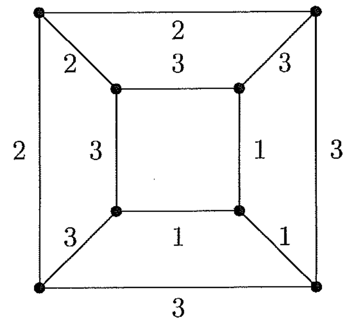
Exercice 2 : Considérons le programme linéaire suivant :

$$
\begin{array}{l}
\max \quad 5x_1 + 1x_2 \\
\text{sous les contraintes} \\
1x_1 \qquad\quad \geq 5 \\
4x_1 + 1x_2 \leq 25 \\
x_1 \geq 0, x_2 \geq 0.
\end{array}
$$

(a) Ecrire le dual de ce programme linéaire.
(b) Appliquer les Phases I et II de l'algorithme du simplexe pour trouver une solution optimale $(\overline{x}_1, \overline{x}_2)$.
(c) En utilisant $(\overline{x}_1, \overline{x}_2)$ et les écarts complémentaires trouver une solution optimale du dual.

Exercice 3 : On considère le graphe G avec fonction de coût sur les arêtes indiqué sur la figure. *Justifier bien toutes les réponses.*

(a) Montrer que G est un graphe biparti.
(b) Exécuter l'algorithme de Kruskal pour trouver un arbre couvrant F de G de coût minimum.
(c) Montrer que l'arbre couvrant F trouvé précédemment ne possède pas de couplage parfait.
(d) Montrer que le graphe G ne possède pas d'ensemble transversal de taille 3.



Exercice 4 : Organisation d'un départ en pique-nique :

"Préparer le panier" consiste à rassembler les boissons, les sandwiches et les fruits dans le panier. "Charger la voiture" consiste à mettre le panier préparé, les couvertures et les affaires de sport dans le coffre de la voiture. La station essence se trouvant à mi-chemin entre le domicile et le lieu du pique-nique, la tâche 8 devra être effectuée après les sept premières, et bien sûr avant la neuvième.

	Tâche	durée en minutes
1	Préparer le thé et le café	15
2	Préparer les sandwiches	10
3	Préparer les fruits	2
4	Préparer le panier	3
5	Prendre des couvertures	2
6	Prendre les affaires de sport	7
7	Charger la voiture	4
8	Mettre de l'essence	6
9	Aller au pique-nique	20

(a) Sachant que tout le monde a rendez-vous à $9h00$ sur le lieu du pique-nique, à quelle heure faut-il commencer à effectuer toutes ces tâches pour ne pas être en retard ? Donner, pour chaque tâche, les dates au plut tôt, au plus tard et les marges. Quelles sont les tâches critiques ?
(b) Si les tâches 5 et 6 étaient effectuées la veille, est-ce que le temps minimal nécessaire aux préparatifs serait changé ?
(c) L'organisateur a oublié d'acheter les fruits. Quelqu'un se propose d'aller les chercher au magasin le plus proche dès son ouverture à $8h15$, pendant que les autres effectueront les autres tâches, mais il devra revenir pour préparer ces fruits. Ce magasin se situe à 10 minutes du domicile. Quelle est l'influence de ce contretemps sur l'heure d'arrivée sur le lieu du pique-nique ? Et sur les tâches critiques ?

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

Mercredi 2 juillet 2014, 8h30-10h30.

POLYCOPIE, NOTES DE COURS ET CALCULATRICES AUTORISES

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

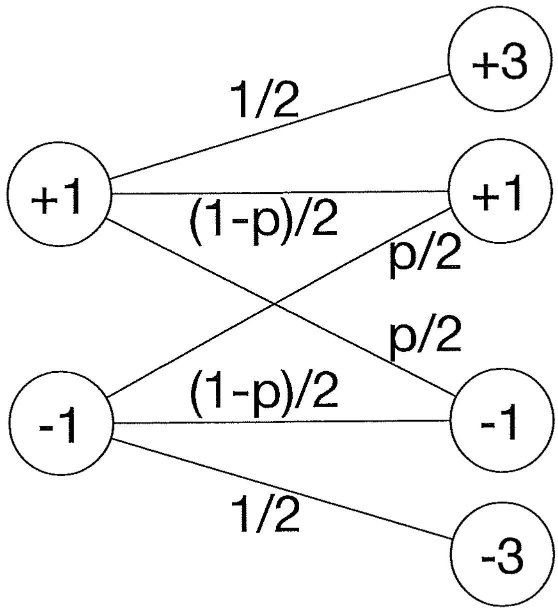
1 Questions générales (4 points)

1. On considère une variable aléatoire de Bernoulli (binaire) :
   (a) Dans quels cas porte-t-elle exactement 1 bit d'information ? (0.5 point).
   (b) Dans quel cas ne porte-t-elle aucune information ? (0.5 point)
2. On considère un « octet », 8-uplet (X1,...,X8) de variables aléatoires binaires à valeurs dans 0 ; 1.
   (a) Si l'octet porte 8 bits d'information, quelle est la loi de la variable X7 ? (0.5 point)
   (b) Si toutes les variables sont liées par X1=... =X8, donner un encadrement de l'information portée par l'octet ? (0.5 point)
3. On considère une source simple à 4 états, notés A, T, G et C, de probabilités respectives 1/8 ; 1/8 ; 1/4 ; 1/2.
   (a) Quelle est la longueur moyenne minimale des mots pour un codage binaire (compression sans perte) de cette source ? (0.5 point)
   (b) Pour cette source, proposer un code de Huffman binaire (alphabet du code 0 ; 1) et donner la longueur moyenne de ses mots. (1 point)
   (c) Commenter la relation entre les réponses aux 2 questions précédentes. (0.5 point)

2 Capacité d'un canal (9 points)

On considère le canal sans mémoire représenté par le diagramme de transition ci-contre.

On note les probabilités en entrée $P_X(+1), P_X(-1)$ et $P_Y(+1), P_Y(-1), P_Y(+3), P_Y(-3)$ celles de sortie.



1. (a) Ecrire la matrice de transition de ce canal (2 points)
   (b) Est-il uniforme par rapport à son entrée ? (0,5 point)
   (c) Est-il uniforme par rapport à sa sortie ? (0,5 point)
2. Lorsque la loi d'entrée est uniforme, calculer la loi de la sortie ? (1 point)
3. Montrer qu'il existe une unique loi d'entrée pour laquelle la sortie suit une loi uniforme ? (1 point)
4. Exprimer $H(Y|X)$ (l'entropie conditionnelle de sa sortie Y par rapport à son entrée X) (1 point)
5. En considérant le nombre d'entrées et le nombre de sorties de ce canal, donner un encadrement de sa capacité. (1 point)
6. Calculer la capacité de ce canal. Tracer cette capacité en fonction de p. (2 points)

3 Code de Hamming (7 points)

On considère un code de Hamming C de rendement $11/15$

1. Ce code est-il un bon candidat sur un canal de capacité 0.5 bit ? (justifier en une ligne) (1 point)
2. Pour ce code C, donner une matrice de parité H sous forme systématique. (1 point)
3. Calculer la distance minimale de C. (2 points)
4. Quelle est la capacité de correction du code C. (1 point)
5. Quelle est la capacité de détection du code C. (1 point)
6. A quoi correspond le syndrome lorsqu'il y a 1 erreur de transmission ? (0.5 point)
7. A quoi correspond le syndrome lorsqu'il y a 2 erreurs de transmission ? (0.5 point)

Théorie des langages 1 – Rattrapages

Durée : 2h.
Documents : tous documents autorisés.

Nb : le barême est donné à titre indicatif; la clarté de la rédaction et la propreté des preuves seront prises en compte dans la notation.

Exercice 1 - Transformations d'automates (4 points)

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

δ	a	b	c
q_0	q_0, q_1	q_0, q_2	q_3
q_1	q_0	-	-
q_2	-	q_0	-
q_3	q_3, q_4	q_3	q_6
q_4	-	-	q_5
q_5	q_6	q_6	q_6
q_6	q_6	q_6	q_6

▷ **Question 1** (4 points) Construire un automate minimal équivalent à A, en développant les étapes de la construction. Caractériser en français le langage reconnu par cet automate.

Exercice 2 - Types d'un programme (9 points)

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

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

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

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

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

▷ **Question 1** (3 points) Donner l'arbre de dérivation de

array(integer range cst .. cst) of character

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

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

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

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

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

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

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

et contient les règles supplémentaires suivantes :

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

Les règles correspondant au non-terminal `decl_type` deviennent

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

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

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

Exercice 3 - Manipulations de grammaires (7 points)

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

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

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

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

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

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

Et on pose :

$$\begin{aligned} E_S &= \{w \mid w \in V_T^+ \wedge \exists n \in \mathbb{N} : Val(w) = 3 \times n\} \\ E_A &= \{w \mid w \in V_T^+ \wedge \exists n \in \mathbb{N} : Val(w) = 3 \times n + 1\} \\ E_B &= \{w \mid w \in V_T^+ \wedge \exists n \in \mathbb{N} : Val(w) = 3 \times n + 2\} \end{aligned}$$

À quels ensembles appartiennent c_1, c_2 et c_3 ?

▷ **Question 4** (2 points)

Dans cette question w désigne un mot de V_T^*.

a) Montrer que si $S \Longrightarrow^1 w$ alors $w \in E_S$.

b) Montrer que si $A \Longrightarrow^1 w$ alors $w \in E_A$.

c) Montrer que si $S \Longrightarrow^{n+1} w$ alors $w \in E_S$, en utilisant l'hypothèse d'induction suivante :

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

Théorie des langages 2

Durée : 1h30.
Documents : tous documents autorisés.

Exercice 1 (5 points) Calculabilité

▷ **Question 1 (2 points)** On considère un vocabulaire Σ ayant au moins 2 éléments. Pour un mot w de Σ^* et un symbole a de Σ, on note $|w|_a$ le nombre d'occurrences de a dans w. Le langage $L = \{w \in \Sigma^* : \forall a \in \Sigma, |w|_a \bmod 2 = 0\}$ est-t-il récursif (décidable) ? ($n \bmod 2$ vaut 0 si n est pair, 1 sinon). On justifiera précisément la réponse.

▷ **Question 2 (3 points)** Les affirmations suivantes sont-elles vraies ou fausses ? On justifiera précisément les réponses. Les 3 premières questions ont pour préliminaire : pour tout vocabulaire Σ , $\forall L \in \Sigma^*$, $\forall M \in \Sigma^*$.

2.1 si L et M sont non récursifs, alors $L \cup M$ est non récursif.

2.2 si $L \subseteq M$ et M est non récursivement énumérable, alors L est non récursivement énumérable.

2.3 si $L \subseteq M$ et M est récursivement énumérable, alors L est récursivement énumérable.

2.4 une réduction est soit injective, soit surjective.

Exercice 2 (11 points) Langages hors-contexte et reconnaisseurs

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

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

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

▷ **Question 1 (1 point)**

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

▷ **Question 2 (3 points)**

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

▷ **Question 3 (3 points)**

Ecrire un analyseur LL(1) qui reconnaît les programmes et les commandes. On utilisera la fonction **lire_mot return token** avec **token** $= VT \cup \{\$\}$, où *$* représente le marqueur de fin de texte. On pourra utiliser les procédures **exp** et **affectation**, qui reconnaissent respectivement les expressions et les affectations, sans les écrire.

▷ **Question 4 (4 points)**

On veut étendre la notation pour permettre l'affectation simultanée d'un nombre non-nul quelconque de variables. Exemples :

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

- Proposer une nouvelle définition du non-terminal **affectation**, sous forme LL(1), prenant en compte cette extension. **La grammaire proposée devra garantir qu'il y a autant d'éléments de chaque côté du signe** :=. On étend le vocabulaire terminal à l'aide du symbole **,**.
- On ajoute la contrainte suivante pour l'affectation simultanée : **les identificateurs apparaissant en partie gauche du signe := doivent être distincts deux à deux.** Comment peut-on décrire une telle contrainte ?

Exercice 3 (4 points) Analyse LL(1)

On s'intéresse ici à composer 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 : $(V_T, \{S\} \cup V_{N_1} \cup V_{N_2}, S, \{S \rightarrow S_1S_2\} \cup R_1 \cup R_2)$, 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) :

$$\mathit{Directeur}(A \rightarrow w) =$$
$$\{\mathbf{x} \in (V_T \cup \{\$\}) : \exists w_1, w_2, w_3 \ . \ (S\$ \Rightarrow^* w_1 A w_2 \Rightarrow w_1 w w_2 \Rightarrow^* w_1 \mathbf{x} w_3)\}$$

1. Définir le langage $L(G)$ en fonction des mots dans $L(G_1)$ et $LG_2)$.
2. Donner un contre-exemple montrant que G n'est pas toujours LL(1).
3. Donner des conditions sur les ensembles Directeur déjà calculés pour G_1 et G_2 permettant de garantir que G est LL(1). Ces conditions sont-elles nécessaires ?