Examen – LI101 – janvier 2005

Aucun document ni machine électronique n'est permis à l'exception de la carte de référence de Scheme. Les téléphones doivent être éteints et rangés dans les sacs.

L'examen dure deux heures. Ce sujet comporte 12 pages.

Les questions peuvent être résolues de façon indépendante. Il est possible, voire même utile, pour répondre à une question, d'utiliser les fonctions qui sont l'objet des questions précédentes.

Répondre sur la feuille même, dans les cadres appropriés. La taille des cadres suggère le nombre de lignes de la réponse attendue (utiliser le dos de la feuille précédente si la réponse déborde des cadres). Le barème (total sur 60) apparaissant dans chaque cadre n'est donné qu'à titre indicatif.

La clarté des réponses et la présentation des programmes seront appréciées. Sauf mention contraire, les fonctions qui apparaîtront dans vos réponses devront être accompagnées de leur spécification.

Ne pas désagrafer les feuilles.

Exercice 1

Ce problème s'intéresse à des listes de nombres.

Question 1.1 – Écrire la signature et la définition d'un prédicat, nommé `decroissante?`, prenant une liste de nombres et vérifiant si ces nombres décroissent (non nécessairement strictement).

```
(decroissante? '(10 5 1 -2 -4))  → #T
(decroissante? '(10 10 9))  → #T
(decroissante? '(10 8 9))  → #F
(decroissante? '())  → #T
```

[4/60]

Question 1.2 – Écrire la signature et la définition d'une fonction, nommée `liste-differences`, prenant une liste d'au moins deux nombres et calculant la liste de leurs différences. Si la liste de nombres est notée $(u_0\ u_1\ \ldots\ u_{n-1})$ alors la liste des différences est la liste $(u_0 - u_1\ \ u_1 - u_2\ \ldots\ u_{n-2} - u_{n-1})$.

```
(liste-differences '(10 5 1 -2 -4))  →  (5 4 3 2)
```

[4/60]

Section	Numéro d'anonymat

Question 1.3 – On dira qu'une liste de nombres rétrécit si elle est formée de nombres dont les différences diminuent (au sens large) en valeur absolue. Écrire la signature et la définition d'un prédicat, nommé `retrecissante?`, prenant une liste de nombres contenant au moins deux termes et vérifiant que cette liste rétrécit.

```
(retrecissante? '(1 2 3 4))  → #T
(retrecissante? '(10 5 1 -2 -4))  → #T
(retrecissante? '(10 12 9))  → #F
(retrecissante? '(10 11))  → #T
```

[3/60]

Exercice 2

Ce problème s'intéresse aussi à des listes de nombres.

Question 2.1 – Écrire une définition du prédicat, nommé `positive-negative?`, prenant une liste de nombres et vérifiant que cette liste est formée d'un nombre positif ou nul suivi d'un nombre strictement négatif suivi d'un nombre positif ou nul et ainsi de suite. Ainsi

```
(positive-negative? '())  → #T
(positive-negative? '(1))  → #T
(positive-negative? '(1 -2))  → #T
(positive-negative? '(1 2 3))  → #F
(positive-negative? '(1 -2 3))  → #T
(positive-negative? '(-1 2 -3))  → #F
(positive-negative? '(1 -2 3 -4 5))  → #T
```

Section Numéro d'anonymat

[4/60]

Question 2.2 – Écrire une définition du prédicat, nommé `alternee?`, prenant un nombre (que l'on qualifiera de *pivot*) et une liste de nombres et vérifiant que cette liste est formée de nombres alternativement égaux ou plus grands puis strictement plus petits que ce pivot. Ainsi

```
(alternee? 5 '(15 2 16 3 17))  → #T
(alternee? 5 '(1 15 2 16 3 17))  → #F
```

[4/60]

Question 2.3 – Écrire, à l'aide du prédicat `alternee?`, une nouvelle définition du prédicat `positive-negative?`.

[2/60]

Section	Numéro d'anonymat

Question 2.4 – Écrire une spécification et une définition de la fonctionnelle, nommée `alternativement?`, ayant la signature suivante.

`;;;` *alternativement?: (alpha -> bool) * LISTE[alpha] -> bool*

Cette fonctionnelle généralise les prédicats précédents. Cette fonctionnelle prend en arguments un prédicat et une liste. Elle vérifie que la liste comporte un terme satisfaisant le prédicat puis un terme ne le satisfaisant pas et ainsi de suite.

```
(alternativement? odd? '(1 2 3 4 5))  → #T
(alternativement? string? '("1" 2 "33" #T))  → #T
(alternativement? string? '("a" "bb"))  → #F
```

[5/60]

Question 2.5 – Écrire, à l'aide de la fonctionnelle `alternativement?`, une nouvelle définition du prédicat `positive-negative?`.

[2/60]

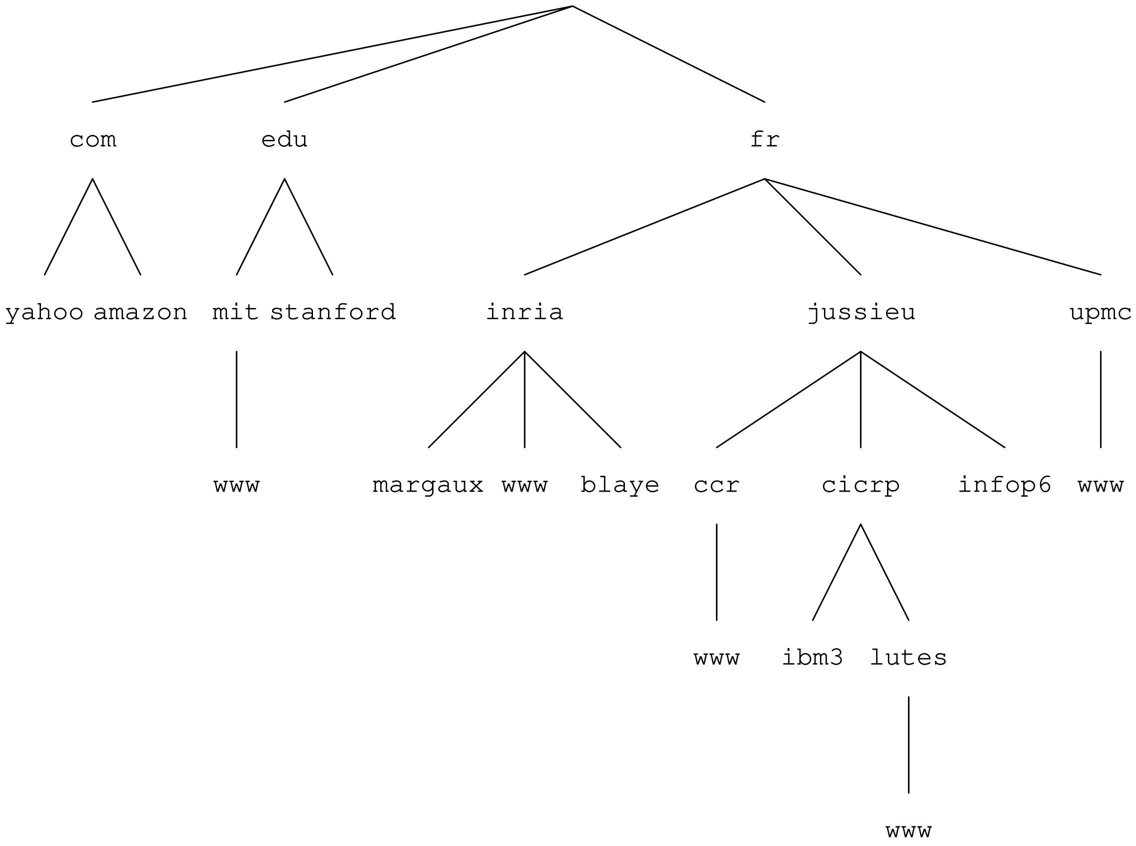
Question 2.6 – Écrire, à l'aide de la fonctionnelle `alternativement?`, une nouvelle définition du prédicat `alternee?`.

[5/60]

Exercice 3

Cet exercice travaille autour de l'arbre des noms de domaines et de machines d'Internet. Cet arbre que l'on nommera un « arbre Internet » sera représenté par un arbre général dont les étiquettes sont des chaînes de caractères, la racine a une étiquette qui est la chaîne vide. Voici l'arbre qui sera utilisé dans les exemples qui suivent et qui est produit par l'expression `(ai)`. Dans ce dessin, les étiquettes sont imprimées sans guillemets.

Section Numéro d'anonymat



L'arbre Internet a des nœuds qui peuvent être des noms de domaines ou des noms de machines. Un nom de domaine ou de machine comme `ccr.jussieu.fr` est représenté par le nœud obtenu en partant de la racine de l'arbre Internet et en suivant les arêtes menant à `fr`, `jussieu` puis `ccr`. Ce chemin suivi dans l'arbre Internet sera représenté par la liste `("fr" "jussieu" "ccr")`. Un nœud est un nom de domaine si sa forêt n'est pas vide (par exemple `com` ou `jussieu.fr`), un nœud est une machine si sa forêt est vide (par exemple, `www.ccr.jussieu.fr` ou `yahoo.com`).

On pourra, sans le redéfinir, utiliser le prédicat `ag-feuille?` vu en TD pour toutes les questions qui suivent.

Question 3.1 – Écrire la signature et une définition de la fonction, nommée `nombre-de-fils-sous-racine`, qui prend un arbre Internet et calcule le nombre de sous-arbres issus de la racine. Ainsi

`(nombre-de-fils-sous-racine (ai))` $\rightarrow$ `3`

[3/60]

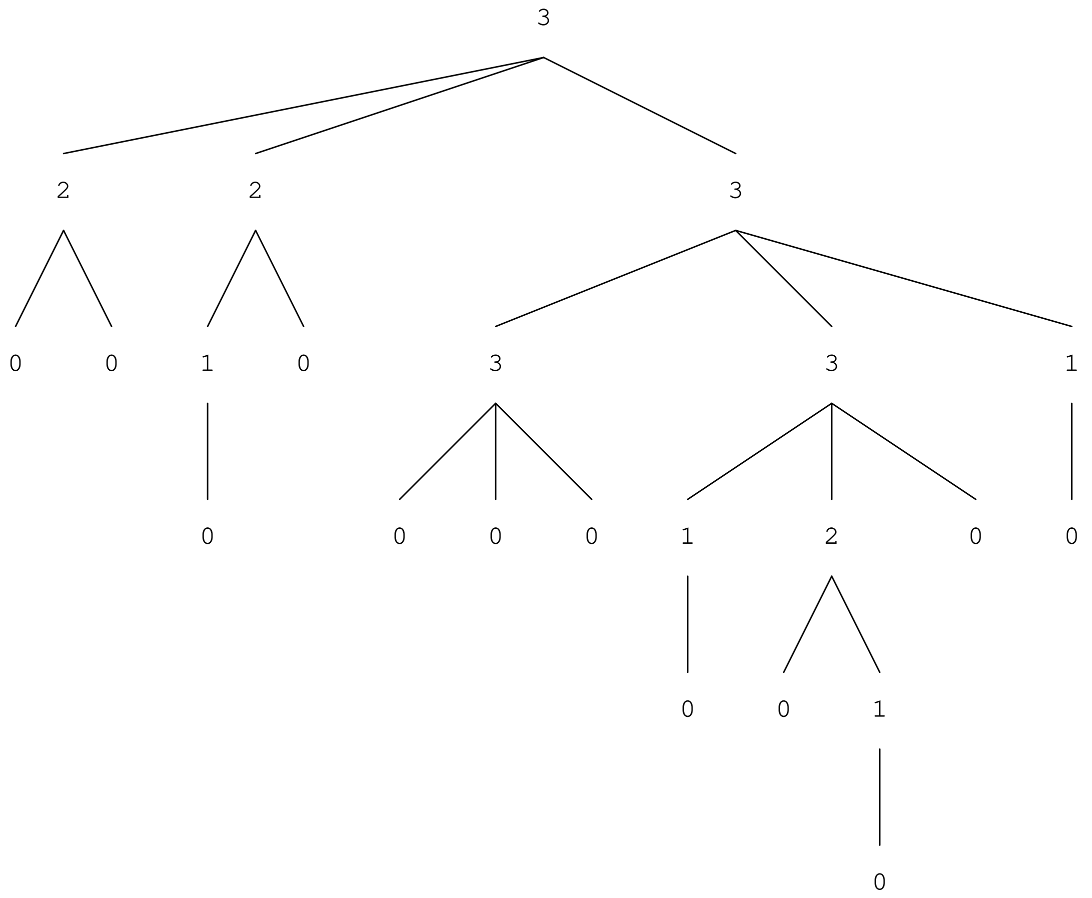
Section Numéro d'anonymat

Question 3.2 – Écrire la signature et une définition de la fonction, nommée `nombre-total-de-fils`, qui prend un arbre Internet et calcule le nombre de descendants de la racine présents dans cet arbre (c'est-à-dire le nombre total de nœuds total de l'arbre diminué de 1) Ainsi

    (nombre-total-de-fils (ai)) → 22

[3/60]

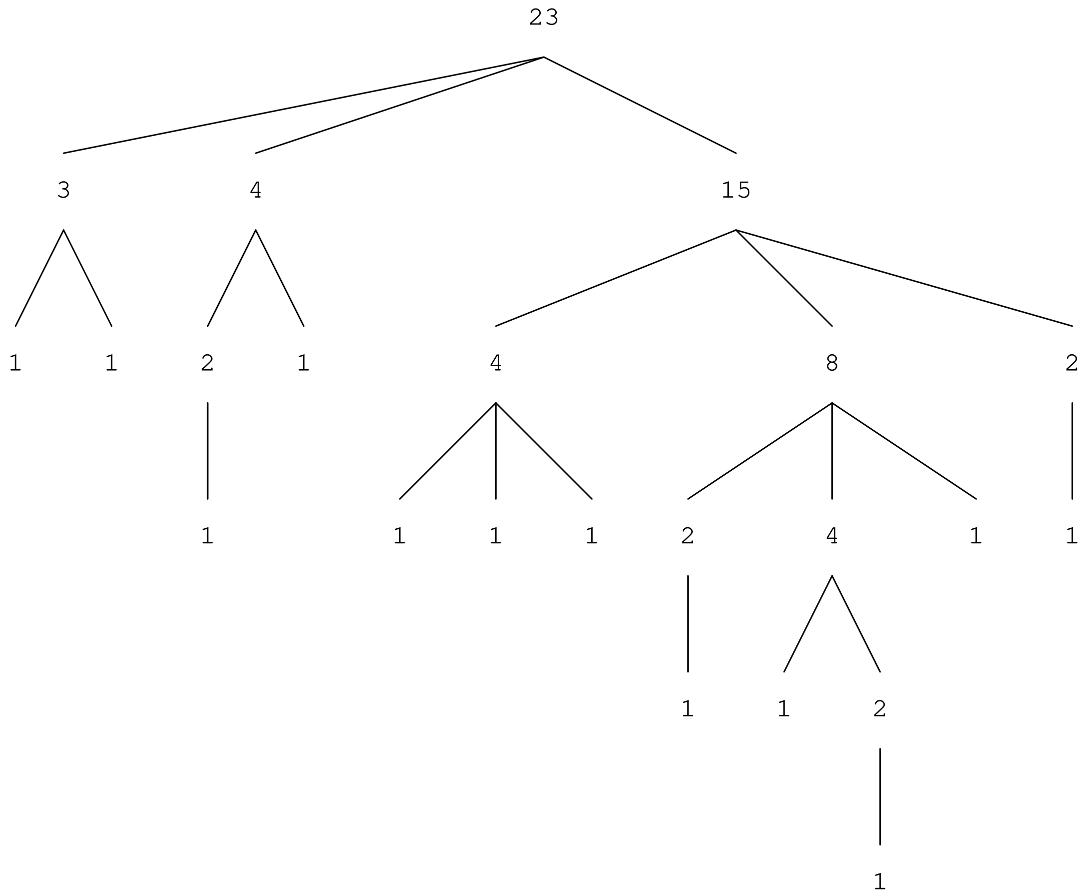
Question 3.3 – Écrire la signature et une définition de la fonction, nommée `arbre-des-nombres-de-fils`, transformant tout nœud de l'arbre Internet reçu en argument en un nouveau nœud dont l'étiquette est le nombre de fils qu'il possède. Ainsi l'application de cette fonction sur l'arbre produit par l'expression `(ai)` donne l'arbre suivant :



Section Numéro d'anonymat

[4/60]

Question 3.4 – Écrire une fonction, nommée `arbre-des-nombres-de-descendants`, transformant tout nœud de l'arbre reçu en argument en un nouveau nœud dont l'étiquette est le nombre de descendants (y compris lui-même) qu'il possède. Ainsi l'application de cette fonction sur l'arbre produit par l'expression `(ai)` donne l'arbre suivant :



Section　　　Numéro d'anonymat

[4/60]

Question 3.5 – Écrire la signature et une définition d'un prédicat, nommé `machine-existe?`, prenant un arbre Internet et une liste de chaînes de caractères représentant le nom d'une machine et vérifiant si cette machine est l'une des machines de cet arbre Internet. Le nom d'une machine, disons `www.upmc.fr`, sera représenté par une liste `("fr" "upmc" "www")`. Ainsi

```
(machine-existe? (ai) '("fr" "upmc" "www"))  → #T
(machine-existe? (ai) '("fr" "www"))  → #F
(machine-existe? (ai) '("upmc" "www"))  → #F
```

[4/60]

Question 3.6 – Écrire la signature et une définition d'une fonction, nommée `machines`, prenant un arbre Internet et calculant la liste des noms de toutes les machines présentes dans cet arbre.

```
(car (machines (ai)))  → "yahoo.com"
(cadr (machines (ai)))  → "amazon.com"
(caddr (machines (ai)))  → "www.mit.edu"
```

Section	Numéro d'anonymat

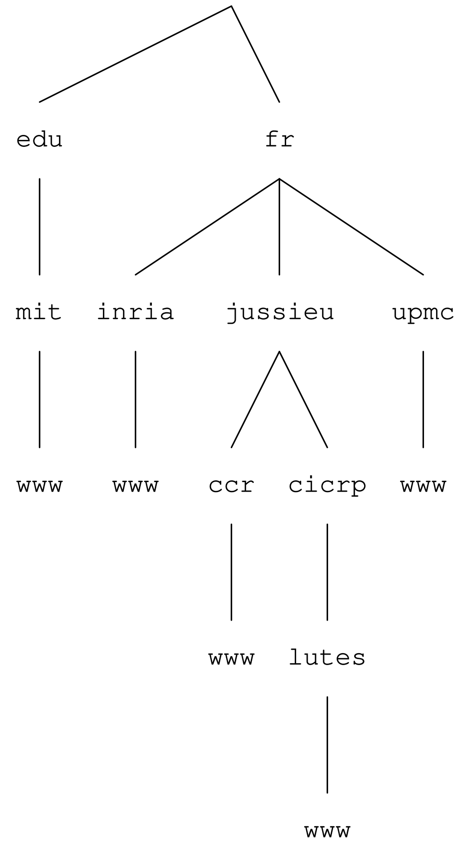
[4/60]

Question 3.7 – Les serveurs Web sont des machines dont, la plupart du temps, le nom débute par `www`. On souhaite déterminer les serveurs Web existant dans un arbre Internet. On souhaite donc élaguer l'arbre Internet de toute branche qui ne mène pas vers une feuille étiquetée `www`. Écrire la fonction, nommée `www-seules`, qui prend un arbre internet et calcule un tel nouvel arbre internet (on retournera Faux si l'arbre internet ne contient aucun serveur Web). Voici sa spécification :

```
;;; www-seules: ArbreInternet -> ArbreInternet + #f
;;; (www-seules ai) calcule un nouvel arbre Internet ne comportant que des nœuds
;;; d'étiquette "www" comme feuilles.
```

Ainsi l'application de cette fonction sur l'arbre produit par l'expression `(ai)` donne l'arbre suivant

Section Numéro d'anonymat



Section Numéro d'anonymat

[5/60]