Programmation applicative – L2
Première série de TP : Premiers programmes en Scheme

G.Artignan {artignan@lirmm.fr}   M.Lafourcade {lafourcade@lirmm.fr}
S.Daudé {sylvain.daude@univ-montp2.fr}   A.Chateau {chateau@lirmm.fr}
B.Paiva Lima da Silva {bplsilva@gmail.com}   C.Dony {dony@lirmm.fr}

1 Préambule

Cet enoncé concerne 2 séances de TP. L'évaluation des TPs se fera en deux phases :

- Pendant les TPs, votre présence et votre progression seront notées. Il y a 10 séances de TP, la note de présence est donc sur 10. Seules les absences justifiées seront acceptées. À la fin de chaque séance de TP, vous indiquerez sur la feuille de présence le numéro de la dernière question à laquelle vous avez entièrement répondu.
- Pendant la dernière séance, une évaluation de votre travail tout au long des séances sera effectuée, sous la forme d'un petit problème auquel il faudra répondre en vous servant des fonctions que vous aurez écrites durant les TPs. Nous noterons cette évaluation finale également sur 10.

En conséquence, il faudra sauvegarder précieusement votre travail au fur et à mesure des séances, pour pouvoir l'utiliser lors de l'examen final.

2 Variables et contextes d'évaluation

Exercice 1 *Dans la zone* interactions*, évaluez (dans l'ordre) les expressions suivantes en essayant de prévoir leur résultat :*

```
(+ taille 7)
(define taille 4)
taille
(puis2 taille)
taille
(define taillebis taille)
taillebis
(define taille 8)
taille
taillebis
(+ taille taillebis)
```

Voyons maintenant ce qui se passe dans le cas où plusieurs contextes d'évaluation sont inclus les uns dans les autres.

Exercice 2 *Pouvez-vous prévoir la valeur renvoyée suite à l'évaluation, dans la zone de définitions, de la dernière expression de cette séquence :*

```
(define x 5)
(define (f x) (* x x))
(f 10)
```

Vérifiez.

Exercice 3 *Même question pour la suite d'expressions :*

```
(define d 1)
(define (plusd x) (+ x d))
(plusd 10)
```

Exercice 4 *Même question si à la suite on ajoute les expressions :*

```
(define d 5)
(plusd 10)
```

Exercice 5 *Passons maintenant à la composition de fonctions. Quel est le résultat de l'évaluation si la séquence suivante est tapée :*

```
(define (h x) (* x x))
(define (g x) (+ 1 (h x)))
(g 10)
```

Se passe-t-il quelque chose si on inverse l'ordre de déclaration des fonctions `h` *et* `g` *?*

3 Les nombres

Scheme distingue les nombres entiers et les nombres flottants.

- `1` est un nombre entier.
- `1.0` est un nombre flottant.

En Scheme, les opérations sur des entiers renvoient des valeurs exactes alors que les opérations sur les flottants renvoient des valeurs approchées.

Exercice 6 *Evaluer les expressions suivantes :*

```
(/ 1 3)
(/ 1.0 3)
(/ 2 6)
(exact->inexact (/ 1 3))
(inexact->exact 0.5)
(/ 3 2)
```

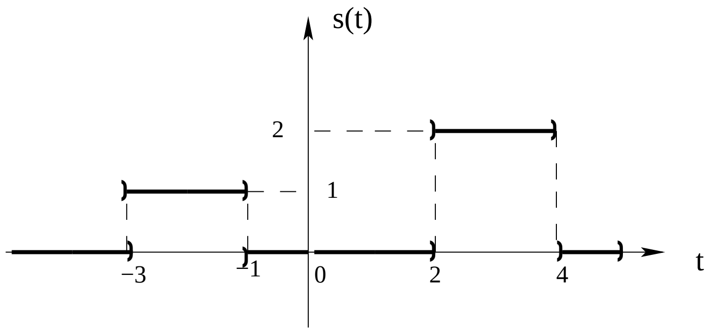
Remarque — On peut utiliser les nombres rationnels en scheme : $\frac{1}{2}$ se note `1/2`. Notez la différence entre `(/ 1 2)` qui est l'application de l'opérateur de division et `1/2` qui est un nombre rationnel.

4 Les conditions

Exercice 7 *Définir une fonction* `monabs` *qui renvoie la valeur absolue d'un nombre.*

Exercice 8 *Définir la fonction* `care-div` *qui gére le problème de la division par* 0 *(utiliser* `display` *pour renvoyer un message d'erreur).*

Exercice 9 *La figure suivante illustre un signal s fonction d'une variable t. Définir à l'aide de* `cond` *une fonction qui calcule la valeur du signal s en un point. Faire varier t de $-\infty$ à $+\infty$.*



5 Calcul de taux d'intérêt

Exercice 10 *Définir une fonction* `placement` *qui prend en argument un capital, un taux, et un nombre d'années. Le résultat de cette fonction doit être le montant que recevra quelqu'un qui a placé ce capital à ce taux pendant cette durée.*

Rappel : Si un placement est au taux d'intérêt t, alors un euro placé pendant n années vaut à la fin $(1+t)^n$ euros.

6 La tortue dans DrScheme

La tortue est un petit animal qui permet de faire des graphiques en la déplaçant au moyen de commandes *simples* (ce n'est qu'une tortue).

Avant de pouvoir utiliser la tortue, sélectionnez « Assez gros Scheme » dans le menu `langage` pour disposer du langage graphique. Entrez ensuite la ligne suivante dans la fenêtre des définitions (en haut) :

```
(require (lib "turtles.ss" "graphics"))
```

et cliquez ensuite sur le bouton `Execute` pour que votre choix soit pris en compte.

Entrez dans la fenêtre d'évaluation la commande :
`(turtles #t)`
Une fenêtre graphique apparaît avec une petite flèche indiquant la position actuelle de la tortue.
Celle-ci accepte les ordres suivants :
- `(turn x)` où x est un angle en degrés : tourne de x degrés dans le sens inverse des aiguilles d'une montre.
- `(turn/radians x)` idem mais cette fois l'angle x est indiqué en radians.
- `(move x)` avance de x unités.
- `(draw x)` avance de x unités en traçant un trait
- `(clear)` efface l'aire de jeu de la tortue

Pour fermer la fenêtre de la tortue, il suffit d'évaluer `(turtles #f)`.

Exercice 11 *Essayez de manipuler la tortue en lui demandant de faire deux tours de spirale, en changeant de direction par morceaux de 30 degrés et en avançant d'une unité de plus à chaque changement de direction.*

Exercice 12 *Dans la fenêtre de définitions, écrivez une fonction* `carré` *qui prend un argument* `lgr` *et fait dessiner à la tortue un carré de côté* `lgr`.

Exercice 13 *Écrivez maintenant une fonction* `hexagone`*, qui lui fait dessiner un hexagone dont la longueur des côtés est indiqué en paramètre.*

Exercice 14 *Généralisez à n côtés : n devient l'argument d'une fonction* `figure`*, que vous devez écrire.*

6.1 Calcul du jour de l'année

Pour savoir si le jour de sa naissance était par exemple un lundi ou un dimanche, on va écrire un programme SCHEME général qui répondra à ce type de questions. Pour cela, il est considéré une date de référence à partir de laquelle les fonctions suivantes feront références : le lundi 01 01 1900. On pourra aussi exécuter la commande Unix tt cal, par exemple : `cal 1900`.

Exercice 15 *Définir une fonction* `bissextile?` *qui prend en argument une année et qui répond #t si elle est bissextile. Pour écrire cette fonction, se rappeler qu'une année est bissextile si elle est divisible par 4 mais pas par 100, sauf si elle est multiple de 400.*

Exercice 16 *Définir une fonction* `nb-annees-bissextiles` *qui prend en argument une année et qui calcule le nombre d'années bissextiles entre 1900 et cette année (incluse). On vérifiera qu'il y en avait 24 en 1999 et 25 en 2001.*

Exercice 17 *Définir une fonction* `nb-jours-au-1-jan` *qui prend en argument un mois (un nombre de 1 à 12), et qui calcule le nombre de jours écoulés entre le premier janvier et le premier jour de ce mois. On supposera que l'année considérée n'est pas bissextile. Par exemple, le nombre de jours pour le mois de janvier (01) sera 0, pour le mois de février (02) sera 31, pour le mois de mars (03) sera 31+28, etc.*

Exercice 18 *Définir une fonction* `nb-jours` *qui prend en argument une date sous la forme de 3 entiers (par exemple, 21 05 2006 pour le 21 mai de l'année 2006). Cette fonction doit calculer le nombre de jours écoulés depuis le 01 01 1900 jusqu'à cette date. Par exemple, le nombre de jours pour le 5 01 1900 sera 4 jours. Vous devrez vous poser la question : au sein de l'année donnée en argument, a-t-on dépassé ou non la fin février ?*

Exercice 19 *Ecrire une fonction* `jour-de-semaine` *qui rend en valeur la bonne chaîne de caractères parmi "Lundi", "Mardi", ... "Dimanche" correspondant à la date passée en argument. Par exemple, l'évaluation de* `(jour-de-semaine 30 09 2008)` *donnera "Mardi".*

A Deux petits exercices supplémentaires pour les gourmands...

Exercice 20 *Calcul de volumes*
On veut calculer le volume d'une sphère de rayon r. Prenons d'abord une valeur de rayon, par exemple 10, en saisissant l'expression : `(define r 10)`*.*

Puis, donnez l'expression qui a pour valeur le volume de la sphère de rayon r. Modifiez la valeur de r, et réévaluez l'expression. De même, definissez une hauteur h et un rayon r et donner une expression qui a pour valeur le volume du cylindre de rayon r et de hauteur h.

Exercice 21 *Relativité*
La fonction `(einstein u v)` *: loi d'addition des vitesses en relativité restreinte, avec la vitesse de la lumière* $c \approx 300000$ $km.s^{-1}$ *:*

$$(u, v) \to \frac{u + v}{1 + \frac{uv}{c^2}}$$

Application numérique Rien ne vaut un problème concret : un voyageur court (assez vite ;-)) à 250000 $km.s^{-1}$ *dans le couloir d'un train, dans le sens de la marche. Et le train roule à* 270000 $km.s^{-1}$*. Quelle est la vitesse du voyageur par-rapport à la voie ?*