TDTP 1 : débuts en Scheme

Utilisation de DrRacket

En salle de TDTP, connectez-vous à l'aide de votre numéro d'étudiant. Vous pouvez lancer DrRacket *via* le menu *Démarrer>Tous les programmes>Racket*.

Configuration de DrRacket : dans le menu Langage, Sélectionnez le langage *Niveau débutant avec abréviations pour les listes*, puis dans *Montrer les détails*, choisissez *write* pour *Style d'impression des résultats* (si vous ne voyez pas le bouton OK, cachez les détails). Utilisez le bouton *Exécuter* pour valider votre choix de langage.

La partie inférieure de la fenêtre vous permet d'évaluer des expressions.

La partie supérieure de la fenêtre vous permet de définir des fonctions que vous pouvez enregistrer (bouton *Sauvegarder*) sur votre compte (l'extension ajoutée est `.rkt`). Prenez l'habitude de créer un fichier par TDTP.

Le bouton *Vérifier* vous permet de vérifier la syntaxe de votre fonction. Le bouton *Exécuter* compile la fonction, pour que vous puissiez ensuite la tester dans la partie inférieure de la fenêtre. Le bouton *Stopper* vous permet d'interrompre l'exécution d'une fonction. Le bouton *Pas* vous permet de dérouler pas-à-pas l'exécution d'une fonction.

Pour travailler chez vous

Vous pouvez installer DrRacket sur votre ordinateur depuis https://racket-lang.org/download/

Sinon vous pouvez utiliser le site en ligne https://replit.com/

Les sujets de TDTP sont aussi accessibles sur https://c5.univ-lyon1.fr/

À préparer avant la séance

1 Évaluation d'expressions en Scheme

Pour ce premier exercice, réfléchissez au résultat que les expressions devraient donner. Vous pouvez vérifier avec Racket/Replit le résultat, et nous discuterons en TDTP des cas qui posent question.

Donner la valeur retournée par les expressions Scheme écrites ci-dessous :

- (+ 2 6)
- (+ (* 2 3 5) (- 6 8))
- '(+ 2 6)
- '(1 2 3)
- (1 2 3)
- (= 0 (modulo 17 2))
- (and (> 21 45) (= 3 (/ 12 4)))
- (and (> 21 45) (= 3 (/ 12 0)))
- (and (= 3 (/ 12 0)) (> 21 45))
- (number? 'toto)
- (boolean? (number? 3))
- (boolean? (number? "abc"))

On suppose que les définitions suivantes ont été faites dans l'ordre donné :
(define moineau 5) (define condor 435) (define rapace 'condor) (define oiseau condor)

Trouver les résultats des évaluations suivantes :

- baleine
- 'baleine
- moineau
- 'moineau
- rapace
- (+ oiseau moineau)
- (+ rapace condor)
- (+ (eval rapace) condor)

2 Premières fonctions en Scheme

Définir en Scheme :

- une fonction qui retourne le double d'un nombre passé en paramètre.
- une fonction qui retourne la moyenne de deux nombres passés en paramètres.

À faire pendant la séance

TD Définir en Scheme :

- une fonction qui retourne un booléen spécifiant si le nombre passé en paramètre est positif.
- une fonction qui calcule la valeur absolue d'un nombre.
- une fonction qui retourne la mention pour une note donnée.
- une fonction qui calcule récursivement la somme des n premiers carrés.
  Dérouler pas à pas (somme-carres 3) qui doit donner 14.

TD ☒ Donner la spécification de la fonction Scheme ci-dessous :

```
(define mystere
 (lambda (n)
   (if (= n 0)
       0
       (if (= (modulo n 2) 0)
           (+ n (mystere (- n 1)))
           (mystere (- n 1))))))
```

TP Définir en Scheme :

- une fonction qui calcule le $n^{ième}$ terme de la suite de Fibonacci.
  Nous rappelons que cette suite est définie par : $u_0 = 1$, $u_1 = 1$, $u_n = u_{n-1} + u_{n-2}$.
- une fonction qui teste si un entier strictement positif est une puissance de 2.
  Dérouler pas à pas la fonction pour les valeurs 3 et 4.

Pour s'entrainer (exercices supplémentaires facultatifs)

Soit la suite de Syracuse, définie comme suit :

$u_{n+1} = 1 + 3u_n$ si u_n impair ;
$u_{n+1} = u_n / 2$ si u_n pair.

Quel que soit le nombre entier u_0 strictement positif, il semble que cette suite finit par engendrer le nombre 1. On voudrait le vérifier.

• Écrire une fonction booléenne `syracuse?` testant si la suite partant d'un nombre u_0 donné en paramètre finit par passer par 1.

• Dérouler `(syracuse? 5)` pas-à-pas.

• Définir une fonction qui teste que la fonction `syracuse?` retourne vrai pour **tous** les entiers positifs inférieurs à un nombre n. Jusqu'à quelle valeur de u_0 arrivez-vous à vérifier la conjecture ?

• Définir une fonction qui calcule à quel terme la fonction `syracuse?` arrive à 1 pour un u_0 donné (par exemple cette fonction doit retourner 7 pour u_0=3, 5 pour u_0=5 et 111 pour u_0=1000).