TD n°5

Algorithmique : de la dichotomie

I Réécrire en DrScheme les versions de *mul* et *mul_dicho* vues en cours

II En s'appuyant sur le modèle de la fonction multiplication vue en cours, écrire :

1) la fonction *puis_dicho* : N x N -> N d'élévation à la puissance

```
> (puis_dicho 2 4)
16
>(puis_dicho  3 5)
243
```

2) En utilisant l'outil de trace pour puis et puis_dicho : déterminer le nombre d'appels récursifs pour (puis 2 128) et (puis_dicho 2 128) Conclusion ?

Pour utiliser l'outil de trace
En début de fenêtre de définition, écrire :

```
(require (lib "trace.ss"))
```

Puis après évaluation de la fenêtre de définitions, taper dans la fenêtre d'évaluation :

```
(trace puis puis_dicho)
```
; remarque la fonction trace est une fonction n_aire

Pour arrêter la trace, il suffit de taper dans la fenêtre d'évaluation :

```
(untrace puis puis_dicho)
```

III Vive l'économie de pensée !

On peut remarquer que les fonctions mul_dicho et puis_dicho ont exactement la même forme. Ecrire une fonction travaille permettant de passer en argument la nature de l'opération à faire sur l'argument subissant l'opération essentielle.

On pourra alors définir : `(define (mul_dicho x y) (travaille x y + 0))`
et `(define (puis_dicho x y) (travaille x y * 1))`

*Idée : Bien réfléchir à ce que représente + (et resp *) et 0 (resp 1) dans la fonction mul_dicho (resp puis_dicho) initiale*

IV Soit la fonction somme –vue dans le TD4- qui permet de calculer la somme des nombres : $\sum_{1}^{n} i$

En voici une version dichotomique :

```
(define (somme_dicho n)
  (if (zero? n)
      0
      (+ (mul2 (somme_dicho (div2 n)))
         (somme_impair (if (even? n)(sub1 n) n)))))

;avec somme_impair  définie comme :
(define (somme_impair n)
  ; n est nécessairement impair
  (if (= n 1)  1  (+ n (somme_impair (sub1 (sub1 n))))))
```

1) Que fait la fonction somme_impair ?
2) en utilisant l'outil de trace pour somme et somme_dicho : déterminer le nombre d'appels récursifs pour (somme 10) et (somme_dicho 10) Conclusion ?
3) Faire tracer ensuite la fonction somme_impair (en plus des deux fonctions précédentes) ! déterminer le nombre d'appels récursifs pour (somme 10) et (somme_dicho 10) Conclusion ? La fonction somme_dicho est-elle vraiment économique ? Quel est le problème de la fonction somme_dicho ?

On se propose d'essayer de résoudre ce problème dans le TD suivant