

Interrogation écrite de Programmation Fonctionnelle

Durée : 2h. Documents autorisés : Notes personnelles de cours et de TD. Les fonctions sont à écrire en OCAML sans utiliser ses éléments impurs !

I Récursivité

1. (Puissance entière) Définir une fonction `puissance` qui, étant donné un nombre réel x et un entier relatif n , a pour résultat x^n . Vous devez donner une définition récursive en utilisant et en complétant les équations suivantes :

$$\begin{aligned}x^n &= x^{\frac{n}{2}} \times x^{\frac{n}{2}} & \text{si } n > 0 \text{ et } n \text{ pair,} \\x^n &= \frac{1}{x^{-n}} & \text{si } n < 0, \\x^n &= 1 & \text{si } n = 0.\end{aligned}$$

Exemples :

`puissance 4. 2 = 16.`

`puissance 4. (-2) = 0.0625`

2. (Schéma de Horner) On considère les polynômes à coefficients réels :

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

Définir la fonction `horner` qui, étant donnés la liste $[a_0 ; a_1 ; \dots ; a_{n-1} ; a_n]$ des coefficients d'un tel polynôme et un nombre réel x , retourne la valeur ($P(x)$) du polynôme en x calculée selon le schéma de Horner :

$$P(x) = ((\dots (a_n x + a_{n-1})x + \dots)x + a_1)x + a_0$$

II Algorithme de typage

Dérouler l'algorithme vu en Cours et en TD pour trouver le type de la fonction `f` définie par : `let f = function x -> function y -> (x y y)+y;;`
(On remplira 3 colonnes : une pour l'ensemble \mathcal{H} des hypothèses, une pour l'ensemble \mathcal{E} des équations et une pour l'ensemble \mathcal{T} des expressions à typer).

III Types algébriques

On considère des dessins en noir sur fond blanc obtenus en utilisant deux figures géométriques élémentaires du plan cartésien : les carrés pleins et les disques. Un exemple de tel dessin est donné ci-dessous à gauche (ce dessin est composé de huit figures : quatre disques et quatre carrés comme on peut le voir à droite). Dans la suite, on définit un dessin par une liste de figures. Dans un tel dessin, une figure peut éventuellement être cachée parce qu'elle est incluse dans une autre figure plus grande. On définit un carré par son centre et son coté, un disque par son centre et son rayon.



1. Définir le type `figure` : carré plein ou disque. On utilisera deux constructeurs : `Carre` et `Disque`.
2. Supposons que l'on dispose d'une fonction `distance` à 2 arguments qui donne la distance entre 2 points du plan. Donner le type de cette fonction. Définir une fonction `inclus` qui, étant données deux figures, retourne vrai (`true`) dans le cas où les deux figures sont des disques et l'un est inclus dans l'autre, et faux (`false`) sinon.
3. Supposons que l'on dispose d'une version plus générale de la fonction `inclus` qui, étant données deux figures quelconques, détermine si la première est incluse dans la deuxième. Donner le type de cette fonction. Définir une fonction `simplifie` qui étant donnée une liste de figures, disons `d1` (de type `figure list`), définissant un certain dessin, retourne une liste de figures, disons `d2`, définissant le même dessin et tel qu'aucune figure de `d2` n'est incluse dans une autre figure de `d2`.