Interrogation écrite 2

INF 201 — IMA1— 16/02/2022 — 25 minutes

Exercice 1. (/3) On définit la fonction sinus cardinal de la façon suivante:

$$\begin{array}{rcl} \text{sinc} & : & \mathbb{R} \to \mathbb{R} \\ \text{sinc}(x) & = & \begin{cases} 1 \quad \text{si} \quad x = 0 \\ \frac{\sin(x)}{x} \quad \text{sinon} \end{cases} \end{array}$$

Proposer une implémentation de sinus cardinal en OCaml **qui n'utilise pas** de structure conditionelle `if ...` mais un **filtrage par cas** à l'aide de l'instruction `match`:

Exercice 2. (/3) Donner les signatures des fonctions suivantes:

```
let f (x,y) z = x || y || z;;

let g a = a^".";;

let h b v = if b then v else 0.;;
```

Exercice 3. (/6) Dans cet exercice on souhaite proposer une extension de la droite réelle en incluant les deux infinis, on parle de **droite réelle achevée**, mathématiquement on écrit:

$$\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$$

Question 1. (/2) Proposer un type OCaml `reel_etendu` qui représente un élément appartenant à $\overline{\mathbb{R}}$.

Question 2. (/4) Définir l'addition `add_etendue` pour le type `reel_etendu` en utilisant **l'analyse par cas**. On prendra bien garde de définir l'addition entre infinis et réels de façon cohérente. Par exemple, $+\infty - \infty$ devra lever une exception à l'aide de l'instruction `failwith "erreur"`. *Indication: il y a 9 cas à traiter.*

Problème. (/11) Le but de ce problème est de proposer une gestion des nombres complexes en OCaml:

```
# let  (x:complexe) = (1.,1.);;
val x : complexe = (1., 1.)
```

```
# add (1.,3.) (-2.,0.);;
- : complexe = (-1., 3.)
# mul (1.,1.) (-2.,1.);;
- : complexe = (-3., -1.)
```

Question 3. (/1) Proposer un type `complexe` qui permet la gestion d'un nombre appartenant à $\mathbb{C}$.

Question 4. (/1) Écrire deux fonctions `re` : $\mathbb{C} \to \mathbb{R}$ et `im`: $\mathbb{C} \to \mathbb{R}$ qui donnent les parties réelles et imaginaires d'un nombre complexe z.

Question 5. (/1) Écrire une fonction `conj` : $\mathbb{C} \to \mathbb{C}$ qui a tout z associe $\bar{z}$.

Question 6. (/1) Proposer une fonction `add` : $\mathbb{C} \to \mathbb{C}$ qui permet d'ajouter deux nombres complexes.

Question 7. (/2) Proposer une fonction `mul` : $\mathbb{C} \to \mathbb{C}$ qui permet de multiplier deux nombres complexes. (Remarquer que pour $x, y \in \mathbb{C}$, alors $x\,y = (a + i\,b)(c + i\,d) = a\,c - b\,d + i\,(a\,d + b\,c)$)

Question 8. (/2) Proposer une fonction `norm` : $\mathbb{C} \to \mathbb{R}$ qui calcule la norme d'un complexe. On pourra utiliser la fonction `sqrt` définit en OCaml:

```
val sqrt:  float -> float = <fun>
```

Question 9. (/3) Proposer une fonction `arg` : $\mathbb{C}^* \to \mathbb{R}$ qui calcule l'argument d'un complexe. On prendra bien garde à la définition de l'argument qui est pour un complexe non nul:

$$\arg(z) \;=\; \begin{cases} 2\arctan\left(\frac{\Im(z)}{\Re(z)+|z|}\right) & \text{si} \quad z \notin \mathbb{R}_- \\ \pi & \text{si} \quad z \in \mathbb{R}_-^* \end{cases}$$

On pourra utiliser la fonction `atan` définit en OCaml et on n'hésitera pas à réutiliser des fonctions précedemment définies.

```
val atan:  float -> float = <fun>
```