

Les Fonctions

1. Expressions fonctionnelles (fonctions anonymes).

Définition : Une expression fonctionnelle est constituée d'un (ou plusieurs) *paramètre* et d'un *corps*. Elle est introduite par les mots réservés *function* (pour les fonctions à un seul argument) ou *fun* (pour les fonctions à plusieurs arguments).

a/ Fonction anonyme à un seul argument :

Syntaxe : $\text{function } p \rightarrow \text{expr} ;;$

Exemple :

la fonction qui élève au carré son argument s'écrit en CAML :

```
# function x -> x*x ;;
- : int -> int = <fun>
```

L'application d'une fonction à un argument s'écrit comme la fonction suivie par l'argument.

```
# (function x -> x * x) 5 ;;
- : int = 25
```

b/ Fonction anonyme à plusieurs arguments :

Syntaxe : $\text{fun } p_1 \dots p_n \rightarrow \text{expr} ;;$

Exemple :

```
# fun x y -> 3*x + y ;;
-: int -> int -> int = <fun>
```

L'application d'une fonction à plusieurs arguments s'écrit comme la fonction suivie par les arguments.

```
##(fun x y -> 3*x + y) 4 5 ;;
-: int = 17
```

2. Définitions de fonctions

Comme dans le cas des valeurs simples, il est possible d'assigner un nom à une fonction.

Syntaxe : $\text{let } \text{nom} = \text{expr_fonc} ;;$

Où *nom* est le nom de la fonction et *expr_fonc* est une expression fonctionnelle (*function* ou *fun*)

TP 2 CaML Light

Comme le montrent les exemples suivants (définitions et applications):

```
# let carre = function x -> x*x ;;
carre : int -> int = <fun>
# carre 5 ;;
- : int = 25
```

```
# let f = fun x y -> 3 * x + y ;;
f : int -> int -> int = <fun>
# f 2 4;;
- : int = 10
```

Simplification de la syntaxe:

Pour simplifier l'écriture, la syntaxe suivante est acceptée pour la définition d'une fonction d'arité n (avec $n \geq 1$).

Syntaxe : $\text{let } nom\ p_1 \dots p_n = expr \ ;;$

Exemple : les deux fonctions précédentes peuvent être définies comme suit :

```
# let carre x = x*x ;;
carre : int -> int = <fun>
```

```
# let f x y = 3 * x + y ;;
f : int -> int -> int = <fun>
```

3. Ecrire en langage CAML les fonctions suivantes

- Succ qui calcule le successeur d'un entier.
- Pred qui calcule le prédécesseur d'un entier.
- Sum qui calcule la somme de 2 entiers.
- Max qui calcule le maximum de 2 réels.
- Max3 qui calcule le maximum de 3 réels de 2 façons différentes (Sans utiliser Max, puis en utilisant Max).
- MinMax qui donne le min et le max en même temps de 2 entiers.
- Carre qui calcule le carré d'un entier.
- SCarre qui calcule la somme des carrés de 2 entiers (en utilisant la fonction Carre).
- ValAbs qui calcule la valeur absolue d'un entier.
- Abs qui calcule la fonction : $Abs(x, y) = |x - y|$.
- Surf qui calcule la surface d'un cercle de rayon r ($\pi = 3.14$).
- Pair qui retourne vrai si son argument est un entier pair, faux sinon.
- Prem qui retourne vrai si son argument est un entier premier, faux sinon.