Travaux Pratiques de Logique n°4

Licence d'informatique

Exemple de fonctions

▸ **Exercice 1.** Écrire une fonction `sum` qui calcule la somme de deux entiers

▸ **Exercice 2.** Écrire une fonction `incr` qui incrémente de 1 un entier en utilisant la fonction `sum`. Comparer avec `let g = sum 1`.

▸ **Exercice 3.** Écrire une fonction `xor` qui calcule le ou exclusif de deux booléens.

▸ **Exercice 4.** Écrire une fonction `fact` qui calcule la factorielle d'un entier.

▸ **Exercice 5.** Écrire une fonction `comp` telle que `comp f g` soit la composée $f \circ g$ des fonctions f et g. Observer le type donné.

▸ **Exercice 6.** Écrire la fonction `iter` qui calcule l'itérée d'une fonction. On devra avoir `iter f n x` vaut `f(f ... (f x)...)` (n fois).

Système de typage

▸ **Exercice 7.** Donner des fonctions Caml qui produisent les types suivants :

1. `int -> int -> int`
2. `(int -> int) -> int`
3. `int -> int -> int -> int`
4. `int -> (int -> int) -> int`
5. `(int -> int -> int) -> int`
6. `(int -> int) -> int -> int`

▸ **Exercice 8.** Déterminer les types des fonctions suivantes, sans les évaluer sur machine :

1. `let h (f,g) = function x -> f(g x);;`
2. `let i (a,b,c) = function x -> (a b (c * x));;`
3. `let j a b c = if (a>1 || c) then (b c a) else (a,c);;`

▸ **Exercice 9.** Ecrire une fonction `pgcd` qui calcule le plus grand diviseur commun de deux entiers. Définir ensuite une fonction `ppcm` qui calcule le plus petit commun diviseur de deux entiers. On pourra utiliser la fonction `pgcd`.