TP1 : Premiers Pas en OBJECTIVE CAML

Objectifs Connatre la syntaxe d'OBJECTIVE CAML, les notions de typage de base et savoir écrire des fonctions (récursives) simples. Le TP est rendre votre encadrant de TP le lundi 5 octobre.

Exercice 1 - Syntaxe, typage et évaluation

Les expressions suivantes sont elles syntaxiquement correctes ? Le cas échéant, sont-elles bien typées ? Si non, pourquoi ? Si oui, donner leur type et leur valeur.

Expressions

1. `let _ = 1 + 2;;`
2. `let _ = max_int + 1;;`
3. `let _ = 1.0 +. 2.0;;`
4. `let _ = 1.0 + 2;;`
5. `let _ = 1 / 2;;`
6. `let _ = 1.0 /. 2.0;;`
7. `let _ = 3 % 2;;`
8. `let _ = 3 mod 2.0;;`
9. `let _ = 42 mod 7;;`
10. `let _ = 0b0 + 2;;`
11. `let _ = false || 1;;`
12. `let _ = false and false;;`
13. `let _ = true && (1 = 0);;`
14.
```
let _ =
    (1.0e100 +. 1e-100) -. 1e100 =
       (1.0e100 -. 1e100) +. 1e-100;;
```
15. `let _ = 1 <> 'a';;`
16.
```
let _ =
    if true || false then false else true;;
```
17. `let _ = if true then 1 else 0 = 1;;`
18. `let _ = if true then false;;`
19. `let _ = "blabla" = "bla" ^ "bla";;`

Identificateurs

1. `let a = 1;;`
2. `let A = 2;;`
3. `let a_prime = 1;;`
4. `let a' = 1;;`
5. `let _a = 1;;`
6. `let 'a = 1;;`
7. `let a1 = 1;;`
8. `let 1a = 1;;`

Exercice 2 - Liaison, portée

Vérifier le respect des règles de portée sur les expressions suivantes. Si les expressions sont typable, donner leur type et leur valeur, sinon dites quel est le problme.

1.
```
let _ =
    let x = 1 in
    let y = x in x + y;;
```
2.
```
let _ =
    let x = y in
    let y = 1 in x + y;;
```
3.
```
let _ =
    let x = 1 in
    let y = 2 in
    let x = 3 in x + y;;
```
4.
```
let _ =
    let x =
      let y = 1 in
      y + 1
    in x + y;;
```
5.
```
let _ =
    let x =
      let y = 1 in y + 1
    in
    let y = 2 in x + y;;
```

6.
```
let _ =
    let x =
      let y = 1 in
      x + y
    in x * 2;;
```
7.
```
let _ =
    let x = 1 in
    let x = true in x;;
```
8.
```
let _ =
    let x =
      if true then
        let tmp = 5.0 in
        tmp /. 2.0
      else 0.0
    in x;;
```
9.
```
let _ =
    let x =
      if true then
        let tmp = 5.0 in
        0.0
      else tmp /. 2.0
    in x;;
```
10.
```
let x = 1;;
let _ = let x = 2 in x;;
let _ = let y = x in y;;
```
11.
```
let _ =
    let f =
      let x = 1
      let y = 2 in
      x
    in f;;
```

Exercice 3 - Fonctions, fonctions n-aires

Donner le type des fonctions suivantes ou expliquer pourquoi elles ne sont pas typables.

1.
```
let f x = x = 1 || x = 1.0;;
```
2.
```
let f x y = x + y;;
```
3.
```
let f x y z w = x + y + z * w;;
```
4.
```
let f x =
    let y = 2 * x in
    if x = 0 then max_int
    else x + y;;
```
5.
```
let abs x =
    if x < 0 then -x else x;;
```
6.
```
let succ = fun x -> x + 1;;
```
7.
```
let incr_or_decr b =
    if b then (fun x -> x + 1)
    else (fun x -> x - 1);;
```
8.
```
let g y =
    let f x = x * y in
    f 0 + f 1;;
```
9.
```
let f x =
    let y = x * x in
    print_int y;;
```
10.
```
let affiche opt x =
    if opt then print_int x
    else ();;
```

Application

1. `affiche true 42`
2. `affiche (true, 42)`
3. `(affiche true) 42`

Définition de fonctions

Écrire le corps des fonctions suivantes.

1. La fonction `xor x y` doit implémenter le ou exclusif.
   `val xor : bool -> bool -> bool`
2. La fonction `max x y` doit donner le maximum de deux entiers.
   `val max : int -> int -> int`
3. On vous donne la fonction `String.length : string -> int` ; implémenter une fonction imprimant la longueur d'une chaine :
   `val affiche_longueur : string -> unit`

4. On vous donne la fonction `String.sub : string -> int -> int -> string`.
   `String.sub s pos len` renvoie la sous chaine de `s` de longueur `len` commençant en `pos`. Implémenter une fonction `affiche_tronque n s` qui affiche les `n` premiers caractères de `s` :
   `val affiche_tronque : int -> string -> unit`

Exercice 4 - Fonctions récursives

On vous donne les fonctions `Char.chr : int -> char` et `print_char : char -> unit`. Écrire la fonction `affiche_ascii n m` qui affiche les caractères ASCII entre `n` et `m` :
`val affiche_ascii : int -> int -> unit`

Question subsidiaire Écrire trois fonctions différentes de multiplication de deux entiers.