Séminaire CAML
QCM n° 6
vendredi 18 sept. 2015

1. **Laquelle de ces phrases n'est pas correcte ?**

   (a) `let x = 2 in let y = 2 * x in y + 5`
   (b) `let x = 2 and y = 3 * x in y + 10`
   (c) `let x = 2 and y = 4 in x + y`
   (d) `let x = 2 in let y = let y = 5 in 2 * x + y in y + 5`
   (e) `let x = 2 in let y = 3 in x + y`

---

2. **Soient les "morceaux" de phrases suivants :**
   `1 : 15` `2 : y + 12` `3 : 2 * y` `4 : let x =` `5 : let y =` `6 : in`
   Parmi les phrases obtenues avec les ordres suivants, lesquelles permettent de définir x lié à la valeur 42 ?

   (a) 5 5 1 6 3 6 4 2
   (b) 4 5 1 6 5 2 6 3
   (c) 4 5 5 1 6 3 6 2
   (d) 5 1 6 5 3 6 4 2
   (e) 4 5 1 6 5 3 6 2

---

3. **Que calcule la fonction suivante ?**

```
let f x =
   let g x = x + 1 in
      g x*2 ;;
```

   (a) $2x + 1$
   (b) $2(x + 1)$
   (c) $(x + 1)^2$
   (d) $x^2 + 1$
   (e) Rien, la fonction est incorrecte.

---

4. **Quels doivent être les types des fonctions `f` et `g` pour que l'expression suivante soit correcte ?**

```
f ((g (3*2) 4)+1) (5 - f 1 2) ;;
```

   (a) `f : int -> int` et `g : int -> int`
   (b) `f : int -> int -> int` et `g : int -> int -> int`
   (c) `f : int -> int` et `g : int -> int -> int`
   (d) `f : int -> int -> int` et `g : int -> int`
   (e) Aucune des propositions ci-dessus.

---

5. **Parmi les fonctions suivantes, lesquelles ont pour type `int -> int -> int` ?**

   (a) `let average a = function b -> (a+b)/2 ;;`
   (b) `let average = (function a -> function b -> (a+b)/2) 4 ;;`
   (c) `let average = function a -> b -> (a+b)/2 ;;`
   (d) `let average a b = (a+b)/2 ;;`
   (e) `let function a -> let average b = (a+b)/2 ;;`

6. **Parmi les phrases suivantes, quelle est l'intruse ?**

(a) `let even = function n -> if n mod 2 = 0 then true else false ;;`
(b) `let even n = let r = n - n/2*2 in r = 0 ;;`
(c) `let even n = n mod 2 ;;`
(d) `let even = function n -> n mod 2 = 0 ;;`
(e) `let even = function n -> n - n/2*2 = 0 ;;`

---

7. **Que contient le résultat de l'évaluation de la phrase suivante ?**

```
let f x = function
     0 -> failwith "divisor is zero"
   | y -> if y > x then y / x
          else if x = 0 then failwith "divisor is zero"
               else x / y
   | _ -> failwith "impossible operation" ;;
```

(a) *val f : int -> int -> string = <fun>*
(b) *val f : int -> int -> int = <fun>*
(c) *Warning U : this match case is unused.*
(d) Une erreur.

---

8. **Que contient le résultat de l'évaluation de la fonction g ?**

```
let g x y = match x with
              0 -> 0
            | y -> 1
            | x -> -1 ;;
```

(a) *val g : int -> int -> int = <fun>*
(b) *val g : int -> 'a -> int = <fun>*
(c) *... Warning U : this match case is unused.*
(d) *... Warning P : this pattern-matching is not exhaustive.*

---

9. **Quel est le type de la fonction f définie ci-dessous ?**

```
let f = function
      ((1,true),_) -> failwith "incorrect"
    | (x, "aa")    -> let (a,b) = x in a ;;
```

(a) `(int * bool) * string -> 'a`
(b) `int * bool * string -> int`
(c) `(int * bool) * string -> int`
(d) `(int * bool) * string -> string`
(e) La fonction est incorrecte.

---

10. **Quel est le résultat de l'évaluation de la fonction suivante ?**

```
let h c = match c with
     (x, y) when x = y -> true
   | _                 -> false  ;;
```

(a) *val h : 'a * 'a -> bool = <fun>*
(b) *val h : 'a -> 'a -> bool = <fun>*
(c) *val h : 'a -> 'b -> bool = <fun>*
(d) Une erreur.
