

**Exercice 1 :** L'environnement global  $EG$  contient les deux liaisons  $EG = \{ \langle x, 2 \rangle; \langle y, 5 \rangle \}$ . Les expressions `let` modifient l'environnement local dans lequel l'expression `(+ (* x x) (* y y))` est interprétée. Il faut donc connaître l'état de cet environnement local.

Rappelons que lors de l'évaluation d'un `let` l'interpréteur évalue d'abord toutes les expressions en partie droite des associations avant de définir les variables locales correspondantes.

- L'expression `(let ((x 3))` crée une nouvelle variable dans l'environnement local et masque la variable globale  $\langle x, 2 \rangle$ . L'environnement local est donc  $EL = \{ \langle x, 3 \rangle; \langle y, 5 \rangle \}$  et l'interprète évalue : `(+ (* 3 3) (* 5 5))` et renvoie 34.
- L'expression `(let ((x y))` est évaluée en deux temps : d'abord l'interprète évalue `y` qui renvoie 5, puis crée la variable locale `x` qui est liée à la valeur 5. L'environnement local est donc  $EL = \{ \langle x, 5 \rangle; \langle y, 5 \rangle \}$  et l'interprète évalue finalement : `(+ (* 5 5) (* 5 5))` et renvoie 50.
- La aussi l'expression `let` est évaluée en deux temps, l'interprète évalue d'abord `y` et `x` dans l'environnement global, puis crée les associations locales `(let ( (x y) (y x) )`  $\rightarrow$  `(let ( (x 5) (y 3) )`. L'environnement local est alors  $EL = \{ \langle x, 5 \rangle; \langle y, 3 \rangle \}$  et l'interprète évalue l'expression : `(+ (* 5 5) (* 3 3))` et renvoie 34.
- Lors de l'évaluation d'un `let*`, contrairement au `let`, les variables locales sont créées séquentiellement dans l'environnement local. Ainsi, l'interprète évalue d'abord `y` dont la valeur est 5 dans l'environnement global est crée la variable locale  $\langle x, 5 \rangle$ , puis il évalue `x` dans ce nouvel environnement et lie la variable locale `y` avec cette valeur, soit 5. On obtient finalement l'environnement local  $EL = \{ \langle x, 5 \rangle; \langle y, 5 \rangle \}$  et l'interprète renvoie 50.

**Exercice 2 :** On crée une variable locale `x2` dont la valeur est  $x^2$ .

```
(lambda (x)
  (let ( (x2 (* x x))
        (/ (+ x2 1) (+ x2 2)) ))
```

**Exercice 3 :** À l'aide d'un `let*` on définit successivement les variables locales : `x2` dont la valeur est  $x^2$ , `x4` dont la valeur est  $x^2 * x^2$ , etc ...

```
(lambda (x)
  (let* ( (x2 (* x x))
         (x4 (* x2 x2))
         (x8 (* x4 x4))
         (x16 (* x8 x8)) )
    (+ x x2 x4 x8 x16)))
```