

Esercizi

1) Si scriva in OCaml un'espressione che introduce una variabile x con valore 3, quindi introduce una variabile y con valore $x+2$, quindi una variabile x con valore $x+y$ ed infine una variabile z con valore l'astratto rispetto y (un funzionale di parametro y) di $x+y$. In questo ambiente vale $y+z(x)$.

2) Utilizzando il linguaggio didattico funzionale puro si mostri la sintassi astratta definita dalla espressione del punto 1.

3) Utilizzando l'interprete del linguaggio didattico si mostri la catena dei record di attivazione generata dall'interprete applicato alla valutazione dell'espressione in 2.

Ex1: Soluzione

1) Si scriva in Caml un'espressione che introduce una variabile x con valore 3, quindi introduce una variabile y con valore $x+2$, quindi una variabile x con valore $x+y$ ed infine una variabile z con valore l'astratto rispetto y di $x+y$. In questo ambiente vale $y+z(x)$.

```
let x=3 in
  let y=x+2 in
    let x = x+y in
      let z = function y -> x+y in
        y+z(x)
```

Ex2: Soluzione

2) Utilizzando il linguaggio didattico funzionale puro si mostri la sintassi astratta definita dalla espressione del punto 1.

```
let x=3 in
  let y=x+2 in
    let x = x+y in
      let z = function y -> x+y in
        y+z(x)
```

```
let("x", Eint 3,
  let("y", sum(Den "x", Eint 2),
    let("x", sum(Den "x", Den "y"),
      let("z", fun(["y"], sum(Den "x", Den "y")),
        sum(Den "y", appl(Den "z", [Den "x"]))6
```

Ex3: Soluzione

3) Utilizzando l'interprete del linguaggio didattico si mostri l'intera catena dei record di attivazione generata dall'interprete applicato alla valutazione dell'espressione in 2.

```

let("x", Eint 3,
  let("y", sum(Den "x", Eint 2),
    let("x", sum(Den "x", Den "y"),
      let("z", fun(["y"], sum(Den "x", Den "y")),
        E sum(Den "y", appl(Den "z", [Den "x"]))6
      )
    )
  )
)
  
```

envstack	cstack	Tempvalstack
_ = unbound	Expr1 let("x", Eint 3, E)	Unbound

envstack	cstack	Tempvalstack
_ = unbound	Expr2 let("x", Eint 3, E) Expr1 Eint 3	Unbound

Ex3: Soluzione - continua

```

let("x", Eint 3,
    let("y", sum(Den "x", Eint 2),
        let("x", sum(Den "x", Den "y"),
            let("z", fun(["y"], sum(Den "x", Den "y")),
                E
                sum(Den "y", appl(Den "z", [Den "x"]))_6
            )
        )
    )

```

envstack	cstack	Tempvalstack
_ = unbound	Expr2 let("x",Eint 3, E) Expr1 Eint 3	Unbound

envstack	cstack	Tempvalstack
_ = unbound	Expr2 let("x",Eint 3, E) Expr2 Eint 3	Unbound

envstack	cstack	Tempvalstack
_ = unbound	Expr2 let("x",Eint 3, E)	Unbound Int(3)

Ex3: Soluzione - continua 2

```

let("x", Eint 3,
  let("y", sum(Den "x", Eint 2),
    let("x", sum(Den "x", Den "y"),
      let("z", fun(["y"], sum(Den "x", Den "y")),
        E sum(Den "y", appl(Den "z", [Den "x"]))6
      )
    )
  )
)

```

envstack	cstack	Tempvalstack
_ = unbound	Expr2 let("x", Eint 3, E)	Unbound Int(3)

envstack	cstack	Tempvalstack
_ = unbound	--	Unbound
_ = unbound x = Int(3)	Expr1 E	Unbound