

PROGRAMMAZIONE II - a.a. 2019-20

Esercitazione — 4 dicembre 2020

Esercizio 1

Si estenda il linguaggio didattico funzionale in modo da includere espressioni e valori di tipo **record**. Un *valore (di tipo) record* è un dato strutturato composto da un numero finito di coppie *nome-valore*, detti **campi**. Analogamente, una *espressione (di tipo) record* è composta da un numero finito di coppie *nome-valore*. La valutazione di una espressione record produce un valore record.

Un identificatore può esser legato a un valore record tramite il costrutto **let**: nel seguente esempio (in sintassi OCaml-like) si evidenzia che l'espressione record che compare nel **let** è valutata in un valore record

```
# let rect = record{base = 5 * 5, altezza = 10 - 6}
val rect = record{base = 25, altezza = 4}
```

Sui record è definita l'operazione di selezione di una componente. Continuando l'esempio precedente

```
# let b = rect.base
val b = 25
```

1. Si estenda la sintassi astratta del linguaggio didattico funzionale in modo da includere valori ed espressioni record e l'operatore di selezione.

Esercizio 1

Una funzione a dominio finito è una funzione che è definita solo per un numero finito di elementi. Ad esempio si consideri la seguente funzione con una sintassi nello stile di OCaml

```
let sum = fun y -> 50 + y for y in [0; 1; 2; 3; 4];;
```

La funzione **sum** è definita solamente per valori del parametro attuale che appartengono all'insieme $\{0, 1, 2, 3, 4\}$, insieme che è calcolato al momento della definizione della funzione stessa.

1. Si estenda la sintassi astratta del linguaggio didattico funzionale senza funzioni ricorsive in modo da includere tali funzioni.
2. Si definiscano le regole OCaml dell'interprete per trattare la valutazione di dichiarazione e la chiamata di funzioni a dominio finito.

Esercizio 3

Si estenda il linguaggio didattico funzionale introducendo il tipo di dato **IntSet** che permette di dichiarare insieme di interi di cardinalità finita. In aggiunta, il linguaggio è esteso con le operazioni primitive **insert myset elem** e **remove myset elem** che permettono di operare su insiemi finiti di interi.

1. Si mostri come deve essere modificato l'interprete del linguaggio didattico funzionale.

Esercizio 4

Si consideri il seguente programma OCaml

```
let z =1;;

let f1 = fun x y -> x + y *z;;

let rec apply_n_times f n x =
  if n<=0 then x
  else apply_n_times f (n-1) (f x);;

let rec map_n_times g n = function
| [] -> []
| h1::ls -> (apply_n_times g n h1) :: map_n_times g n ls;;

let z = 2;;

let ff = f1 1;;

map_n_times ff z [10;20];;
```

1. Si simuli la valutazione del programma mostrando la struttura della pila dei record di attivazione.
2. Si determini il valore calcolato dal programma.