

# PROGRAMMAZIONE II - a.a. 2019-20

## Esercitazione —5 Dicembre 2019

### 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 2.** Si consideri il seguente programma OCaml, che realizza l'elevazione a potenza (la funzione `power`) con moltiplicazioni successive

```
let rec iterate n f d =
  if n = 0 then d
  else iterate (n-1) f (f d);;    (**)

let power i n =
  let i_times a = a * i in
  iterate n i_times 1;;

# power 3 2;;
- : int = 9
```

1. Si indichi il tipo inferito dall'interprete OCaml per le funzioni `iterate` e `power`.
2. Quante volte viene eseguita l'istruzione marcata con `(**)` valutando l'espressione `power 3 2`?
3. Simulando la valutazione dell'espressione `power 3 2`, si mostri la struttura della pila dei record di attivazione subito dopo l'invocazione di `f` e subito dopo l'invocazione di `iterate` per ogni esecuzione della linea marcata con `(**)`.