

Chiusure (semantica)

Scoping statico

Nei linguaggi con scoping statico le variabili libere nel corpo delle funzioni vengono legate a tempo di compilazione costruendo delle chiusure.

```
var pippo: Int = 10
func moltiplica(let n: Int) -> Int {
    return pippo * n
}
```

La dichiarazione di `moltiplica` estende l'ambiente come segue:

```
[(moltiplica,  $\lambda(\text{let } n) . \{[(\text{pippo}, l_2)]; \text{return pippo} * n\}$ )]
```

questo prende il nome di *λ -astrazione*. Le due parti evidenziate sono rispettivamente la lista dei parametri formali da associare agli attuali al momento della chiamata e la chiusura, ovvero la composizione sequenziale dell'ambiente contenente i legami delle variabili libere all'interno della funzione con il corpo della funzione stessa.

È un errore a tempo di compilazione utilizzare all'interno della funzione un identificatore non dichiarato.

Scoping dinamico

Le variabili libere vengono legate a tempo di esecuzione quando vengono utilizzate. Nella chiusura si registra solo il corpo della funzione.

```
[(moltiplica,  $\lambda(\text{let } n) . \{\text{return pippo} * n\}$ )]
```

Errori nel riferimento a variabili libere nel corpo della funzione possono essere individuati solo a tempo di esecuzione.