

PROGRAMMAZIONE II – A.A. 2019-20: Secondo Progetto Intermedio

[consegna entro il 17 gennaio 2020]

Il progetto ha l'obiettivo di applicare a casi specifici i concetti e le tecniche di programmazione esaminate durante la seconda parte del corso, e consiste nella progettazione e realizzazione di alcuni moduli software.

Descrizione: Progettazione e sviluppo di un interprete in OCaml

Si consideri un'estensione del linguaggio didattico funzionale presentato a lezione che permetta di manipolare *dizionari*. Un dizionario è una collezione di valori identificati univocamente da una chiave. Pertanto, un dizionario è una collezione di coppie chiave-valore dove la chiave è unica.

Un esempio concreto di dizionario è riportato di seguito

```
Magazzino = {'mele': 430, 'banane': 312, 'arance': 525, 'pere': 217}
```

I dizionari sono caratterizzati da diverse operazioni primitive.

L'operazione **insert** inserisce una coppia chiave-valore in un dizionario. Per esempio l'operazione **insert('kiwi', 300, Magazzino)** produce come risultato il dizionario:

```
{'mele': 430, 'banane': 312, 'arance': 525, 'pere': 217, kiwi: 300}
```

L'operazione **delete** rimuove una coppia chiave-valore da un dizionario. Per esempio: l'operazione **delete Magazzino('pere')** produce come risultato il dizionario:

```
{'banane': 312, 'arance': 525, 'mele': 430}
```

L'operazione **has_key** controlla l'esistenza della chiave in un dizionario. Per esempio l'operazione **has_key(banane, Magazzino)** restituisce il valore booleano `true`.

L'operazione **Iterate(f,d)** applica la funzione `f` a tutte le coppie chiave-valore presenti nel dizionario, restituendo un nuovo dizionario con i valori ottenuti come risultato della funzione.

Per esempio l'operazione **iterate((fun val -> val +1), Magazzino)** produce come risultato il dizionario:

```
{'mele': 431, 'banane': 313, 'arance': 526, 'pere': 218}
```

L'operazione **fold(f d)** calcola il valore ottenuto applicando la funzione `f` sequenzialmente a tutti gli elementi del dizionario. Al passo `i`-esimo il valore calcolato `f_i` è ottenuto come somma del valore della funzione `f` applicata al valore `v_i-1` nel dizionario e del valore `f_i-1`.

Ad esempio l'operazione **fold((fun val -> val +1), Magazzino)** produce come risultato il valore 1488.

L'operazione **filter(key list, d)** restituisce come risultato il dizionario ottenuto dal dizionario `d` eliminando tutte le coppie chiave-valore per cui la chiave non appartiene alla lista delle chiavi passata come parametro. Ad esempio l'applicazione di **filter(['mele'; 'pere'] d)** produce come risultato il dizionario:

```
{'mele': 430, 'pere': 217}
```

1. Definire le regole operazionali per la gestione del dizionario
2. Estendere l'interprete OCaml del linguaggio funzionale assumendo la regola di scoping statico.
3. Opzionale : definire il type checker dinamico del linguaggio risultante.
4. Si verifichi la correttezza dell'interprete progettando ed eseguendo una quantità di casi di test sufficiente a testare tutti gli operatori aggiuntivi.

Modalità di consegna

- Il progetto deve essere svolto e discusso col docente individualmente. Il confronto con colleghi mirante a valutare soluzioni alternative durante la fase di progetto è incoraggiato.
- Il progetto deve essere costituito da
 - i file sorgente contenenti il codice sviluppato e le corrispondenti batterie di test, ove tutto il codice deve essere adeguatamente commentato;
 - una relazione di massimo una pagina che descrive le principali scelte progettuali ed eventuali istruzioni per eseguire il codice.
- La consegna va fatta inviando per email tutti i file in un archivio entro il 17 Gennaio 2020. Per il corso A, inviare l'email al Prof. Ferrari con oggetto "[PR2A] Consegna progetto 2". Per il corso B, inviare l'email al Prof.ssa Levi con oggetto contenente la stringa "[PR2B] Consegna progetto 2".

Altre informazioni

- Per quanto riguarda il progetto, i docenti risponderanno solo a eventuali domande riguardanti l'interpretazione del testo, e non commenteranno soluzioni parziali prima della consegna.