

Relazione secondo progetto Programmazione 2

Serafino Gabriele – 564411 – Corso B

Introduzione

Il progetto richiede di realizzare un'estensione dell'interprete didattico funzionale in OCaml che permetta di manipolare come tipi di dato primitivi dei dizionari, ovvero delle collezioni di elementi non ordinati formati da coppie nome-valore dove il nome è un identificatore unico utilizzato per accedere al valore ad esso associato, mentre i valori sono di tipo non omogeneo.

Sintassi concreta

La sintassi concreta secondo le regole definite nel testo del progetto è la seguente:

```
Dict ::= Nil | (Ide, Exp) ; Dict
DictSet ::= (Dict, Ide, Exp)
DictGet ::= (Dict, Ide)
DictRm ::= (Dict, Ide)
ApplyOver ::= (Ide, Exp, Dict)
DictClear ::= Dict
DictLen ::= Dict
DictMerge ::= (Dict, Dict)
```

Interpretazione del testo

La formulazione del testo insieme agli esempi forniti sono sufficienti ad interpretare univocamente ciò che viene richiesto. L'unica operazione che potrebbe essere realizzata in più modi è la funzione **ApplyOver**.

Questa operazione richiede, data una funzione **F** ed un dizionario **D**, di applicare **F** a tutti i valori di **D**.

Poichè però i valori di **D** non sono omogenei non sarà sempre possibile applicare la funzione senza errori.

Nell'implementazione dell'interprete da me fornito l'applicazione di **ApplyOver** restituisce come risultato un dizionario **D'**, in cui la funzione **F** è applicata solo ai valori di **D** che appartengono al suo dominio.

$$D' = \begin{cases} (i, f(v)) & , \text{se } v \in \text{Dom}(F) \\ (i, v) & , \text{altrimenti} \end{cases} \quad \forall (i, v) \in D$$

ApplyOver permette l'applicazione di funzioni semplici e ricorsive, nei test a fine progetto è presente un caso con l'utilizzo di una semplice funzione ricorsiva. Notare che la sintassi astratta per le funzioni ricorsive richiede di inserire **ApplyOver** all'interno del **letBody** dell'operatore **Letrec**.

Funzioni aggiuntive

Ho deciso di estendere ulteriormente la sintassi fornita, permettendo l'utilizzo di due funzioni utili alla manipolazione dei dizionari:

- **DictLen** : restituisce la **lunghezza** del dizionario (numero di coppie nome-valore);
- **DictMerge** : prende come argomento due dizionari **d1** e **d2**, restituisce un dizionario **d3** che è il risultato del **merge** dei due dizionari di partenza. La funzione di merge segue il comportamento dell'analoga funzione in Python, perciò se entrambi i dizionari hanno una coppia con uguale nome, il dizionario risultante conterrà la coppia con il valore del dizionario **d2**.

Casi di test

Alla fine del file, contenente l'interprete OCaml, sono presenti alcuni casi di test, ogni test verifica la correttezza di una funzione implementata. Notare che per ogni caso di test è presente un commento che mostra come il dizionario viene modificato dall'esecuzione di quella determinata operazione.