Federico Ramacciotti - 582646
PR2 Corso A - Prof. G. Ferrari

UNIVERSITÀ DI PISA

Relazione

Secondo progetto - Dizionario

Dettagli sul progetto

Per realizzare il progetto, è stato esteso il linguaggio didattico (nello specifico, il codice della Professoressa Levi) con i dizionari. Questi sono stati definiti con una lista di coppie <chiave, valore>, con la chiave di tipo stringa e il valore di tipo intero o booleano. I valori di un dizionario possono errore non omogenei, ma nelle operazioni fold e iterate, che applicano delle funzioni a tutto il dizionario, serve un tyechecker dinamico che controlli se tutti i valori sono omogenei, in modo tale da poter applicare la funzione al dizionario. Infatti, funzioni somma, ad esempio, non possono essere applicate a valori bool.
Dal momento che l'operazione Fold richiede una funzione binaria, che accetta due parametri, è stata aggiunta sia la sua dichiarazione che l'applicazione alle funzioni realizzabili.

Operazioni

Le operazioni che sono state implementate sono Insert, che prende in input un dizionario, una chiave e un valore e lo aggiunge al dizionario, se esso non esiste già. Delete, che rimuove da un dizionario una chiave, entrambi passati in input. HasKey controlla se in un dato dizionario esiste o meno una chiave, restituendo true se esiste e false altrimenti. Iterate applica una funzione a ogni valore del dizionario, mentre Fold fa la stessa cosa ma sequenzialmente, tramite una funzione a doppio parametro, con accumulatore. Per ultima, Filter filtra un dizionario attraverso una maschera, lasciando all'interno dello stesso solo le coppie di con le chiavi specificate nella maschera (passata in input come lista di stringhe).

Regole Operazionali

Dict

$$env \triangleright Dict(d) \Rightarrow v$$

Insert

$$\frac{env \triangleright dict \Rightarrow DictVal(d)}{env \triangleright Insert(k, v, dict) \Rightarrow DictVal(d \cup (k, val))}$$

Delete

$$\frac{env \triangleright dict \Rightarrow DictVal(d)}{env \triangleright Insert(k, v, dict) \Rightarrow DictVal(d - (k, val))}$$

HasKey

$$\frac{\begin{array}{c} env \triangleright dict \Rightarrow DictVal(d) \\ (\exists (k \in d).\, k = key) \Rightarrow true\ else\ false \Rightarrow v \end{array}}{env \triangleright HasKey(key, dict) \Rightarrow v}$$

Iterate

$$\frac{\begin{array}{c} env \triangleright dict \Rightarrow DictVal(d) \quad env \triangleright funct \Rightarrow FunVal(x, e) \\ \forall [(k1, v1), (k2, v2)] \in dict \Rightarrow type(v1) = type(v2) \end{array}}{env \triangleright Iterate(funct, dict) \Rightarrow DictVal(\forall (k, val) \in dict \Rightarrow env[val/x])}$$

Fold

$$\frac{\begin{array}{c} env \triangleright dict \Rightarrow DictVal(d) \quad env \triangleright funct \Rightarrow FunAccVal(acc, x, e) \\ \forall [(k1, v1), (k2, v2)] \in dict \Rightarrow type(v1) = type(v2) \end{array}}{env \triangleright Fold(funct, dict) \Rightarrow DictVal(\forall (k, val) \in dict \Rightarrow env[val/x, acc])}$$

Filter

$$\frac{\begin{array}{c} env \triangleright dict \Rightarrow DictVal(d) \\ (DictVal(\forall (k, val) \in dict\,.\, k \in keylist)) \Rightarrow v \end{array}}{env \triangleright Filter(keylist, dict) \Rightarrow v}$$