RELAZIONE SECONDO PROGETTO [PR2]

Al linguaggio didattico è stata applicata un’estensione che permette la manipolazione dei dizionari, insiemi di valori identificati univocamente da una chiave. La sintassi introdotta è la seguente:

* **Dictionary of assocList**, permette di dichiarare un dizionario passando come argomento una assocList, una lista di associazioni chiave-valore. Dichiarando un dizionario si ottiene un tipo valutabile **DicClosure of (ide \* evT) list**
* **InsertDic** **of ide \* exp \* exp**, permette di inserire all’interno di un dizionario una nuova associazione chiave-valore
* **DeleteDict of ide \* exp**, permette di rimuovere dal dizionario un’associazione passando come argomento una chiave
* **HashKey of ide \* exp**, permette di vedere se una chiave è presente all’interno del dizionario
* **Iterate of exp \* exp**, applica una funzione a tutti gli elementi del dizionario
* **Fold exp \* exp**, calcola il valore ottenuto applicando la funzione passata come argomento sequenzialmente a tutti gli elementi del dizionario. La funzione deve avere esattamente due parametri formali
* **Filter of ideList \* exp**, restituisce solo le associazioni del dizionario che sono presenti anche nella lista di chiavi ideList

Per realizzare l’operazione **Fold** è stato inoltre necessario estendere il linguaggio didattico in modo da poter supportare anche funzioni a più parametri. È stata definita la nuova espressione **FunArgs** **of ideList \* exp** che prende come argomenti una lista di valori ed il corpo della funzione. È stato inoltre creato il nuovo tipo valutabile **FunArgs of ide list \* exp \* evT env**. Per applicare una funzione a più argomenti è stata definita l’espressione **ApplyArgs of exp \* expList**, costituita dalla funzione e dalla lista dei parametri attuali da associare ai parametri formali.