Relazione Secondo Progetto Programmazione II [Corso A]
Raffaele Apetino - N.M. 549220

Estensione del linguaggio

Ho iniziato il mio progetto estendendo le espressioni "*exp*" del linguaggio con i nuovi costruttori:

- ***Dict of (ide * exp) list*** (definisco il dizionario come una lista di coppie identificatore, valore)
- ***SelectDict of exp * ide***, per accedere a un elemento del dizionario (prende come parametro un dizionario e una etichetta)
- ***AddDict of exp * ide * exp***, per poter aggiungere un nuovo elemento nel dizionario (parametri: dizionario, nuova etichetta, valore da associare all'etichetta)
- ***UpdateDict of exp * ide * exp***, per poter aggiornare il valore di una etichetta con un nuovo valore (parametri: dizionario, etichetta da aggiornare, nuovo valore)
- ***RmDict of exp * ide***, per poter rimuovere una coppia etichetta valore dal dizionario (parametri: dizionario, etichetta)
- ***ClearDict of exp***, per rendere il dizionario nuovamente vuoto (parametro: dizionario)
- ***ApplyOverDict of exp * exp*** che applica la funzione (primo parametro) a tutti gli elementi del dizionario (secondo parametro)

Implementazione Dizionario

L'implementazione del dizionario è basata su una lista di coppie (etichetta,valore) utilizzando le liste native di OCaml. Il valore inizialmente trasportato dalla coppia è una espressione *exp*, che poi va in seguito valutata tramite l'*eval* nell'ambiente corrente. Per questo ho incluso nel linguaggio un tipo esprimibile ***DictVal of (ide * evT) list*** che è una lista contenente (etichetta,valore nell'ambiente corrente), in questo modo mi è permesso trasportare come valore tutto ciò che è esprimibile nel mio linguaggio.
Nella mia scelta di progetto riguardante il dizionario ogni etichetta deve essere univoca. Sia nella creazione che nell'inserimento controllo che non ci siano duplicati. Nell'istante della prima creazione del dizionario, se nella lista sono presenti delle etichette duplicate, decido di accettare, e quindi inserire, l'ultima definizione. Nel caso in cui l'utente cercasse di aggiungere (*AddDict*) una coppia dove l'etichetta è già presente, lancio un'eccezione. Nel caso in cui l'utente volesse aggiornare il valore di una coppia, farà riferimento al costruttore *UpdateDict*.

Informazioni aggiuntive

L'ambiente è definito come una lista polimorfa **(string * 't) list** e il type checking dei tipi viene eseguito dinamicamente.
I casi di test sono inclusi nel file mydict.ml.