
Relazione Progetto OcaML

ANNO 2018-2019

Selene Gerali - MATRICOLA: 546091

INTRODUZIONE:

Il progetto richiede di estendere il linguaggio didattico funzionale (presentato a lezione) in modo da permettere di gestire dizionari di elementi "Dict" come dati primitivi del linguaggio.

SCELTE PROGETTUALI:

Ho esteso la sintassi astratta del linguaggio didattico funzionale aggiungendo nel type *exp*: *Dict of (ide*exp)list* che rappresenta una collezione di coppie <chiave, valore> e successivamente ho aggiunto: *Set of Dict * ide * exp*, *Remove of Dict * ide*, *Get of Dict * ide*, *Clear of Dict* e *ApplyOver of Dict * exp* che permettono di eseguire determinate operazioni su un dizionario.

Successivamente ho definito le regole di OcaML dell'interprete per trattare la valutazione di "Dict" modificando la funzione ricorsiva "eval" e inserendo le funzioni ausiliarie necessarie. Ho ritenuto opportuno implementare una funzione *EvalList* che mi valuta ogni elemento del dizionario, *Lookup* che restituisce il valore associato ad una chiave se presente nel dizionario, *rm* che se una certa chiave è presente nel dizionario la elimina, *aggiornaVal* che va a modificare il valore di un campo se coincidente al parametro selezionato, una funzione booleana *cerca* che mi restituisce true se una chiave passata come parametro è presente nel dizionario, un'altra funzione booleana *contiene* che ha lo stesso comportamento della *cerca* ma lavora su liste di tipo *(ide*evT)list* mentre la precedente lavora su *(ide*exp)list* (non ho potuto farne una unica che prendeva una lista di tipo *(ide* 'a)list* perché avevo un errore di tipo durante la compilazione). Ho inoltre implementato una funzione *contrDict* che controlla se il dizionario ha tutti elementi distinti (ho ipotizzato che non ci possono essere campi duplicati) e infine la funzione *apply* nel quale valuto una funzione applicandola ai campi di un mio dizionario, tenendo conto che ci possono essere campi a cui non posso applicare la funzione (esempio se ho una funzione che lavora su interi non la posso applicare a campi che sono stringhe e viceversa), per questo motivo ho inserito un "try excep" nella valutazione del corpo della funzione.

SINTASSI CONCRETA:

Dict ::= (ide * exp) Dict | ε

Remove ::= (Dict * ide)

Clear ::= Dict

Set ::= (Dict * ide * exp)

Get ::= (Dict * ide)

ApplyOver ::= (Dict * exp)

