**RELAZIONE PROGETTO INTERMEDIO OCAML**

Superbo Rebecca

Matricola 560504

Il progetto si compone di un file .ml contenente l’interprete.

**DIZIONARIO:** il dizionario è implementato tramite un tipo dictionary.

**TIPO EXP:** al tipo exp sono stati aggiunti i costruttori:

* EDict -> definisce un dizionario
* Get -> richiede un elemento all’interno di un dizionario
* Remove -> rimuove un elemento all’interno del dizionario
* Add -> aggiunge un nuovo elemento nel dizionario
* Clear -> rimuove tutti gli elementi di un dizionario
* ApplyOver -> applica una funzione a tutti gli elementi presenti in un dizionario

**TIPI ESPRIMIBILI:** ai tipi esprimibili sono stati aggiunti i costruttori:

* String -> identifica le stringhe
* Dict -> identifica un dizionario che può essere Empty (vuoto) oppure formato da una tripla ide \* evT \* dicti

**TYPECHECKING:** al typechecking sono stati aggiunti i pattern matching:

* String -> controlla che il parametro v sia una stringa
* Dict -> controlla che il parametro v sia un dizionario

**FUNZIONI PRIMITIVE:** alle funzioni primitive sono state aggiunte:

* get -> cerca un elemento nel dizionario e ritorna il valore ad esso associato
* rm -> cerca una chiave nel dizionario e ne rimuove il valore ad esso associato
* add -> aggiunge un elemento nel dizionario (se non è già presente) oppure aggiorna il valore ad esso associato
* clear -> elimina tutti gli elementi presenti nel dizionario

**INTERPRETE:** sono stati aggiunti i seguenti casi al pattern matching:

* EDict -> tramite l’uso di una funzione ausiliaria evalDict, essa valuta ricorsivamente le espressioni presenti nel dizionario e restituisce lo stesso con le espressioni valutate
* Get -> viene controllato che il parametro d sia un dizionario, se lo è restituisce il valore associato alla chiave considerata, altrimenti lancia un errore di tipo
* Remove -> viene controllato che il parametro d sia un dizionario, se lo è rimuove la chiave considerata dal dizionario, altrimenti lancia un errore di tipo
* Add -> controlla e valuta value e d, in seguito a esito positivo aggiunge il valore scelto al dizionario, altrimenti lancia un errore di tipo
* Clear -> viene controllato che il parametro d sia un dizionario, se lo è elimina tutti gli elementi presenti nel dizionario
* ApplyOver -> controlla e valuta i parametri d e f, in seguito a esito positivo applica la funzione f ad ogni elemento presente nel dizionario d e restituisce il nuovo dizionario aggiornato

**TEST:** sono stati forniti test case che valutano la corretta esecuzione di tutti gli operatori aggiuntivi, anche nel caso del lancio di eccezioni.