SECONDO PROGETTO PROGRAMMAZIONE 2

Università di Pisa, A.A. 2019-2020
Maria Vitali, 548154

REGOLE OPERAZIONALI

`evalPairlist( pairlist pl ) genera una lista di coppie chiave-valore del tipo` $[(k_1, v_1); (k_2, v_2); ...; (k_n, v_n)]$ `dove` $n = list.length$

Dict

$$env \triangleright Dict(d) \Rightarrow Dictval(dict), \quad dict = evalPairlist(d)$$

Insert

$$\frac{env \triangleright evalPairlist(d) = dict, d \Rightarrow Dictval(dict), v \Rightarrow val, dictMember(k, dict) \Rightarrow false}{env \triangleright Insert(k, v, d) \Rightarrow Dictval((k, v) :: dict)}$$

Delete

$$\frac{env \triangleright d \Rightarrow Dictval(dict), dict = evalPairlist(d) \wedge \exists (key, value) \in dict \,.\, key = k}{env \triangleright Delete(d, k) \Rightarrow Dictval(dict \setminus (k, v))}$$

Has_Key

$$\frac{env \triangleright d \Rightarrow Dictval(dict), dict = evalPairlist(d)}{env \triangleright Has\_Key(k, d) \Rightarrow Bool(dictMember(k, dict))}$$

and $dictMember(k, dict) = \begin{vmatrix} true & if\ k \in dict \\ false & if\ k \notin dict \end{vmatrix}$

Iterate

con funzione non ricorsiva

$$\frac{\begin{array}{c} env \triangleright f = Fun(i, a) \Rightarrow Funval(i, a, r) \\ env \triangleright d \Rightarrow Dictval(dict), dict = evalPairlist(d) \\ \forall i, 1 \leq i \leq n, n = dict.length \wedge (k_i, v_i) \in dict \wedge FunCall(f, v_i) \Rightarrow v_i' \\ . \ dd = [(k_1, v_1'); (k_2, v_2'); \ldots; (k_n, v_n')] \end{array}}{env \triangleright Iterate(f, d) \Rightarrow Dictval(dd)}$$

con funzione ricorsiva

$$\frac{\begin{array}{c} env \triangleright f \Rightarrow RecFunval(g, (arg, fBody, fDecEnv)) \\ env \triangleright d \Rightarrow Dictval(dict), dict = evalPairlist(d) \\ \forall i, 1 \leq i \leq n, n = dict.length \wedge (k_i, v_i) \in dict \wedge FunCall(f, v_i) \Rightarrow v_i' \\ . \ dd = [(k_1, v_1'); (k_2, v_2'); \ldots; (k_n, v_n')] \end{array}}{env \triangleright Iterate(f, d) \Rightarrow Dictval(dd)}$$

Fold

$$\frac{\begin{array}{c} env \triangleright f = BinFun(acc, i, a) \Rightarrow BinFunVal(acc, i, a, r) \\ env \triangleright d \Rightarrow Dictval(dict), dict = evalPairlist(d) \\ \forall i, 1 \leq i \leq n, n = dict.length \wedge (k_i, v_i) \in dict \\ . \ BinFunCall(f, \ldots(BinFunCall(f, (BinFunCall(f, 0, v_1)), v_2)\ldots, v_n) = v \end{array}}{env \triangleright Fold(f, d) \Rightarrow v}$$

Filter

$$\frac{\begin{array}{c} env \triangleright d \Rightarrow Dictval(dict), dict = evalPairlist(d) \\ \{(k_i, v_i) \mid 0 \leq i \leq dict.length \ \wedge \ (k_i, v_i) \in dict \ \wedge \ k_i \in kl\} = filteredDict \end{array}}{env \triangleright Filter(kl, d) \Rightarrow Dictval(filteredDict)}$$