[Corso A] Secondo Progetto Intermedio

sre990

June 27, 2021

1 Empty(t):

$$\frac{t \in typeSet}{env \triangleright Empty(t) \Longrightarrow Set(t,\emptyset)}$$

2 Singleton(e):

$$\frac{t \in typeSet, env \triangleright e \to i \qquad gettype(i)=t}{env \triangleright Singleton(e) \Longrightarrow Set(t,\{i\} \cup \emptyset)}$$

3 Of(t,e):

$$\frac{t \in typeSet, e \in collection, env \triangleright e \to lst \qquad lst=(\forall x \in lst.gettype(x)=t \wedge (\nexists y \in lst.y=x))}{env \triangleright Of(t,e) \Longrightarrow Set(t,lst)}$$

4 Union(s1,s2):

$$\frac{env \triangleright s1 \to Set(t1,l1) \qquad s2 \to Set(t2,l2) \wedge t1=t2}{env \triangleright Union(s1,s2) \Longrightarrow Set(t,l1 \cup l2)}$$

5 Intersection(s1,s2):

$$\frac{env \triangleright s1 \to Set(t1,l1) \qquad s2 \to Set(t2,l2) \wedge t1=t2}{env \triangleright Intersection(s1,s2) \Longrightarrow Set(t,l1 \cap l2)}$$

6 Difference(s1,s2):

$$\frac{env \triangleright s1 \to Set(t1,l1) \qquad s2 \to Set(t2,l2) \wedge t1=t2}{env \triangleright Difference(s1,s2) \Longrightarrow Set(t,l1-l2)}$$

7 IsSubset(s1,s2):

$$\frac{env \triangleright s1 \rightarrow Set(t1,l1) \qquad s2 \rightarrow Set(t2,l2) \wedge t1=t2}{env \triangleright IsSubset(s1,s2) \implies l1 \subseteq l2}$$

8 Add(set,v):

$$\frac{env \triangleright set \rightarrow Set(t1,l1) \qquad v \rightarrow v1 \wedge t1=gettype(v1)}{env \triangleright Add(set,v) \implies Set(t,l1 \cup \{v1\})}$$

9 Remove(set,v):

$$\frac{env \triangleright set \rightarrow Set(t1,l1) \qquad v \rightarrow v1 \wedge t1=gettype(v1)}{env \triangleright Remove(set,v) \implies Set(t1,l1-\{v1\})}$$

10 IsInside(set,v):

$$\frac{env \triangleright set \rightarrow Set(t1,l1) \qquad v \rightarrow v1 \wedge t1=gettype(v1)}{env \triangleright IsInside(set,v) \implies v1 \in l1}$$

11 IsEmpty(set):

$$\frac{env \triangleright set \rightarrow Set(t1,l1)}{env \triangleright IsEmpty(set) \implies l1=\emptyset}$$

12 GetMax(set):

$$\frac{env \triangleright set \rightarrow Set(t1,l1) \qquad v=(\exists v \in l1.(\forall x \in l1 \, x <= v))}{env \triangleright GetMax(set) \implies v}$$

13 GetMin(set):

$$\frac{env \triangleright set \rightarrow Set(t1,l1) \qquad v=(\exists v \in l1.(\forall x \in l1 \, x >= v))}{env \triangleright GetMin(set) \implies v}$$

14 ForAll(pred,set):

caso true:

$$\frac{env \triangleright pred \rightarrow Closure(arg,body,fDecEnv) \qquad set \rightarrow Set(t,l),(\forall x \in l.fDecEnv[l \backslash x] \triangleright body \rightarrow true) \rightarrow b=true}{env \triangleright ForAll(pred,set) \implies b}$$

caso false:

$$\frac{env \triangleright pred \rightarrow Closure(arg,body,fDecEnv) \qquad set \rightarrow Set(t,l),(\exists x \in l.fDecEnv[l \backslash x] \triangleright body \rightarrow false) \rightarrow b=false}{env \triangleright ForAll(pred,set) \implies b}$$

15   Exists(pred,set):

caso true:

$$\frac{env \triangleright pred \rightarrow Closure(arg,body,fDecEnv) \qquad set \rightarrow Set(t,l),(\exists x \in l.fDecEnv[l \backslash x] \triangleright body \rightarrow true) \rightarrow b=true}{env \triangleright Exists(pred,set) \Longrightarrow b}$$

caso false:

$$\frac{env \triangleright pred \rightarrow Closure(arg,body,fDecEnv) \qquad set \rightarrow Set(t,l),(\forall x \in l.fDecEnv[l \backslash x] \triangleright body \rightarrow false) \rightarrow b=false}{env \triangleright Exists(pred,set) \Longrightarrow b}$$

16   Filter(pred,set):

$$\frac{env \triangleright pred \rightarrow Closure(arg,body,fDecEnv) \qquad set \rightarrow Set(t,l),(\forall x \in l.fDecEnv[l \backslash x] \triangleright body \rightarrow true) \rightarrow x}{env \triangleright Filter(pred,set) \Longrightarrow Set(t,x \in l1)}$$

17   Map(pred,set):

$$\frac{env \triangleright pred \rightarrow Closure(arg,body,fDecEnv) \qquad set \rightarrow Set(t,l),(\forall x \in l.fDecEnv[l \backslash x] \triangleright body \rightarrow v) \rightarrow v}{env \triangleright Map(pred,set) \Longrightarrow Set(t,v \in l1)}$$