

Regole operazionali

Nota: gli operatori insiemistici (\cup , \cap , $-$, \subseteq , ecc...) sono qui liberamente usati per indicare le rispettive operazioni su liste, come definite all'interno del progetto.

Type(ev) è definita come la seguente funzione:

```
Type(ev: evT) : setType =  
  match ev with  
    | Int(_) -> IntSet  
    | Bool(_) -> BoolSet  
    | String(_) -> StringSet  
;;
```

che associa un evT al corrispondente setType.

Empty:

$$\frac{-}{env \vdash Empty(t) \Rightarrow Set(t, EmptySet)}$$

Singleton:

$$\frac{env \vdash e \Rightarrow v}{env \vdash Singleton(t, e) \Rightarrow Set(t, Elements([v]))}$$

Of:

$$\frac{env \vdash lis \Rightarrow evLis}{env \vdash Of(t, lis) \Rightarrow Set(t, Elements(evLis))}$$

Union:

$$\frac{env \vdash i1 \Rightarrow Set(t1, Elements(evLis1)), env \vdash i2 \Rightarrow Set(t2, Elements(evLis2)), t1 = t2}{env \vdash Union(i1, i2) \Rightarrow Set(t1, Elements(evLis1 \cup evLis2))}$$

$$\frac{env \vdash i1 \Rightarrow Set(t1, EmptySet), env \vdash i2 \Rightarrow Set(t2, e), t1 = t2}{env \vdash Union(i1, i2) \Rightarrow Set(t2, e)}$$

$$\frac{env \vdash i1 \Rightarrow Set(t1, e), env \vdash i2 \Rightarrow Set(t2, EmptySet), t1 = t2}{env \vdash Union(i1, i2) \Rightarrow Set(t1, e)}$$

Intersec:

$$\frac{env \vdash i1 \Rightarrow Set(t1, Elements(evLis1)), env \vdash i2 \Rightarrow Set(t2, Elements(evLis2)), t1 = t2}{env \vdash Intersec(i1, i2) \Rightarrow Set(t1, Elements(evLis1 \cap evLis2))}$$

$$\frac{env \vdash i1 \Rightarrow Set(t1, EmptySet), env \vdash i2 \Rightarrow Set(t2, e), t1 = t2}{env \vdash Intersec(i1, i2) \Rightarrow Set(t1, EmptySet)}$$

$$\frac{env \vdash i1 \Rightarrow Set(t1, e), env \vdash i2 \Rightarrow Set(t2, EmptySet), t1 = t2}{env \vdash Intersec(i1, i2) \Rightarrow Set(t2, EmptySet)}$$

Diff:

$$\frac{env \vdash i1 \Rightarrow Set(t1, Elements(evLis1)), env \vdash i2 \Rightarrow Set(t2, Elements(evLis2)), t1 = t2}{env \vdash Diff(i1, i2) \Rightarrow Set(t1, Elements(evLis1 - evLis2))}$$

$$\frac{env \vdash i1 \Rightarrow Set(t1, EmptySet), env \vdash i2 \Rightarrow Set(t2, e), t1 = t2}{env \vdash Diff(i1, i2) \Rightarrow Set(t1, EmptySet)}$$

$$\frac{env \vdash i1 \Rightarrow Set(t1, e), env \vdash i2 \Rightarrow Set(t2, EmptySet), t1 = t2}{env \vdash Diff(i1, i2) \Rightarrow Set(t1, e)}$$

Add:

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash e \Rightarrow v, v \notin evLis, Type(v) = t}{env \vdash Add(i, e) \Rightarrow Set(t, Elements(v :: evLis))}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(v :: evLis)), env \vdash e \Rightarrow v}{env \vdash Add(i, e) \Rightarrow Set(t, Elements(v :: evLis))}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash e \Rightarrow v, Type(v) = t}{env \vdash Add(i, e) \Rightarrow Set(t, Elements([v]))}$$

Remove:

$$\frac{env \vdash i \Rightarrow Set(t, Elements(v :: evLis)), env \vdash e \Rightarrow v}{env \vdash Remove(i, e) \Rightarrow Set(t, Elements(evLis))}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash e \Rightarrow v, v \notin evLis, Type(v) = t}{env \vdash Remove(i, e) \Rightarrow Set(t, Elements(evLis))}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements([v])), env \vdash e \Rightarrow v}{env \vdash Remove(i, e) \Rightarrow Set(t, EmptySet)}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash e \Rightarrow v, Type(v) = t}{env \vdash Remove(i, e) \Rightarrow Set(t, EmptySet)}$$

IsEmpty:

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet)}{env \vdash IsEmpty(i) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis))}{env \vdash IsEmpty(i) \Rightarrow Bool(false)}$$

Member:

$$\frac{env \vdash i \Rightarrow Set(t, Elements(v :: evLis)), env \vdash e \Rightarrow v}{env \vdash Member(i, e) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash e \Rightarrow v, v \notin evLis, Type(v) = t}{env \vdash Member(i, e) \Rightarrow Bool(false)}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash e \Rightarrow v, Type(v) = t}{env \vdash Member(i, e) \Rightarrow Bool(false)}$$

Subset:

$$\frac{env \vdash i1 \Rightarrow Set(t1, EmptySet), env \vdash i2 \Rightarrow Set(t2, e), t1 = t2}{env \vdash Subset(i1, i2) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i1 \Rightarrow Set(t1, Elements(eli1)), env \vdash i2 \Rightarrow Set(t2, Elements(eli2)), eli1 \subseteq eli2}{env \vdash Subset(i1, i2) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i1 \Rightarrow Set(t1, Elements(eli1)), env \vdash i2 \Rightarrow Set(t2, Elements(eli2)), eli1 \not\subseteq eli2, t1 = t2}{env \vdash Subset(i1, i2) \Rightarrow Bool(false)}$$

Min:

$$\frac{env \vdash i \Rightarrow Set(t, Elements(v :: evLis)), env \vdash e \Rightarrow v, \forall x \in evLis. v \leq x}{env \vdash Min(i) \Rightarrow v}$$

Max:

$$\frac{env \vdash i \Rightarrow Set(t, Elements(v :: evLis)), env \vdash e \Rightarrow v, \forall x \in evLis. v \geq x}{env \vdash Max(i) \Rightarrow v}$$

For_all:

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash p \Rightarrow Closure(-), \forall v \in evLis. p(v) = true}{env \vdash For_all(p, i) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash p \Rightarrow Closure(-), \exists v \in evLis. p(v) = false}{env \vdash For_all(p, i) \Rightarrow Bool(false)}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash p \Rightarrow Closure(-)}{env \vdash For_all(p, i) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash p \Rightarrow RecClosure(-), \forall v \in evLis. p(v) = true}{env \vdash For_all(p, i) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash p \Rightarrow RecClosure(-), \exists v \in evLis. p(v) = false}{env \vdash For_all(p, i) \Rightarrow Bool(false)}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash p \Rightarrow RecClosure(-)}{env \vdash For_all(p, i) \Rightarrow Bool(true)}$$

Exists:

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash p \Rightarrow Closure(-), \exists v \in evLis. p(v) = true}{env \vdash Exists(p, i) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash p \Rightarrow Closure(-), \nexists v \in evLis. p(v) = true}{env \vdash Exists(p, i) \Rightarrow Bool(false)}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash p \Rightarrow Closure(-)}{env \vdash Exists(p, i) \Rightarrow Bool(false)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash p \Rightarrow RecClosure(-), \exists v \in evLis. p(v) = true}{env \vdash Exists(p, i) \Rightarrow Bool(true)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(evLis)), env \vdash p \Rightarrow RecClosure(-), \nexists v \in evLis. p(v) = true}{env \vdash Exists(p, i) \Rightarrow Bool(false)}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash p \Rightarrow RecClosure(-)}{env \vdash Exists(p, i) \Rightarrow Bool(false)}$$

Filter:

$$\frac{env \vdash i \Rightarrow Set(t, Elements(eli1)), env \vdash p \Rightarrow Closure(-), \exists v \in eli1. p(v) = true}{env \vdash Filter(p, i) \Rightarrow Set(t, Elements(eli2)), \forall v \in eli2. v \in eli1 \wedge p(v) = true}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(e1i1)), env \vdash p \Rightarrow Closure(-), \nexists v \in eli1. p(v) = true}{env \vdash Filter(p, i) \Rightarrow Set(t, EmptySet)}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash p \Rightarrow Closure(-)}{env \vdash Filter(p, i) \Rightarrow Set(t, EmptySet)}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(e1i1)), env \vdash p \Rightarrow RecClosure(-), \exists v \in eli1. p(v) = true}{env \vdash Filter(p, i) \Rightarrow Set(t, Elements(eli2)), \forall v \in eli2. v \in eli1 \wedge p(v) = true}$$

$$\frac{env \vdash i \Rightarrow Set(t, Elements(e1i1)), env \vdash p \Rightarrow RecClosure(-), \nexists v \in eli1. p(v) = true}{env \vdash Filter(p, i) \Rightarrow Set(t, EmptySet)}$$

$$\frac{env \vdash i \Rightarrow Set(t, EmptySet), env \vdash p \Rightarrow RecClosure(-)}{env \vdash Filter(p, i) \Rightarrow Set(t, EmptySet)}$$

Map:

$$\frac{env \vdash i \Rightarrow Set(t1, Elements(evLis1)), env \vdash p \Rightarrow Closure(-)}{env \vdash Map(f, i) \Rightarrow Set(t2, Elements(evLis2)), \forall v \in evLis2. v = f(x) \text{ dove } x \in evLis1}$$

$$\frac{env \vdash i \Rightarrow Set(t1, EmptySet), env \vdash p \Rightarrow Closure(-)}{env \vdash Map(f, i) \Rightarrow Set(t2, EmptySet)}$$

$$\frac{env \vdash i \Rightarrow Set(t1, Elements(evLis1)), env \vdash p \Rightarrow RecClosure(-)}{env \vdash Map(f, i) \Rightarrow Set(t2, Elements(evLis2)), \forall v \in evLis2. v = f(x) \text{ dove } x \in evLis1}$$

$$\frac{env \vdash i \Rightarrow Set(t1, EmptySet), env \vdash p \Rightarrow RecClosure(-)}{env \vdash Map(f, i) \Rightarrow Set(t2, EmptySet)}$$

Dove t2 è un setType che dipende dal tipo del risultato di f.

Luca Lombardo
Mat. 546688
13/12/2020