Regole operazionali

Di seguito sono riportate le regole operazionali per le espressioni implementate nel linguaggio. Verrà adoperata la seguente sintassi:
- i ∈ Types per indicare che "i" appartiene all'insieme dei tipi leciti per gli insiemi;
- v ∈ Types(i) per indicare che "v" appartiene all'insieme dei valori di tipo "i";
- noDuplicate(lst) è una funzione che restituisce una lista senza duplicati, partendo dalla lista "lst";
- addList(list, v) è una funzione che aggiunge l'elemento "v" alla lista "list" e restituisce la lista risutante;
- l'operatore "=" è un predicato con notazione infissa che restituisce "true" sse i due valori sono uguali, "false" altrimenti;
- con "∪", "∩", "\" ci si riferisce, in ordine, agli operatori su insiemi "unione", "intersezione" e "differenza". Questi operatori possono avere come secondo operando anche un valore;

EmptySet:

$$\frac{i \in \text{Types}}{Env \rhd \text{EmptySet(i)} \Rightarrow Set(i, [])}$$

Singleton:

$$\frac{i \in \text{Types},\ \text{Env} \rhd \text{e} \Rightarrow \text{v1},\ \text{v1} \in \text{Types(i)}}{Env \rhd \text{Singleton(i, e)} \Rightarrow \text{Set(i, [v1])}}$$

Set:

$$\frac{i \in \text{Types},\ \text{e} \in \text{valuesList},\ \text{Env} \rhd \text{e} \Rightarrow \text{lst},\ \left(\forall \text{l} \in \text{lst}.\ \text{l} \in \text{Types(i)}\right)}{\text{Env} \rhd \text{Of(i, e)} \Rightarrow \text{Set(i, noDuplicate(lst))}}$$

Val:

$$\frac{Env \rhd \text{e} \Rightarrow \text{v},\ Env \rhd \text{valuesList} \Rightarrow \text{list},\ \text{addList(list, v)} \Rightarrow \text{lst}}{Env \rhd \text{Val(e, valuesList)} \Rightarrow \text{lst}}$$

Union:

$$\frac{\text{Env} \rhd \text{e1} \Rightarrow \text{Set(i, lst1)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{Set(j, lst2)},\ \text{i} = \text{j}}{\text{Env} \rhd \text{Union(e1, e2)} \Rightarrow \text{Set(i, lst1} \cup \text{lst2)}}$$

Intersect:

$$\frac{\text{Env} \rhd \text{e1} \Rightarrow \text{Set(i, lst1)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{Set(j, lst2)},\ \text{i} = \text{j}}{\text{Env} \rhd \text{Intersect(e1, e2)} \Rightarrow \text{Set(i, lst1} \cap \text{lst2)}}$$

Difference:

$$\frac{\text{Env} \rhd \text{e1} \Rightarrow \text{Set(i, lst1)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{Set(j, lst2)},\ \text{i} = \text{j}}{\text{Env} \rhd \text{Difference(e1, e2)} \Rightarrow \text{Set(i, lst1} \setminus \text{lst2)}}$$

Add:

$$\frac{\text{Env} \rhd \text{e1} \Rightarrow \text{Set(i, lst)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{v1},\ \text{v1} \in \text{Types(i)}}{\text{Env} \rhd \text{Add(e1, e2)} \Rightarrow \text{Set(i, lst} \cup \text{v1)}}$$

Remove:

$$\frac{\text{Env} \rhd \text{e1} \Rightarrow \text{Set(i, lst)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{v1},\ \text{v1} \in \text{Types(i)}}{\text{Env} \rhd \text{Remove(e1, e2)} \Rightarrow \text{Set(i, lst} \setminus \text{v1)}}$$

IsEmpty:

$$\frac{\text{Env} \rhd \text{e1} \Rightarrow \text{Set(i, lst)}}{\text{Env} \rhd \text{IsEmpty(e1)} \Rightarrow ((\text{lst} \cup [\,]) = [\,])}$$

Contains:

$$\frac{\text{Env} \rhd \text{e1} \Rightarrow \text{Set(i, lst)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{v1},\ \text{v1} \in \text{Types(i)}}{\text{Env} \rhd \text{Contains(e1, e2)} \Rightarrow ((\text{lst} \cap \text{v1}) = \text{v1})}$$

SubSet:

$$\frac{\text{Env} \rhd \text{e1} \Rightarrow \text{Set(i, lst1)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{Set(j, lst2)},\ \text{i} = \text{j}}{\text{Env} \rhd \text{SubSet(e1, e2)} \Rightarrow ((\text{lst1} \cap \text{lst2}) = lst2)}$$

Max:

$$\frac{\text{Env} \rhd \text{e} \Rightarrow \text{Set(i, lst)},\ (\exists\ \text{elem} \in \text{lst}.\ (\forall\ \text{l} \in\ \text{lst}.\ \text{elem} \geq \text{l}))}{\text{Env} \rhd \text{Max(e)} \Rightarrow \text{elem}}$$

Min:

$$\frac{\text{Env} \rhd \text{e} \Rightarrow \text{Set(i, lst)}, (\exists\ \text{elem} \in \text{lst}.\ (\forall\ \text{l} \in\ \text{lst}.\ \text{elem} \leq \text{l}))}{\text{Env} \rhd \text{Min(e)} \Rightarrow \text{elem}}$$

For_all:

$$\frac{\begin{array}{c}\text{Env} \rhd \text{e1} \Rightarrow \text{Closure(x, body, fDecEnv)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{Set(i, lst)},\\ \left(\forall\ \text{l} \in\ \text{lst}.\ fDecEnv\left[{}^{l}/_{x}\right] \rhd \text{body} \Rightarrow \text{v} \in \text{Types("bool")}\right),\ (\forall\ \text{v}\,.\,\text{v} = \text{Bool(true)}) \Rightarrow \text{res}\end{array}}{\text{Env} \rhd \text{For\_all(e1, e2)} \Rightarrow \text{res}}$$

Exists:

$$\frac{\begin{array}{c}\text{Env} \rhd \text{e1} \Rightarrow \text{Closure(x, body, fDecEnv)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{Set(i, lst)},\\ \left(\forall\ \text{l} \in\ \text{lst}.\ fDecEnv\left[{}^{l}/_{x}\right] \rhd \text{body} \Rightarrow \text{v} \in \text{Types("bool")}\right),\ (\exists\ \text{v}\,.\,\text{v} = \text{Bool(true)}) \Rightarrow \text{res}\end{array}}{\text{Env} \rhd \text{Exists(e1, e2)} \Rightarrow \text{res}}$$

Filter:

$$\frac{\begin{array}{c}\text{Env} \rhd \text{e1} \Rightarrow \text{Closure(x, body, fDecEnv)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{Set(i, lst)},\\ \left(\forall\ \text{l} \in\ \text{lst}.\ fDecEnv\left[{}^{l}/_{x}\right] \rhd \text{body} \Rightarrow \text{v} \in \text{Types("bool")}\right),\ (\forall\ \text{l} \in\ \text{lst}.\ (\text{v} = \text{Bool(true)}) \Rightarrow \text{l}) \Rightarrow \text{res}\end{array}}{\text{Env} \rhd \text{Filter(e1, e2)} \Rightarrow \text{res}}$$

Map:

$$\frac{\begin{array}{c}\text{Env} \rhd \text{e1} \Rightarrow \text{Closure(x, body, fDecEnv)},\ \text{Env} \rhd \text{e2} \Rightarrow \text{Set(i, lst)},\\ (\forall\ \text{l} \in\ \text{lst}.\ fDecEnv[{}^{l}/_{x}] \rhd \text{body} \Rightarrow \text{v}) \Rightarrow \text{res}\end{array}}{\text{Env} \rhd \text{Map(e1, e2)} \Rightarrow \text{res}}$$