Regole operazionali

Ivan Lo Greco Mat.597002

December 14, 2020

$$\textsc{ESet}\quad \frac{type \in TypesSet \quad \forall x \in elist.(guessType(env \triangleright x) = type))}{env \triangleright ESet(elist, type) \Rightarrow Set(env \triangleright elist, type)}$$

$$\textsc{Empty}\quad \frac{type \in TypesSet}{env \triangleright Empty(type) \Rightarrow Set(\emptyset, type)}$$

$$\textsc{Singleton}\quad \frac{type \in TypesSet \quad guessType(env \triangleright e) = type}{env \triangleright Singleton(type, e) \Rightarrow Set(\emptyset \cup env \triangleright e, type)}$$

$$\textsc{Of}\quad \frac{type \in TypesSet \quad \forall x \in elist.(guessType(env \triangleright x) = type))}{env \triangleright Of(type, elist) \Rightarrow Set(env \triangleright elist, type)}$$

$$\textsc{Insert}\quad \dfrac{\dfrac{env \triangleright e \Rightarrow v \quad guessType(v) = type}{env \triangleright set \Rightarrow Set(list, type) \quad type \in TypesSet}}{env \triangleright Insert(set, e) \Rightarrow Set(list \cup \{v\}, type)}$$

$$\textsc{Remove}\quad \dfrac{\dfrac{env \triangleright e \Rightarrow v \quad guessType(v) = type}{env \triangleright set \Rightarrow Set(list, type) \quad type \in TypesSet}}{env \triangleright Remove(set, e) \Rightarrow Set(list \setminus \left\{v\right\}, type)}$$

$$\textsc{IsEmpty}\quad \frac{env \triangleright set \Rightarrow Set(list, type) \quad list \subseteq \emptyset \Rightarrow b}{env \triangleright IsEmpty(set) \Rightarrow Bool(b)}$$

$$\textsc{IsSubset}\quad \dfrac{\dfrac{\dfrac{type1 = type2 \quad list1 \subseteq list2 \Rightarrow b}{env \triangleright set2 \Rightarrow Set(list2, type2) \quad type2 \in TypesSet}}{env \triangleright set1 \Rightarrow Set(list1, type1) \quad type1 \in TypesSet}}{env \triangleright IsSubset(set1, set2) \Rightarrow Bool(b)}$$

$$\textsc{Contains}\quad \dfrac{\dfrac{env \triangleright e \Rightarrow v \quad guessType(v) = type \quad v \in list \Rightarrow b}{env \triangleright set \Rightarrow Set(list, type) \quad type \in TypesSet}}{env \triangleright Contains(set, e) \Rightarrow Bool(b)}$$

$$\frac{\mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{Set}(\mathit{list}, \mathit{type}) \quad \mathit{type} \in \mathit{TypesSet} \quad \exists! x \in \mathit{list}.(\forall y \in \mathit{list} \setminus \{x\}.(x < y)))}{\mathit{env} \triangleright \mathit{Min}(\mathit{set}) \Rightarrow v \mid \mathit{Unbound}}\ \textsc{Min}$$

$$\frac{\mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{Set}(\mathit{list}, \mathit{type}) \quad \mathit{type} \in \mathit{TypesSet} \quad \exists! x \in \mathit{list}.(\forall y \in \mathit{list} \setminus \{x\}.(x > y)))}{\mathit{env} \triangleright \mathit{Max}(\mathit{set}) \Rightarrow v \mid \mathit{Unbound}}\ \textsc{Max}$$

$$\textsc{UnionSet}\ \frac{\dfrac{\dfrac{\mathit{type}1 = \mathit{type}2}{\mathit{env} \triangleright \mathit{set}2 \Rightarrow \mathit{Set}(\mathit{list}2, \mathit{type}2) \quad \mathit{type}2 \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{set}1 \Rightarrow \mathit{Set}(\mathit{list}1, \mathit{type}1) \quad \mathit{type}1 \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{Union}(\mathit{set}1, \mathit{set}2) \Rightarrow \mathit{Set}(\mathit{list}1 \cup \mathit{list}2, \mathit{type}1)}$$

$$\frac{\dfrac{\dfrac{\mathit{type}1 = \mathit{type}2}{\mathit{env} \triangleright \mathit{set}2 \Rightarrow \mathit{Set}(\mathit{list}2, \mathit{type}2) \quad \mathit{type}2 \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{set}1 \Rightarrow \mathit{Set}(\mathit{list}1, \mathit{type}1) \quad \mathit{type}1 \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{IntersectionSet}(\mathit{set}1, \mathit{set}2) \Rightarrow \mathit{Set}(\mathit{list}1 \cap \mathit{list}2, \mathit{type}1)}\ \textsc{IntersectionSet}$$

$$\textsc{SubtractSet}\ \frac{\dfrac{\dfrac{\mathit{type}1 = \mathit{type}2}{\mathit{env} \triangleright \mathit{set}2 \Rightarrow \mathit{Set}(\mathit{list}2, \mathit{type}2) \quad \mathit{type}2 \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{set}1 \Rightarrow \mathit{Set}(\mathit{list}1, \mathit{type}1) \quad \mathit{type}1 \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{SubtractSet}(\mathit{set}1, \mathit{set}2) \Rightarrow \mathit{Set}(\mathit{list}2 - \mathit{list}1, \mathit{type}1)}$$

$$\frac{\dfrac{\dfrac{\forall x \in \mathit{list}.(\mathit{applyFun}(\mathit{env} \triangleright \mathit{pred}, x) = \mathit{Bool}(\mathit{true})) \Rightarrow b}{\mathit{env} \triangleright \mathit{pred} \Rightarrow \mathit{FunClosure}(\mathit{param}, \mathit{body}, \mathit{ev}) \mid \mathit{RecClosure}(\mathit{id}, \mathit{param}, \mathit{body}, \mathit{ev})}}{\mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{Set}(\mathit{list}, \mathit{type}) \quad \mathit{type} \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{ForAll}(\mathit{pred}, \mathit{set}) \Rightarrow \mathit{Bool}(b)}\ \textsc{ForAll}$$

$$\frac{\dfrac{\dfrac{\exists x \in \mathit{list}.(\mathit{applyFun}(\mathit{env} \triangleright \mathit{pred}, x) = \mathit{Bool}(\mathit{true})) \Rightarrow b}{\mathit{env} \triangleright \mathit{pred} \Rightarrow \mathit{FunClosure}(\mathit{param}, \mathit{body}, \mathit{ev}) \mid \mathit{RecClosure}(\mathit{id}, \mathit{param}, \mathit{body}, \mathit{ev})}}{\mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{Set}(\mathit{list}, \mathit{type}) \quad \mathit{type} \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{Exists}(\mathit{pred}, \mathit{set}) \Rightarrow \mathit{Bool}(b)}\ \textsc{Exists}$$

$$\frac{\dfrac{\dfrac{\mathit{result} = \{x \in \mathit{list} \mid \mathit{applyFun}(\mathit{env} \triangleright \mathit{pred}, x) = \mathit{Bool}(\mathit{true})\}}{\mathit{env} \triangleright \mathit{pred} \Rightarrow \mathit{FunClosure}(\mathit{param}, \mathit{body}, \mathit{ev}) \mid \mathit{RecClosure}(\mathit{id}, \mathit{param}, \mathit{body}, \mathit{ev})}}{\mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{Set}(\mathit{list}, \mathit{type}) \quad \mathit{type} \in \mathit{TypesSet}}}{\mathit{env} \triangleright \mathit{Filter}(\mathit{pred}, \mathit{set}) \Rightarrow \mathit{Set}(\mathit{result}, \mathit{type})}\ \textsc{Filter}$$

$$\begin{array}{c}
\textsc{Map} \\
\forall newX \in result.(guessType(newX) = newType) \\
\hline
result = \{x \in list \mid applyFun(env \triangleright pred, x) \in TypesSet\} \quad TypeSet(result) \Rightarrow newType \\
\hline
env \triangleright pred \Rightarrow FunClosure(param, body, ev) \mid RecClosure(id, param, body, ev) \\
\hline
env \triangleright set \Rightarrow Set(list, type) \quad type \in TypesSet \\
\hline
env \triangleright Map(pred, set) \Rightarrow Set(result, newType)
\end{array}$$