Regole Operazionali
Roberta Zucca Mat.560158

SET

$$\begin{array}{c} env \triangleright list \Rightarrow elist \\ \hline isPrimitive(type)\ \land\ \ \forall\, v \in\ elist\ typecheck(type,v) \\ \hline env \triangleright Set(type,list) \Rightarrow Valset(type,elist) \end{array}$$

EMPTYSET

$$\begin{array}{c} isPrimitive(type) \\ \hline env \triangleright EmptySet(type) \Rightarrow Valset(type,[]) \end{array}$$

SINGLETON

$$\begin{array}{c} isPrimitive(type) \\ \hline env \triangleright v \Rightarrow ev \\ \hline typecheck(type,ev) \\ \hline env \triangleright Singleton(type,v) \Rightarrow Valset(type,ev) \end{array}$$

HAS_ELEMENT

$$\begin{array}{c} isSet(Set) \qquad env \triangleright set \Rightarrow ValSet(type,elist) \qquad env \triangleright el \Rightarrow ev \\ \hline isPrimitive(type) \land\ typecheck(type,ev) \\ \hline \exists v1 \in values \Rightarrow true \qquad !(\ \exists v1 \in values) \Rightarrow false \\ \hline env \triangleright Has\_Element(el,set) \Rightarrow b \end{array}$$

INSERT

$$\begin{array}{c} isSet(Set) \qquad env \triangleright set \Rightarrow ValSet(type,elist) \qquad env \triangleright el \Rightarrow ev \\ \hline isPrimitive(type) \land\ typecheck(type,ev) \\ \hline !(\ \exists ev \in elist).\ newlist=newlist \cup \{ev\} \\ \hline env \triangleright Insert(el,set) \Rightarrow ValSet(type,newlist) \end{array}$$

REMOVE

$$\begin{array}{c} isSet(Set) \qquad env \triangleright set \Rightarrow ValSet(type,elist) \qquad env \triangleright el \Rightarrow ev \\ \hline isPrimitive(type) \land\ typecheck(type,ev) \\ \hline (\ \exists ev \in elist).\ newlist=newlist - \{ev\} \\ \hline env \triangleright Remove(el,set) \Rightarrow\ ValSet(type,newlist) \end{array}$$

Regole Operazionali
Roberta Zucca Mat.560158

IS_EMPTY

$$\begin{array}{c}
\mathit{isSet(Set)} \qquad \mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{ValSet(type,elist)} \\ \hline
\mathit{elist=[\,]} \Rightarrow \mathit{true} \qquad \mathit{elist\ !=[\,]} \Rightarrow \mathit{false} \\ \hline
\mathit{env} \triangleright \mathit{isEmpty(set)} \Rightarrow b
\end{array}$$

IS_SUBSET

$$\begin{array}{c}
\mathit{isSet(set1)} \wedge \mathit{isSet(set2)} \\ \hline
\mathit{env} \triangleright \mathit{set1} \Rightarrow \mathit{ValSet(type1,elist1)} \qquad \mathit{env} \triangleright \mathit{set2} \Rightarrow \mathit{ValSet(type2,elist2)} \\ \hline
\mathit{isPrimitive(type1)} \wedge \mathit{type1=type2} \\ \hline
\forall v \in \mathit{elist1}\ \exists w \in \mathit{elist2}\ |\ v = w \Rightarrow \mathit{true} \qquad !(\forall v \in \mathit{elist1}\ \exists w \in \mathit{elist2}\ |\ v = w \Rightarrow \mathit{false}) \\ \hline
\mathit{env} \triangleright \mathit{isSubset(set1,set2)} \Rightarrow b
\end{array}$$

MIN

$$\begin{array}{c}
\mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{ValSet(type,elist)} \\ \hline
\mathit{isPrimitive(type)} \\ \hline
\exists w \in \mathit{elist}.\ \forall v \in \mathit{elist},\ w \leq v \\ \hline
\mathit{env} \triangleright \mathit{Min(set)} \Rightarrow w
\end{array}$$

MAX

$$\begin{array}{c}
\mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{ValSet(type,elist)} \\ \hline
\mathit{isPrimitive(type)} \\ \hline
\exists w \in \mathit{elist}.\ \forall v \in \mathit{elist},\ w \geq v \\ \hline
\mathit{env} \triangleright \mathit{Max(set)} \Rightarrow w
\end{array}$$

MAP

$$\begin{array}{c}
\mathit{env} \triangleright \mathit{set} \Rightarrow \mathit{ValSet(type,elist)} \qquad \mathit{isPrimitive(type)} \\ \hline
\mathit{env} \triangleright \mathit{funct} \Rightarrow \mathit{Closure(arg,fbody,s)} \vee \mathit{RecClosure(f,arg,fBody,s)} \\ \hline
\forall v\_i \in \mathit{elist}\ \ \mathit{env} \triangleright \mathit{Apply(funct,v\_i)} \Rightarrow e\_i \\ \hline
\forall e\_i\ \mathit{newlist= newlist}\ U\ \{ei\} \\ \hline
\mathit{env} \triangleright \mathit{Map(funct,set)} \Rightarrow \mathit{ValSet(type,newlist)}
\end{array}$$

Regole Operazionali
Roberta Zucca Mat.560158

FILTER

$$\begin{array}{c}
\textit{env} \triangleright \textit{set} \Rightarrow \textit{ValSet(type,elist)} \qquad \textit{isPrimitive(type)} \\ \hline
\textit{env} \triangleright \textit{pred} \Rightarrow \textit{Closure(arg,fbody,s)} \vee \textit{RecClosure(f,arg,fBody,s)} \\ \hline
\forall \textit{v\_i} \in \textit{elist} \ \ \textit{env} \triangleright \textit{Apply(funct,v\_i)} \Rightarrow \textit{e\_i} \\ \hline
\forall \textit{e\_i . e\_i= true : newlist=newlist U \{v\_i\}} \\ \hline
\textit{env} \triangleright \textit{Filter(pred,set)} \Rightarrow \textit{ValSet(type,newlist)}
\end{array}$$

FORALL

$$\begin{array}{c}
\textit{env} \triangleright \textit{set} \Rightarrow \textit{ValSet(type,elist)} \qquad \textit{isPrimitive(type)} \\ \hline
\textit{env} \triangleright \textit{pred} \Rightarrow \textit{Closure(arg,fbody,s)} \vee \textit{RecClosure(f,arg,fBody,s)} \\ \hline
\forall \textit{v\_i} \in \textit{elist} \ \ \textit{env} \triangleright \textit{Apply(funct,v\_i)} \Rightarrow \textit{e\_i} \\ \hline
\forall \textit{e\_i , e\_j | i !=j} \ \ \textit{e\_i and e\_j} \\ \hline
\textit{env} \triangleright \textit{ForAll(pred,set)} \Rightarrow \textit{b}
\end{array}$$

EXISTS

$$\begin{array}{c}
\textit{env} \triangleright \textit{set} \Rightarrow \textit{ValSet(type,elist)} \qquad \textit{isPrimitive(type)} \\ \hline
\textit{env} \triangleright \textit{pred} \Rightarrow \textit{Closure(arg,fbody,s)} \vee \textit{RecClosure(f,arg,fBody,s)} \\ \hline
\forall \textit{v\_i} \in \textit{elist} \ \ \textit{env} \triangleright \textit{Apply(funct,v\_i)} \Rightarrow \textit{e\_i} \\ \hline
\forall \textit{e\_i , e\_j | i !=j} \ \ \textit{e\_i or e\_j} \\ \hline
\textit{env} \triangleright \textit{Exists(pred,set)} \Rightarrow \textit{b}
\end{array}$$