



UNIVERSITÀ DI PISA

Dipartimento di Informatica
Corso di Laurea in Informatica

Regole di semantica operativa

Programmazione II

Prof. Gianluigi Ferrari

Giacomo Trapani
600124 - Corso A

Anno Accademico 2020/2021

Creazione di insiemi.¹

Empty

$$\frac{(str \in \{"int", "string", "bool"\})}{env \triangleright Empty(str) \Longrightarrow Set(str, \emptyset)}$$

Singleton

$$\frac{(\Gamma = (env \triangleright aElem) : str) \quad (str \in \{"int", "string", "bool"\})}{env \triangleright Singleton(aElem, str) \Longrightarrow Set(str, [env \triangleright aElem])}$$

Of

$$\frac{\frac{(str \in \{"int", "string", "bool"\}) \quad (\forall L \in aLis. (\exists! M \in aLis \mid M = L))}{(\forall L \in aLis. \Gamma = (env \triangleright L) : str) \quad (\forall L \in aLis. (env \triangleright L) \in eLis)}}{env \triangleright Of(str, aLis) \Longrightarrow Set(str, eLis)}$$

Operazioni su insiemi.

Add

$$\frac{(env \triangleright aSet \Longrightarrow Set(str, eLis)) \quad ((\Gamma = (env \triangleright aElem) : str) \Longrightarrow v = Set(str, [env \triangleright aElem]) \cup (env \triangleright aSet))}{\frac{(\neg(\Gamma = (env \triangleright aElem) : str)) \Longrightarrow v = (env \triangleright aSet)}{env \triangleright Add(aSet, aElem) \Longrightarrow v}}$$

Remove

$$\frac{(env \triangleright aSet \Longrightarrow Set(str, eLis))}{env \triangleright Remove(aSet, aElem) \Longrightarrow ((env \triangleright aSet) \setminus \{env \triangleright aElem\})}$$

IsEmpty

$$\frac{(env \triangleright aSet \Longrightarrow Set(str, eLis)) \quad ((eLis \neq \emptyset) \Longrightarrow v = false)}{\frac{(eLis = \emptyset) \Longrightarrow v = true}{env \triangleright IsEmpty(aSet) \Longrightarrow v}}$$

BelongsTo

$$\frac{(env \triangleright aSet \Longrightarrow Set(str, eLis)) \quad (\neg((env \triangleright aElem) \in eLis) \Longrightarrow v = false)}{\frac{((env \triangleright aElem) \in eLis) \Longrightarrow v = true}{env \triangleright BelongsTo(aSet, aElem) \Longrightarrow v}}$$

¹Per ragioni di spazio, alcune regole operazionali presentano premesse divise in più righe; si segnala che in questi casi l'unica e sola conclusione è definita dall'ultima riga.

IsSubset

$$\frac{(env \triangleright aSet1 \implies Set(str1, eLis1)) \quad (env \triangleright aSet2 \implies (str2, eLis2))}{((\forall L \in eLis1. L \in eLis2) \implies v = true) \quad ((\exists L \in eLis1 \mid L \notin eLis2) \implies v = false)} env \triangleright IsSubset(aSet1, aSet2) \implies v$$

Maximum

$$\frac{(env \triangleright aSet \implies Set(str, eLis) \quad (eLis \neq \emptyset) \quad (\exists L \in eLis \mid (\forall M \in eLis. L \geq M) \implies v = L))}{env \triangleright Maximum(aSet) \implies v}$$

Minimum

$$\frac{(env \triangleright aSet \implies Set(str, eLis) \quad (eLis \neq \emptyset) \quad (\exists L \in eLis \mid (\forall M \in eLis. L \leq M) \implies v = L))}{env \triangleright Minimum(aSet) \implies v}$$

Operatori funzionali.²

For_all

Caso 1: insiemi non vuoti

$$\frac{\frac{(env \triangleright predicate \implies Closure(x, body, fDecEnv))}{(env \triangleright aSet \implies Set(str, eLis)) \quad (eLis \neq \emptyset)} \quad \frac{((\forall L \in eLis. fDecEnv[(env \triangleright L)/x] \triangleright body \implies true) \implies v = true)}{((\exists L \in eLis \mid fDecEnv[(env \triangleright L)/x] \triangleright body \implies false) \implies v = false)}} env \triangleright For_all(predicate, aSet) \implies v$$

Caso 2: insiemi vuoti

$$\frac{(env \triangleright predicate \implies Closure(x, body, fDecEnv))}{(env \triangleright aSet \implies Set(str, eLis) \quad (eLis = \emptyset))} env \triangleright For_all(predicate, aSet) \implies true$$

Exists

Caso 1: insiemi non vuoti

$$\frac{\frac{(env \triangleright predicate \implies Closure(x, body, fDecEnv))}{(env \triangleright aSet \implies Set(str, eLis)) \quad (eLis \neq \emptyset)} \quad \frac{((\exists L \in eLis \mid fDecEnv[(env \triangleright L)/x] \triangleright body \implies true) \implies v = true)}{((\forall L \in eLis. fDecEnv[(env \triangleright L)/x] \triangleright body \implies false) \implies v = false)}} env \triangleright Exists(predicate, aSet) \implies v$$

Caso 2: insiemi vuoti

$$\frac{(env \triangleright predicate \implies Closure(x, body, fDecEnv))}{(env \triangleright aSet \implies Set(str, eLis) \quad (eLis = \emptyset))} env \triangleright Exists(predicate, aSet) \implies false$$

²Si sceglie, per evitare di avere regole operazionali particolarmente lunghe e di difficile lettura, di dividere - quando necessario - le regole in più casi (nello specifico si distinguono le operazioni e i controlli da fare nel caso di operatori funzionali applicati a insiemi vuoti e non.)

Filter

Caso 1: insiemi non vuoti

$$\frac{\frac{\frac{(env \triangleright predicate \implies Closure(x, body, fDecEnv))}{(env \triangleright aSet \implies Set(str, eLis)) \quad (eLis \neq \emptyset)}}{(\forall L \in eLis. (fDecEnv[(env \triangleright L)/x] \triangleright body \implies true) \implies L \in v)}}{env \triangleright Filter(predicate, aSet) \implies v, \text{ con } \Gamma \vdash v : Set}$$

Caso 2: insiemi vuoti

$$\frac{\frac{(env \triangleright predicate \implies Closure(x, body, fDecEnv))}{(env \triangleright aSet \implies Set(str, eLis)) \quad (eLis = \emptyset)}}{env \triangleright Filter(predicate, aSet) \implies \emptyset}$$

Map

$$\frac{\frac{\frac{(env \triangleright predicate \implies Closure(x, body, fDecEnv))}{(env \triangleright aSet \implies Set(str, eLis)) \quad (eLis \neq \emptyset)}}{(\forall L \in eLis. (fDecEnv[(env \triangleright L)/x] \triangleright body \implies res) \implies res \in v)}}{env \triangleright Map(func, aSet) \implies v, \text{ con } \Gamma \vdash v : Set}$$