



Scelte progettuali

Riferimento	
Versione	
Data	04/11/2019
Destinatario	Prof. G. Costagliola
Presentato da	Giammaria Giordano, Valeria Pontillo
Approvato da	



Modifiche apportate a livello sintattico del compilatore

Non sono state create delle produzioni *Arith_op*, *Bool_op* e *Rel_op* che riportassero agli operatori delle diverse operazioni, ma è stata creata una produzione per ogni operatore sotto *Expr*.

Modifiche apportate a livello semantico del compilatore

I tipi *IN*, *OUT* e *INOUT* del nodo *ParType* sono stati trattati nel seguente modo:

- *IN*: sono dichiarate di tipo *IN* le variabili che possono essere solo lette ma non possono essere modificate;
- *OUT*: sono dichiarate di tipo *OUT* le variabili che non possono essere lette ma possono essere modificate;
- *INOUT*: sono dichiarate di tipo *INOUT* le variabili che possono sia essere lette che modificate

Type System

Per le diverse operazioni aritmetiche, booleane e relazioni sono state create delle tabelle di compatibilità tra i tipi. Per quanto riguarda le espressioni aritmetiche, sono state create due tabelle diverse: una per le addizioni e una per tutte le altre operazioni. Per quanto riguarda le espressioni booleane, sono state create due tabelle, una per l'AND e una per l'OR. Infine, per quanto riguarda le espressioni relazionali, sono state create due tabelle, una per l'operatore *EQ* (*=*) e una per tutti gli altri operatori. Di seguito sono riportate le diverse tabelle.

Add Operation

	Integer	Double	String	Char	Bool
Integer	Integer	Double	String	Integer	Undefined
Double	Double	Double	String	Undefined	Undefined



String	String	String	String	String	String
Char	Integer	Undefined	String	Integer	Undefined
Bool	Undefined	Undefined	String	Undefined	Undefined

Arith Operation

	Integer	Double	String	Char	Bool
Integer	Integer	Double	Undefined	Undefined	Undefined
Double	Double	Double	Undefined	Undefined	Undefined
String	Undefined	Undefined	Undefined	Undefined	Undefined
Char	Undefined	Undefined	Undefined	Undefined	Undefined
Bool	Undefined	Undefined	Undefined	Undefined	Undefined



Eq Operation

	Integer	Double	String	Char	Bool
Integer	Boolean	Boolean	Undefined	Undefined	Undefined
Double	Boolean	Boolean	Undefined	Undefined	Undefined
String	Undefined	Undefined	Boolean	Undefined	Undefined
Char	Undefined	Undefined	Undefined	Boolean	Undefined
Bool	Undefined	Undefined	Undefined	Undefined	Boolean

Relop Operation

	Integer	Double	String	Char	Bool
Integer	Boolean	Boolean	Undefined	Undefined	Undefined
Double	Boolean	Boolean	Undefined	Undefined	Undefined



String	Undefined	Undefined	Boolean	Undefined	Undefined
Char	Undefined	Undefined	Undefined	Boolean	Undefined
Bool	Undefined	Undefined	Undefined	Undefined	Undefined

Uminus Operation

Integer	Double	String	Char	Boolean
Integer	Double	Undefined	Undefined	Undefined

NotOperation

Integer	Double	String	Char	Boolean
Undefined	Undefined	Undefined	Undefined	Boolean

BoolOperation (And e Or Operation)

	Integer	Double	String	Char	Bool
Integer	Undefined	Undefined	Undefined	Undefined	Undefined
Double	Undefined	Undefined	Undefined	Undefined	Undefined
String	Undefined	Undefined	Undefined	Undefined	Undefined



Char	Undefined	Undefined	Undefined	Undefined	Undefined
Bool	Undefined	Undefined	Undefined	Undefined	Boolean

Assign Operation

	Integer	Double	String	Char	Bool
Integer	Integer	Integer	Undefined	Integer	Undefined
Double	Double	Double	Undefined	Undefined	Undefined
String	Undefined	Undefined	String	Undefined	Undefined
Char	Char	Undefined	Undefined	Char	Undefined
Bool	Undefined	Undefined	Undefined	Undefined	Boolean

Type System nodo ReadOperation

IF il tipo del nodo figlio è tipo compatibile con operatore

THEN nodo.type= getVars.type

ELSE restituisci “Error”

Type System nodo Vars

IF il tipo del nodo figlio è tipo compatibile con operatore



THEN nodo.type= VOID
ELSE restituisci “Error”

Type System nodo Args

IF il tipo del nodo figlio è tipo compatibile con operatore
THEN nodo.type= VOID
ELSE restituisci “Error”

Type System nodo WriteOperation

IF il tipo del nodo figlio è tipo compatibile con operatore
THEN nodo.type= getArgs.type
ELSE restituisci “Error”

Type System nodi CallOperationWithParams, CallOperationWithoutParams

IF il dominio della funzione coincide con la firma del metodo
THEN nodo.type=VOID
ELSE restituisci “Error”

Type System nodo VarInitValue

IF il tipo dell’espressione è compatibile con il tipo della variabile
THEN nodo.type=VOID
ELSE restituisci “Error”