Università –redacted–
Dipartimento di Informatica

Type system del linguaggio NewLang

Autori
Carbonic

Anno Accademico 2022-2023

Indice

Capitolo 1

Regole di tipo

1.1 Espressioni

1.1.1 Identificatore

$$\frac{\Gamma(id)=\tau}{\Gamma\vdash id:\tau}$$

1.1.2 Costanti

$\Gamma \vdash IntegerConstExpr : integer$
$\Gamma \vdash RealConstExpr : real$
$\Gamma \vdash StringConstExpr : string$
$\Gamma \vdash CharConstExpr : char$
$\Gamma \vdash TrueConstExpr : boolean$
$\Gamma \vdash FalseConstExpr : boolean$

1.1.3 FunCall: Chiamata a funzione

Con tipo di ritorno

$$\frac{\Gamma\vdash f:\tau_1\times...\times\tau_n\rightarrow\tau \;\; \Gamma\vdash Expr_i:\tau_i^{i\in 1..n}}{\Gamma\vdash f(Expr_1,...,Expr_n):\tau}$$

Senza tipo di ritorno

$$\frac{\Gamma\vdash f:\tau_1\times...\times\tau_n\rightarrow notype \Gamma\vdash Expr_i:\tau_i^{i\in 1..n}}{\Gamma\vdash f(Expr_1,...,Expr_n):notype}$$

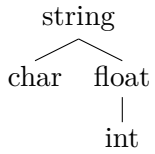
1.1.4 Operatori unari

$$\frac{\Gamma \vdash e{:}\tau_1 \; optype1(op_1, \tau_1) = \tau}{\Gamma \vdash op_1 \;\; e{:}\tau}$$

1.1.5 Tabella op_1

op_1	operando	risultato
MinusExpr	integer	integer
MinusExpr	float	float
NotExpr	boolean	boolean
Cast<string>	char	string
Cast<string>	float	string
Cast<float>	integer	float
Cast<string>	integer	string

Nella tabella precedente è incluso anche l'operatore Cast, poiché matematicamente rappresenta una operazione unaria. Per facilitare la comprensione, si può vedere il seguente albero che mostra le regole di type widening permesse dal linguaggio.



1.1.6 Operatori binari

$$\frac{\Gamma \vdash e_1{:}\tau_1 \;\; \Gamma \vdash e_2{:}\tau_2 \;\; optype2(op_2, \tau_1, \tau_2) = \tau}{\Gamma \vdash e_1 op_2 e_2{:}\tau}$$

1.1.7 Tabella op_2

op	operando1	operando2	risultato
AddExpr, SubExpr, TimesExpr, DivExpr, PowExpr	integer	integer	integer
AddExpr, SubExpr, TimesExpr, DivExpr, PowExpr	integer	float	float
AddExpr, SubExpr, TimesExpr, DivExpr, PowExpr	float	integer	float
AddExpr, SubExpr, TimesExpr, DivExpr, PowExpr	float	float	float
StrConcatExpr	string	string	string
StrConcatExpr	string	int	string
StrConcatExpr	int	string	string
StrConcatExpr	int	int	string
StrConcatExpr	string	float	string
StrConcatExpr	float	string	string
StrConcatExpr	float	float	string
StrConcatExpr	string	char	string
StrConcatExpr	char	string	string
StrConcatExpr	char	char	string
StrConcatExpr	string	bool	string
StrConcatExpr	bool	string	string
StrConcatExpr	bool	bool	string
AndExpr, OrExpr	boolean	boolean	boolean
GTExpr, GEExpr, LTExpr, LEExpr, EQExpr, NEExpr	integer	integer	boolean
GTExpr, GEExpr, LTExpr, LEExpr, EQExpr, NEExpr	integer	float	boolean
GTExpr, GEExpr, LTExpr, LEExpr, EQExpr, NEExpr	float	integer	boolean
GTExpr, GEExpr, LTExpr, LEExpr, EQExpr, NEExpr	float	float	boolean
EQExpr, NEExpr	string	string	boolean

1.2 Istruzioni

1.1.1 IfStat: Istruzione if-then

Senza Else

$$\frac{\Gamma \vdash Expr : boolean \quad \Gamma \vdash Body : notype}{\Gamma \vdash \text{if Expr then Body: notype}}$$

Con Else

$$\frac{\Gamma \vdash Expr : boolean \quad \Gamma \vdash Body_{if} : notype \quad \Gamma \vdash Body_{else} : notype}{\Gamma \vdash \text{if Expr then } Body_{if} \text{else} Body_{else} : \text{notype}}$$

1.2.2 ForStat: Istruzione for

$$\frac{\Gamma\vdash ICExpr_1{:}integer\ \ \Gamma\vdash ICExpr_2{:}integer \Gamma[id\mapsto integer]\vdash Body{:}notype}{\Gamma\vdash \text{for id}\ll ICExpr_1 \text{ to } ICExpr_2 \text{ loop Body: notype}}$$

Dove $ICExpr = IntegerConstExpr$ e $id = Identifier$

1.2.3 ReadStat: Istruzione read

Senza messaggio di output

$$\frac{\Gamma(Identifier_i){:}\tau_i^{i\in 1\ldots n}}{\Gamma\vdash Identifier_1,\ldots,Identifier_n<{-}{-}\ :notype}$$

Con messaggio di output

$$\frac{\Gamma(Identifier_i){:}\tau_i^{i\in 1\ldots n}\ \ \Gamma\vdash StringConstExpr_1{:}string}{\Gamma\vdash Identifier_1,\ldots,Identifier_n<{-}{-}StringConstExpr_1\ :notype}$$

1.2.4 WriteStat: Istruzione write

Senza newline

$$\frac{\Gamma\vdash Expr{:}\tau_i^{i\in 1\ldots n}}{\Gamma\vdash (Expr_1,\ldots,Expr_n){-}{-}{>}\ :notype}$$

Con newline

$$\frac{\Gamma\vdash Expr{:}\tau_i^{i\in 1\ldots n}}{\Gamma\vdash (Expr_1,\ldots,Expr_n){-}{-}{>}!\ :notype}$$

1.2.5 AssignStat: Assegnazione

$$\frac{\Gamma(Identifier_i){:}\tau_i^{i\in 1\ldots n}\ \ \Gamma\vdash Expr_i{:}\tau_i^{i\in 1..n}}{\Gamma\vdash Identifier_1,\ldots,Identifier_n\ll Expr_1,\ldots,Expr_n{:}notype}$$

1.2.6 WhileStat: Istruzione while

$$\frac{\Gamma\vdash Expr{:}boolean\ \ \Gamma\vdash Body{:}notype}{\Gamma\vdash \text{while Expr loop Body: notype}}$$

1.2.7 ReturnStat: Istruzione return

In entrambi i casi, f indica la funzione contenente il costrutto ReturnStat in analisi.

Con ritorno di una espressione

$$\frac{\Gamma \vdash f{:}\tau_1 \times ... \times \tau_n \rightarrow \tau \;\; \Gamma \vdash e{:}\tau}{\Gamma \vdash \text{return e: notype}}$$

Senza ritorno di una espressione

$$\frac{\Gamma \vdash f{:}\tau_1 \times ... \times \tau_n \rightarrow notype}{\Gamma \vdash \text{return: notype}}$$

1.3 Altri costrutti

1.3.1 Body: Blocco dichiarazione-istruzione

$$\frac{\Gamma \vdash VarDecl_i^{i \in 1...n}{:}notype \;\; \Gamma \vdash Stat_j^{j \in 1...m}{:}notype}{\Gamma \vdash \tau Body{:}notype}$$

1.3.2 Program: Programma

$$\frac{\Gamma \vdash VarDecl_i^{i \in 1...n}{:}notype \;\; \Gamma \vdash f_j^{j \in 1...m}{:}\tau_1 \times ... \times \tau_p \rightarrow \tau}{\Gamma \vdash Program{:}notype}$$

1.3.3 VarDecl: Dichiarazione di variabile

Senza inferenza di tipo, senza inizializzazione

$$\frac{}{\Gamma[Identifier_i^{i \in 1...n} \rightarrow \tau_i^{i \in 1...n}] \vdash \tau_i^{i \in 1...n} Identifier_i{:}notype}$$

Senza inferenza di tipo, con inizializzazione

$$\frac{\Gamma[Identifier_i^{i \in 1...n} \rightarrow \tau_i^{i \in 1...n}] \vdash Expr_i^{i \in 1...n}{:}\tau^{i \in 1...n}}{\Gamma \vdash \tau_i^{i \in 1...n} Identifier_i^{i \in 1...n} << Expr_i^{i \in 1...n}{:}notype}$$

Con inferenza di tipo, con inizializzazione

$$\frac{\Gamma[Identifier_i^{i\in 1...n} \rightarrow \tau_i^{i\in 1...n}] \vdash Expr_i^{i\in 1...n} : \tau^{i\in 1...n}}{\Gamma \vdash varIdentifier_i^{i\in 1...n} << Expr_i^{i\in 1...n} : notype}$$

1.3.4 FunDecl: Definizione di funzione

Gli Identifier in questo caso rappresentano i parametri della funzione, che sono presenti nel type environment prima del controllo del tipo del Body.

Con valore di ritorno

$$\frac{\Gamma[Identifier_i^{i\in 1...n} : \tau_i^{i\in 1...n}] \vdash Body : notype}{\Gamma[f : \tau_1 \times ... \times \tau_n \rightarrow \tau] \vdash FunDecl : notype}$$

Senza valore di ritorno

$$\frac{\Gamma[Identifier_i^{i\in 1...n} : \tau_i^{i\in 1...n}] \vdash Body : notype}{\Gamma[f : \tau_1 \times ... \times \tau_n \rightarrow notype] \vdash FunDecl : notype}$$