Regole di inferenza

Dario Mazza, Nicolò Delogu

1 Regole di inferenza

1.1 Identificatore

$$\frac{\Gamma(id) = \tau}{\Gamma \vdash id : \tau} \tag{1}$$

1.2 Costanti

$$\Gamma \vdash int\_const : integer \tag{2}$$

$$\Gamma \vdash float\_const : float \tag{3}$$

$$\Gamma \vdash string\_const : string \tag{4}$$

$$\Gamma \vdash true : boolean \tag{5}$$

$$\Gamma \vdash false : boolean \tag{6}$$

$$\Gamma \vdash char\_const : char \tag{7}$$

1.3 Lista di Istruzioni

$$\frac{\Gamma(stmt_1) : notype \;\; \Gamma(stmt_2) : notype}{\Gamma \vdash stmt_1; stmt_2 : notype} \tag{8}$$

1.4 Chiamata a funzione con o senza tipo di ritorno (espressione o istruzione) senza controllo del parametro out

$$\frac{\Gamma \vdash f : \tau_1 \times \ldots \times \tau_n \rightarrow \tau \;\; \Gamma \vdash e_i : \tau_i^{i \in 1 \ldots n}}{\Gamma \vdash f(e_1, \ldots e_n) : \tau} \tag{9}$$

$$\frac{\Gamma \vdash f : \tau_1 \times \ldots \times \tau_n \rightarrow notype \;\; \Gamma \vdash e_i : \tau_i^{i \in 1 \ldots n}}{\Gamma \vdash f(e_1, \ldots e_n) : notype} \tag{10}$$

1.5 Chiamata a funzione con controllo del parametro di output

$$\frac{\Gamma \vdash f : p_1 \;\; x \ldots \;\; x \;\; p_n \rightarrow \tau \;\; p_i : (Out?^1, \;\; \tau_i)^{i \in 1 \ldots n} \;\; \Gamma \vdash compatibility(e_i, p_i)^{i \in 1 \ldots n}}{\Gamma \vdash f(e_1, \ldots, e_n) : \tau} \tag{11}$$

1.6 Assegnazione

$$\frac{\Gamma(id)_i : \tau_i^{i \in 1 \ldots n} \;\; \Gamma \vdash e_i : \alpha_i^{i \in 1 \ldots n} \;\; \Gamma \vdash compatibility(\tau_i, \alpha_i)^{i \in 1 \ldots n}}{\Gamma \vdash id_1, \ldots, id_n << e_1, \ldots, e_n : notype} \tag{12}$$

1.7 Blocco dichiarazione-istruzione

$$\frac{\Gamma[id \rightarrow \tau] \vdash stmt : notype}{\Gamma \vdash \tau \;\; id; stmt : notype} \tag{13}$$

1.8 Istruzione while

$$\frac{\Gamma \vdash e : boolean \;\; \Gamma \vdash body : notype}{\Gamma \vdash while(e) \;\; loop \;\; body : notype} \tag{14}$$

1.9 Istruzione for

$$\frac{\Gamma \vdash e_1 : integer\_const \;\; \Gamma \vdash e_2 : integer\_const \;\; \Gamma[id \rightarrow integer] \vdash body : notype}{\Gamma \vdash for \;\; id << e_1 \;\; to \;\; e_2 \;\; loop \;\; body : notype} \tag{15}$$

1.10 Istruzione if-then-else

$$\frac{\Gamma \vdash e : boolean \;\; \Gamma \vdash body_{if} : notype \;\; \Gamma \vdash body_{else} : notype}{\Gamma \vdash if(e) \;\; then \;\; body_{if} \;\; else \;\; body_{else} : notype} \tag{16}$$

1.11 Istruzione return

$$\frac{\Gamma \vdash e : \tau \quad \tau = funReturnType}{\Gamma \vdash return \; e : \tau} \tag{17}$$

$$\frac{void = funReturnType}{\Gamma \vdash return : notype} \tag{18}$$

1.12 Dichiarazione

$$\frac{\Gamma \vdash decl_i^{i \in 1...n} : notype}{\Gamma \vdash \{decl_1; ...; decl_n\} : notype} \tag{19}$$

1.13 Body

$$\frac{\Gamma \vdash decl_i^{i \in 1...n} : notype \;\; \Gamma \vdash stmt_j^{j \in 1...m} : notype}{\Gamma \vdash \{decl_1; ...; decl_n; stmt_1; ...; stmt_m\} : notype} \tag{20}$$

1.14 Istruzione read

$$\frac{\Gamma(id)_i : \tau_i^{i \in 1...n}}{\Gamma \vdash id_1, \ldots, id_n \longleftarrow string\_const : notype} \tag{21}$$

1.15 Dichiarazione di funzione

$$\frac{p_i : (Out?,^2 \;\; \tau_i)^{i \in 1...n} \;\; \Gamma[param_j^{j \in 1...n} \rightarrow p_i] \vdash body : notype}{\Gamma \vdash deffunId(param_1, \ldots, param_n)" : "\tau \;\; body : notype \rightarrow \tau} \tag{22}$$

1.16 Compatibility

Type1	Type2	Result
τ	τ	true
integer	float	true
float	integer	true

1.17 Operatori Unari

$$\frac{\Gamma \vdash e : \tau_1 \;\; Op_1(op_1, \tau_1) = \tau}{\Gamma \vdash op_1 \;\; e : \tau} \tag{23}$$

Op	**Arg**	**Result**
MINUS	integer	integer
MINUS	float	float
NOT	boolean	boolean

Table 1: Operatori unari

1.18 Operatori Binari

$$\frac{\Gamma \vdash e_1 : \tau_1 \;\; \Gamma \vdash e_2 : \tau_2 \;\; Op_2(op, \tau_1, \tau_2) = \tau}{\Gamma \vdash e_1 \;\; op \;\; e_2 : \tau} \tag{24}$$

Op	**Operando1**	**Operando2**	**Result**
PLUS, TIMES, ...	integer	integer	integer
PLUS, TIMES, ...	integer	float	float
PLUS, TIMES, ...	float	integer	float
PLUS, TIMES, ...	float	float	float
STR_CONCAT	(string, integer, float, boolean, char)	(string, integer, float, boolean, char)	string
AND, OR	boolean	boolean	boolean
LT,LE,GT,GE	integer	integer	string
LT,LE,GT,GE	integer	float	boolean
LT,LE,GT,GE	float	integer	boolean
LT,LE,GT,GE	float	float	boolean
EQ,NE	integer	integer	boolean
EQ, NE	integer	float	boolean
EQ, NE	float	integer	boolean
EQ, NE	float	float	boolean
EQ, NE	string	string	boolean
EQ, NE	char	char	boolean
EQ, NE	boolean	boolean	boolean

Table 2: Operatori binari