Type Checking

processo che verifica se le espressioni sono tipate correttamente

Tipo Es. $\mathbb{B}$, int, ...

- Un insieme di valori (Es. numeri interi)
- Un insieme di operazioni che si possono effettuare su quei valori (Es. $+, -, *, /$)

Il nostro obiettivo è: **data un'espressione, verificare che sia tipata correttamente**

Ciò viene fatto attraverso l'uso di REGOLE D'INFERENZA

Una REGOLA D'INFERENZA è composta da un insieme di premesse $A_1, \dots, A_n$ e da una conclusione A

$$\frac{A_1 \quad \dots \quad A_n}{A}$$

se sono vere $A_1 \dots A_n$ allora A è vera

Una regola senza premesse è chiamata assioma $\rightarrow$ $\dfrac{}{A}$

Simile alla definizione che si usa in logica, noi siamo interessati ai tipi

quindi $A_1, \dots, A_n, A$ in realtà ci dicono il tipo di una espressione

e si indicano con $\Gamma \vdash e : T$

Γ: dal contesto Gamma; $\vdash$: deduciamo; $e : T$: che e ha tipo T

Un esempio di regola è quella dell'AND

$$\frac{\Gamma \vdash e_1 : bool \qquad \Gamma \vdash e_2 : bool}{\Gamma \vdash e_1 \,\&\&\, e_2 : bool} \qquad \frac{\Gamma \vdash e_1 : int \qquad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 + e_2 : int}$$

Come capiamo il tipo delle variabili?

Dipende dal contesto (chiamato Γ) in cui si trovano

hint: gamma nelle regole verrà utilizzato solo per accedere al tipo di una variabile! Possiamo pensarla come una funz che presa in input la var. mi dice il tipo nell'ambiente

$$\frac{\Gamma(x) = T}{\Gamma \vdash x : T}$$

dove T può essere bool, int, ...

Nota: per rendere vera la conclusione, la premessa deve valere, altrimenti la nostra exp. è tipata male!

Γ: $x \mapsto int$

$$\frac{\Gamma(x) : int}{\Gamma \vdash x : bool} \times$$

Introduciamo le regole d'inferenza per un ling. con interi, bool, somma e uguaglianza e variabili

Sono tutti assiomi, non servono premesse
. 7 è un int
. true è un bool
. false è un bool

$$\dfrac{}{\Gamma \vdash num : int}\,[Num] \qquad \dfrac{}{\Gamma \vdash true : bool}\,(true) \qquad \dfrac{}{\Gamma \vdash false : bool}\,[false]$$

Nome della regola

$$\dfrac{\Gamma(x) : T}{\Gamma \vdash x : T}\,[Id]$$

qui si accede a gamma per verificare il tipo di x

Nota: queste regole sono sempre nelle foglie dell'albero, immaginate di scomporre una espressione

\+ deve prendere due interi e tornare un intero

$$\dfrac{\Gamma \vdash e_1 : int \quad \Gamma \vdash e_2 : int \quad + : int \times int \to int}{\Gamma \vdash e_1 + e_2 : int}\,[sum]$$

non T, la somma non è definita nei booleani

$$\dfrac{\Gamma \vdash e_1 : T_1 \quad \Gamma \vdash e_2 : T_2 \quad T_1 = T_2 \quad == : T_1 \times T_2 \to bool}{\Gamma \vdash e_1 == e_2 : bool}$$

è bool, un'uguaglianza può essere vera o falsa, ma vogliamo def. sia su bool che su int!

tipiamo $((x+2)+3) == (x+2)$ assumendo $x \to int$

(non ci interessa il valore!)

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma(x) : int}{\Gamma \vdash x : int} \quad \dfrac{}{\Gamma \vdash 2 : int} \quad + \ldots}{\Gamma \vdash x+2 : int}\,[sum] \quad \dfrac{}{\Gamma \vdash 3 : int} \quad + : int \times int \to int}{\Gamma \vdash (x+2)+3 : \overset{int}{T_1}}\,[sum] \qquad \dfrac{\dfrac{\Gamma(x) : int}{\Gamma \vdash x : int} \quad \dfrac{}{\Gamma \vdash 2 : int} \quad + \ldots}{\Gamma \vdash x+2 : \overset{int}{T_2}} \quad \overset{int}{T_1} = \overset{int}{T_2} \quad == : \overset{int}{T_1} \times \overset{int}{T_2} \to bool}{\Gamma \vdash ((x+2)+3) == (x+2) : bool}\,[eq]$$

(le scritte in rosso si chiamano unifications)

Complichiamo

Regole per if then else e dichiaraz. di funz.

$$\dfrac{\Gamma \vdash e_1 : T_1 \quad \Gamma \vdash e_2 : T_2 \quad \Gamma \vdash e_3 : T_3 \quad T_1 = bool \quad T_2 = T_3}{\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : T_2}\,[if]$$

formali e attuali

$$\dfrac{\Gamma \vdash f : T_1 \times \ldots \times T_n \to T \quad (\Gamma \vdash e_i : T_i')_{i \in 1..n} \quad (T_i = T_i')_{i \in 1..n}}{\Gamma \vdash f(x_1, \ldots, x_n) : T}$$

dichiarazioni

Se dichiaro una var.
↓
Aggiorno l'ambiente

quindi le dichiaraz. tornano un ambiente aggiornato

$$\frac{\Gamma \vdash e : T' \quad x \notin dom(top(\Gamma)) \quad T = T'}{\Gamma \vdash T\ x = e;\ : \Gamma'} \ [VarD]$$

$$\Gamma' \to \Gamma[x \to T]$$

$$\frac{\Gamma \vdash d : \Gamma' \quad \Gamma' \vdash D : \Gamma''}{\Gamma \vdash d\,D : \Gamma''} \ [SeqD]$$

simile per le dichiarazioni di funzioni

$$\frac{\Gamma[x_1 \to T_1, \dots, x_n \to T_n] \vdash e : T' \quad T' = T \quad f \notin dom(top(\Gamma))}{\Gamma \vdash T\ f(T_1\, x_1, \dots, T_n\, x_n) = e} \ [fun]$$

← non ammette ricorsive
tipiamo e senza
avere f nel contesto

Slide 46
pacco 6

$A \to BC$

$B \to aB \mid \varepsilon$

$C \to CbB \mid c$

$\Downarrow$

$C \to cC'$

$C' \to bBC'$

$C' \to \varepsilon$

$N = \{B, C'\}$

$N(A) = N(B) \land N(C) = T \land F = F$

$N(B) = N(aB) \lor N(\varepsilon) = F \lor T = T$

$N(C) = N(cC') = F$

$N(C') = N(bBC') \lor N(\varepsilon) = F \lor T = T$

$A \to BC$

$B \to aB \mid \varepsilon$

$C \to CbB \mid c$

$\Downarrow$

$C \to cC'$

$C' \to bBC'$

$C' \to \varepsilon$

$N = \{B, C'\}$

$Fo(A) = \{\$\}$

$Fo(B) = \{c, b, \$\}$

$Fo(C) = \{\$\}$

$Fo(C') = \{\$\}$

$Fi(A) = Fi(B) \setminus \{\varepsilon\} \cup FI(C)$

$= \{a\} \cup \{c\} = \{a, c\}$

$Fi(B) = \{a, \varepsilon\}$

$Fi(C) = \{c\}$

$Fi(C') = \{b, \varepsilon\}$

	a	b	c	$
A	$A \to BC$		$A \to BC$	
B	$B \to aB$	$B \to \varepsilon$	$B \to \varepsilon$	$B \to \varepsilon$
C			$C \to cC'$	
C'		$C' \to bBC'$		$C' \to \varepsilon$