

## IDENTIFICATORI:

Nomi Per Denotare Oggetti  $\Rightarrow$  c: Sarà Quindi un Associazione / Legame Tra Identificatore ed Oggetto Denotato

IL Contenitore di Questi legami è l'Ambiente

Ci Sono legami::

**STATICI**  $\Rightarrow$  Creati a Tempo di Compilazione (Fissi)

**DINAMICI**  $\Rightarrow$  Creati a Tempo di Esecuzione (Cambiano)

Per Questo ci Sono 2 Ambienti  $\neq$ :

**STATICO**

**DINAMICO**

## VALORE DENOTABILE:

Che Noi Possiamo denotare/Riferire Attraverso Identificatori

tutti i Valori Esprimibili Sono DENOTABILI, ma NON SONO LO STESSO INSIEME

DVal  $\rightarrow$  Insieme dei Valori Denotabili

Occorrenze di Identificatori:

**FREE:** Non Sono nello SCOPE di una definizione

M Di Uso o DEFINIZIONE: Occorrenza che definisce il Significato  
e Accocrenza Nello Scope di una definizione

Se un Programma non Contiene Identificatori → è un Termine

### GROUND

Se un Programma Contiene identificatori ma non in Condizione Libera allora è un TERMINE CHIUSO, ovvero ha Significato Senza Accadere ad un Contesto (NON HA BISOGNO DI ALTRE INFORMAZ.)

Se un PROG. ha Identificatori liberi, non ha Significato se non accede ad un Ambiente (DEVE ACCEDERE AD UN AMBIENTE PER AVERE SIGNIFICATO)

# AMBIENTE (DINAMICO)

Generalmente Sempre dinamico Sottointeso

Denotato da ENV.

V insieme di Identif.

$\text{Env}_v : V \rightarrow \text{DVal} \cup \{\perp\}$

Associazione Tra Ambiente e Valore denotabile

Associa ad Ogni Identificatore in V un Valore in DVal  $\cup \{\perp\}$

$$\overbrace{\text{Env} = \bigcup_{V \subseteq_f Id} \text{Env}_v}$$

Unione degli Ambienti definiti Su Insiemi finiti di identificatori

$V = \{x, y\}$   $\text{Env}_v$  funzione che Associa a x e a y il Valore denotato

Integro Ora la Semantica Usando Gli Ambienti:

Exp:  $E \rightarrow A | B$  dove

$A \rightarrow x | n | A \text{ op } A$  Quindi  $N \subseteq \text{DVal}$

$B \rightarrow \text{True} | \text{false} | x | B \text{ or } B | A = A | \text{Not } B$

Ed Allora  $B = \{\text{True}, \text{false}\} \subseteq \text{DVal}$  (Identif. Statici)

E così Sono Aggiunti Idenf. alla Grammatica

Genera Solo Espressioni Con Identificatori liberi

Ed Abbiamo Appena detto che non ha Significato Se Non Viene legata Ad un Ambiente ed Allora:

$$\beta \vdash e_1 \rightarrow e_2$$

Solo Quando Serve

La Regola Si legge Come: Nell' Ambiente  $\beta$  l' Exp  $e_1$  viene Valutata in  $e_2$

Serve Ora: ~~M~~ Assioma per Identificatori

SINTASSI/Simboli

$$\# \quad \beta \vdash m \text{ op } n \rightarrow p$$

SEMANTICA/Significato

$$\text{Se } p = n \text{ op } m$$

$$\# \quad \beta \vdash \text{Not } t_0 \rightarrow t$$

$$\text{Se } t_0 = t_0 = \text{Not } t_0$$

$$\# \quad \beta \vdash t_0 \text{ OR } t_1 \rightarrow t$$

$$\text{Se } t = t_0 \text{ OR } t_1$$

$$\# \quad \beta \vdash m = n \rightarrow t$$

$$\begin{cases} \text{True se } m = n \\ \text{False se } m \neq n \end{cases}$$

$$\# \quad \beta \vdash x \rightarrow K$$

$$\text{Se } \beta(x) = K \text{ che } \in DVal$$

Bisogna Aggiornare Con il Contesto:

$$\frac{\rho \vdash e_1 \rightarrow e'_1}{\rho \vdash e_1 \text{ op } e_2 \rightarrow e'_1 \text{ op } e_2}$$

$$\frac{\rho \vdash e_2 \rightarrow e'_2}{\rho \vdash m \text{ op } e_2 \rightarrow m \text{ op } e'_2}$$

$$\frac{\rho \vdash e_1 \rightarrow e'_1}{\rho \vdash e_1 \text{ bop } e_2 \rightarrow e'_1 \text{ bop } e_2}$$

$$\frac{\rho \vdash e_2 \rightarrow e'_2}{\rho \vdash m \text{ bop } e_2 \rightarrow m \text{ bop } e'_2}$$

- t OR e'\_2
- m = e'\_2

$$\frac{\rho \vdash e_0 \rightarrow e'_0}{\rho \vdash \text{Not } e_0 \rightarrow \text{Not } e'_0}$$

## TIPO:

È l'Insieme Dei **Valori** che (e Quindi delle **Operazioni**) che possiamo Associare ad un Identificatore.

Questi Valori devono CONDIVIDERE una proprietà STRUTTURALE, devono Avere Qualcosa in Comune.

Tipo è una forma di Astrazione per discriminare cosa si può FARÉ e cosa NO

$\{n \mid n \text{ è Pari}\}$  Non determina un Tipo, dunque Rappresentere tutti i Valori che hanno Quella proprietà Strutturelle

Integer / int  
Boolean  
String  
Int → Bool  
- - - -

SONO TIPI

Non È TIPO DIRE UN INSIEME DI ELEMENTI:  
 $\{troe, 5\}$

$\{ \text{IN} \in \text{Pari} \}$

funzioni Specifiche Tipo  $f(x) = x^2 + x$

- - - - -

Al livello di **Progettazione** permettono di **Organizzare l'Informazione**

A livello di **Implementazione** (Nel Programma) quello di **PREVENIRE GLI ERROTI DI TIPO** [controllo dimensionale]

5 + 'ciao' è SEMPRE sbagliato anche se + è OVERLOAD

A livello di **Implementazione** → Permettono **ottimizzazioni dell'uso della Memoria**

Devo Allora **Creare Legami Tra IDENTIF. e TIPO** ( $id \leftrightarrow tipo$ )  
anche detta **TYPE BINDING**

Se Abbiamo il Costrutto per la **dichiarazione è Esplicito** allora  
il **legame di Tipo Sono STATICI** (gestiti a **Tempo di Compilaz.**)  
Altrimenti **TYPE BINDING** è dinamico, vale a dire **che i LEGAMI DI TIPO Sono DINAMICI** (gestiti a **Tempo di Esecuzione**)

Uso = su INTERI/NUMERI  
bool      ↙  
Uso OR su BOOLEANI

$$x = y$$

$$x \text{ OR } y$$

## SEMANTICA STATICÀ:

Mi permette di Verificare l'uso Corretto degli IDENTIFICATORI (se le operazioni sono USATE CORRETTAMENTE)

Ambiente Statico Associa ad Ogni Identificatore il Suo Tipo

DType = Insieme dei Tipi dei Valori Denotabili

$$DType = \{ Int, Bool \}$$

$V \subseteq Id$  insieme finito di identificatori

$\Delta$  Metavariabile per Ambiente Statico

$T_{Env_v} : V \rightarrow DType$  associa ad Ogni Identificatore in  $V$  un tipo Denotabile

$$\Delta \in T_{Env_v} = \bigcup_{V \subseteq Id} T_{Env_v}$$

Regole Controllano la CORrettezza del TIPO

SEMANTICA STATICÀ DI EXP:

EXP di cui verificare CORrettezza del TIPO

$$\Delta \vdash e : \tau \quad \tau \in DType$$

Id su cui Abbiamo (dominio di  $\Delta$ , ovvero l'insieme di ID a cui è definito Significato)  $\Delta$  attribuisce un TIPO, non OBBLIGATORIO

Ambiente Statico in cui Verifichiamo il TIPO di e

ho Regole che fanno CONTROLLI, PATTERN MATCHING Sarà Alla Base

ASSIOMI: (Servono per Categorie Verifiche)

$$\Delta \vdash n : \text{INT}$$

$\hookrightarrow n \in \mathbb{N}$

$$\Delta \vdash t : \text{Bool}$$

$\hookrightarrow t \in \begin{cases} \text{true}, \text{false} \\ B \end{cases}$

$$\Delta \vdash x : \textcolor{blue}{T}$$

Tipi del Valore

$$\Delta(x) = \textcolor{blue}{T}$$

Sono Gli Stessi

Con gli Assiomi non Posso Introdurre ERRORI

$$\Delta \vdash e_1 : \text{Bool}$$

$$\Delta \vdash e_2 : \text{Bool}$$

$$\Delta \vdash e_1 \text{ OR } e_2 : \text{Bool}$$

per forza, ma  
con che Ipotesi?

Se ENTRAMBI Sono  
BOOLEANI

$$\Delta \vdash e_1 : \text{INT} \quad \Delta \vdash e_2 : \text{INT}$$

$$\Delta \vdash e_1 \text{ op } e_2 : \text{INT}$$

$\hookrightarrow e \in \{+, -, *\}$

STESO AMBIENTE PER LA VERIFICA

e devono Essere tutti INT

$$\Delta \vdash e_1 : \text{Bool}$$

e

Per Come Abbiamo Scritto NOT // linguaggio:

$$\Delta \vdash e_1 : \text{INT} \quad \Delta \vdash e_2 : \text{INT}$$

$$\Delta \vdash \text{Not } e_1 : \text{Bool}$$

$$\Delta \vdash e_1 = e_2 : \text{Bool}$$

Ho 2 Sematiche Ora.

DINAMICA usa  $p \in Env$  che  $V \rightarrow DVal$

STATICA usa  $\Delta \in TEnv$  che  $V \rightarrow Dtype$

Devo dare un Contatto di COMPATIBILITÀ.

Sono COMPATIBILI ( $\Delta$  e  $p$ ) se  $\exists v \subseteq_f Id . \Delta : V \rightarrow Dtype$   
 $p : V \rightarrow DVal$

e  $\forall . ID \in V .$  se  $\Delta(ID) = \tau$   $p(ID) \in \tau$  [valore di TIPO  $\tau$ ]  
quindi ci deve ESSERE COERENZA Tra TIPO associato dall'  
ambiente Statico e Valore dell' Ambiente dinamico

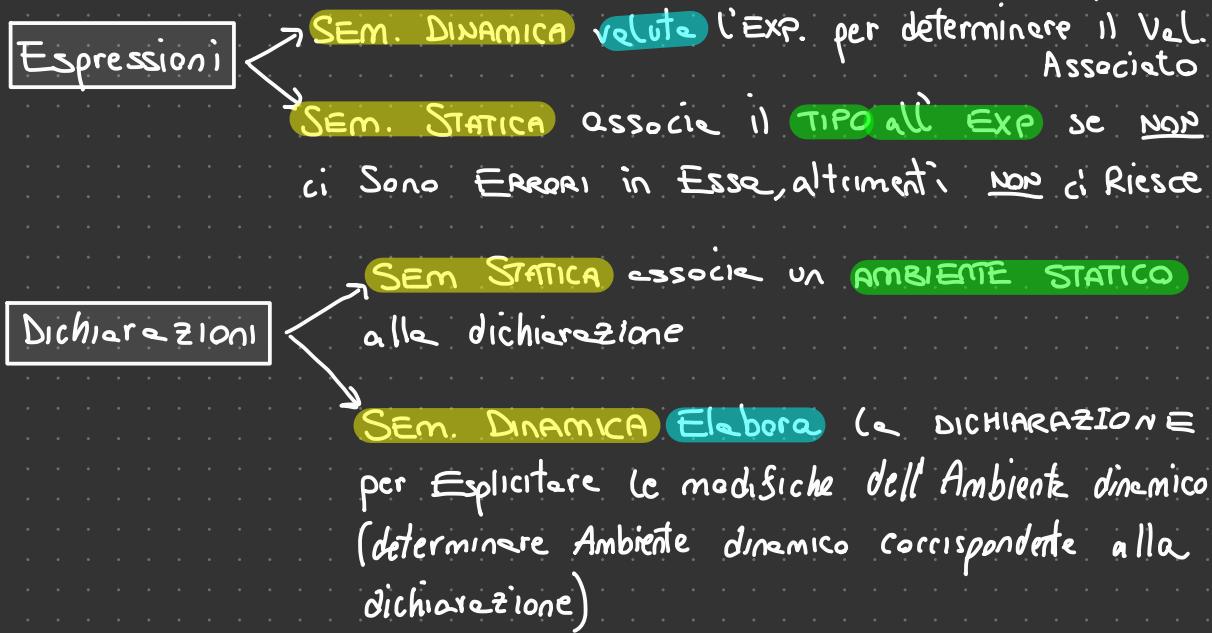
$P \vdash_a$

Valutazione dell' Ambiente dinamico  $P$  Compatibile con l'Ambiente  
Statico  $\Delta$

## DICHIARAZIONI:

Strumento SINTATTICO per Creare e Manipolare LEGAMI e quindi AMBIENTI

Possono Essere Sia IMPLICITI che ESPICITI



## SINTASSI DELLE DICHIARAZIONI:

Dichiarazioni in IMP

Dec               $D \rightarrow \text{Nil}$  *[Servirà Come Punto di Terminazione Nella Semantica (AMBIENTE vuoto)]*

$D \rightarrow \underline{\text{Costante } x:r = e}$  *[non useremo Mai La Locaz. di Memoria Vista che Sarà Per Sempre Costante]*

dichiarazione di  $x$  costante di TIPO  $r$  con Valore Rappresentabile  $e$

Andranno Poi Aricchite Quando Tratteremo la memoria

$D \rightarrow D; D$  [Composizione Seq. di dichiarazioni]

$D \rightarrow D \text{ in } D$  [Composizione Privata]

$D \rightarrow p \in \text{ENV}$  [ci Serve Come Valore Terminale]