Tempi di bindings I bindings possono essere creati:

- A tempo di compilazione (early binding), ottenendo un'esecuzione più veloce ma un programma poco flessibile in quanto l'allocazione della memoria è statica (uso dello stack). Quindi tutti i bindings hanno indirizzi assoluti;
- A tempo di esecuzione (late binding), ottenendo un'esecuzione più lenta ma un programma più flessibile grazie all'allocazione dinamica della memoria. Tutti i bindings, quindi, hanno indirizzi non assoluti.

Esistono linguaggi che adottano entrambe le soluzioni.

ATTENZIONE: affinchè un identificatore abbia significato, deve essere legato a qualcosa.

Semantica delle dichiarazioni

Per riferire gli oggetti denotabili (valori riferibili tramite identificatore) usiamo il concetto di **ambiente**, che indica l'insieme delle associazioni tra nomi e oggetti denotabili esistenti a runtime in uno specifico punto del programma ed in uno specifico momento dell'esecuzione.

Le associazioni vengono create nell'ambiente tramite le **dichiarazini**. Infatti queste permettono di passare da free-occurrences, ovvero occorrenze di uso senza legame, ad applied-occurrences, dove l'identificatore non è più libero.

Identificatori liberi Un identificatore è libero se non c'è nessuna dichiarazione che lo coinvolge.

Definizione per induzione

La funzione $FI : Exp \to Id$ che associa ad ogni espressione l'insieme degli identificatori liberi in essa contenuti è definita come segue:
$FI(k) = \emptyset$
$FI(id) = \{id\}$
$FI(e_0 \; bop \; e_1) = FI(e_0) \cup FI(e_1)$
$FI(not \; e) = FI(e)$

Termine chiuso In un linguaggio, un termine (programma) in cui non ci sono identificatori liberi è detto chiuso.

Termine groud In un linguaggio, un termine (programma) in cui non ci sono identificatori è detto ground.

Ambiente dinamico Un ambiente dinamico è un elemento dello spazio di funzioni

$$Env = \cup_{V \subseteq_f Id} Env_V$$

dove $Env_V : V \to DVal$ e $\cup\{\perp\}$ ha metavariabile p.

LE REGOLE →

Ora riportiamo le regole aggiornate con l'ambiente, quelle già descritte non verranno ulteriormente commentate, essendo esattamente le stesse già descritte per le espressioni.

$$\mathscr{E}_1: \rho \vdash m \ \mathbf{op} \ n \rightarrow_e p \quad \text{se} \quad m \ op \ n = p, m,n,p \in \mathcal{N}$$

La seconda regola adesso cambia, in quanto ora ci serve l'assioma per gli identificatori:

$$\mathscr{E}_2: \rho \vdash I \rightarrow_e n \quad \text{se} \quad \rho(I) = n$$

In questo assioma quindi diciamo che quando incontriamo un identificatore, questo viene valutato nel valore che l'ambiente associa all'identificatore. È chiaro che questa regola è applicabile solo se I è un identificatore per il quale in ρ esiste una associazione.

$$\mathscr{E}_3: \frac{\rho \vdash e \rightarrow_e e'}{\rho \vdash e \ \mathbf{op} \ e_0 \rightarrow_e e' \ \mathbf{op} \ e_0}$$

$$\mathscr{E}_4: \frac{\rho \vdash e \rightarrow_e e'}{\rho \vdash m \ \mathbf{op} \ e \rightarrow_e m \ \mathbf{op} \ e'}$$

$$\mathscr{E}_5: \rho \vdash t_1 \ \mathbf{bop} \ t_2 \rightarrow_e t \quad \text{se} \quad t_1 \ op \ t_2 = t, t_1, t_2, t \in \mathcal{B}$$

$$\mathscr{E}_{3'}: \frac{\rho \vdash e \rightarrow_e e'}{\rho \vdash e \ \mathbf{bop} \ e_0 \rightarrow_e e' \ \mathbf{bop} \ e_0}$$

$$\mathscr{E}_6: \frac{\rho \vdash e \rightarrow_e e'}{\rho \vdash t \ \mathbf{op} \ e \rightarrow_e t \ \mathbf{op} \ e'}$$

$$\mathscr{E}_7: \rho \vdash \mathbf{not} \ t_1 \rightarrow_e t \quad \text{se} \quad not \ t_1 = t, t_1 \in \mathcal{B}$$

$$\mathscr{E}_8: \frac{\rho \vdash e \rightarrow_e e'}{\rho \vdash \mathbf{not} \ e \rightarrow_e \mathbf{not} \ e'}$$