

Tipi Il tipo fa parte dei vincoli sintattici che la grammatica non è in grado di descrivere. Il tipo determina l'insieme di valori (che condividono una certa proprietà strutturale) dotato di un insieme di operazioni definite per i valori di quel tipo.

Il concetto di tipo è utile:

- A livello di progetto, dove è impiegato per organizzare l'informazione;
- A livello di programmazione, dove permette di prevenire determinati errori (non posso assegnare un int ad una variabile String);
- A livello di implementazione, dove favorisce l'ottimizzazione (in base al tipo, riserva alla variabile una zona di memoria con dimensione differente);

Type bindings Il tipo è un oggetto denotabile associato ad un identificatore per descrivere le sue proprietà, e quindi i suoi vincoli semantici.

Il tipo dell'identificatore può essere specificato in modo:

- Esplicito -> il linguaggio contiene il costrutto del tipo;
- Implicito -> viene assegnato dal linguaggio (minore affidabilità).

Il legame identificatore - tipo può essere stabilito in modo:

- Dinamico -> il tipo è associato con l'assegnamento;
- Statico -> il tipo è associato a tempo di compilazione).

Semantica statica e dinamica La semantica statica è utilizzata come strumento per creare i legami di tipo. Si occupa di gestire le associazioni di tipo e i vincoli sintattici non context free (valuta il tipo delle espressioni all'interno di un ambiente statico che associa gli identificatori ai tipi).

La semantica dinamica, invece, valuta le espressioni all'interno di un ambiente dinamico, permettendo di modificare i legami con gli oggetti denotabili (valori, locazioni, ...) -> valuta le espressioni, elabora le dichiarazioni ed esegue i comandi.

Ambiente statico Un ambiente statico è una funzione

$$\Delta \in TEnv = \bigcup_{V \subseteq fEnvId} TEnvId \quad \text{con} \quad TEnvId : V \rightarrow Typ$$

$TEnv$ indica l'ambiente di tipi. $DTyp$ indica i tipi denotabili, rappresentati come:

$$\tau \in DTyp = \{int, bool, \perp \text{ (valore non definito)}\}$$

La notazione $\Delta \vdash_V$ indica che valuto l'espressione nell'ambiente statico Δ che ha dominio V .

Semantica statica delle espressioni

Vediamo allora come sono fatte le regole della semantica statica delle espressioni nel nostro linguaggio. Le prime regole sono due assiomi che dicono che, nell'ambiente vuoto (e quindi in ogni ambiente possibile), un numero intero ha tipo intero (int), e una costante booleana ha tipo booleano (bool):

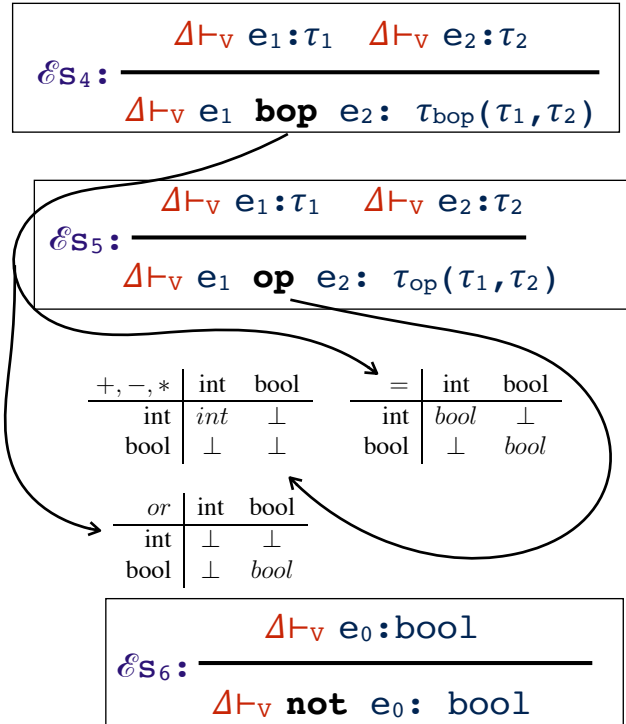
$$\mathcal{E}_1: \vdash n:\text{int}$$

$$\mathcal{E}_2: \vdash t:\text{bool}$$

L'altro assioma riguarda gli identificatori, in tal caso la regola determina che l'identificatore ha come tipo proprio quello che l'ambiente statico Δ del contesto gli associa:

$$\mathcal{E}_3: \Delta \vdash I:\tau \quad \text{se } \Delta(I):\tau, I \in V$$

Adesso vediamo le regole. Esse riguardano le operazioni booleane e aritmetiche, e le definiamo usando delle funzioni che determinano il tipo del risultato di un operatore booleano o aritmetico in funzione del tipo degli operandi. Analogamente diamo la regola per il not booleano.



Compatibilità d'ambiente L'ambiente dinamico ρ e l'ambiente dinamico Δ sono compatibili se

$$\forall id \in I \quad (\Delta(I) = \tau) \text{ and } \rho(I) \in \tau(\text{valore di tipo } \tau)$$

regole definitive

Ora riportiamo le regole aggiornate anche con l'ambiente statico. Le regole non verranno ulteriormente commentate, essendo esattamente le stesse già descritte in precedenza.

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

$$\mathcal{E}_2: \rho \vdash_{\Delta} I \rightarrow_e n \quad \text{se } \rho(I) = n$$

$$\mathcal{E}_3: \frac{\rho \vdash_{\Delta} e \rightarrow_e e'}{\rho \vdash_{\Delta} e \text{ op } e_0 \rightarrow_e e' \text{ op } e_0}$$

$$\mathcal{E}_4: \frac{\rho \vdash_{\Delta} e \rightarrow_e e'}{\rho \vdash_{\Delta} m \text{ op } e \rightarrow_e m \text{ op } e'}$$

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

$$\mathcal{E}_3': \frac{\rho \vdash_{\Delta} e \rightarrow_e e'}{\rho \vdash_{\Delta} e \text{ bop } e_0 \rightarrow_e e' \text{ bop } e_0}$$

$$\mathcal{E}_6: \frac{\rho \vdash_{\Delta} e \rightarrow_e e'}{\rho \vdash_{\Delta} t \text{ op } e \rightarrow_e t \text{ op } e'}$$

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

$$\mathcal{E}_8: \frac{\rho \vdash_{\Delta} e \rightarrow_e e'}{\rho \vdash_{\Delta} \text{not } e \rightarrow_e \text{not } e'}$$