

# Schemi di Traduzione

# (SDT: Syntax Directed Translation scheme)

Gli **schemi di traduzione (SDT)** sono un'utile notazione per specificare la traduzione durante la parsificazione.

Uno **schema di traduzione** è una definizione guidata dalla sintassi in cui le azioni semantiche, racchiuse tra parentesi graffe, sono inserite nel corpo delle produzioni, in posizione tale che, durante il processo di parsificazione, il valore di un attributo sia disponibile quando un'azione fa ad esso riferimento.

Gli schemi di traduzione presuppongono un ordine di valutazione che corrisponde a una visita dell'albero sintattico in profondità da sinistra a destra e permettono che nelle azioni semantiche siano contenuti frammenti di programma e in generale side-effect che non influiscano sulla valutazione degli attributi.

# Schemi di traduzione e SDD

Trasformare una SDD S-attribuita in un SDT :

- Inserire le azioni che calcolano gli attributi sintetizzati della variabile di testa alla fine delle produzioni

Trasformare una SDD L-attribuita in un SDT :

- Inserire le azioni che calcolano gli attributi ereditati per un non terminale A immediatamente prima dell'occorrenza di A nel corpo della produzione.
- Se diversi attributi ereditati per A dipendono uno dall'altro, ordinare la valutazione degli attributi in modo che quelli necessari prima siano calcolati per primi.
- Porre le azioni che calcolano un attributo sintetizzato per la variabile di testa di una produzione alla fine del corpo della produzione stessa.

# Schemi di traduzione e Grammatiche LR (bottom-up) e LL

Uno schema di traduzione può essere implementato costruendo *prima* l'albero di parsificazione e poi eseguendo le azioni *nell'ordine di visita in profondità, da sinistra verso destra*.

Tipicamente però gli SDT sono implementati **durante la parsificazione**, senza costruzione esplicita dell'albero.

4

Vi sono due casi importanti:

- SDD S-attribuita con grammatiche parsificabili bottom-up (LR)
- SDD L-attribuita con grammatiche parsificabili top-down (LL)

# Schema di traduzione S-attribuiti

**Esempio:** Espressione aritmetica  $\rightarrow$  valore numerico

$$E \rightarrow E_1 + T \quad \{E.val = E_1.val + T.val\}$$
$$E \rightarrow T \quad \{E.val = T.val\}$$
$$T \rightarrow T_1 * \text{num} \quad \{T.val = T_1.val * \text{num.val}\}$$
$$T \rightarrow \text{num} \quad \{T.val = \text{num.val}\}$$

# Schema di traduzione S-attribuiti: esempio

Il seguente schema traduce una stringa in  $\{0,1\}^*$  nel suo valore numerico.

$$N \rightarrow N_1 D \quad \{N.val = N_1.val \times 2 + D.val\}$$

$$N \rightarrow D \quad \{N.val = D.val\}$$

$$D \rightarrow 0 \quad \{D.val = 0\}$$

$$D \rightarrow 1 \quad \{D.val = 1\}$$

# Schema di traduzione L-attribuiti: esempio

**Esempio:** Lista delle differenze I

$C \rightarrow N \# \quad \{L.elem = N.val\} \quad L \quad \{C.list = L.list\}$

$L \rightarrow N ; \{L_1.elem = L.elem\} \quad L_1 \quad \{L.list = \underline{cons} (N.val - L.elem, L_1.list)\}$

$L \rightarrow \epsilon \quad \{L.list = \underline{null}\}$

$N \rightarrow \text{digit} \quad \{N.val = \text{num.val}\}$

La grammatica è LL(1):

$C \rightarrow N\#L$

$L \rightarrow N;L_1$

$L \rightarrow \epsilon$

$N \rightarrow \text{num}$

Insiemi guida

$\{\text{num}\}$

$\{\text{num}\}$

$\{\$ \}$

$\{\text{num}\}$

La grammatica è stata ottenuta da quella di un esempio visto in precedenza sostituendo la produzione  $L \rightarrow N$  con  $L \rightarrow \epsilon$  e modificando di conseguenza la regola semantica associata. Questa grammatica, a differenza della precedente, è LL(1).

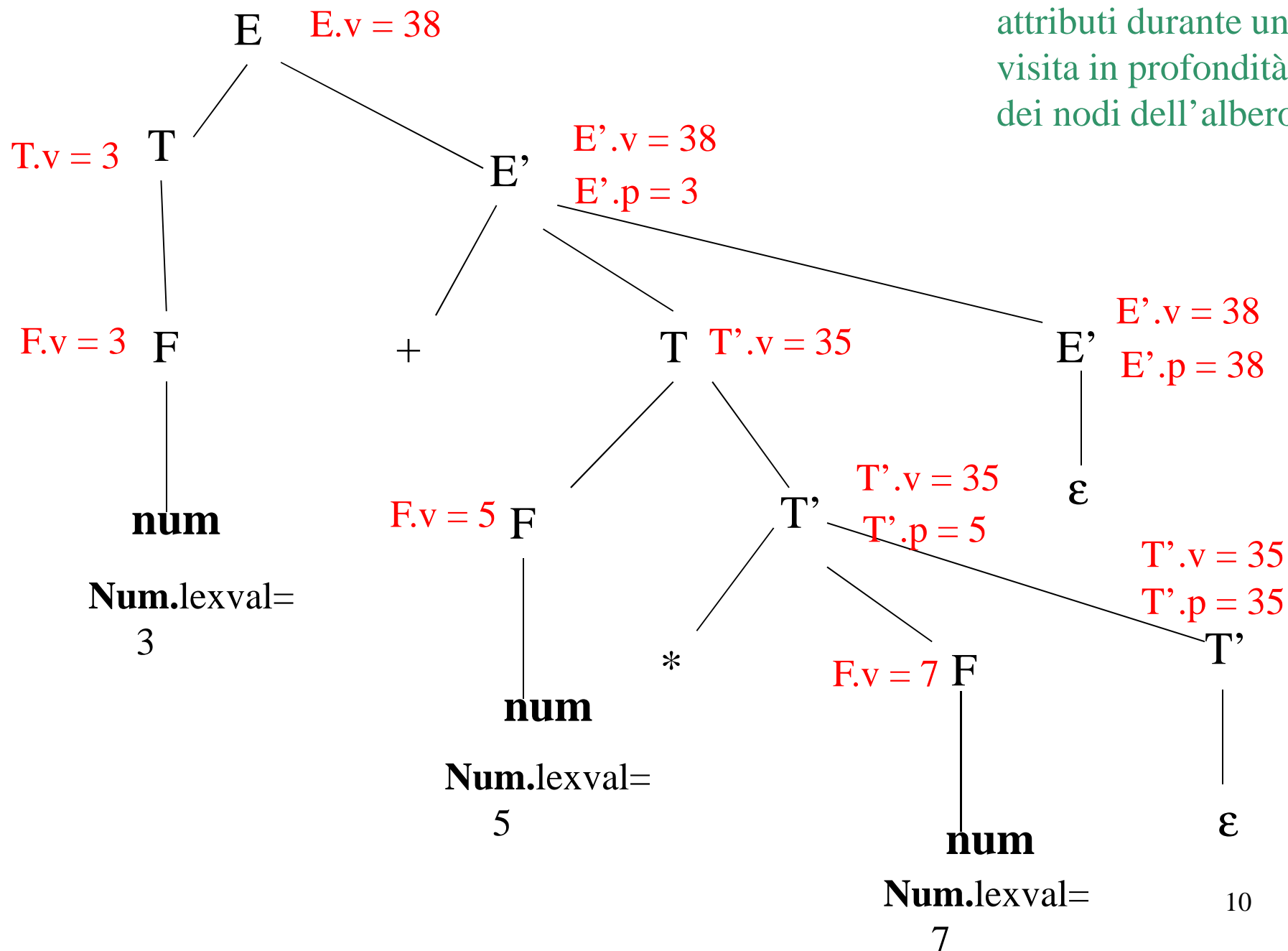
La funzione *cons* aggiunge un elemento in testa a una lista.

## Valutazione con grammatica L-attribuita

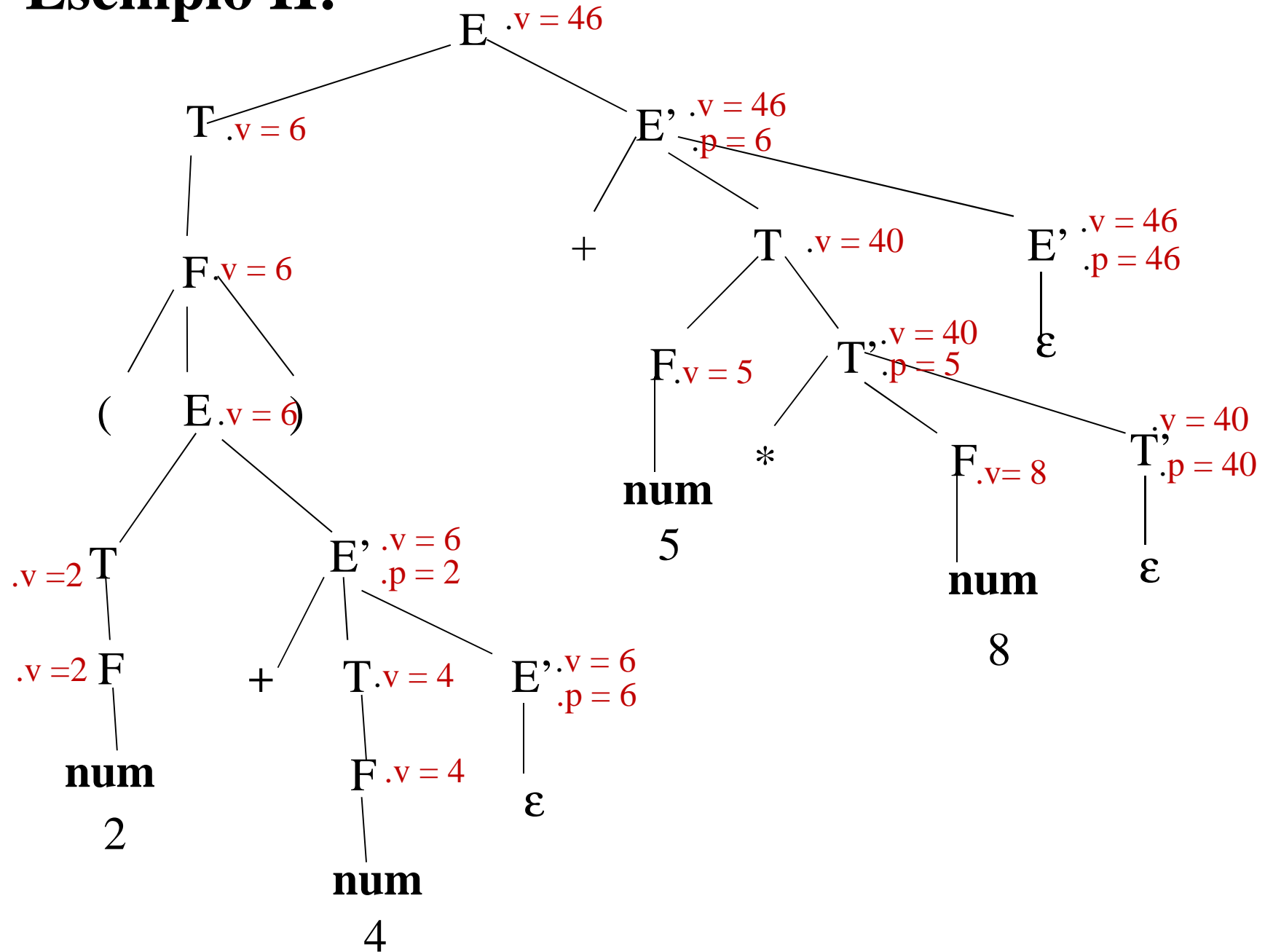
$$\begin{aligned} E &\rightarrow T \{E'.\text{prec} = T.\text{val}\} \\ &\quad E' \{E.\text{val} = E'.\text{val}\} \end{aligned}$$
$$\begin{aligned} E' &\rightarrow + T \{E_1'.\text{prec} = E'.\text{prec} + T.\text{val}\} \\ &\quad E_1' \{E'.\text{val} = E_1'.\text{val}\} \end{aligned}$$
$$E' \rightarrow \epsilon \{E'.\text{val} = E'.\text{prec}\}$$
$$\begin{aligned} T &\rightarrow F \{T'.\text{prec} = F.\text{val}\} \\ &\quad T' \{T.\text{val} = T'.\text{val}\} \end{aligned}$$
$$\begin{aligned} T' &\rightarrow * F \{T_1'.\text{prec} = T'.\text{prec} \cdot F.\text{val}\} \\ &\quad T_1' \{T'.\text{val} = T_1'.\text{val}\} \end{aligned}$$
$$T' \rightarrow \epsilon \{T'.\text{val} = T'.\text{prec}\}$$
$$F \rightarrow (E) \{F.\text{val} = E.\text{val}\}$$
$$F \rightarrow \mathbf{num} \{F.\text{val} = \mathbf{num}.\text{lexval}\}$$



Valutazione degli  
attributi durante una  
visita in profondità  
dei nodi dell'albero



## Esempio II:



## Traduzione L-attribuita in notazione postfissa

$E \rightarrow T \{E'.prec = T.code$   
 $\quad E' \{E.code = E'.code\}$

$E' \rightarrow + T \{E_1'.prec = E'.prec \parallel T.code \parallel '+'\}$   
 $\quad E' \{E'.code = E_1'.code\}$

$E' \rightarrow \epsilon \{E'.code = E'.prec\}$

$T \rightarrow F \{T'.prec = F.code\}$   
 $\quad T' \{T.code = T'.code\}$

$T' \rightarrow * F \{T_1'.prec = T'.prec \parallel F.code \parallel '*'\}$   
 $\quad T_1' \{T'.code = T_1'.code\}$

$T' \rightarrow \epsilon \{T'.code = T'.prec\}$

$F \rightarrow (E) \{F.code = E.code\}$

$F \rightarrow \mathbf{num} \{F.code = \mathbf{num.lexval}\}$

Valutazione degli  
attributi durante una  
visita in profondità  
dei nodi dell'albero

