



UNIVERSITÀ
DEGLI STUDI
DI TORINO

Traduzione guidata dalla sintassi

a.a. 2018-2019

Compilatori: principi, tecniche e strumenti, A.V. Aho, M. S. Lam, R. Sethi, J.D. Ullman

5. Traduzione guidata dalla sintassi

5.1 Definizioni guidate dalla sintassi

5.1.1 Attributi ereditati e sintetizzati

5.1.2 Valutazione di una SDD ai nodi di un albero di parsing

5.2 Ordine di valutazione delle SDD

5.2.1 Grafi delle dipendenze

5.2.2 Ordine di valutazione degli attributi

Le tecniche di traduzione guidate dalle grammatiche libere possono essere applicate al type checking, alla generazione del codice intermedio o usate nell'implementazione di piccoli linguaggi per compiti particolari.

- si associa informazione a un costrutto di un linguaggio associando “**attributi**” al simbolo della grammatica che rappresenta il costrutto.
- in una **definizione guidata dalla sintassi** (syntax-directed definition) i valori degli attributi sono calcolati da “**regole semantiche**” associate alle produzioni della grammatica.

L'approccio più generale alla traduzione syntax-directed consiste nel:

- parsificare la sequenza di token in input costruendo l'albero sintattico;
- visitare i nodi dell'albero per calcolare il valore degli attributi usando le regole semantiche.

La traduzione è il risultato della valutazione delle regole semantiche.

Notazione infissa → Notazione postfissa

Produzioni

Regole semantiche

$E \rightarrow E_1 + T$

$E.code = E_1.code \parallel T.code \parallel '+'$

$E \rightarrow E_1 - T$

$E.code = E_1.code \parallel T.code \parallel '-'$

$E \rightarrow T$

$E.code = T.code$

$T \rightarrow \mathbf{digit}$

$T.code = \mathbf{digit.lexval}$

- nelle prime due produzioni l'indice 1 in E_1 distingue le diverse occorrenze di E nelle produzioni stesse.
- E e T hanno un attributo *code*, che è una stringa.
- \parallel indica la concatenazione, ad esempio la regola semantica associata alla prima produzione specifica che $E.code$ è ottenuto dalla concatenazione di $E_1.code$, di $T.code$ e del carattere $+$

Gli attributi possono essere di un tipo qualunque: numeri, tipi, riferimenti a tabelle, stringhe, sequenze di codice, . . .

Se X è un simbolo e a un suo attributo, usiamo $X.a$ per denotare il valore di a in un particolare nodo dell'albero di parsificazione etichettato X .

Le definizioni guidate dalla sintassi, quando i valori degli attributi sono definiti solo in funzione di costanti e dei valori di altri attributi, vengono anche chiamate **grammatiche ad attributi**.

Gli **attributi per i simboli terminali** hanno i valori lessicali forniti dall'analizzatore lessicale. Nelle SDD non vi sono regole semantiche per calcolare i valori degli attributi per i simboli terminali.

Per i **simboli non terminali** consideriamo due tipi di attributi:

- ***sintetizzati***: un attributo sintetizzato per una variabile A in un nodo n dell'albero di parsificazione è definito da una regola semantica associata alla produzione in n e il suo valore è calcolato solo in termini dei valori degli attributi nei nodi figli di n e in n stesso.
(A è il simbolo a sinistra nella produzione, cioè la testa).
- ***ereditati***: un attributo ereditato per una variabile A in un nodo n dell'albero di parsificazione è definito da una regola semantica associata alla produzione nel nodo padre di n e il suo valore è calcolato solo in termini dei valori degli attributi del padre di n , di n stesso e dei suoi fratelli.
(A è un simbolo nel corpo della produzione, cioè al membro destro).

Notazione infissa → Notazione postfissa

Produzioni	Regole semantiche
$E \rightarrow E_1 + T$	$E.code = E_1.code \parallel T.code \parallel '+'$
$E \rightarrow E_1 - T$	$E.code = E_1.code \parallel T.code \parallel '-'$
$E \rightarrow T$	$E.code = T.code$
$T \rightarrow \mathbf{digit}$	$T.code = \mathbf{digit.lexval}$

L'unico attributo *code* per le variabili E e T è sintetizzato, mentre il terminale **digit** ha l'attributo *lexval*, che è un valore fornito dall'analizzatore lessicale.

Per visualizzare la traduzione specificata da un SDD (definizione guidata dalla sintassi) può essere utile usare gli alberi di parsificazione, costruendo prima l'albero e poi usando le regole per valutare gli attributi in ogni nodo dell'albero. Un albero di parsificazione che mostra i valori degli attributi è chiamato ***albero di parsificazione annotato***.

Notazione infissa \rightarrow Notazione postfissa

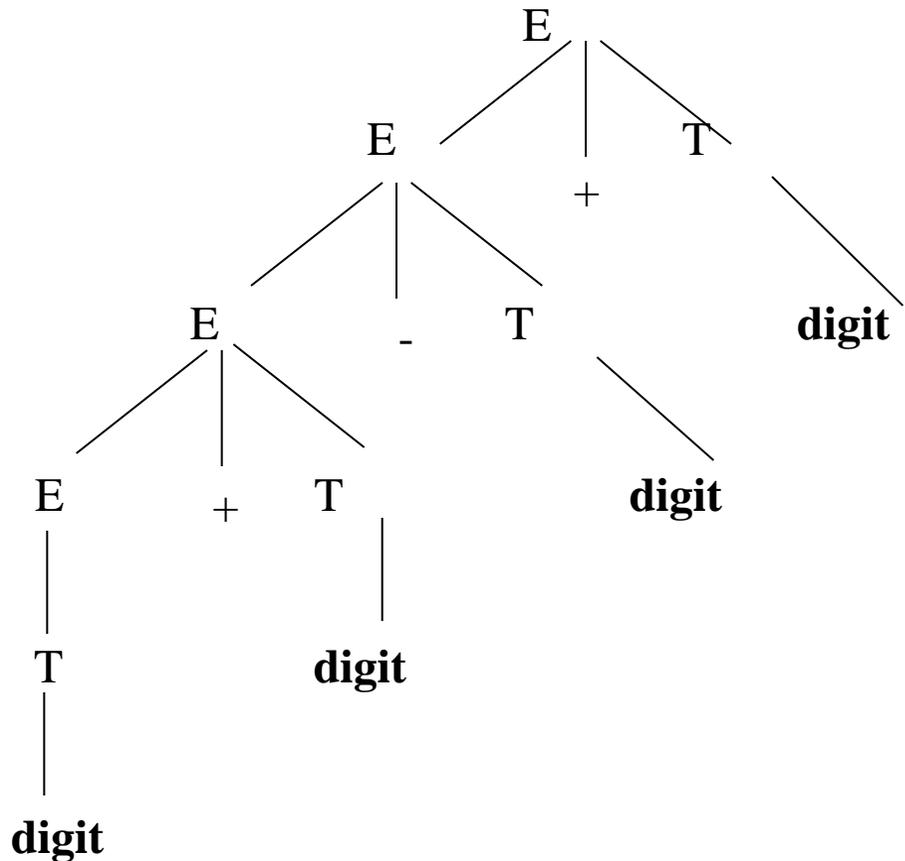
$E \rightarrow E_1 + T$ $E.code = E_1.code \parallel T.code \parallel '+'$

$E \rightarrow E_1 - T$ $E.code = E_1.code \parallel T.code \parallel '-'$

$E \rightarrow T$ $E.code = T.code$

$T \rightarrow \mathbf{digit}$ $T.code = \mathbf{digit.lexval}$

Albero di parsificazione per la stringa
digit+digit-digit+digit



Notazione infissa → Notazione postfissa

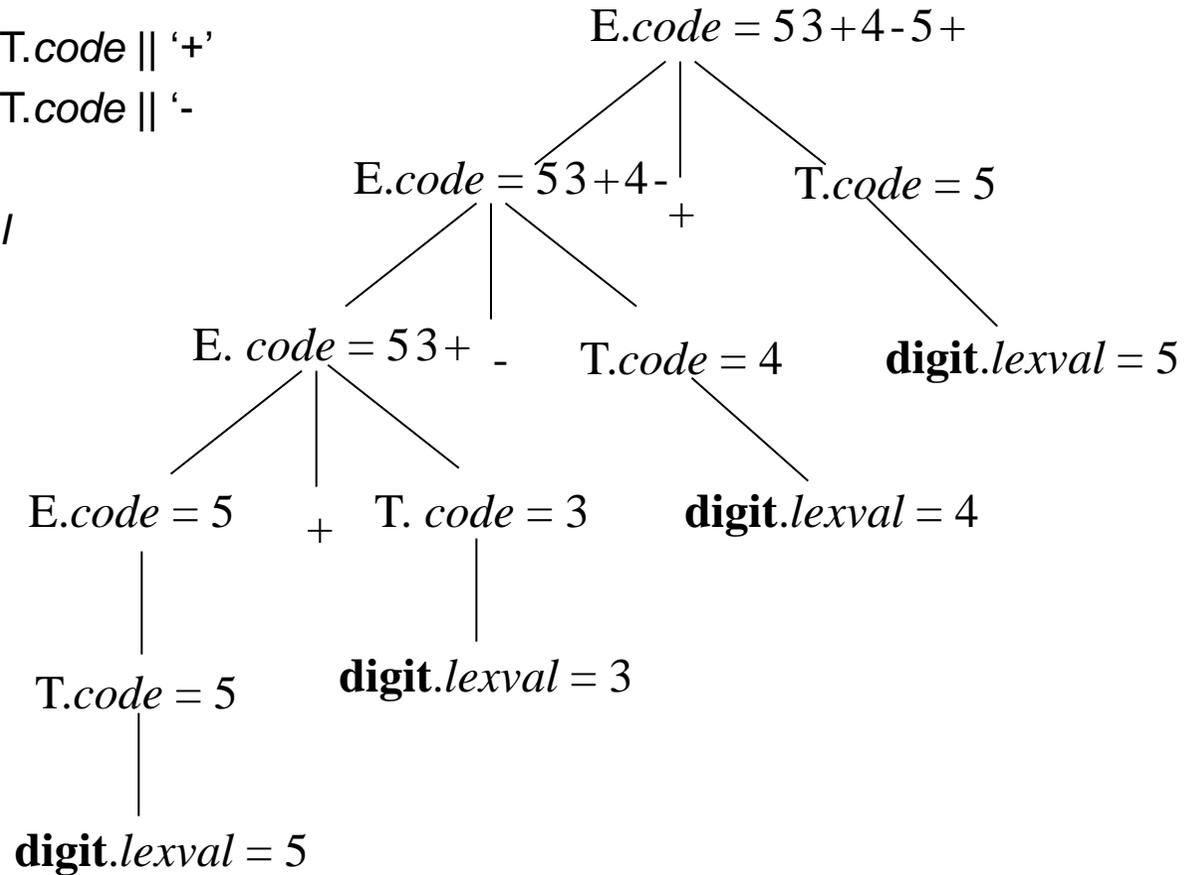
$E \rightarrow E_1 + T$ $E.code = E_1.code \parallel T.code \parallel '+'$

$E \rightarrow E_1 - T$ $E.code = E_1.code \parallel T.code \parallel '-'$

$E \rightarrow T$ $E.code = T.code$

$T \rightarrow \mathbf{digit}$ $T.code = \mathbf{digit.lexval}$

Albero di parsificazione
annotato per la stringa
5+3-4+5



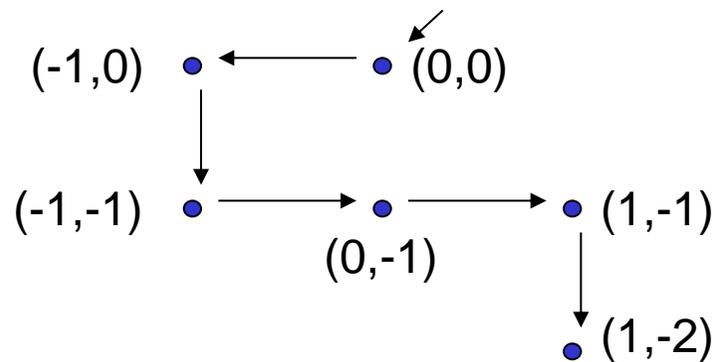
Movimenti di un robot

Seq \rightarrow Seq Istr | **begin**

Istr \rightarrow **est** | **ovest** | **nord** | **sud**

begin ovest sud est est sud

Traduciamo la sequenza in una posizione relativa al punto di partenza:



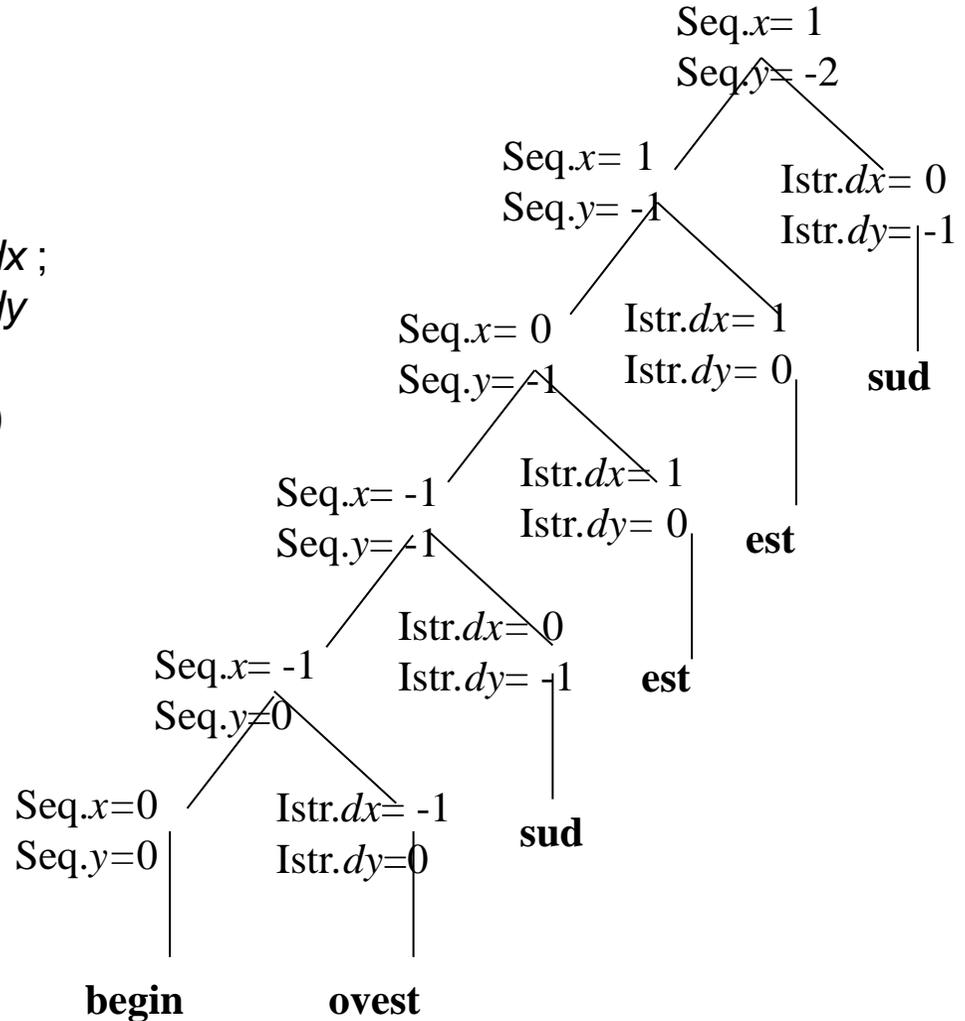
Movimenti di un robot

Produzioni	Regole semantiche
$\text{Seq} \rightarrow \mathbf{begin}$	$\text{Seq}.x = 0$ $\text{Seq}.y = 0$
$\text{Seq} \rightarrow \text{Seq}_1 \text{ Istr}$	$\text{Seq}.x = \text{Seq}_1.x + \text{Istr}.dx$ $\text{Seq}.y = \text{Seq}_1.y + \text{Istr}.dy$
$\text{Istr} \rightarrow \mathbf{est}$	$\text{Istr}.dx = 1, \text{Istr}.dy = 0$
$\text{Istr} \rightarrow \mathbf{ovest}$	$\text{Istr}.dx = -1, \text{Istr}.dy = 0$
$\text{Istr} \rightarrow \mathbf{nord}$	$\text{Istr}.dx = 0, \text{Istr}.dy = 1$
$\text{Istr} \rightarrow \mathbf{sud}$	$\text{Istr}.dx = 0, \text{Istr}.dy = -1$

Movimenti di un robot

$\text{Seq} \rightarrow \text{begin}$ $\text{Seq}.x = 0 ; \text{Seq}.y = 0$
 $\text{Seq} \rightarrow \text{Seq}_1 \text{ Istr}$ $\text{Seq}.x = \text{Seq}_1.x + \text{Istr}.dx ;$
 $\text{Seq}.y = \text{Seq}_1.y + \text{Istr}.dy$
 $\text{Istr} \rightarrow \text{est}$ $\text{Istr}.dx = 1 ; \text{Istr}.dy = 0$
 $\text{Istr} \rightarrow \text{ovest}$ $\text{Istr}.dx = -1 ; \text{Istr}.dy = 0$
 $\text{Istr} \rightarrow \text{nord}$ $\text{Istr}.dx = 0 ; \text{Istr}.dy = 1$
 $\text{Istr} \rightarrow \text{sud}$ $\text{Istr}.dx = 0 ; \text{Istr}.dy = -1$

begin ovest sud est est sud



Come costruire un albero annotato?

In quale ordine devono essere valutati gli attributi?

- prima di valutare un attributo in un nodo, si devono valutare gli attributi dai quali dipende il suo valore
- gli attributi sintetizzati possono essere valutati in ordine bottom-up
- negli SDD che hanno sia attributi sintetizzati, sia ereditati, non vi è garanzia che vi sia almeno un ordine in cui valutare gli attributi nei nodi perché potrebbero essere presenti regole “circolari” che rendono impossibile la valutazione.

Il problema di determinare se, data una SDD, esiste una circolarità in qualche albero di parsificazione da annotare è decidibile, ma ha complessità esponenziale in tempo.

Dato però un albero di parsificazione si può costruire un **grafo delle dipendenze** che mostri il flusso di informazione tra le istanze degli attributi. Tale grafo permette di scoprire se la valutazione è possibile e in tal caso trovare un ordine per la valutazione stessa.

Costruzione del grafo delle dipendenze

for (ogni nodo n dell'albero di parsificazione etichettato X)

for (ogni attributo a del simbolo X della grammatica)

costruisci un nodo per $X.a$

for (ogni nodo n etichettato X in cui è applicata una produzione p)

- se a p è associata una regola semantica che definisce l'attributo sintetizzato $X.c$ in termini del valore $Y.b$, crea un arco da $Y.b$ a $X.c$.
- se a p è associata una regola semantica che definisce l'attributo ereditato $Z.c$ in termini del valore $X.a$, crea un arco da $X.a$ a $Z.c$.

Se il grafo presenta un ciclo non è possibile valutare gli attributi.

L'ordine di valutazione deve rispettare un ordinamento topologico dei vertici del grafo.

Espressioni aritmetiche

Produzioni

$T \rightarrow F T'$

$T' \rightarrow * F T'_1$

$T' \rightarrow \varepsilon$

$F \rightarrow \text{digit}$

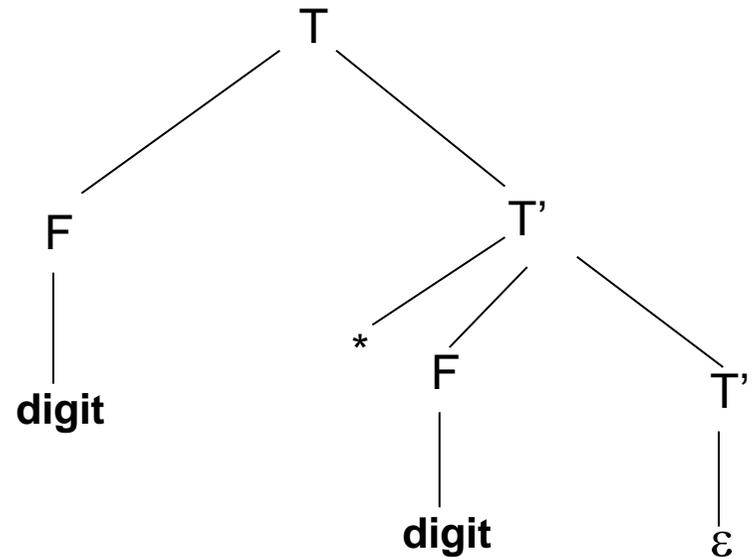
Regole semantiche

$T'.i = F.val$
 $T.val = T'.s$

$T'_1.i = T'.i \times F.val$
 $T'.s = T'_1.s$

$T'.s = T'.i$
 $F.val = \text{digit.lexval}$

Albero di parsificazione
della stringa **digit * digit**



Espressioni aritmetiche

$$T \rightarrow F T'$$

$$T'.i = F.val$$

$$T.val = T'.s$$

$$T' \rightarrow * F T_1'$$

$$T_1'.i = T'.i \times F.val$$

$$T'.s = T_1'.s$$

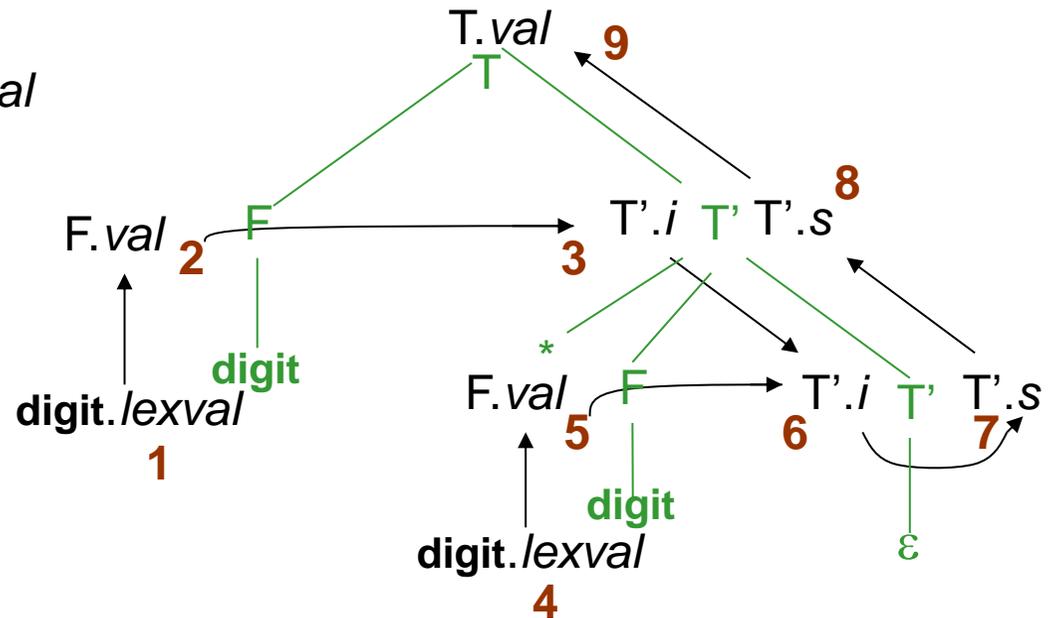
$$T' \rightarrow \varepsilon$$

$$T'.s = T'.i$$

$$F \rightarrow \text{digit}$$

$$F.val = \text{digit.lexval}$$

Grafo delle dipendenze



Lista delle differenze

La grammatica genera frasi formate da un numero seguito da #, a sua volta seguito da una sequenza di numeri separati da “;”. La definizione syntax-directed associa ad ogni frase della grammatica la lista dei numeri ottenuti sottraendo il primo a tutti gli altri.

Produzioni

Regole semantiche

$C \rightarrow N\#L$

$C.list = L.list ; L.elem = N.val$

$L \rightarrow N;L_1$

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

$L \rightarrow N$

$L.list = \underline{cons} (N.val - L.elem, \underline{null})$

$N \rightarrow \mathbf{digit}$

$N.val = \mathbf{digit.lexval}$

list e *val*: sintetizzati

elem: ereditato

Lista delle differenze

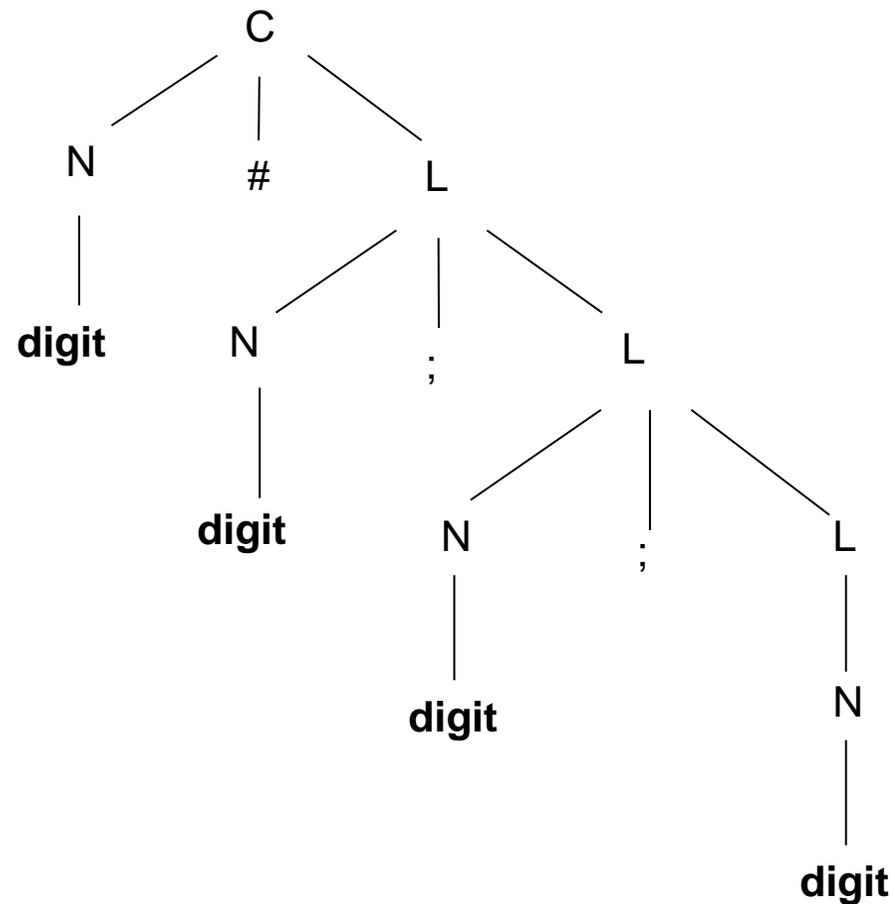
$C \rightarrow N\#L$

$L \rightarrow N;L_1$

$L \rightarrow N$

$N \rightarrow \mathbf{digit}$

Albero di parsificazione per la stringa:
digit # digit;digit;digit



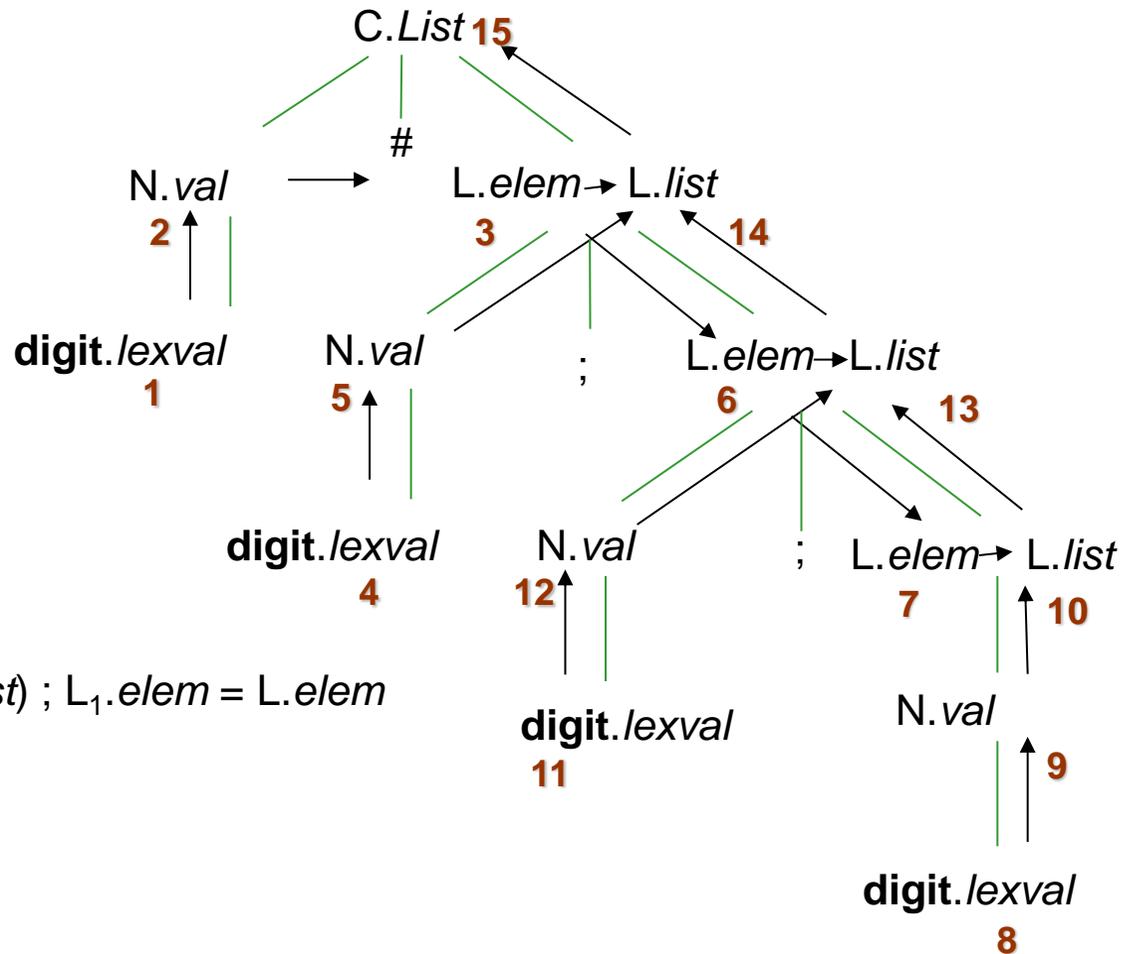
Lista delle differenze

1. $C \rightarrow N\#L$
2. $L \rightarrow N;L_1$
3. $L \rightarrow N$
4. $N \rightarrow \text{digit}$

Regole semantiche

1. $C.list = L.list ; L.elem = N.val$
2. $L.list = \underline{cons} (N.val - L.elem, L_1.list) ; L_1.elem = L.elem$
3. $L.list = \underline{cons} (N.val - L.elem, \underline{null})$
4. $N.val = \text{digit.lexval}$

Grafo delle dipendenze e Ordine di valutazione



Albero di parsificazione annotato

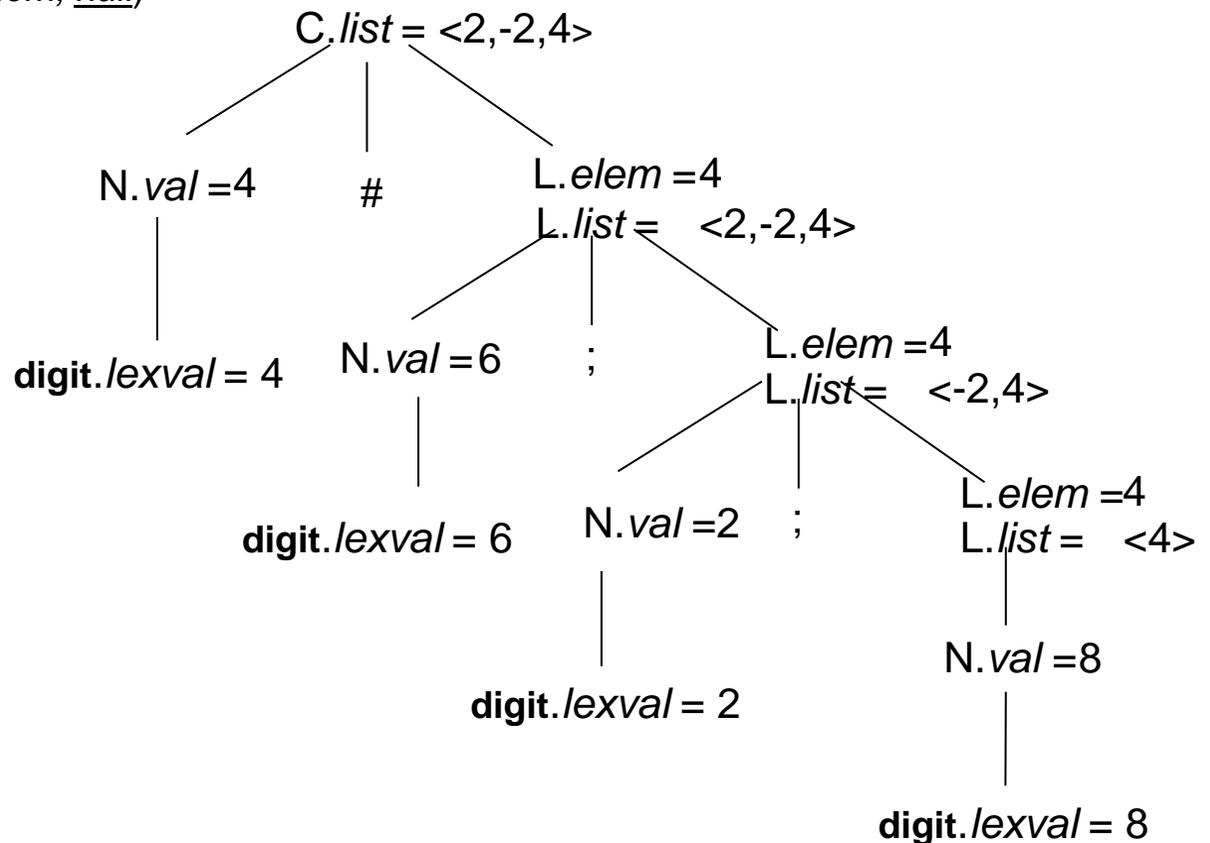
Lista delle differenze

$C \rightarrow N\#L$	$C.list = L.list ; L.elem = N.val$
$L \rightarrow N;L_1$	$L.list = \underline{cons} (N.val - L.elem , L_1.list) ;$ $L_1.elem = L.elem$
$L \rightarrow N$	$L.list = \underline{cons} (N.val - L.elem, \underline{null})$
$N \rightarrow \mathbf{digit}$	$N.val = \mathbf{digit.lexval}$

Albero annotato per la parola
4 # 6 ; 2 ; 8 fornita al
parsificatore dall'analizzatore
lessicale come

digit # digit ; digit ; digit

La traduzione di 4 # 6 ; 2 ; 8 è
la lista <2 , -2 , 4>



1. Data la grammatica con il seguente insieme di produzioni:

$$\begin{array}{ll} L \rightarrow S T & T \rightarrow S T \\ T \rightarrow \varepsilon & S \rightarrow \text{id} := E; \\ E \rightarrow \text{id} & E \rightarrow \text{num } G \\ G \rightarrow + \text{num } G & G \rightarrow \varepsilon \end{array}$$

- trovare gli insiemi guida delle produzioni;
- scrivere l'analizzatore a discesa ricorsiva;
- attribuire la grammatica in modo da associare ad ogni stringa in input il programma in cui le espressioni numeriche sono state sostituite dal loro valore, gli identificatori sostituiti dai lessemi associati e le istruzioni sono separate da una virgola.

Ad esempio la frase:

$$\text{id}_1 := \text{num}_1 + \text{num}_2 + \text{num}_3; \text{id}_2 := \text{id}_3;$$

nell'ipotesi che $\text{id}_1.name = A$, $\text{id}_2.name = B$, $\text{id}_3.name = A$ e $\text{num}_1.val = 10$, $\text{num}_2.val = 6$, $\text{num}_3.val = 2$, deve essere tradotta come:

$$A \leftarrow 18, B \leftarrow A, .$$

2. Data la grammatica con il seguente insieme di produzioni:

$$\begin{array}{lll} A \rightarrow A s B & A \rightarrow A m B & A \rightarrow B \\ A \rightarrow a & B \rightarrow (A) & B \rightarrow b \end{array}$$

Fornire una SDD che traduca ogni frase generata dalla grammatica in un intero ottenuto interpretando s e m come somma e moltiplicazione rispettivamente e supponendo $a.lexval = 2$ e $b.lexval = 5$

3. Data la grammatica con il seguente insieme di produzioni:

$$\begin{array}{ll} S \rightarrow RA \mid A[S] & R \rightarrow E = E \\ E \rightarrow (E+E) \mid a & A \rightarrow bA \mid \varepsilon \end{array}$$

- Calcolare gli insiemi guida delle produzioni, a partire dalla loro definizione, indicando i passaggi del calcolo;
- Se la grammatica risulta LL(1), scrivere la procedura di analisi a discesa ricorsiva per lo start symbol;
- Attribuire la grammatica in modo da calcolare il numero di “a” e il numero di “b” complessivamente presenti in ciascuna stringa del linguaggio e mostrare un esempio di albero annotato.

4. Nella seguente grammatica ad attributi, dire quali attributi sono ereditati e quali sono sintetizzati, motivando la risposta.

Costruire il grafo delle dipendenze e l'albero annotato per la stringa: “(3*5)+1”.

$E \rightarrow T E'$	$E'.ops = T.val$	$E.val = E'.val$
$E' \rightarrow +TE_1'$	$E_1'.ops = E'.ops + T.val$	$E'.val = E_1'.val$
$E' \rightarrow \varepsilon$	$E'.val = E'.ops$	
$T \rightarrow F T'$	$T'.opm = F.val$	$T.val = T'.val$
$T' \rightarrow *F T_1'$	$T_1'.opm = T'.opm * F.val$	$T'.val = T_1'.val$
$T' \rightarrow \varepsilon$	$T'.val = T'.opm$	
$F \rightarrow \mathbf{num}$	$F.val = \mathbf{num.lexval}$	
$F \rightarrow (E)$	$F.val = E.val$	

5. Data la grammatica con le produzioni: $\{S \rightarrow (L) \mid a, L \rightarrow L,S \mid S\}$, definire le opportune azioni semantiche per calcolare, per ciascuna stringa del linguaggio, il numero di coppie di parentesi presenti nella stringa.