

Linguaggi Formali e Traduttori

Esercizi – traduzione diretta dalla sintassi

1. Data la grammatica con il seguente insieme di 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.

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

$$T \rightarrow S \ T \quad T \rightarrow \varepsilon \quad S \rightarrow \text{id} := E;$$

$$E \rightarrow \text{num} \ G \quad G \rightarrow + \ \text{num} \ G \quad G \rightarrow \varepsilon$$

attribuirla in modo da associare ad ogni stringa generata il programma in cui le espressioni numeriche sono sostituite dal loro valore, gli identificatori dai lessemi associati, il simbolo di assegnazione è \leftarrow e le istruzioni sono separate da una virgola.

Ad esempio, nell'ipotesi $\text{id}_1.\text{name} = A$, $\text{id}_2.\text{name} = B$, $\text{num}_1.\text{val} = 10$, $\text{num}_2.\text{val} = 6$, $\text{num}_3.\text{val} = \text{num}_4.\text{val} = 2$, la frase

$\text{id} := \text{num} + \text{num} + \text{num} ; \text{id} := \text{num};$ deve essere tradotta in $A \leftarrow 18, B \leftarrow 2, .$

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

$$S \rightarrow RA \mid A[S] \quad R \rightarrow E = E$$

$$E \rightarrow (E+E) \mid a \quad A \rightarrow bA \mid \varepsilon$$

- 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. Costruire il traduttore a discesa ricorsiva a partire dal seguente schema di traduzione, dove .ok è un attributo di tipo booleano:

$$P \rightarrow D; \{S.\text{dt} = D.\text{t}\} \ S \ \{P.\text{ok} = S.\text{ok}\}$$

$$D \rightarrow x \ \text{Int} \ \{D.\text{t} = \text{Int}\}$$

$$D \rightarrow y \ \text{Bool} \ \{D.\text{t} = \text{Bool}\}$$

$$S \rightarrow x := \mathbf{num} \ \{S.\text{ok} = S.\text{dt} == \text{Int}\}$$

$$S \rightarrow y := \mathbf{true} \ \{S.\text{ok} = S.\text{dt} == \text{Bool}\}$$

5. Dato il seguente schema di traduzione:

$$L \rightarrow \{ S.pc = 1 \} S \{ T.pc = 2 \} T \{ L.out = S.out \parallel T.out \}$$

$$T \rightarrow \{ S.pc = T.pc \} S \{ T_1.pc = T.pc + 1 \} T_1 \{ T.out = S.out \parallel T_1.out \}$$

$$T \rightarrow \varepsilon \quad \{ T.out = ' ' \}$$

$$S \rightarrow \mathbf{id} := E; \{ S.out = S.pc \parallel \mathbf{id.name} \text{ ':=' } E.val \}$$

$$E \rightarrow \mathbf{num} \ G \quad \{ E.val = \mathbf{num.val} + G.val \}$$

$$G \rightarrow + \mathbf{num} \ G_1 \quad \{ G.val = \mathbf{num.val} + G_1.val \}$$

$$G \rightarrow \varepsilon \quad \{ G.val = 0 \}$$

in cui \parallel è l'operatore di concatenazione di stringhe:

- dire quali attributi sono ereditati e quali sono sintetizzati;
- scrivere il traduttore a discesa ricorsiva;
- dire che cosa associa il traduttore ad ogni stringa prodotta dalla grammatica.

6. Scrivere il traduttore deterministico bottom-up per il seguente schema di traduzione:

$$A \rightarrow bA_1A_2 \{ A.num = A_1.num + A_2.num + \langle 0,1,0 \rangle \}$$

$$A \rightarrow cA_1A_2 \{ A.num = A_1.num + A_2.num + \langle 0,0,1 \rangle \}$$

$$A \rightarrow a \{ A.num = \langle 1,0,0 \rangle \}$$

che associa ad ogni stringa generata dalla grammatica la terna formata dal numero di a , di b e di c di cui è formata la stringa.

N.B. Definiamo la somma tra k n-ple A_1, A_2, \dots, A_k come la n-pla ottenuta sommando A_1, A_2, \dots, A_k componente per componente. Ad esempio: $\langle 5,3,4 \rangle + \langle 1,0,7 \rangle = \langle 6,3,11 \rangle$.

7. Data la grammatica con l'insieme di produzione $\{ S \rightarrow E, E \rightarrow \mathbf{id} \ E', E' \rightarrow + \mathbf{id} \ E' \mid \varepsilon \}$,

- a) attribuire la grammatica in modo che, dato in input un insieme L di identificatori, verifichi che l'espressione generata da S contenga solo identificatori appartenenti all'insieme L .

Si supponga di disporre di una funzione $in(x,I)$ che verifica se un elemento x appartiene a un insieme I . Il risultato della traduzione sarà contenuto nell'attributo $S.ok$.

Per esempio, dato $L = \{A, B\}$, e l'espressione $A+B+A$, il risultato sarà *vero*, mentre sarà falso per l'espressione $A+B+C$.

- b) Scrivere le funzione del traduttore a discesa ricorsiva.

8. Fornire la traduzione nel Java bytecode degli statement:

S: $a := 8 ; b := a$

S: $\text{if } (a > b) \ a := a + a$

S: $\text{if } (a < b) \text{ then } x := y \text{ else } x := 2 * y$

In entrambi i casi si supponga $S.next = L_3$.

9. Costruire l'albero di parsificazione per il seguente programma:

$\text{if } (x = 2) \text{ while } (x < 100) \ y := x * 2$

- annotarlo con gli attributi necessari a calcolare la sua traduzione nel bytecode;
- annotarlo per la traduzione on-the fly

10. Individuare le regole semantiche per la traduzione nel bytecode dello statement 'repeat S until B', con la seguente interpretazione "esegui S; se B è vero esegui l'istruzione successiva, altrimenti ripeti il ciclo".

11. Fornire la traduzione in bytecode degli statement:

S = **if** ($a > b$) $a = a + a$

S = **if** ($a < b \ || \ c = d$) $x = y$ **else** $x = -y$

In entrambi i casi si supponga $S.next = L_3$ e che non sia ancora stato generato nessun nome temporaneo.

12. Dato il seguente frammento di codice, tradurlo in bytecode con le regole studiate e annotare il relativo albero sintattico per la traduzione on-the-fly.

S = **while** ($a < b$)

if ($c > d$) $a := a + d$ **else** $c := c + a$

Si assuma $S.next = L_5$.