LINGUAGGI FORMALI E COMPILATORI AA08/09 – UNIVERSITÀ DELLA CALABRIA
DOCENTE: ING. GIANLUIGI GRECO
ESERCITATORE: ING. ENRICO MALIZIA

Esercitazione VII

GRAMMATICHE LL(1)

1. Calcolare FIRST e FOLLOW dei simboli non terminali della seguente grammatica:

   P→begin L end
   L→ST
   T→ST | ε
   S→id := E; | read (id) ; | write (E) ;
   E →FG
   G→+ FG | ε
   F→(E) | id

   Calcolare gli insiemi guida della grammatica dell'esercizio precedente e dire se la grammatica e' LL(1), motivando la risposta.

2. Sia data la seguente grammatica:

   S -> a A c S | c
   A -> c P e | A e c
   P -> c X | c H c
   H -> H a | ε
   R -> k C k | h C
   X -> X c | c a
   C -> i A V | R
   V -> V e | V A V | ε

(a) Verificare se essa genera un linguaggio di tipo LL(1);
(b) Realizzare la tabella di parsing di un automa (anche non deterministico nel caso in cui il linguaggio generato non sia LL(1)) che contenga almeno la riga corrispondente ai non terminali: S e A.

3. Sia data la seguente grammatica:

   S -> h A b S | c
   D -> g E | g H d
   A -> b D D e | D f
   E -> E t | w
   B -> C k | h
   H -> H w H | H H w | i
   C -> i A D | B
   Z -> D Z | E

(a) Verificare se essa genera un linguaggio di tipo LL(1);
(b) Realizzare la tabella di parsing di un automa (anche non deterministico nel caso in cui il linguaggio generato non sia LL(1)) che contenga almeno la riga corrispondente ai non terminali: S e E.

SOLUZIONI

1.

FIRST (S) = { **id**, **read**, **write** }
FIRST (F) = { **(**, **id** }
FIRST (G) = { **+**, ε }
FIRST (T) = { **id**, **read**, **write**, ε }
FIRST (E) = { **(**, **id** }
FIRST (L) = { **id**, **read**, **write** }
FIRST (P) = { **begin** }

FOLLOW (P) = { **$** }
FOLLOW (L) = { **end** }
FOLLOW (E) = { **;** , **)** }
FOLLOW (T) = { **end** }
FOLLOW (G) = { **;** , **)** }
FOLLOW (S) = { **id**, **read**, **write**, **end** }
FOLLOW (F) = { **+** , **;** , **)** }

IG (P→**begin** L **end**) = { **begin** }
IG (L→ST) = { **id**, **read**, **write** }
IG (T→ST) = { **id**, **read**, **write** }
IG (T → ε) = { **end** }
IG (S→**id :=** E**;**) = { **id** }
IG (S→**read (id) ;**) = { **read** }
IG (S→**write (** E **) ;**) = { **write** }
IG (E →FG) = { **(**, **id** }
IG (G→+ FG) = { + }
IG (G→ ε) = { **;** , **)** }
IG (F→**(**E**)**) = { **(** }
IG (F→**id**) = { **id** }

La grammatica e' LL(1) perche' gli insiemi guida delle produzioni per lo stesso non terminale hanno intersezione vuota.

Costruiamo, come prova, la tabella di parsing:

	begin	**end**	**id**	**read**	**write**	**(**	**)**	**;**	**+**	**$**
P	→**begin** L **end**									
L			→ST	→ST	→ST					
T		→ ε	→ST	→ST	→ST					
S			→ **id :=** E**;**	→ **read (id) ;**	→ **write (** E **) ;**					
E			→ FG			→ FG				
G							→ ε	→ ε	→+ FG	

F			**→ id**			**→ (E)**				

2.

Eliminiamo la ricorsione sinistra

S → a A c S | c
A → c P e A’
A’ → e c A’ | ε
P → c X | c H c
H → H’
H’→ a H’ | ε
R → k C k | h C
X → c a X’
X’ → c X’ | ε
C → i A V | R
V → V’
V’ → eV’ | A V V’ | ε

Fattorizziamo

S → a A c S | c
A → c P e A’
A’ → e c A’ | ε
P → c P’
P’ → X | H c
H → H’
H’→ a H’ | ε
R → k C k | h C
X → c a X’
X’ → c X’ | ε
C → i A V | R
V → V’
V’ → eV’ | A V V’ | ε

FIRST(S) = { a, c }
FIRST(A) = { c }
FIRST(A’) = { e, ε}
FIRST(P) = { c }
FIRST(P’) = FIRST(X) U FIRST(H) = { a, c }
FIRST(H) = FIRST(H’) = { a, ε }
FIRST(H’) = { a, ε }
FIRST(R) = { k, h }
FIRST(X) = { c }
FIRST(X’) = { c, ε }
FIRST(C) = { i } U FIRST(R) = { i, h, k }
FIRST(V) = FIRST(V’) = { e, c, ε }
FIRST(V’) = { e, ε } U FIRST(A) = { e, c, ε }

FOLLOW(S) = { $ }
FOLLOW(A) = { c } U FIRST(V) = { c, e } U FOLLOW (C) U FIRST(V') = { c, e } U FOLLOW(C) U FOLLOW(V') = { c, e, k }
FOLLOW(A') = FOLLOW (A) = { c, e, k}
FOLLOW(P) = { e }
FOLLOW(P') = FOLLOW (P) = { e }
FOLLOW(H) = { c }
FOLLOW(H') = FOLLOW(H) = { c }
FOLLOW(R) = FOLLOW(C) = { k }
FOLLOW(X) = FOLLOW (P') ={ e }
FOLLOW (X') = FOLLOW (X) = { e }
FOLLOW (C) = { k } U FOLLOW (R) = { k }
FOLLOW (V) = FOLLOW (C) U FIRST (V') = { k } U FOLLOW (V') = { k, e, c }
FOLLOW(V') = FOLLOW (V) = { k, e, c }

Calcoliamo gli insiemi guida delle produzioni

IG(S → a A c S) = {a}
IG(S → c) = {c}
IG(S → a A c S) = {a}
IG(A → c P e A') = {c}
IG(A' → e c A') = {e}
IG(A' → ε) = FOLLOWS (A') = { c, e, k}
IG(P → c P') = {c}
IG(P' → X) = { c }
IG(P' → Hc) = { a, c }
IG(H → H') = { a, c }
IG(H'→ a H') = { a }
IG(H'→ ε) = { c }
IG(R → k C k) = { k }
IG(R → h C) = { h }
IG(X → c a X') = { c }
IG(X' → c X') = { c }
IG (X' → ε) = { e }
IG(C → i A V) = { i }
IG (C → R) = { k, h }
IG (V → V') = { e, c, k }
IG(V' → eV') = {e }
IG (V' → A V V') = { c }
IG (V' → ε) = { k, e, c }

La grammatica non è LL(1) poichè se consideriamo gli insiemi guida delle produzioni

IG(A' → e c A') = {e}
IG(A' → ε) = FOLLOWS (A') = { c, e, k}

IG(P' → X) = { c }
IG(P' → Hc) = { a, c }

IG(V’ → eV’) = { e }
IG (V’ → ε) = { k, e }

IG(V’ → AVV’) = { c }
IG(V’ → ε) = { e, c, k }

esse non hanno intersezione vuota

	a	C	e	i	k	h	$
S	→ a A c S	→ c					
A		→ c P e A’					
A’		→ ε	→ e c A’ → ε		→ ε		
P		→ c P’					
P’	→ Hc	→ X → Hc					
H	→ H’	→ H’					
H’	→ a H’	→ ε					
R					→ k C k	→ h C	
X		→ c a X’					
X’		→ c X’	→ ε				
C				→ i A V	→ R	→ R	
V		→ V’	→ V’		→ V’		
V’		→ A V V’ → ε	→ eV’ → ε		→ ε		

3.

S -> h A b S | c
D -> g E | g H d
A -> b D D e | D f
E -> E t | w
B -> C k | h
H -> H w H | H H w | i
C -> i A D | B
Z -> D Z | E

Eliminiamo la ricorsione sinistra

S → h A b S | c
D → g E | g H d
A → b D D e | D f
E → wE’
E’ → tE’ | ε
B → hB’ | i A Dk B’
B’ → kB’ | ε
H → i H’
H’ → w H H’ | H w H’ | ε

Z → D Z | E

Fattorizziamo

S → h A b S | c
D → g D’
D’ → E | H d
A → b D D e | D f
E → wE’
E’ → tE’ | ε
B → hB’ | i A Dk B’
B’ → kB’ | ε
H → i H’
H’ → w H H’ | H w H’ | ε
Z → D Z | E

FIRST(S) = { h, c }
FIRST(D) = { g }
FIRST(D’) = FIRST(E) U FIRST (H) = { w, i }
FIRST(A) = { b } U FIRST (D) = { b, g }
FIRST(E) = { w }
FIRST(E’) = { t, ε}
FIRST(B) = { h, i }
FIRST(B’) = { k, ε }
FIRST(H) = { i }
FIRST(H’) = { w, ε } U FIRST (H) = { w, i, ε }
FIRST(Z) = FIRST(D) U FIRST(E) = { g, w }

FOLLOW(S) = { $ }
FOLLOW (D) = { e, f , k} U FIRST(D) U FIRST(Z) = { e, f, k, g, w }
FOLLOW (D’) = FOLLOW(D) = { e, f, k, g, w }
FOLLOW(A) = { b } U FIRST(D) = { b, g }
FOLLOW(E) = FOLLOW (D’) U FOLLOW(Z) = { e, f, k, g, w }
FOLLOW(E’) = FOLLOW(E) = { e, f, k, g, w }
FOLLOW (B) = { }
FOLLOW (B’) = FOLLOW(B) = { }
FOLLOW (H) = { d, w } U FIRST (H’) = { d, w, i } U FOLLOW(H’) = { d, w, i }
FOLLOW (H’) = FOLLOW (H) = { d, w, i }
FOLLOW (Z) = { }

Calcoliamo gli insiemi guida delle produzioni

IG(S → h A b S) = {h}
IG(S → c) = {c}
IG(D → g D’) = {g}
IG(D’ → E) = {w}
IG(D’ → H d) = {i}
IG(A → b D D e) { b}
IG(A → D f) = {g}

IG(E → wE') = { w }
IG(E' → tE') = { t }
IG(E' → ε) = { e, f, k, g, w }
IG(B → hB') = { h }
IG(B → i A Dk B') = { i }
IG(B' → kB') = { k }
IG(B' → ε) = { }
IG(H → i H') = { i }
IG(H' → w H H') = { w }
IG (H' → H w H') = { i }
IG(H' → ε) = { d, w, i }
IG (Z → D Z) = { g }
IG (Z → E) = { w }

La grammatica non è LL(1) poichè se consideriamo gli insiemi guida delle produzioni

IG(H' → w H H') = { w }
IG (H' → H w H') = { i }
IG(H' → ε) = { d, w, i }

esse non hanno intersezione vuota

	h	c	g	w	i	b	t	e	f	d	k	$
S	→ h A b S	→ c										
D			→ g D'									
D'				→ E	→ H d							
A			→ D f			→ b D D e						
E				→ wE'								
E'			→ ε	→ ε			→ tE'	→ ε	→ ε		→ ε	
B	→ hB'				→ i A Dk B'							
B'											→ kB'	
H					→ i H'							
H'				→ w H H' → ε	→ H w H' → ε					→ ε		
Z			→ D Z	→ E								