Seminar 5

1) Se da gramatica G cu productiile

1: S->aSS 2: S-> aAb 3: A-> lambda

Sa se construiasca tabela LR(1) pentru G.

Se extinde G: S’->S; se introduce simbolul terminal nou, #

Se calculeaza multimile canonice LR(1)

I0= S’->.S;# —>I1 // inchidere pentru S

S->.aSS;# (#=First(lambda.#)) —>I2

S->.aAb;# —>I2

I1= S’->S.;#

I2= S->a.SS;# —>I3

S->a.Ab;# —>I4

S->.aSS;a (a=first(S#)) —>I5

S->.aAb;a —>I5

A->.;b (b=First(b#))

I3= S->aS.S;# —>I6

S->.aSS;# —>I2

S->.aAb;# —>I2

I4= S->aA.b;# —>I7

I5= S->a.SS;a —>I8

S->a.Ab;a —>I9

S->.aSS;a —>I5

S->.aAb;a —>I5

A->.;b

I6= S->aSS.;#

I7= S->aAb.;#

I8= S->aS.S;a —>I10

S->.aSS;a —>I5

S->.aAb;a —>I5

I9= S->aA.b;a —>I11

I10= S->aSS.;a

I11= S->aAb.;a

Tabela LR(1) pentru G

Tabela action Tabela goto

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| M | a | b | # | S | A |
| 0 | Shift 2 | error | error | 1 | Error |
| 1 | ... |  | accept |  |  |
| 2 | Shift 5 | Reduce 3 |  | 3 | 4 |
| 3 | Shift 2 |  |  | 6 |  |
| 4 |  | Shift 7 |  |  |  |
| 5 | Shift 5 | Reduce 3 |  | 8 | 9 |
| 6 |  |  | Reduce 1 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 |  |  | Reduce 2 |  |  |
| 8 | Shift 5 |  |  | 10 |  |
| 9 |  | Shift 11 |  |  |  |
| 10 | Reduce 1 |  |  |  |  |
| 11 | Reduce 2 |  |  |  |  |

M nu are intrari multiple (conflicte) <=> G este LR(1)

Analizam sirul aabab

(0, aabab#, lambda) -> (shift 2) (0a2, abab#, lambda) -> (shift 5) (0a2a5, bab#, lambda) -> (reduce 3) (0a2a5A9, bab#, 3) -> (shift 11) (0a2**a5A9b’11’**, ab#, 3) -> (reduce 1) (0a2S3, ab#, 13) -> (shift 2) (0a2S3a2, b#, 13) (reduce 3) (0a2S3a2A4, b#, 313) -> (0a2S3**a2A4b7**, #, 313) -> (reduce 2) (0**a2S3S6**, #, 2313) -> (reduce 1) (0S1, #, 12313) -> accept

2) Sa se construiasca parser SLR(1) pentru gramatica de la punctul 1:

1: S->aSS 2: S-> aAb 3: A-> lambda

I) Se extinde G: S’->S; se introduce simbolul terminal nou, #

Ii) Se calculeaza multimile Follow(X), X neterminal, initializam Follow(S) cu {#}

|  |  |  |
| --- | --- | --- |
| Follow | Pasul 1 | Pasul 2 |
| S | #, a |  |
| A | b |  |

S-> a**S**S Follow(S)+=First(S.Follow(S)) = {a}

S-> aAb Follow(A)+=First(b.Follow(S))={b}

Iii) Calculam multimile canonice LR(0)

I0= S’->.S —>I1

S->.aSS —>I2

S->.aAb —>I2

I1= S’->S.

I2= S->a.SS —>I3

S->a.Ab —>I4

S->.aSS —>I2

S->.aAb —>I2

A->.

I3= S->aS.S —>I5

S->.aSS —>I2

S->.aAb —>I2

I4= S->aA.b —>I6

I5= S->aSS.

I6= S->aAb.

Iv) Se calculeaza SLR(1). Reducerile se efectueaza pentru toate simbolurile din Follow (neterminalul din membrul stang al gramaticii)

Tabela action Tabela goto

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| M2 | a | b | # | S | A |
| 0 | Shift 2 | error | error | 1 | error |
| 1 | ... |  | accept |  |  |
| 2 | Shift 2 | Reduce 3 |  | 3 | 4 |
| 3 | Shift 2 |  |  | 5 |  |
| 4 |  | Shift 6 |  |  |  |
| 5 | Reduce 1 |  | Reduce 1 |  |  |
| 6 | Reduce 2 |  | Reduce 2 |  |  |

M2 nu are intrari multiple (conflicte) <=> G este SLR(1)

Analizam abb

(0, abb#, lambda) -> (shift 2) (0a2, bb#, lambda) -> (reduce 3) (0a2A4, bb#, 3) -> (shift 6) (0a2A4b6, b#, 3) -> error