SEMINAR 5

1) Se da gramatica G cu productiile: E->TR R->+TR | \*TR |lambda T->n

1: 2: 3: 4: 5:

Sa se construiasca tabela LR(1) pentru G.

Extindem G: E’->E

Construim multimile canonice LR(1)

I0= E’->.E;#—>I1 (goto(I0,E)) I1= E’->E.;#

E->.TR;#—>I2 I3= T->n.;+|\*|#

T->.n;+|\*|#- —>I3 //First(R#)

I2= E->T.R;#—>I4 (goto(I2,R))

R->.+TR;#—>I5

R->.\*TR;#—>I6

R->.;#

I4= E->TR.;#

I5= R->+.TR;#—>I7

T->.n;+|\*|#—>I3

I6= R->\*.TR;#—>I8

T->.n;+|\*|#—>I3

I7= R->+T.R;#—>I9

R->.+TR;#—>I5

R->.\*TR;#—>I6

R->.;#

I8= R->\*T.R;#—>I10

R->.+TR;#—>I5

R->.\*TR;#—>I6

R->.;#

I9= R->+TR.;#

I10= R->\*TR.;#

Tabela LR(1) pentru G:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | + | \* | n | # | E | T | R |
| 0 | error | error | Shift 3 | error | 1 | 2 | error |
| 1 | error | error | error | accept | error | error | error |
| 2 | Shift 5 | Shift 6 | error | Reduce 4 | error | error | 4 |
| 3 | Reduce 5 | Reduce 5 | error | Reduce 5 | error | error | error |
| 4 | error | error | error | Reduce 1 | error | error | error |
| 5 | error | error | Shift 3 | error | error | 7 | error |
| 6 | error | error | Shift 3 | error | error | 8 | error |
| 7 | Shift 5 | Shift 6 | error | Reduce 4 | error | error | 9 |
| 8 | Shift 5 | Shift 6 | error | Reduce 4 | error | error | 10 |
| 9 | error | error | error | Reduce 2 | error | error | error |
| 10 | error | error | error | Reduce 3 | error | error | error |

A C T I O N G O T O

Tabela *action* nu are intrari multiple <=> G este LR(1)

Sa se analizeze sirul n\*n+n

(0, n\*n+n#, lambda)->(0n3, \*n+n#, lambda)->(0T2,\*n+n#, 5)

->(0T2\*6, n+n#, 5)->(0T2\*6n3, +n#, 5)->(0T2\*6T8, +n#, 55)

->(0T2\*6T8+5, n#, 55)->(0T2\*6T8+5n3, #, 55)->(0T2\*6T8+5T7, #, 555)

->(0 T2\*6T8+5T7R9, #, 4555)->(0T2\*6T8R’10’, #, 24555)->(0T2R4, #, 324555)

->(0E1, #, 1324555) -> accept2

2) Se da gramatica G2: E-> E+E | E\*E | n

1: 2: 3:

Sa se construiasca tabela SLR(1) pentru G:

Se extinde G2: E’->E

Se calculeaza Follow(E). Se initializeaza Follow(E)={#}

E->E+E Follow(E)+=First(+E.Follow(E))={+}

E->E\*E Follow(E)+=First(\*E.Follow(E))={\*}

Deci Follow(E)={#,+,\*}

Se calculeaza multimile canonice LR(0)

I0= E’->.E—>I1

E->.E+E —>I1

E->.E\*E —>I1

E->.n —>I2

I1= E’->E.

E->E.+E —>I3

E->E.\*E —>I4

I2= E->n.

I3= E->E+.E —>I5

E->.E+E —>I5

E->.E\*E —>I5

E->.n —>I2

I4= E->E\*.E —>I6

E->.E+E —>I6

E->.E\*E —>I6

E->.n —>I2

I5= E->E+E.

E->E.+E —>I3

E->E.\*E —>I4

I6= E->E\*E.

E->E.+E —>I3

E->E.\*E —>I4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | + | \* | n | # | E |
| 0 |  |  | Shift 2 |  | 1 |
| 1 | Shift 3 | Shift 4 |  | accept |  |
| 2 | Reduce 3 | Reduce 3 |  | Reduce 3 |  |
| 3 |  |  | Shift 2 |  | 5 |
| 4 |  |  | Shift 2 |  | 6 |
| 5 | Reduce 1/ shift 3 | Reduce 1/ shift 4 |  | Reduce 1 |  |
| 6 | Reduce 2/  Shift 3 | Reduce 2/ shift 4 |  | Reduce 2 |  |

Tabela are intrari multiple <=> G2 nu este SLR(1)

Conventii pentru operatorii +, \* sunt asociativi la stanga, \* prioritar fata de +

Conflict Reduce 1/ shift 3 in starea 5, pentru ‘+’: se alege reducerea, Reduce 1

Conflict Reduce 1/ shift 4 in starea 5, pentru ‘\*’: se alege deplasarea, shift 4

….