

IMPLEMENTATION OF LR(0) ITEMS

EX. NO. 9

Parth Langalia

Date:16/3/23

AIM: : A program to implement LR(0) items

ALGORITHM:

1. Start.
2. Create structure for production with LHS and RHS.
3. Open file and read input from file.
4. Build state 0 from extra grammar Law $S' \rightarrow S \$$ that is all start symbol of grammar and one Dot (.) before S symbol.
5. If Dot symbol is before a non-terminal, add grammar laws that this non-terminal is in Left Hand Side of that Law and set Dot in before of first part of Right Hand Side.
6. If state exists (a state with this Laws and same Dot position), use that instead.
7. Now find set of terminals and non-terminals in which Dot exist in before.
8. If step 7 Set is non-empty go to 9, else go to 10.
9. For each terminal/non-terminal in set step 7 create new state by using all grammar law that Dot position is before of that terminal/non-terminal in reference state by increasing Dot point to next part in Right Hand Side of that laws.
10. Go to step 5.
11. End of state building.
12. Display the output.
13. End.

PROGRAM WITH OUTPUT:

The screenshot shows the Programiz Online C++ Compiler interface. The code in `main.cpp` is as follows:

```
1 #include<iostream>
2 #include<string.h>
3
4 using namespace std;
5
6 char prod[20][20],listofvar[26]="ABCDEFGHJKLMNOPQR";
7 int novar=1,i=0,j=0,k=0,n=0,m=0,arr[30];
8 int noitem=0;
9
10 struct Grammar
11 {
12     char lhs;
13     char rhs[8];
14 }g[20],item[20],clos[20][10];
15
16 int isvariable(char variable)
17 {
18     for(int i=0;i<novar;i++)
19         if(g[i].lhs==variable)
20             return i+1;
21     return 0;
22 }
23
24 void findclosure(int z, char a)
25 {
26     int n=0,i=0,j=0,k=0,l=0;
27     for(i=0;i<arr[z];i++)
```

The output window displays the following results:

```
A->E
E->E+T
E->T
T->T*F
T->F
F->(E)
F->I
THE SET OF ITEMS ARE

I0
A->.E
E->|.E+T
E->|.T
T->|.T*F
T->|.F
F->|. (E)
F->|. I

I1
A->E.
E->E+.T

I2
```

The browser's taskbar at the bottom shows the date as 16-03-2023 and the time as 15:31.

The screenshot shows the Programiz Online C++ Compiler interface with the same code as the first image. The output window displays the following results:

```
E->T
T->T*F
T->F
F->(E)
F->I
I5
F->I.
I6
E->E+.T
T->T*F
T->F
F->(E)
F->I
I7
T->T*.F
F->(E)
F->I
I8
F->(E.)
E->E+.T
I9
E->E+T.
T->T*.F
I10
T->T*F.
```

The browser's taskbar at the bottom shows the date as 16-03-2023 and the time as 15:33.

RESULT :

The predictive parsing table has been implemented successfully.