

Christian Mesina
Hugo Romero
Luis Escobar Urrutia

Lab 4

Introduction:

- For this lab, we created a program that writes grammars for the following languages: Strings of 0's and 1's with unequal numbers of 0's and 1's, Strings of 0's and 1's that are palindromes (the second half of the string is the reverse of the first half), and Strings of 0's and 1's of the form xx . We also generated LR(0) tables for these grammars.

Question:

Is the grammar SLR? Yes or No. If No, indicate at least one conflict.

- The first language is not an SLR. There are a total of 11 conflicts, 9 shift/reduce and 2 reduce/reduce, which prevents this language from being an SLR.
- The second language is an SLR because there are no shift/reduce or reduce/reduce errors found.

Source Code:

- **lab4.cpp**

```
/*  
 * Christian Mesina, Hugo Romero, Luis Escobar  
 * CSE 570 Compilers Lab 4  
 * Instructor: Dr. Ernesto Gomez  
 *  
 **/  
  
#include <iostream>  
#include <string>  
#include <queue>  
#include <fstream>  
#include <cstdio>  
#include <cstdlib>  
#include <string.h>  
  
using namespace std;  
  
int length;  
int record_array[100];  
int cnt_i;  
  
char array[20][20];  
char array1[20][20];  
char left_vn[100]; //record left grammar symbols  
char right_rule[100]; //the first non-terminal after dot
```

```

char I[40][40]; //to store canonical sets
char I1[40][40];
char vn1[100];

bool vt[100]; //judge whether it is terminal after dot
bool vn[100]; //judge whether it is non-terminal after dot
bool flag[100]; //non-terminal symbol
bool f[100];

queue<int>q;

int init()
{
    FILE *fp;
    int i = 0;
    if((fp = fopen("first_grammar.txt", "r")) == NULL)
    {
        printf("The file grammar.txt does not exist!\n");
        return -1;
    }
    while(fscanf(fp, "%s", array[i]) != EOF)
    {
        puts(array[i++]);
        length++;
    }
    fclose(fp);
    return 1;
}

void left()
{
    int i = 1;
    for(i; i < length; i++)
    {
        left_vn[i] = array[i][0];
    }
}

void insert(int tmp, int pos)
{
    int i = 0;
    for(i; i < length; i++)
    {
        strcpy(array1[i], array[i]);
    }
    for(i = strlen(array1[tmp]); i > pos; i--)
    {
        array1[tmp][i] = array1[tmp][i-1];
    }
    array1[tmp][i] = '.';
}

```

```

    if(array1[tmp][pos+1] >= 'A' && array1[tmp][pos+1] <= 'Z')
    {
        flag[tmp] = true;
        right_rule[tmp] = array1[tmp][pos+1];
    }
}

void closure_init(int tmp,int cnt)
{
    insert(tmp,3);
    strcpy(I[cnt],array1[tmp]);
    strcpy(I1[cnt],array1[tmp]);
    cnt++;
    if(flag[tmp] == true)
    {
        int i = 1;
        for(i; i < length; i++)
        {
            if(right_rule[tmp] == left_vn[i])
            {
                insert(i,3);
                strcpy(I[cnt],array1[i]);
                strcpy(I1[cnt],array1[i]);cnt++;
            }
        }
        cnt_i = cnt;
    }
}

void record(int cnt)
{
    int i = 0;
    for(i; i < strlen(I1[cnt]); i++)
    {
        if(I1[cnt][i] == '.')
        {
            if(i < strlen(I1[cnt])-1 && (I1[cnt][i+1] >= 'A' &&
I1[cnt][i+1] <= 'Z'))
            {
                vn[cnt] = true;
                vn1[cnt] = I1[cnt][i+1];
            }
            else
                vt[cnt] = true;
            break;
        }
    }
}

int go_j(int cnt)

```

```

{
    int i = 0;
    for(i; i < strlen(I[cnt]); i++)
    {
        if (I1[cnt][i] == '.' && (i+1 < strlen(I1[cnt])))
        {
            swap(I1[cnt][i], I1[cnt][i+1]);
            return 1;
        }
    }
    return 0;
}

int judge(int cnt, int j)
{
    int i = 0;
    for(i; i < cnt; i++)
    {
        if(strcmp(I[i], I1[j]) == 0)
            return 1;
    }
    return 0;
}

void closure()
{
    int xu = 0;
    ofstream outfile("result.txt", ios_base::out);
    outfile << "The canonical LR(0) collection of sets for the input
grammar is:" << endl;
    if(!outfile)
    {
        cerr << "error: can not open \"result.txt\" for output\n";
        exit(-1);
    }
    closure_init(0, 0);
    outfile << "I" << xu << ":" << endl;
    int i1;
    for(i1 = 0; i1 < 3; i1++)
    {
        outfile << I1[i1] << endl;
    }
    outfile << endl;
    xu++;
    int t = 0;
    int tmp = cnt_i;
    //cout << tmp << endl;
    int i = 0;
    int temp = 0;
    for(i = 0; i < cnt_i; i++)

```

```

{
    int ju = go_j(i);
    record(i);
    if((ju == 1 && vt[i] == true) || (ju == 1 &&
I1[i][strlen(I1[i])-1] == '.'))
    {
        strcpy(I[tmp],I1[i]);
        strcpy(I1[tmp],I1[i]);
        outfile << "I" << xu << ":" << endl;
        outfile << I1[tmp] << endl << endl;
        xu++;
        tmp++;
        temp = tmp;
        f[i] = true;
    }
    else if(ju == 1 && vn[i] == true)
    {
        strcpy(I[tmp],I1[i]);
        strcpy(I1[tmp],I1[i]);
        f[i] = true;
        tmp++;
        int j = 1;
        for(j; j < length; j++)
        {
            if(vn1[i] == left_vn[j])
            {
                closure_init(j,tmp);
                tmp = tmp + 1;
            }
        }

        outfile << "I" << xu << ":" << endl;
        int i2;
        for(i2 = temp; i2 < tmp+1; i2++)
        {
            outfile << I1[i2] << endl;
        }
        outfile << endl;
        xu++;
        temp = tmp;
    }
}
int cur = cnt_i;
int cal = tmp;
q.push(cal);
int x = tmp;
while(q.front() != cur)
{
    for(i = cur; i < q.front(); i++)
    {

```

```

        if(judge(cur,i))
        {
            continue;
        }
        else
        {
            int ju = go_j(i);
            record(i);
            if((ju == 1 && vt[i] == true) || (ju == 1 &&
I1[i][strlen(I1[i])-1] == '.'))
            {
                strcpy(I[cal],I1[i]);
                strcpy(I1[cal],I1[i]);
                outfile << "I" << xu << ":" << endl;
                outfile << I1[cal] << endl << endl;
                xu++;
                cal++;
                x = cal;
                f[i] = true;
            }
            else if(ju == 1 && vn[i] == true)
            {
                strcpy(I[cal],I1[i]);
                strcpy(I1[cal],I1[i]);
                f[i] = true;
                cal++;
                int j = 1;
                for(j; j < length; j++)
                {
                    if(vn1[i] == left_vn[j])
                    {
                        closure_init(j,cal);
                        cal = cal + 1;
                    }
                }
                outfile << "I" << xu << ":" << endl;
                int i3;
                for(i3 = x; i3 < cal+1; i3++)
                {
                    outfile << I1[i3] << endl;
                }
                outfile << endl;
                xu++;
                x = cal;
            }
        }
    }
    cur = q.front();
    q.push(cal);
    q.pop();

```

```

    }
}

int main()
{
    cout << "The input grammar is:" << endl;
    memset(f, sizeof(f), false);
    init();
    left();

    closure();
    system("pause");
    return 0;
}

```

- **first_grammar.txt**

$X \rightarrow 1 X 1 \mid 0 X 0 \mid e$

0010101010101010101010101010
 00000000001111111111
 0011111001111100

- **second_grammar.txt**

$S \rightarrow 0 0 1 \mid A$
 $A \rightarrow 1 0 0 \mid T$
 $T \rightarrow 0 1 1 \mid B$
 $B \rightarrow 1 1 0 \mid U$
 $U \rightarrow 0 1 0 \mid e$

Outputs:

- first_grammar.txt

```
[005319687@csusb.edu@jlb359-2 lab4]$ g++ -o lab4 lab4.cpp
[005319687@csusb.edu@jlb359-2 lab4]$ ./lab4
The input grammar is:
X
->
1
X
1
|
0
X
0
|
e
001010101010101010101010101010
00000000001111111111
0011111001111100
sh: pause: command not found
[005319687@csusb.edu@jlb359-2 lab4]$
```

```
The canonical LR(0) collection of sets for the input grammar is:
I0:
X.
```


- second_grammar.txt

```
[005319687@csusb.edu@jlb359-2 lab4]$ g++ -o lab4 lab4.cpp
[005319687@csusb.edu@jlb359-2 lab4]$ ./lab4
The input grammar is:
S
->
0
0
1
|
A
A
->
1
0
0
|
T
T
->
0
1
1
|
B
B
->
1
1
0
|
U
U
->
0
1
0
|
e
sh: pause: command not found
[005319687@csusb.edu@jlb359-2 lab4]$
```

The canonical LR(0) collection of sets for the input grammar is:

I0:

S.

1.

B.

I1:

S.

1.

B.

BA.A-100|TT-011|BB-1->

BA.A-100|TT-011|BB-1->

A.

A.

U.

I2:

BAA.-100|TT-011|BB-1->

I3:

BAA-.100|TT-011|BB-1->

I4:

BAA-1.00|TT-011|BB-1->

I5:

BAA-10.0|TT-011|BB-1->

I6:

BAA-100.|TT-011|BB-1->

I7:

BAA-100|.TT-011|BB-1->

T.

T.

I8:

BAA-100|T.T-011|BB-1->

T.

T.

I9:

BAA-100|TT.-011|BB-1->

I10:
BAA-100|TT-.011|BB-1->

I11:
BAA-100|TT-0.11|BB-1->

I12:
BAA-100|TT-01.1|BB-1->

I13:
BAA-100|TT-011.|BB-1->

I14:
BAA-100|TT-011|.BB-1->
1.
B.
B.AA-100|TT-011|BB-1->

I15:
BAA-100|TT-011|B.B-1->
1.
B.
B.AA-100|TT-011|BB-1->

I16:
BAA-100|TT-011|BB.-1->

I17:
BAA-100|TT-011|BB-.1->

I18:
BAA-100|TT-011|BB-1.->

I19:
BAA-100|TT-011|BB-1-.>

I20:
BAA-100|TT-011|BB-1->.

