Week 5 CD Lab

Rhea Adhikari
190905156
Roll NO 23
Batch: D1

Q1)

S->a | > | (T)
T->T, S|S

We remove left recursion before parsing.
Updated grammar is as follows:
S->a | > | (T)
T -> ST'
T' -> ,ST' | ε

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int curr = 0;
char str[100];

void invalid()
{
	printf("error\n");
	exit(0);
}

void valid()
{
	printf("successful\n");
	exit(0);
}

void S()
{
	if (str[curr] == 'a' || str[curr] == '>')
	{
		curr++;
		return;
	}
	else if (str[curr] == '(')
	{
		curr++;
		T();
		if (str[curr] == ')')
		{
			curr++;
			return;
		}
```

```
                else
                {
                        invalid();
                        printf("1");
                }
        }
        else
        {
                invalid();
                printf("2");
        }
}

void T()
{
        S();
        Tprime();
}

void Tprime()
{
        if (str[curr] == ',')
        {
                curr++;
                S();
                Tprime();
        }
}

void main()
{
        printf("Enter string: \n");
        scanf("%s", str);
        S();
        if (str[curr] == '$')
        {
                valid();
        }
        else
        {
                invalid();
        }
}
```

```
ugcse@prg28:~/Documents/190905156/Lab5$ gcc 1.c -o 1
ugcse@prg28:~/Documents/190905156/Lab5$ ./1
Enter String:
a>$
error
ugcse@prg28:~/Documents/190905156/Lab5$ ./1
Enter String:
(a,>)$
successful
ugcse@prg28:~/Documents/190905156/Lab5$
```

Q2)

S->UVW
U -> (S) | aSb | d
V ->aV | ε
W->cW | ε

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int curr = 0;
char str[100];

void S();
void U();
void V();
void W();

void invalid()
{
	printf("error\n");
	exit(0);
}
void valid()
{
	printf("successful");
	exit(0);
}

void S()
{
	U();
	V();
	W();
}
```

```
void U()
{
	if (str[curr] == '(')
	{
		curr++;
		S();
		if (str[curr] == ')')
		{
			curr++;
			return;
		}
		else
		invalid();
	}
	else if (str[curr] == 'a')
	{
		curr++;
		S();
		if (str[curr] == 'b')
		{
			curr++;
			return;
		}
		else
			invalid();
	}
	else if (str[curr] == 'd')
	{
		curr++;
		return;
	}
	else
		invalid();
}

void V()
{
	if (str[curr] == 'a')
	{
		curr++;
		V();
	}
}

void W()
{
	if (str[curr] == 'c')
	{
		curr++;
		W();
	}
}
```

```
void main()
{
	printf("Enter String: \n");
	scanf("%s", str);
	S();
	if (str[curr] == '$')
		valid();
	else
		invalid();
}
```

```
ugcse@prg28:~/Documents/190905156/Lab5$ ./2
Enter String:
daac$
successfulugcse@prg28:~/Documents/190905156/Lab5$ ./2
Enter String:
daac
error
ugcse@prg28:~/Documents/190905156/Lab5$
```

Q3)

S->aAcBe
A->Ab|b
B->d

We remove left recursion before parsing.
Updated grammar is as follows:
S->aAcBe
A->bA'
A'->bA'|empty
B->d

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int curr = 0;
char str[100];

void S();
void A();
void A2();
void B();

void invalid()
{
	printf("error\n");
```

```
	exit(0);
}
void valid()
{
	printf("successful");
	exit(0);
}

void S()
{
	if (str[curr] == 'a')
	{
		curr++;
		A();
		if (str[curr] == 'c')
		{
			curr++;
			B();
			if (str[curr] == 'e')
			{
			curr++;
			return;
			}
			else
			invalid();
		}
		else
			invalid();
	}
	else
		invalid();
}

void A()
{
	if (str[curr] == 'b')
	{
		curr++;
		A2();
	}
	else
		invalid();
}
void A2()
{
	if (str[curr] == 'b')
	{
		curr++;
		A2();
	}
}
void B()
```

```
{
        if (str[curr] == 'd')
        {
                curr++;
                return;
        }
        else
                invalid();
}
void main()
{
        printf("Enter String: \n");
        scanf("%s", str);
        S();
        if (str[curr] == '$')
                valid();
        else
                invalid();
}
```

```
ugcse@prg28:~/Documents/190905156/Lab5$ gcc 3.c -o 3
ugcse@prg28:~/Documents/190905156/Lab5$ ./3
Enter String:
abcde
error
ugcse@prg28:~/Documents/190905156/Lab5$ ./3
Enter String:
abcde$
successfulugcse@prg28:~/Documents/190905156/Lab5$
```

Q4)
S ->(L) | a
L ->L,S | S

We remove left recursion before parsing.
Updated grammar is as follows:
S ->(L) | a
L ->SL'
L'->,SL'| ε

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int curr = 0;
char str[200];

void S();
void L();
void L2();
```

```
void invalid()
{
	printf("error\n");
	exit(0);
}
void valid()
{
	printf("successful");
	exit(0);
}

void S()
{
	if (str[curr] == 'a'){
		curr++;
		return;
	}
	else if (str[curr] == '(')
	{
		curr++;
		L();
		if (str[curr] == ')')
		{
			curr++;
			return;
		}
		else
			invalid();
	}
	else
		invalid();
}

void L()
{
	S();
	L2();
}

void L2()
{
	if (str[curr] == ',')
	{
		curr++;
		S();
		L2();
	}
}

void main()
{
	printf("Enter String: \n");
```

```
        scanf("%s", str);
        S();
        if (str[curr] == '$')
                valid();
        else
                invalid();
}
```