Wasfi Momen

1.

2.

void localVariable() {

printf(“Enter local var statement\n”);

if(currentToken == FINAL\_Code) {

lex();

type();

} else {

type();

if(nextToken == COMMA\_CODE) {

variable\_declarator();

} else {

if(nextToken != SEMICOLON\_CODE) {

error();

}

}

}

}

3.

S -> aS’ | bS’

S’ -> aS’ | AS’ | ἐ

A -> bA | aS

4.

a) S -> bBA

-> bBaAB

-> bBaAb

-> bBaabb

-> babaabb

b) aAB, bBA, ab, b

c) ab, b

d) b

5.

a) The test fails since cBB = aaA and the first rule has aB so the two have both a in the beginning which fails the test.

b) The test fails since aB and aBb both begin with the letter a and fails the test.

c) the test passes since there are no rules that begin with the same letter.

6.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Stack | Input | Action |
| 1 | 0 | Id + (id \* id)$ | Shift 5 |
| 2 | 0 id 5 | + (id\*id)$ | Reduce 6 (0. F) |
| 3 | 0 F 3 | + (id\*id)$ | Reduce 4 (0, T) |
| 4 | 0 T 2 | + (id\*id)$ | Reduce 2 (0,E) |
| 5 | 0 E 1 | + (id\*id)$ | Shift 6 |
| 6 | 0 E 1 + 6 | (id\*id)$ | Shift 4 |
| 7 | 0 E 1 + 6 ( 4 | id\*id)$ | Shift 5 |
| 8 | 0 E 1 + 6 ( 4 id 5 | \*id)$ | Reduce 6 (4, F) |
| 9 | 0 E 1 + 6 ( 4 F 3 | \*id)$ | Reduce 4 (4,T) |
| 10 | 0 E 1 + 6 ( 4 T 2 | \*id)$ | Shift 7 |
| 11 | 0 E 1 + 6 ( 4 T 2 \* 7 | id)$ | Shift 5 |
| 12 | 0 E 1 + 6 ( 4 T 2 \* 7 id 5 | )$ | Reduce 6 (7, F) |
| 13 | 0 E 1 + 6 ( 4 T 2 \* 7 F 10 | )$ | Reduce 3 (0, T) |
| 14 | 0 E 1 + 6 ( 4 T 2 | )$ | Reduce 2 (4,T) |
| 15 | 0 E 1 + 6 ( 4 E 8 | )$ | Shift 11 |
| 16 | 0 E 1 + 6 ( 4 E 8 ) 11 | $ | Reduce 5 (6, F) |
| 17 | 0 E 1 + 6 T 9 | $ | Reduce 1 (0, E) |
| 18 | 0 E 1 | $ | Accept |

7.

a) language implementation time

b) run time

c) run time

d) language design time

8.

a)

#include <iostream>

using namespace std; stack

int n; static

int sum(int a[], int n);

int main(void) {

int i; stack dynamic

cout << "Enter number of integers to be summed: ";

cin >> n;

int \*a = new int[n]; explicit heap dynamic

for (i = 0; i < n; i++) {

cout << "Enter an integer: ";

cin >> a[i];

}

cout << "The sum is " << sum(a, n) << endl;

cout << "The sum is " << sum(a, n) << endl;

return 0;

}

int sum(int a[], int n) {

static int total = 0; static

for (int j = 0; j < n; j++) j here is stack dynamic

total += a[j];

return total;

}

b) since total is declared static, it keeps its value when the subprogram sum is ran a second time in main()