НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №2

з дисципліни **«**Системне програмування**»**

Виконав:

студент 3 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Київ – 2014 р.

Лабораторна робота 2.2

Тема роботи: Побудова і використання об’єктів вузлів деревоподібних та ієрархічних графів

Мета роботи: вивчення методів створення та використання вузлів графів автоматів, а також деревоподібних та ієрархічних графів, організації доступу до інформації, реконструкції вхідного тексту та скорочення графів через вилучення повторних вузлів.

Завдання на роботу

Завдання на підготовку до роботи на комп’ютері

1. Визначити варіант завдання для основних задач за таблицею 3.1. Визначити приклади лексем через константи в модулі тестування spLb3.cpp.

2. Ознайомитись з шаблоном програмного проекту spLb3.dsp. Настроїти відповідні дані символьних позначень та структур вузлів графів в програмному проекті на мові С.

3. Підготовити настройку управляючих таблиць мов програмування oprtrC[] або oprtrP[], cprC[] або cprP[], fpr[] і gpr[], а також згідно з варіантами, заданими в табл. 2.1. Також

згідно з варіантами визначити масиви лексем struct lxNode token[] і образів char \*imgs[] для оператора, заданого у варіанті.

4. Використати масив матриці переходів char nxtSts [Se+1][sgE+1] з файлу automat.cpp шаблону програмного проекту spLb3 для формування станів автомату, які будуть пройдені за

варіантом, заданим в таблиці 2.2.

5. Підготувати настройку програми реконструкції для еквівалентних конструкцій на альтернативній мові програмування.

6. Підготувати програмний модуль контрольної задачі, який виконує задані варіанти програм реконструкції і управління станом автомата і дозволяє перевірити коректність виконання програми і її окремих модулів.

Завдання на роботу на комп’ютері

7. Побудувати програмний проект, ввівши програмні модулі у відповідні файли проекту і налагодити синтаксис.

8. Побудувати виконуваний модуль тестової програми і налагодити змістовне виконання програми для перевірки результатів контрольних прикладів.

9. Одержати результати виконання, проаналізувати їх і зро-

бити висновки.

|  |  |  |  |
| --- | --- | --- | --- |
| 12 | case c begin 0: b:=2\*a[n]; else: b:=d end | Состояния 2..9; 8->2(*dlm*),  5->7(*cfr*), 3->7(*ltr*) | Pascal |

**Лістинг програми**

// spLb3.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <stdlib.h>

#include "visgrp.h"

#include "token.h"

#include "automat.h"

char \*imgs[100]={"c","b","2","a[n]","d","0"};

struct lxNode token[]={

{\_else, &token[1] ,&token[13], 0, 0, 0, 0, 0, NULL, 0},//0

{\_case, &token[2] ,&token[3], 0, 0, 0, 0, 0, NULL, 0},//1

{\_nam, (struct lxNode\*)imgs[0] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//2

{\_EOS, &token[15] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//3

{\_nam, (struct lxNode\*)imgs[5] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//4

{\_ass, &token[7] ,&token[6], 0, 0, 0, 0, 0, NULL, 0},//5

{\_mul, &token[9] ,&token[8], 0, 0, 0, 0, 0, NULL, 0},//6

{\_nam, (struct lxNode\*)imgs[1] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//7

{\_nam, (struct lxNode\*)imgs[3] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//8

{\_nam, (struct lxNode\*)imgs[2], 0, 0, 0, 0, 0, NULL, 0},//9

{\_nam, (struct lxNode\*)imgs[4] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//10

{\_ass, &token[12] ,&token[14], 0, 0, 0, 0, 0, NULL, 0},//11

{\_nam, (struct lxNode\*)imgs[1] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//12

{\_EOS, &token[11] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//13

{\_nam, (struct lxNode\*)imgs[4] ,NULL, 0, 0, 0, 0, 0, NULL, 0},//14

{\_cln, &token[4] ,&token[5], 0, 0, 0, 0, 0, NULL, 0},//15

};

// управляючі таблиці мов програмування

extern char \*oprtrC[], \*oprtrP[], \*oprtrV[],

\*cprC[], \*cprP[], \*cprV[];

char \*\*oprtr=oprtrP, \*\*cpr=cprP,

modeP=0, // тип роздільника операторних дужок для Паскаля

modeC=1, // тип роздільника операторних дужок для С

modeL=modeC;

int n=3, b, a[3]={0,1,2};

int main(int argc, char\* argv[])

{ prLxTxt(token);

printf("\n\n");

printf("S0");

enum autSgn ASgn[12]={dlm,dlm,rtl,dlm,sg0,dlm,rtl,sg0,rtl,dlm,rtl,sgE};

for(n=0;n<12;n++)printf("%5d->%2d ",ASgn[n],nxtStat(ASgn[n]));

printf("\n\n");

system("pause");

}Automat.h

enum autStat

{S0, // S0 - Початковий стан

S1, // S1 - Перший стан

S2, // S2 - Другий стан

S3,

S4,

S5,

S6,

S7,

S8,

S9,

Se // Se - Останній стан

};

enum autSgn

{sg0, // sg0 - Нульовий сигнал

dlm, // dlm - Перший сигнал

rtl, // rtl - Другий сигнал

sgE // sgE - Останній сигнал

};

enum autStat nxtStat(enum autSgn sgn);

automat.cpp

#include "stdafx.h"

#include "automat.h"

enum autStat nxtSts[Se+1][sgE+1] =

{// dlm,rtl

{S0,S3,S5,S7}, //для S0

{S1,S5,S4,S2}, // S1

{S1,S4,S2,S3}, // S2

{S1,S3,S4,S2}, // S3

{S1,S6,S5,S5},

{S1,S4,S9,S4}, // S5

{S1,S4,S6,S5},

{S1,S5,S4,S4},

{S1,S4,S6,S5}, // S8

{S1,S7,S8,S3}, // S9

{Se,Se,Se,Se} // Se

};

enum autStat nxtStat(enum autSgn sgn)

{static enum autStat s=S0;// поточний стан лексеми

return s=nxtSts[s][sgn];// новий стан лексеми

}