|  |
| --- |
| Київський національний університет  «Київський політехнічний інститут» |
| Факультет інформатики та обчислювальної техніки |
| Кафедра обчислювальної техніки |
| Лабораторна робота №3 |
| з дисципліни «Системне програмування» |
| *Залікова книжка № 0103* |
|  |
| Виконав студент 3-го курсу |
| групи ІО-01 |
| Редько Олександр |

|  |
| --- |
| 2012 |

Мета: Вивчення методів створення вузлів дерево­по­діб­них та ієрархічних графів, організації доступу до інформації, ре­кон­струкції вхідного тексту та скорочення графів через вилу­чення повторних вузлів

|  |  |  |  |
| --- | --- | --- | --- |
| № вар. | Вираз, який відтворюється в графі внутрішнього подання | Настроювання графа автомата з послідовними станами | Мова відтво­рення |
| 3 | do b+=a[--n]; while(n); | Стани 1..8; 3->8(*dlm*), 5->5(*ltr*) | С |

# Лістинг програми

## Sysprog\_Lab3.cpp

// Sysprog\_Lab3.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] = {"b", "a", "n"};

struct lxNode token[] = // do b+=a[--n]; while(n);

{

{\_endloop, &token[9] ,NULL, 0, 0, 0, 0, 0, NULL, 0},

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

{\_asAdd, &token[1] ,&token[3], 0, 0, 0, 0, 0, NULL, 0},

{\_ixbz, &token[6] ,&token[7], 0, 0, 0, 0, 0, NULL, 0},

{\_whileN, &token[8] ,&token[5], 0, 0, 0, 0, 0, NULL, 0},

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

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

{\_dcr, NULL ,&token[5], 0, 0, 0, 0, 0, NULL, 0},

{\_myEOS, &token[2] ,NULL, 0, 0, 0, 0, 0, NULL, 0},

{\_myEOS, &token[4] ,NULL, 0, 0, 0, 0, 0, NULL, 0}

};

// управляючі таблиці мов програмування

extern char \*oprtrC[], \*oprtrP[], \*oprtrV[], \*cprC[], \*cprP[], \*cprV[];

char \*\*oprtr = oprtrC, \*\*cpr = cprC,

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");

}

## automat.h

enum autStat

{

S0, // S0 - Початковий стан

S1, // S1 - Перший стан

S2, // S2 - Другий стан

S3,

S4,

S5,

S6,

S7,

S8,

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,S8,S4,S2}, // S3

{S1,S6,S5,S5},

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

{S1,S4,S6,S5},

{S1,S5,S4,S4},

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

{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

}