Міністерство освіти та науки України Національний технічний університет України «КПІ»

Факультет інформатики та обчислювальної техніки

**Автоматизація проектування комп'ютерних систем**

Лабораторна робота №6

Автоматизація мінімізації булевих функцій

Виконав

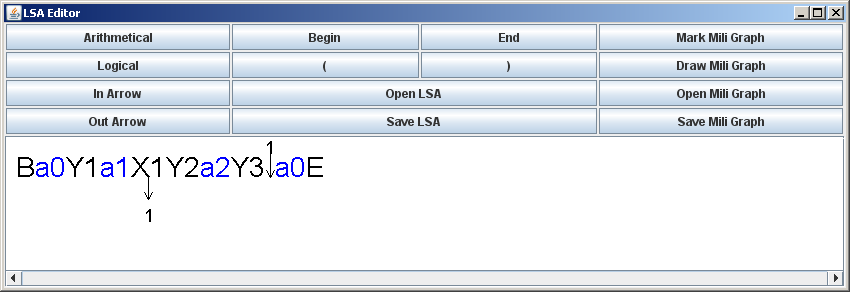
ст. гр. ІО-71

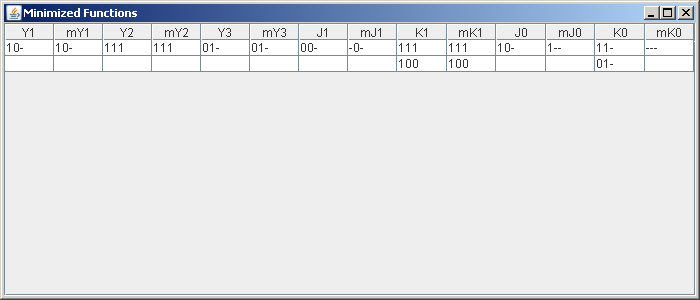
Ювженко Д.

**Київ 2010**

Мета: здобуття навичок автоматизації процедури мінімізації булевих функцій методом Квайна-МакКласкі.

Скріншоти програми:





**private** String[] minimizeFunction(ArrayList<String> Constituents, ArrayList<String> CoveringTableConstituents) {

ArrayList<String> NewCube = (ArrayList<String>) Constituents.clone();

ArrayList<String> CurrentCube;

ArrayList<Integer> CurrentAbsorptionCube = **new** ArrayList<Integer>();

ArrayList<ArrayList<String>> Cubes = **new** ArrayList<ArrayList<String>>();

ArrayList<ArrayList<Integer>> AbsorptionCubes = **new** ArrayList<ArrayList<Integer>>();

**boolean** Splicing = **false**;

**do** {

Splicing = **false**;

CurrentCube = (ArrayList<String>) NewCube.clone();

NewCube.clear();

CurrentAbsorptionCube.clear();

**for** (**int** i = 0; i<CurrentCube.size()-1; i++)

**for** (**int** j = i+1; j<CurrentCube.size(); j++) {

**boolean** OneBitDifferece = **false**;

**int** OneBitDiffereceIndex = 0;

**for** (**int** k = 0; k<CurrentCube.get(i).length(); k++)

**if** (CurrentCube.get(i).charAt(k)!=CurrentCube.get(j).charAt(k))

**if** ((CurrentCube.get(i).charAt(k)!='-') && (CurrentCube.get(j).charAt(k)!='-') && (!OneBitDifferece)) {

OneBitDifferece = **true**;

OneBitDiffereceIndex = k;

}

**else** {

OneBitDifferece = **false**;

**break**;

}

**if** (OneBitDifferece) {

**if** ((!NewCube.contains(CurrentCube.get(i).substring(0, OneBitDiffereceIndex)+"-"+CurrentCube.get(i).substring(OneBitDiffereceIndex+1, CurrentCube.get(i).length()))))

NewCube.add(CurrentCube.get(i).substring(0, OneBitDiffereceIndex)+"-"+CurrentCube.get(i).substring(OneBitDiffereceIndex+1, CurrentCube.get(i).length()));

**if** (!CurrentAbsorptionCube.contains(i))

CurrentAbsorptionCube.add(i);

**if** (!CurrentAbsorptionCube.contains(j))

CurrentAbsorptionCube.add(j);

}

}

Cubes.add((ArrayList<String>) CurrentCube.clone());

AbsorptionCubes.add((ArrayList<Integer>) CurrentAbsorptionCube.clone());

**if** (!NewCube.isEmpty())

Splicing = **true**;

} **while** (Splicing);

ArrayList<String> TDNF = **new** ArrayList<String>();

**for** (**int** i = 0; i<Cubes.size(); i++)

**for** (**int** j = 0; j<Cubes.get(i).size(); j++)

**if** (!AbsorptionCubes.get(i).contains(j))

TDNF.add(Cubes.get(i).get(j));

Integer[][] CoveringTable = **new** Integer[TDNF.size()][CoveringTableConstituents.size()];

**for** (**int** i = 0; i<CoveringTable.length; i++)

**for** (**int** j = 0; j<CoveringTable[i].length; j++) {

**boolean** Cover = **true**;

**for** (**int** k = 0; k<TDNF.get(i).length(); k++)

**if** ((TDNF.get(i).charAt(k)!=CoveringTableConstituents.get(j).charAt(k)) && (TDNF.get(i).charAt(k)!='-')) {

Cover = **false**;

**break**;

}

**if** (Cover)

CoveringTable[i][j] = 1;

**else**

CoveringTable[i][j] = 0;

}

Boolean[] CoveredConstituents = **new** Boolean[CoveringTableConstituents.size()];

**for** (**int** i = 0; i<CoveredConstituents.length; i++)

CoveredConstituents[i] = **false**;

Boolean AllConstituentsAreCovered = **false**;

ArrayList<String> MDNF = **new** ArrayList<String>();

**while** (!AllConstituentsAreCovered) {

**int** MinConstituentOnesQuantity = Integer.*MAX\_VALUE*;

ArrayList<Integer> MinConstituentOnesNumbers = **new** ArrayList<Integer>();

**for** (**int** j = 0; j<CoveringTable[0].length; j++)

**if** (!CoveredConstituents[j]) {

**int** CurrentConstituentOnesQuantity = 0;

**for** (**int** i = 0; i<CoveringTable.length; i++)

**if** (CoveringTable[i][j]==1)

CurrentConstituentOnesQuantity++;

**if** (CurrentConstituentOnesQuantity<MinConstituentOnesQuantity) {

MinConstituentOnesQuantity = CurrentConstituentOnesQuantity;

MinConstituentOnesNumbers.clear();

}

**if** (CurrentConstituentOnesQuantity==MinConstituentOnesQuantity)

MinConstituentOnesNumbers.add(j);

}

**int** MaxUncoveredConstituentsQuantity = 0;

**int** MaxUncoveredConstituentsNumber = 0;

**for** (**int** j = 0; j<MinConstituentOnesNumbers.size(); j++) {

ArrayList<Integer> CurrentConstituentCoveringImplicantsNumbers = **new** ArrayList<Integer>();

**for** (**int** k = 0; k<CoveringTable.length; k++)

**if** (CoveringTable[k][MinConstituentOnesNumbers.get(j)] == 1)

CurrentConstituentCoveringImplicantsNumbers.add(k);

ArrayList<Integer> CurrentConstituentCoveringConstituentsNumbers = **new** ArrayList<Integer>();

**for** (**int** k = 0; k<CurrentConstituentCoveringImplicantsNumbers.size(); k++)

**for** (**int** l = 0; l<CoveringTable[CurrentConstituentCoveringImplicantsNumbers.get(k)].length; l++)

**if** ((CoveringTable[CurrentConstituentCoveringImplicantsNumbers.get(k)][l]==1) && (!CoveredConstituents[l]) && (!CurrentConstituentCoveringConstituentsNumbers.contains(l)))

CurrentConstituentCoveringConstituentsNumbers.add(l);

**if** (CurrentConstituentCoveringConstituentsNumbers.size()>MaxUncoveredConstituentsQuantity) {

MaxUncoveredConstituentsQuantity = CurrentConstituentCoveringConstituentsNumbers.size();

MaxUncoveredConstituentsNumber = MinConstituentOnesNumbers.get(j);

}

}

**for** (**int** i = 0; i<CoveringTable.length; i++)

**if** (CoveringTable[i][MaxUncoveredConstituentsNumber]==1) {

MDNF.add(TDNF.get(i));

**for** (**int** j = 0; j<CoveredConstituents.length; j++)

**if** (CoveringTable[i][j]==1)

CoveredConstituents[j] = **true**;

}

AllConstituentsAreCovered = **true**;

**for** (**int** i = 0; i<CoveredConstituents.length; i++)

**if** (!CoveredConstituents[i]) {

AllConstituentsAreCovered = **false**;

**break**;

}

}

String[] ArrayMDNF = **new** String[MDNF.size()];

ArrayMDNF = MDNF.toArray(ArrayMDNF);

**return** ArrayMDNF;

}

Висновок: під час виконання даної лабораторної роботи були отримані навички автоматизації процедури мінімізації булевих функцій методом Квайна-МакКласкі. Була розроблена процедура мінімізації булевих функцій методом Квайна-МакКласкі, реалізовані засоби збереження результатів мінімізації у файл.