VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

MATEMATINĖS INFORMATIKOS KATEDRA

Tiesioginio ir atbulinio išvedimo (Forward and Backward Chaining) produkcijų sistemoje programa C# kalba

Darbą atliko 4 kurso kompiuterių moklso studentas

Saulius Karmanovas

Vilnius

2018

# Turinys

# Įvadas

Dažnai tenka sutikti paieškos uždavinių. Viena ar kita forma jie susiveda į tokias dalis: taisyklių sąrašas, pradinių faktų sąrašas ir tikslas. Norint išspręsti šį uždavinį reikia nustatyti, ar naudojant duotas taisykles iš pradinių faktų galima išvesti duotąjį tikslą.

Šiame darbe nagrinėjami du šios užduoties sprendimo būdai. Juose naudojama produkcijų sistema susidedanti iš:

* Globalios duomenų bazės, tai yra teisingų teiginių arba faktų sąrašas. Dažnai žymimi didžiosiomis abecėlės raidėmis.
* Produkcijų aibės, kurios kiekviena produkcija turi sąrašą prielaidų ir vieną išvadą. Produkciją galima taikyti tik tuomet, kai visos jos prielaidos yra tarp faktų arba išvestos iš jų naudojant kažkurias kitas produkcijas. Pritaikius taisyklę jos išvada yra patalpinama į faktų aibę.

# Duomenų įvesties failo formatas

Tiesioginio ir atbulinio išvedimo uždaviniams spręsti progrma naudoja tą patį įvesties failą. Įvesties failas aprašomas pagal tokias taisykles:

1. Failo pirmoje eilutėje yra vienas simbolis, kuris reiškia tikslą
2. Antroje eilutėje faktų sąrašas (be tarpų)
3. Trečia eilutė ir visos likusios yra išvedimo taisyklės. Eilutėje pirma eina prielaidų sąrašas (be tarpų), o po vieno tarpo – taisyklės išvada.

# Tiesioginio išvedimo uždavinys

Šis uždavinys formuluojams taip: surasti taisyklių seką, kurias panaudojus kartu su pradiniu faktų aibe yra išvedama nauja aibė, kurioje yra tikslas. Jeigu tokios sekos suformuoti neįmanoma, šį uždavinį sprendžianti programa turi apie tai pranešti.

# Algoritmo pseudokodas

Įvestis:

1. Tikslas
2. Aibė fauktų
3. Aibė taisyklių

Algoritmo išvestis: Sąrašas taisyklių, kurias taikant išprendžiamas uždavinys

Kodas:

FC(tikslas, faktai, taisyklės)

while(faktuose nėra tikslo)

if(taisykliu nera)

return Neimanoma išvesti

for(taisyklė iš taisyklių){

if (taisyklė panaudota)

imame sekančią taisyklę

if (taisyklės išvada tarp faktų)

imame sekančią taisyklę

if (taisyklės visos prielaidos tarp faktų)

pridedame išvadą prie faktų

pažymime taisyklę kaip panaudotą

nutraukiame ciklą

if (cikle nei vienos taisyklės nepanaudojo)

return tikslas nepasiekiamas

return tikslas pasiektas

# Pavyzdžiai:

## Pirmas pavyzdys:

Įvestis:

Z

ABC

A L

L K

D A

D M

BF Z

DC F

A D

Išvestis

Tikslas: Z

Faktai:

A B C

Taisyklės:

R1: 'A'->'L'

R2: 'L'->'K'

R3: 'D'->'A'

R4: 'D'->'M'

R5: 'B'->'F'->'Z'

R6: 'D'->'C'->'F'

R7: 'A'->'D'

1 ITERACIJA

R1:'A'->'L' taikoma, Pakeliama flag1 A B C ir L

2 ITERACIJA

R1:'A'->'L' praleidziama, nes pekelta flag1

R2:'L'->'K' taikoma, Pakeliama flag1 A B C ir L K

3 ITERACIJA

R1:'A'->'L' praleidziama, nes pekelta flag1

R2:'L'->'K' praleidziama, nes pekelta flag1

R3:'D'->'A' netaikoma, nes konsekventas faktuose. Pakeliama flag2

R4:'D'->'M' netaikoma, nes truksta D

R5:'B'->'F'->'Z' netaikoma, nes truksta F

R6:'D'->'C'->'F' netaikoma, nes truksta D

R7:'A'->'D' taikoma, Pakeliama flag1 A B C ir L K D

4 ITERACIJA

R1:'A'->'L' praleidziama, nes pekelta flag1

R2:'L'->'K' praleidziama, nes pekelta flag1

R3:'D'->'A' praleidziama, nes pekelta flag2

R4:'D'->'M' taikoma, Pakeliama flag1 A B C ir L K D M

5 ITERACIJA

R1:'A'->'L' praleidziama, nes pekelta flag1

R2:'L'->'K' praleidziama, nes pekelta flag1

R3:'D'->'A' praleidziama, nes pekelta flag2

R4:'D'->'M' praleidziama, nes pekelta flag1

R5:'B'->'F'->'Z' netaikoma, nes truksta F

R6:'D'->'C'->'F' taikoma, Pakeliama flag1 A B C ir L K D M F

6 ITERACIJA

R1:'A'->'L' praleidziama, nes pekelta flag1

R2:'L'->'K' praleidziama, nes pekelta flag1

R3:'D'->'A' praleidziama, nes pekelta flag2

R4:'D'->'M' praleidziama, nes pekelta flag1

R5:'B'->'F'->'Z' taikoma, Pakeliama flag1 A B C ir L K D M F Z

Tikslas Z gautas

Kelias R1 R2 R7 R4 R6 R5