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:

* Failo pirmoje eilutėje yra vienas simbolis, kuris reiškia tikslą
* Antroje eilutėje faktų sąrašas (be tarpų)
* 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:

* Tikslas
* Aibė fauktų
* 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:**

Iš Intelektualių sitemų konspekto

**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

**Antras pavyzdys:**

Įvestis:

Z

A

G Z

A G

B C

C D

D Z

Išvestis:

Tikslas: Z

Faktai:

A

Taisykles:

R1: 'G'->'Z'

R2: 'A'->'G'

R3: 'B'->'C'

R4: 'C'->'D'

R5: 'D'->'Z'

1 ITERACIJA

R1:'G'->'Z' netaikoma, nes truksta G

R2:'A'->'G' taikoma, Pakeliama flag1 A ir G

2 ITERACIJA

R1:'G'->'Z' taikoma, Pakeliama flag1 A ir G Z

Tikslas Z gautas

Kelias R2 R1

**Trečias pavyzdys:**

Įvestis:

Z

A

D Z

C D

B C

A B

A G

Išvestis:

Tikslas: Z

Faktai:

A

Taisykles:

R1: 'D'->'Z'

R2: 'C'->'D'

R3: 'B'->'C'

R4: 'A'->'B'

R5: 'A'->'G'

1 ITERACIJA

R1:'D'->'Z' netaikoma, nes truksta D

R2:'C'->'D' netaikoma, nes truksta C

R3:'B'->'C' netaikoma, nes truksta B

R4:'A'->'B' taikoma, Pakeliama flag1 A ir B

2 ITERACIJA

R1:'D'->'Z' netaikoma, nes truksta D

R2:'C'->'D' netaikoma, nes truksta C

R3:'B'->'C' taikoma, Pakeliama flag1 A ir B C

3 ITERACIJA

R1:'D'->'Z' netaikoma, nes truksta D

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

4 ITERACIJA

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

Tikslas Z gautas

Kelias R4 R3 R2 R1

**Ketvirtas pavyzdys:**

Įvestis:

Z

AZ

Išvestis:

Tikslas: Z

Faktai:

A Z

Taisykles:

Tikslas Z gautas

Kelias

**Penktas pavyzdys:**

Įvestis:

Z

A

A B

BC Z

Išvestis:

Tikslas: Z

Faktai:

A

Taisykles:

R1: 'A'->'B'

R2: 'B'->'C'->'Z'

1 ITERACIJA

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

2 ITERACIJA

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

R2:'B'->'C'->'Z' netaikoma, nes truksta C

Tikslas nepasiekiamas

**Atbulinis išvedimas:**

Atbulinis išvedimo algoritmas eina nuo tikslo ir bando įrodyti (išvesti) vieną faktą. Tarp taisyklių ieško tų, kurių išvada atitinka tikslą ir poto bando išvesti visas taisyklės prielaidas.

**Pseudo kodas:**

bool Prove(taisyklės, faktai, tikslas, paliesti\_tikslai, įrodyti\_faktai)

if (tikslas tarp paliesti\_faktai)

return false

if (tikslas tarp faktai)

return true

if (tikslas tarp įrodyti\_faktai)

return true

taisyklė = pasirinkti pirmą taisyklę iš taisyklės, kuri:

taisyklė.išvada == tikslas and visos taisyklė.prielaidos tenkina:

Prove(taisyklės, gaktai, prielaida,

paliesti faktai + tikslas, įrodyti\_faktai)

if(taisyklė == null)

return false

įrodyti\_faktai + tikslas

return true;

**Pavyzdžiai:**

Iš Intelektualių sitemų konspekto

**Pirmas pavyzdys:**

Įvestis:

Z

T

CD Z

T C

T Z

Išvestis:

Tikslas: Z

Faktai:

T

Taisykles:

R1: 'C'->'D'->'Z'

R2: 'T'->'C'

R3: 'T'->'Z'

Trying rule R1

Skipping rule R1

Trying rule R2

Original facts contain goal T

Proved new fact C

Skipping rule R1

Skipping rule R2

Skipping rule R3

All rules exhausted FAIL

Skipping rule R2

Trying rule R3

Original facts contain goal T

Proved new fact Z

Goal was proved

Rules: R2 R3

