

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINIERIJOS KATEDRA**

**Andrej Afonin**

**ABSTRAKČIŲ AUTOMATŲ STEBIMUMO  
NUSTATYMO BEI PADIDINIMO TYRIMAS**

Magistro darbas

**Vadovas  
habil. dr. R. Šeinauskas**

**KAUNAS, 2005**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINIERIJOS KATEDRA**

**TVIRTINU  
Katedros vedėjas  
dr. E. Bareiša  
2005-05-25**

**ABSTRAKČIŲ AUTOMATŲ STEBIMUMO  
NUSTATYMO BEI PADIDINIMO TYRIMAS**

Informatikos magistro baigiamasis darbas

**Kalbos konsultantė  
Lietuvių kalbos katedros lektorė  
dr. J. Mikelionienė  
2005-05-23**

**Recenzentas  
2005-05-23**

**Vadovas  
habil.dr.R. Šeinauskas  
2005-05-23**

**Atliko  
IFM 9/2 gr. stud.  
A. Afonin  
2005-05-23**

**KAUNAS, 2005**

**KVALIFIKACINĖ KOMISIJA**

**Pirmininkas:** Raimundas Stulpinas, UAB „Strauja“ generalinis direktorius

**Sekretorius:** Kęstutis Motiejūnas, docentas

**Nariai:** Eduardas Bareiša, docentas

Rimantas Butleris, docentas

Egidijus Kazanavičius, profesorius

Dalius Rubliauskas, docentas

Rimantas Šeinauskas, profesorius

Vytautas Štuikys, profesorius

## SUMMARY

A circuit testing nowadays is expensive and complex process. That's why circuits testing, errors finding and fixing require more and more investments. One of possible ways of reducing cost and speed up testing process is increasing controllability and observability of circuits. It takes a lot of time to find out circuit's controllability and observability that's why that process have to be computerized.

For that purpose was decided to create software which will be helpful for circuit designers in that process. As a result it will help designers in making design for testability schemes.

Research and training action for system on chip using internet software is dedicated for users that have C/C++ system on chip code from every place in the world, using only web browser, would be able put that code in to the server, test it and retrieve it's observability results in text and visual modes and also increase system's on chip observability level. And as well review and get acquainted with systems on chip that are already on the server and are stored on it.

Software architecture uses client-server mode. All computations are performed on a server side. System is realized on Apache server with Linux OS. System modules are realized using HTML, JAVA, PHP, JavaScript, C++ and CGI programming languages.

Web page is working independent from users OS, user needs only web browser (Internet Explorer not older than 3.0 ver., Opera not older than 6.0 ver., Netscape Navigator not older than 3.0 ver.). If user wants put, compile and get acquainted with results in text and forms mode, the only requirement is having Internet browser. If user wants to see visual mode hi needs to have Internet browser that supports Java. If user wants save work results and take them to another PC, hi have option to save it as .pdf type file.

Andrej AFONIN

**TURINYS**

1. Įvadas.....	10
1.1 Temos aktualumas.....	10
1.2 Projekto kūrimo pagrindas.....	10
1.3 Darbo tikslas .....	11
1.4 Darbo struktūra .....	12
1.5 Santrauka .....	12
2. Analitinė dalis .....	14
2.1 Schemos testuojamumas.....	14
2.2 Loginės schemos stebimumas.....	14
2.2.1 Mazgo stebimumo apibrėžimas .....	14
2.2.2 Stebimumo sąvokos analizė.....	14
2.3 Stebimumo apskaičiavimo metodai.....	16
2.3.1 TMEAS Stivensono ir Greisono metodas (Stephenson & Grason’s Approach— TMEAS) .....	16
2.3.2 SCOAP Goldsteino metodas(Goldstein’s approach - SCOAP).....	17
2.3.3 CAMELOT – Bennets‘o, Mauder‘io ir Robinson‘o metodas (Computer-Aided MEasure for LOGic Testability - Kompiuterio pagalba įvertinimas loginiam testavimui) .....	19
2.4 Testuojamumo matų (kontroliškumo ir stebimumo matų) svarbumas [12] [15] .....	19
2.5 SPF apskaičiavimas .....	20
2.6 Papildomų jungčių pridėjimas [13] .....	20
2.7 Sprendimas.....	23
2.7.1 Iteracinio modelio sąvoka.....	23
2.7.2 Stebimumo apskaičiavimas.....	24
2.7.3 Stebimumo pagerinimas .....	24
3. Projektinė dalis.....	26
3.1 Sistemos paskirtis .....	26
3.2 Apribojimai sprendimui .....	26
3.3 Diegimo aplinka .....	27
3.4 Bendradarbiaujančios sistemos .....	28
3.5 Veiklos sfera.....	28
3.5.1 Veiklos kontekstas.....	28
3.5.2 Veiklos pavadinimas .....	29

3.5.3	Panaudojimo atvejų diagrama.....	30
3.6	Bendra sistemos architektūra .....	31
3.7	Stebimumo nustatymo detali architektūra .....	33
3.8	Stebimumo padidinimo detali architektūra.....	34
4.	Eksperimentinė dalis .....	36
4.1	Tyrimo objektas .....	36
4.2	Terminų apibrėžimas .....	36
4.3	Tyrimo prielaidos ir metodika .....	36
4.3.	Eksperimentų rezultatai .....	38
4.3.1.	Automato M03 tyrimas.....	38
4.3.2.	Automato M05 tyrimas.....	47
4.3.3.	Automato M05 tyrimas.....	57
5.	Išvados .....	60
6.	Literatūra .....	61
7.	Santrumpų ir terminų žodynas.....	66
8.	Priedai.....	67

**PAVEIKSLŲ SĄRAŠAS**

1.1.2 pav Testavimo išlaidų priklausomybė.....	11
2.2.2.1 pav Tiriama schema.....	15
2.3.2.1 pav Tiriama schema 2.....	17
2.6.1 pav Originali schema.....	21
2.6.2 pav Stebimumo didinimas. Pirmas būdas.....	22
2.6.3 pav Stebimumo didinimas. Antras būdas.....	23
2.7.1 pav Iteracinis modelis.....	24
2.7.3.1 pav Automato stebimumo pagerinimo schema.....	25
3.5.1.1 pav Konteksto diagrama.....	28
3.5.3.1 pav PA diagrama.....	30
3.6.1 pav Sistemos struktūra.....	31
3.6.2 Sistemos architektūra.....	32
3.7.1 pav stebimumo nustatymo struktūrinė schema.....	33
3.8.1 pav komponento darbo struktūra.....	34
3.8.2 skaičiavimų eiliškumas.....	35
4.3.1.3 pav M03 automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė.....	40
4.3.1.4 pav M03 pagerinto automato stebimumo priklausomybė nuo iteracijų skaičiaus.....	41
4.3.1.5 pav M03 pagerinto automato tikslumo priklausomybė nuo iteracijų skaičiaus.....	42
4.3.1.6 pav M03 pagerinto automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė.....	43
4.3.1.7 pzv M03 automato bazinio ir pagerinto modelio stebimumo palyginimas.....	44
4.3.1.8 pav M03 automato bazinio ir pagerinto automatų stebimumo skirtumas vienetais.....	45
4.3.1.9 pav M03 automato bazinio ir pagerinto automatų stebimumo skirtumas vienetais.....	46
4.3.2. Automato M05 tyrimas.....	47
4.3.2.1 pav M05 automato Stebimumo priklausomybė nuo iteracijų skaičiaus.....	47
4.3.2.2 pav M05 automato tikslumo priklausomybė nuo iteracijų skaičiaus.....	48
4.3.1.3 pav M05 automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė.....	49
4.3.2.4 pav M05 pagerinto automato stebimumo priklausomybė nuo iteracijų skaičiaus.....	50
4.3.2.5 pav M05 pagerinto automato tikslumo priklausomybė nuo iteracijų skaičiaus.....	51
4.3.2.6 pav M05 pagerinto automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė.....	52
4.3.2.7 pav M05 automato bazinio ir pagerinto modelio stebimumo palyginimas.....	53

4.3.2.8 pav M05 automato bazinio ir pagerinto automatų stebimumo skirtumas vienetais.....	54
4.3.2.9 pav Bazinio ir pagerinto automatų stebimumo skirtumas vienetais.....	55
4.3.3.1 pav c499 automato Stebimumo priklausomybė nuo iteracijų skaičiaus.....	57
4.3.3.2 pav c499 automato tikslumo priklausomybė nuo iteracijų skaičiaus.....	58
4.3.3.3 pav c499 automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė .....	59

### **LENTELIŲ SĄRAŠAS**

2.2.2.1 lentelė Y taško stebimumas .....	15
2.2.2.1 lentelė X taško stebimumas .....	15
3.5.2.1 lentelė Veiklos įvykių sąrašas .....	29





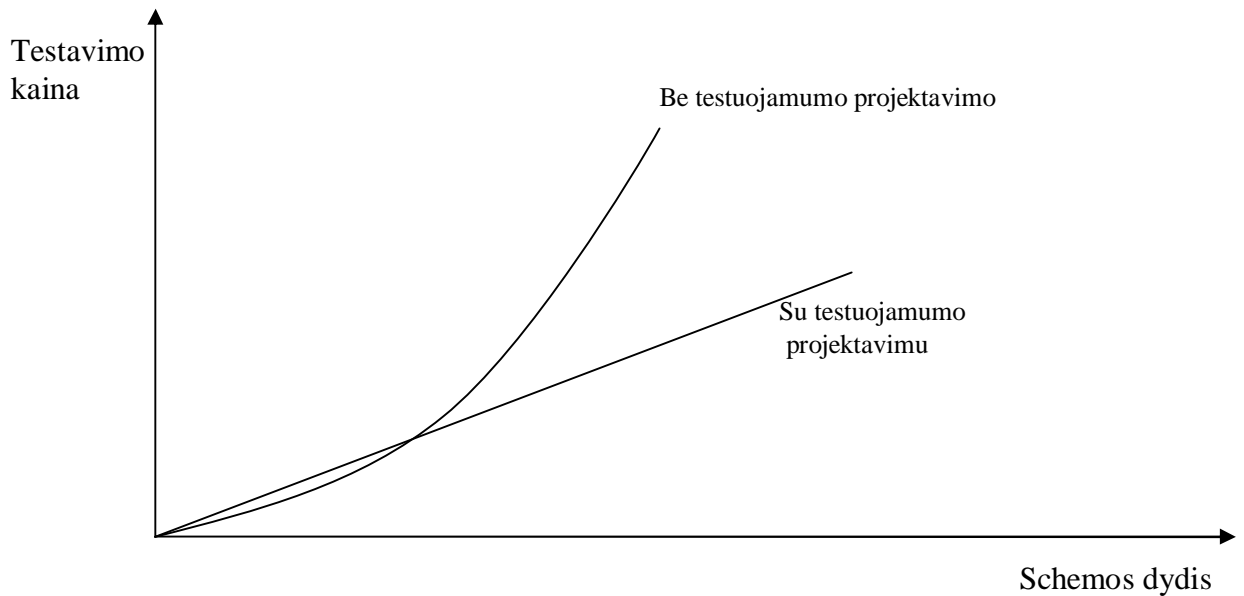
## 1. ĮVADAS

### 1.1 Temos aktualumas

Šiuolaikinių elektroninių schemų testavimas tampa brangus ir sudėtingas. Kuriamų schemų sudėtingumas didėja eksponentiškai. Dėl to elektroninių schemų testavimas, galimų klaidų aptikimas bei jų ištaisymas reikalauja vis didesnių ir didesnių sąnaudų. Vienas iš galimų variantų, kaip sumažinti sąnaudas ir pagreitinti šį procesą, yra testuoti ne pati įrenginį ar elektroninę schemą, o jos aprašytą modelį ir didinti šios schemos *testuojamumo* lygį t.y. padidinti schemos *stebimumą* (*observability*) ir *valdomumą* (*controllability*). Be to, testuojant aprašytą schemos modelį, nėra būtinas galutinės schemos egzistavimas arba testinio modelio sukūrimas. Pakanka turėti abstraktaus automato kuria nors programavimo kalba aprašytą modelį. Turint šį abstraktų modelį galima schema testuoti taip pat efektyviai, kaip ir turint fizinį modelio analogą. Privalumas yra tas, kad testavimas bus žymiai greitesnis ir pigesnis. Be to testavimas gali vykti tuo pat metu, kaip ir schemos fizinis kūrimas, kas pagreitina jos atsiradimą.

### 1.2 Projekto kūrimo pagrindas

Efektyvios mikroelektronikos technologijos tyrimo priemonės bei projektavimo automatizavimo įrankiai įgalina inžinierius projektuoti vis didesnes, sudėtingesnes integruotas schemas. Sistemos lustuose yra nauja kryptis puslaidininkių pramonėje. Naujų produktų projektavimas ir testavimas yra labai brangus. Kuo yra sudėtingesnė schema, tuo dramatiškiau išauga schemos testavimo bei klaidos aptikimo kaina [1]. Todėl sudėtingų elektroninių sistemų testavimas ir verifikavimas yra labai svarbus. Projektavimas ir testavimas yra vientisas procesas. Iš **1.1.2pav.** matyti, kiek lėšų galima sutaupyti projektuojant testuojamumą, ir kiek išauga testavimo bei klaidos aptikimo išlaidos didėjant schemos sudėtingumui ir blogai projektuojant schemos testuojamumą arba apskritai vengiant šio proceso [2].



### 1.1.2 pav Testavimo išlaidų priklausomybė

Europos Komisijos surartimi IST - 2000 – 30193 kuriamas projektas REASON, “ Research and Training Action for System on Chip Design” skirtas sistemų luste projektavimo ir testavimo specialistų rengimo tobulinimui Europoje. Projekto eigoje tyrinėjamos įvairios mokymo technologijos ir įrankiai. Propaguojamas nuotolinis mokymas ir žiniatinklio galimybių taikymas projektuojant ir tiriant sistemas.

Reikalingos lengvai naudojamos, vaizdžios ir paprastos e. mokymo priemonės, skirtos sistemoms lustuose tirti ir mokyti. Profesionalumui ugdyti būtina, kad studentas pats spręstų problemas, analizuotų ir projektuotų. Siekiama skatinti studentų kūrybiškumą, lavinti meistriškumą. Taip pat projektas yra skirtas arčiau supažindinti studentus su testavimo specifika bei metodikom, vaizdžiai parodyti ką galima pasiekti naudojant tokio tipo įrankius. Priemonės bus naudojamos ir universitetiniame mokyme, ir tęstiniame kvalifikacijos kėlime.

Suprojektuoto e.mokymo įrankio kūrimas yra šio darbo tikslas.

### 1.3 Darbo tikslas

Sistemos paskirtis edukacinė. Studentai individualiai mokysis įvertinti automato stebimumo koeficientą, padėti tyrinėtojai padaryti specifines išvadas apie abstraktaus automato sudėtingumo lygį, jo testuojamumą. Dėstytojai naudosis sukurta įranga kaip vaizdine priemone paskaitose. Sistema bus skirta nustatyti abstraktaus automato stebimumo rodiklius bei padidinti to automato stebimumą. Sistemos paskirtis yra sudominti studentus testavimu ir vaizdžiai parodyti testavimo aspektus, skirtingų metodologijų darbo rezultatus, padėti aiškiau suprasti testavimo projektavimo vieta realiuose projektuose, viską tai pateikti aiškiai, paprastai ir vaizdžiai.

Todėl darbo tikslas yra sugalvoti metodika abstrakčių automatų stebimumui nustatyti, padidinti abstrakčių automatų stebimumo koeficientą bei pateikti pagerinimo pasiūlymus ir sukurti programinę įrangą, kuri leistų apskaičiuoti abstraktaus automato stebimumo lygį bei padidinti jį.

Tiriamąjį darbo tikslą yra iširti ir įsitikinti, kaip sukurta programinė įrangą nustato iteracijų skaičių, kai yra pasiekiamas maksimalus automato stebimumas, o taip pat iširti ar iš tiesų sukurta programinė įrangą padaro teisingus sprendimus dėl abstrakčių automatų stebimumo lygio padidinimo ir kiek ji padidina automatų stebimumo lygį.

### **1.4 Darbo struktūra**

Pirmame darbo skyriuje, įvade, aptariami teorinės darbo prielaidos bei metodologija, apibrėžiami tyrimo objektas, aptariamas temos aktualumas. Taip pat apibrėžiami tyrimo tikslai bei uždaviniai, minima apie straipsnius ir konferencijas, kur buvo pristatoma šio darbo problematika.

Antrajame skyriuje, analitinėje dalyje, aprašomi metodai ir algoritmai, kurie yra naudojami tiriamojoje dalyje. Yra formuluojami reikalavimai kuriams modeliams, algoritmams, programinei įrangai ir atliekamiems eksperimentams. Pateikiamas metodų, modelių ir programinės įrangos pasirinkimo pagrindimas.

Trečiame skyriuje, projektinėje dalyje, yra pristatoma sukurta programinė įrangą. Trumpai aprašomos sistemos tiekiamos funkcijos. Išnagrinėtos stebimumo nustatymo bei pagerinimo realizacijos ypatumai. Pateikiami sukurtos programinės įrangos techninės-projektinės dokumentacijos esminiai aspektai.

Ketvirtas skyrius, tiriamoji dalis, yra skirtas išnagrinėti sukurtos sistemos patobulinimo galimybes, pagrįsti bei realizuoti išnagrinėtus patobulinimus.

Penktas skyrius, eksperimentinė dalis, yra skirtas aprašyti ir išanalizuoti atliktus tyrimus, panaudojus sukurta programinę įrangą.

Šeštame skyriuje, išvadoje, glaudžiai pateikiama nuomonė apie svarbiausius tezęse paliestus klausimus: projektavimo metu priimtus sprendimus, atliktų tyrimų esmę ir gautus rezultatus.

Toliau pateikiamas naudotos literatūros sąrašas, terminų bei santrumpų žodynas ir priedai, kuriuose pateikiama tyrimo medžiaga bei papildoma medžiaga, reikalinga tezių tvirtinimams pagrįsti.

### **1.5 Santrauka**

Elektronika ir mikroelektronika įgauna vis didesnę ir svarbesnę vaidmenį mūsų kasdienybėje. Technologijos tampa vis brangesnės ir sudėtingesnės. Augant mikroschemų sudėtingumui auga ir išlaidos, susijusios su schemų projektavimu, testavimu bei gamyba. Klaidos, nepastebėtos arba praleistos projektavimo metu vėliau kainuoja nepalyginamai daugiau vėlesnėse projektų fazėse. Todėl yra ypač svarbu kreipti didelį dėmesį į testavimą ir klaidų atradimą. Pastaruoju metu šiai problemai

yra skiriama vis daugiau dėmesio, šioje vietoje kyla naujos mokslo ir tyrimo šakos. Kalbant apie testavimą vis dažniau galima išgirsti sąvoka „*testuojamumo projektavimas*“. Testuojamumo projektavimas leidžia labai sumažinti klaidų skaičių būsimoj įrangoj dar produkto projektavimo metu.

Testuojamumo projektavimas apibrėžiamas tokiomis sąvokomis kaip schemos *stebimumas* (*observability*), schemos *valdomumas* (*controllability*) ir schemos *nuspėjamumas* (*predictability*). Magistro studijų metu buvo sukurta programinė įranga, kuri tiria schemų programinių modelių stebimumą ir valdomumą, pateikia pasiūlymą vartotojui, kaip schema gali būti patobulinta didinant jos stebimumą ir valdomumą bei testuoja programos modelį atsižvelgdama į pasiūlytus stebimumo/valdomumo pagerinimo faktorius.

Ši programinė įranga gali būti naudinga sistemoms lustuose tirti ir mokyti. Profesionalumui ugdyti būtina, kad studentas pats spręstų problemas, analizuotų ir projektuotų. Siekiama skatinti studentų kūrybiškumą, lavinti meistriškumą. Priemonės bus naudojamos ir universitetiniame mokyme, ir tęstiniame kvalifikacijos kėlime.

Ketvirtą magistro studijų semestrą buvo atliekamas sukurto programinės įrangos tyrimas. Buvo keliamos prielaidos dėl įrangos vykdymo charakteristikų, atliekami praktiniai eksperimentai ir įsitikinama tų prielaidų teisingumu arba klaidingumu.

Atliekant šį darbą buvo skaitomi pranešimai, susiję su jame nagrinėjamais klausimais. Buvo dalyvauta šiose konferencijose:

[1] Septintoji jaunųjų mokslininkų konferencija, VGTU, Vilnius, 2004 03 18. Pranešimo tema: „Abstrakčių automatų valdomumo ir stebimumo lygis“.

[2] 9-ji magistrantų ir doktorantų konferencija, IVUS 9, VDU, Kaunas, 2004 04 16. Pranešimo tema: „Abstrakčių automatų valdomumo ir stebimumo lygio nustatymas“.

Taip pat, šios srities darbai buvo publikuojami [16].

## 2. ANALITINĖ DALIS

### 2.1 Schemos testuojamumas

#### Apibrėžimas 1

Defektas yra *testuojamas* tada, kai egzistuoja tiksliai specifikuota jo atskleidimo procedūra, kuri įvykdoma su priimtinais išlaidomis panaudojus šiuolaikines technologijas.

#### Apibrėžimas 2

*Testuojamumo projektavimas (design for testing-DFT)* yra projektavimo metodų rinkinys, kuris uždeda apribojimus projektavimo procesui, tam kad būtų pagerintas testuojamumas ir diagnostika.

Testuojamumo sudedamosios dalys:

- kontroliškumas (*controlability*)
- stebimumas (*observability*)
- nuspėjamumas (*predictability*) – galimybė aptikti žinomas išėjimo reikšmes paduodant žinomas įėjimo reikšmes.

Taigi, testuojamumą galima būtų apibrėžti kaip trijų sudedamųjų, kontroliškumo, stebimumo ir nuspėjamumas, sumą.

### 2.2 Loginės schemos stebimumas

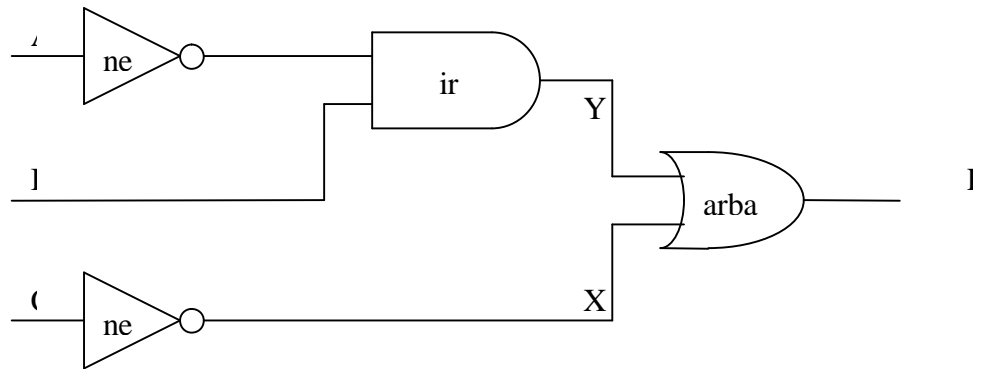
#### 2.2.1 Mazgo stebimumo apibrėžimas

Loginės schemos mazgo (taško) stebimumas - tiksliai apibūdina kaip, sunku ar lengva perduoti  $n$  taško reikšmę į tiesiogiai stebimą išėjimą[10].

Analizuojant schemą, savoka *schemos stebimumas* yra neatskiriama nuo tokių sąvokų kaip *schemos kontroliškumas*, *schemos nuspėjamumas* ir *schemos testuojamumas*.

#### 2.2.2 Stebimumo sąvokos analizė

Tam, kad tiksliau apibrėžti, kas yra schemos stebimumas, panagrinėkime pavyzdį. Turime loginę schemą, pateikta **1.pav.** Šioje schemoje nesudėtinga nustatyti, kad 6 iš 8 galimų įėjimų kombinacijų (2.2.2.1 lentelė) leidžia mums stebėti arba spręsti taško Y reikšmę stebint išėjimą F. Iš kitos pusės, taško X reikšmę stebint išėjimą F galima nusakyti su puse galimų įėjimų kombinacijų (2.2.2.2 lentelė), dėl to daroma išvada, kad taškas Y yra geriau stebimas negu taškas X.



2.2.2.1 pav Tiriama schema

2.2.2.1 lentelė Y taško stebimumas

A	B	C	Y Stebimumas
X	X	0	-
X	X	1	P

2.2.2.1 lentelė X taško stebimumas

A	B	C	X Stebimumas
0	0	X	P
0	1	X	-
1	0	X	P
1	1	X	P

Šio pavyzdžio dėka mes matome, kad testuojant logines schemas labai svarbus rodiklis yra schemos stebimumo lygis. Bet iš kitos pusės, nėra jokios naudos iš schemos aukšto lygio stebimumo, jeigu negalima keičiant įėjimų lygio keisti stebimo mazgo loginio lygio.

### 2.3. Stebimumo apskaičiavimo metodai

Yra naudojama daug įvairių stebimumo apskaičiavimo metodologijų ir matavimo vienetų. Visi išnagrinėti metodai (HLTS, TMEAS, SCOAP, CAMELOT) yra skirti „baltoms“ dėžėms. Tačiau jie visi (išskyrus SCOAP) skaičiavimams naudoja SPF- Stebimumo Perdavimo Faktorių (Observability transfer factor). VPF ir SPF yra faktoriai, kurie nurodo nedalomo elemento valdomumą ir stebimumą su tam tikra tikimybe. Nedalomą elementą traktuojame kaip „juodą“ dėžę ir tada VPF ir SPF faktoriai gali būti naudojami „juodos“ dėžės valdomumo ir stebimumo įvertinimui. Kadangi išnagrinėti metodai skirti tik baltų dėžių tyrimams, todėl šioje apžvalgoje pateikiamos metodų dalys kurias galima pritaikyti ir „juodos“ dėžės tyrimams.

#### 2.3.1 TMEAS Stivensono ir Greisono metodas (Stephenson & Grason's Approach—TMEAS)

Ü Sukurtas registru-perdavimo-lygio (*register-transfer-level*) schemoms, bet gali būti pritaikytas ir įėjimams (*gate level*).

Ü Matai yra normalizuoti tarp 0 ir 1, tam kad atspindėti vidinio mazgo stebimumo paprastumą.

1. kiekvienai signalo linijai  $s$ , mes nustatome  $s$  linijos stebimumą  $OY(s)$ .
2.  $OYs$  visų linijų reikšmes yra gaunami sprendžiant lygiagrečių lygybių sistema imant kad  $OYs$ .

Ø Tegul  $N_j(0)$  ir  $N_j(1)$  bus įėjimų kombinacijų skaičius kurių  $z_j$  reikšmė būtų atitinkamai 0 ir 1.

Ø Apskaičiuosim kiekvieno įėjimo  $x_j$  reikšmę  $OY$  pagal formulę, kur  $OTF$  yra stebimumo perdavimo koeficientas.

$$OY(x_i) = OTF \times \frac{1}{m} \sum_{j=1}^m OY(z_j)$$

**2.3.1.1 formulė**  $OY$  reikšmės apskaičiavimas kiekvienam įėjimui.

Ø Tegul  $NS_i$  yra įėjimų kombinacijų skaičius, kai  $x_i$  pakeitimas keičia išėjimą.

Tuomet

$$OTF \equiv \frac{1}{n} \sum_{i=1}^n \frac{NS_i}{2^n}$$

**2.3.1.2 formulė** Stebimumo koeficiento apskaičiavimas



- §  $NS_i$  taip pat reiškia įėjimo kombinacijų skaičių kurie veikia kelią  $x_i$ -išėjimas.
- §  $OTF$  matas atspindi tikimybę, kad klaidinga reikšmė įėjime persiduos į išėjimą
- §  $0 \leq OTF \leq 1$
- § kiekvieno įėjimo stebimumas yra pastovus

3. **Išsišakojimai(fanouts).** Tegu  $s$  yra šakuota ir  $k$  yra jos šakų skaičius. Tada kiekvienos šakos stebimumas  $OY$  bus, kur  $b_i$  yra is  $s$  išsišakojusios šakos.

$$OY(s) = 1 - \prod_{i=1}^k [1 - OY(b_i)],$$

**2.3.1.3 formulė** Kiekvienos šakos stebimumas

4. **Nuoseklūs komponentai(sequential components).** Nuoseklūs komponentai yra formuojami pridendant atbulinius ryšius komponentams, kurie atspindi vidinę būseną.

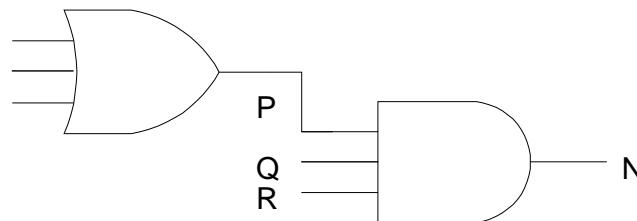
**2.3.2 SCOAP Goldsteino metodas(Goldstein's approach - SCOAP)**

ü SCOAP - Sandia Controllability Observability Analysis Program.

ü Vienas iš plačiausiai naudojamų metodų.

ü Išmatavimai atspindi vidinių mazgų kontroliškumo ir stebimumo *sudėtingumą*; didesnis skaičius parodo, kad mazgas yra *sudėtingiau* valdomas arba stebimas.

ü Iš esmės, šis matavimas reikalauja *minimalias* išlaidas kontroliškumui ir stebimumui.



**2.3.2.1 pav Tiriama schema 2**

1.  $Y = \text{AND}(A, B, C)$  kombinaciniam 1- ir 0-stebimumui:

$$\emptyset CO^1(Y) = CO^1(A) + CO^1(B) + CO^1(C) + 1$$

$$\emptyset CO^0(Y) = \min\{CO^0(A), CO^0(B), CO^0(C) + 1\}$$

2. Rezultatas yra didinamas 1, todėl šis skaičius atspindi (dalinai) nuotolį iki įėjimo.
3. Einant nuo įėjimo link išėjimo, mes apskaičiuojame  $CO$  išėjimo linijos kiekvieno elemento kaip įėjimo linijos  $CO$  funkciją.
4. Nuoseklus stebimumas duoda mums įverti kiek laiko taktų reikės tam, kad pakeisti 0 į 1 tam tikram mazge.

Jei  $Y=XOR(A,B)$ :

$$\emptyset CO^0(Y) = \min\{CO^0(A)+CO^0(B), CO^1(A)+CO^1(B)\}+1$$

$$\emptyset CO^1(Y) = \min\{CO^0(A)+CO^1(B), CO^1(A)+CO^0(B)\}+1$$

$$\emptyset SO^0(Y) = \min\{SO^0(A)+SO^0(B), SO^1(A)+SO^1(B)\}+1$$

$$\emptyset SO^1(Y) = \min\{SO^0(A)+SO^1(B), SO^1(A)+SO^0(B)\}+1$$

5. Kai yra skaičiuojami nuoseklūs stebimumai kombinaciniuose schemose, reikšmės nėra didinamos-nereikia papildomų laiko taktų.
6. Kai yra išvedamos lygtis nuoseklioms schemoms, SO reikšmės yra didinamos 1, o CO reikšmės nėra didinamos.

Teigiamo sužadinto trigerio (Positive edge-triggered) su aktyvių žemo lygio apnulinimų (reset):

$$\emptyset CO^0(Q) = \min\{CO^0(R), CO^1(R)+CO^0(D)+CO^0(C)+CO^1(C)\}$$

$$\emptyset CO^1(Q) = CO^1(R)+CO^1(D)+CO^0(C)+CO^1(C)$$

$$\emptyset CC^1(Q) = CC^1(R)+CC^1(D)+CC^0(C)+CO^1(C)$$

$$\emptyset SO^0(Q) = \min\{SO^0(R), SO^1(R)+SO^0(D)+SO^0(C)+SO^1(C)\}+1$$

$$\emptyset SO^1(Q) = \min\{SO^1(R)+SO^1(D)+SO^0(C)+SO^1(C)+1$$

$$\emptyset SC^1(Q) = \min\{SC^1(R)+SC^1(D)+SC^0(C)+SC^1(C)+1$$

7. Grįžtant prie tiriamos schemos 3.pav turime:

$$\emptyset CO(P) = CO(N)+CC^1(Q)+CC^1(R)+1$$

$$\emptyset SO(P) = SO(N)+SC^1(Q)+SC^1(R)$$

Teigiamo sužadinto trigerio (Positive edge-triggered) su aktyvių žemo lygio apnulinimų (reset):

$$\emptyset CO(R) = CO(Q)+CC^1(Q)+CC^0(R)$$

$$\emptyset SO(R) = SO(R)+SC^1(Q)+SC^0(R)+1$$

- ✓ Tam kad stebėti  $R \Rightarrow$  stebim Q ir paverčiam FF į 1, ir po to apnulinam jį į 0,.

**2.3.3 CAMELOT – Bennets‘o, Mauder‘io ir Robinson‘o metodas (Computer-Aided MEasure for LOGic Testability - Kompiuterio pagalba įvertinimas loginiam testavimui)**

Û Stebimumo reikšmė ( $OY$ ) yra intervale  $[0,1]$ , kas atspindi mazgo stebėjimo paprastumą.

Û  $OY(\text{išėjime}) = OTF \times OY(\text{įėjime}) \times g(CYs(\text{palaikomi įėjimai}))$ , kur  $OTF$  yra stebimumo perdavimo koeficientas komponentei nuo išėjimo.  $OTF$  iš įėjimo  $I$  į išėjimą  $O$  pazymimas kaip  $OTF(I-O)$ , ir atvaizduoja kiek lengva perduoti klaidos poveikį iš  $I$  į  $O$ .

$$OTF(I-O) = \frac{N(SP:I-O)}{N(SP:I-O) + N(IP:I-O)},$$

**2.3.3.1 formulė** Stebimumo perdavimo koeficientas

kur  $N(SP:I-O)$  yra bendras skaičius jautrių kelių iš  $I$  į  $O$ , ir  $N(IP:I-O)$  yra bendras skaičius nejautrių kelių. Perdavimas laikomas jautrių tada kai vienas įėjimas turi atitinkama išėjimą.

**2.4 Testuojamumo matų (kontroliškumo ir stebimumo matų) svarbumas [12] [15]**

∅ Testavimo matai gali padėti projektuotojui pagerinti projektuojamų schemų testuojamumą.

- Po testavimo visi vidiniai mazgai yra išanalizuoti, todėl galima daug efektyviau pasirinkti ir įgyvendinti *testuojamumo projektavimą* (įtraukiant Ad Hoc metodus, DFT technologijas, dalinį ir pilną testavimą, ir BIST).

∅ Testų generavimo euristiciniai algoritmai dažniausiai naudoja tam tikrus testuojamumo matus jų euristicinėms operacijoms (pavyzdžiui priimant paieškos sprendimus), kas žymiai padidina testų generavimo procesą

## 2.5 SPF apskaičiavimas

SPF- *stebimumo perdavimo faktorius*, yra skaičiuojamas schemos įėjimams. Jis nusako kaip lengvai tam tikro įėjimo pakeitimas stebimas išėjimuose. SPF skaičiuojamas pagal formulę:

$$SPF(I-O) = \frac{N(PDC : I - O)}{N(PDC : I - O) + N(NPDC : I - O)}$$

### 2.5.1 formule Stebimumo perdavimo faktorius

kur  $N(PDC : I - O)$  yra bendras skaičius skirtingų jautrių ryšių tarp vieno įėjimo ir visų išėjimų, o  $N(NPDC:I-O)$ - nejautrių ryšių skaičius.

Pastabos:

$$0 \leq SPF \leq 1$$

$$SPF = 1, \text{ kai } N(NPDC:I-O) = 0$$

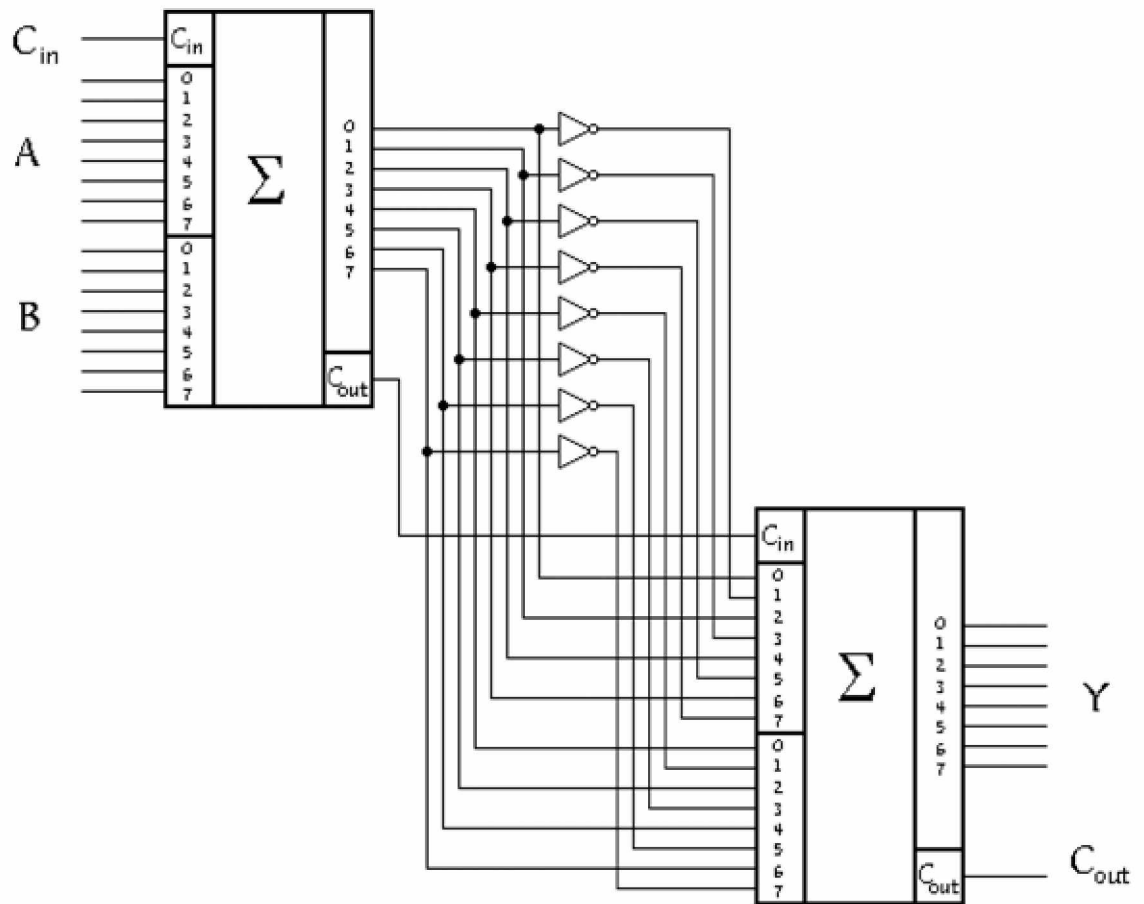
$$SPF = 0, \text{ kai } N(PDC : I - O) = 0$$

Kiekvienas įėjimas turi savo SPF reikšmę (turi atitinkamą sąrašą išėjimų, kuriuose šis įėjimas yra stebimas).

## 2.6 Papildomų jungčių pridėjimas [13]

Testavimo lengvumas yra labai svarbus kriterijus. Sudėtingas testuojamumas padidina gaminimo kaštus ir padidina gaminimo laiką, iškeldamas dideles problemas funkcionalumo testavimo metu. Dar vienas iš testavimo pagerinimo metodų yra jungčių pridėjimas prie tam tikrų schemos vidinių taškų, kad būtų galima juos stebėti.

Pavyzdys:

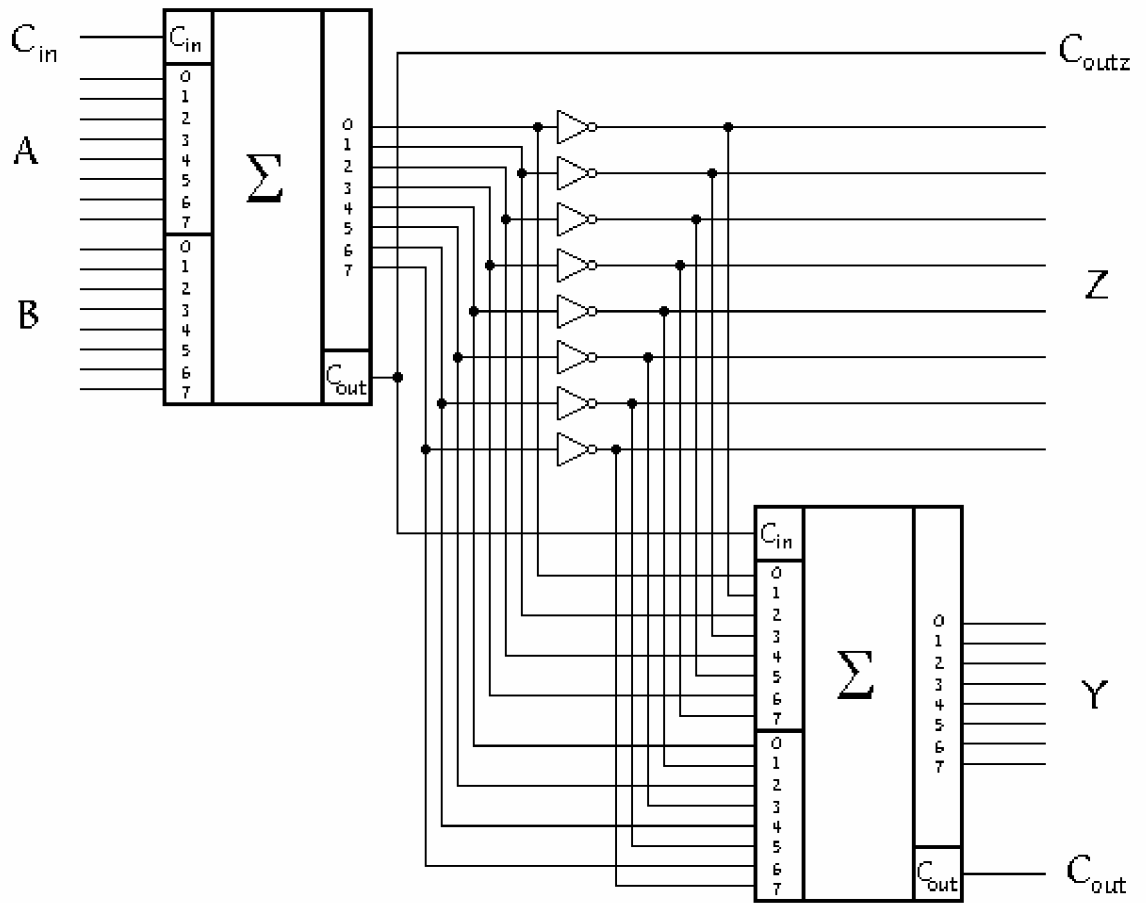


2.6.1 pav Originali schema

Šiame pavyzdyje nagrinėjama schema yra du nuosekliai sujungti sumatoriai. Pirmo sumatoriaus išėjimai yra sujungti su antro sumatoriaus įėjimais. Šios schemas stebimumas yra sudėtingas. Tam, kad padidinti schemas stebimumą, o kartu ir schemas testuojamumą, reikia įnešti tam tikrus pakeitimus.

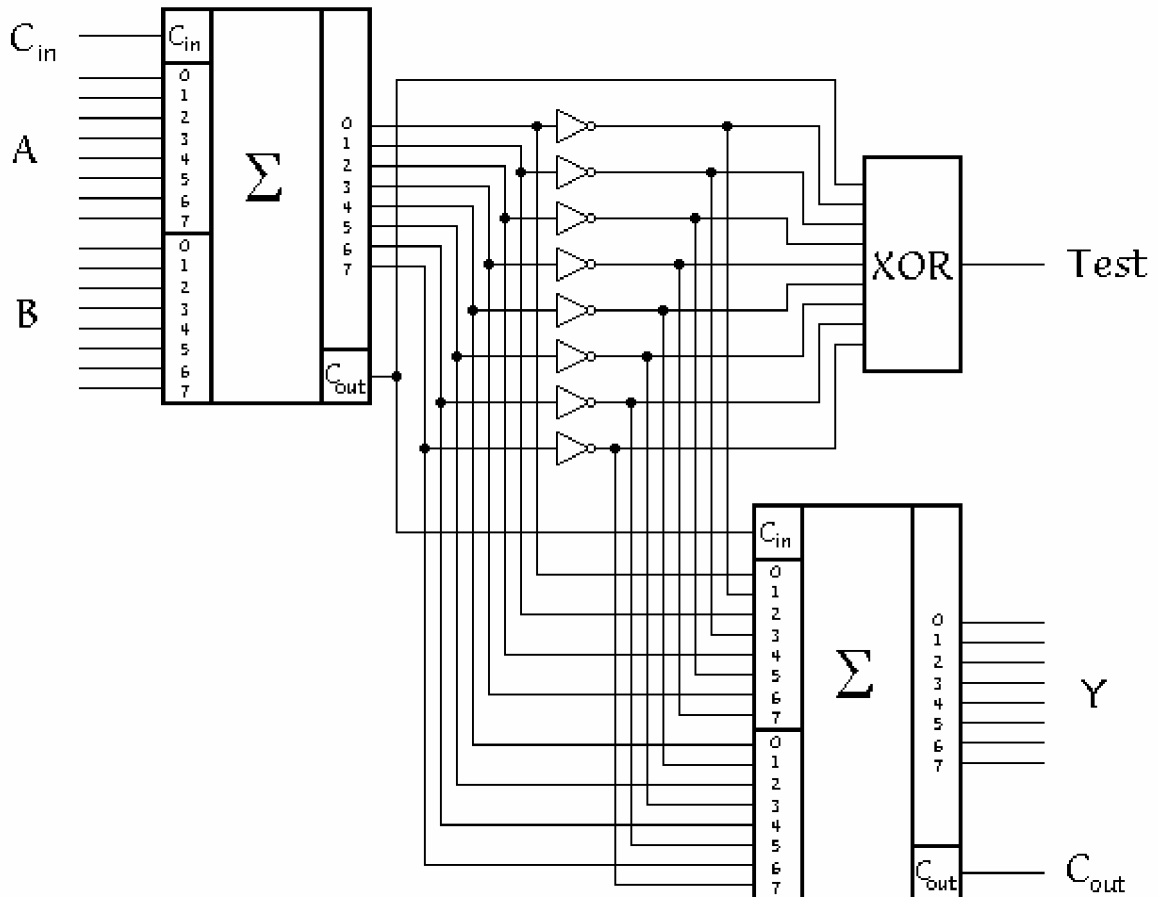
Egzistuoja du būdai stebimumo projektavimo.

Pirmų atveju, devynios sumatorių sujungimo linijos yra išvedamos į išorę į derinius išėjimus:  $C_{outz}$  ir Z ( $z_0 - z_7$ ).



2.6.2 pav Stebimumo didinimas. Pirmas būdas

Tačiau, yra brangu turėti daug išėjimų, kurie būtų neišnaudojami įprasto darbo režimo metu. Todėl modifikuosim schema pavaizduota **2.6.3 pav** įvesdami papildomą XOR elementą, kuris jungtų šiuos išėjimus ir išvestų juos per vieną išėjimą pavadintą *Test*.



2.6.3 pav Stebimumo didinimas. Antras būdas

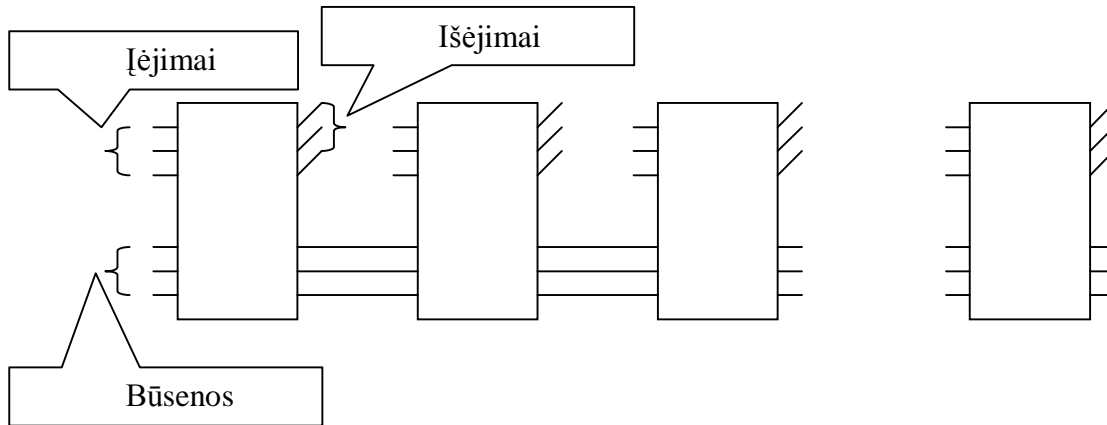
## 2.7 Sprendimas

### 2.7.1 Iteracinio modelio sąvoka

Kaip matome iš išnagrinėtų metodų stebimumas yra skaičiuojamas baltoms dėžėms. Kadangi schemas imituojamos C++ kalba ir schemas struktūra yra nežinoma, todėl schemas bus tiriamos kaip „juodos“ dėžės. Šiam tikslui buvo nuspręsta naudoti metodai, kurie yra naudojami nustatyti baltų dėžių nedalomų elementų stebimui. Kadangi imituojamos schemas paprastai turi daugiau įėjimų negu „baltų“ dėžių nedalomi elementai, todėl buvo nuspręsta nenaudoti pilno įėjimų perrinkimo nes tai gali užimti daug laiko ir resursų. Dėl šios priežasties tyrimai bus atliekami dalinio perrinkimo būdu. Taip bus prarandamas rezultatų tikslumas, tačiau tai neturėtų būti pastebimas nukrypimas nuo realių reikšmių.

Kad gauti korektiškus rezultatus C++ kalba realizuotos schema turi būti nekombinacinės. Kad atlikti tyrimus su kombinacinėmis schemomis turi būti papildomi priėjimai prie modelio būsenų. Tai leistų naudojant modelį be atminties imtuoti nekombinacinės schemas darbą.

Imituojant modelio darbą naudojamas iteracinis modelis:



**2.7.1 pav Iteracinis modelis**

Šiame modelyje yra naudojamos schemas modelio identiškos kopijos. Iš kopijos gautos būsenų reikšmės yra perduodamos sekančiai kopijai ir taip toliau. Tokiu būdu yra imituojama schema su atmintimi. Šio modelio privalumas, kad be įėjimų stebimumo taip pat galima nustatyti ir būsenų stebimumą.

### **2.7.2 Stebimumo apskaičiavimas**

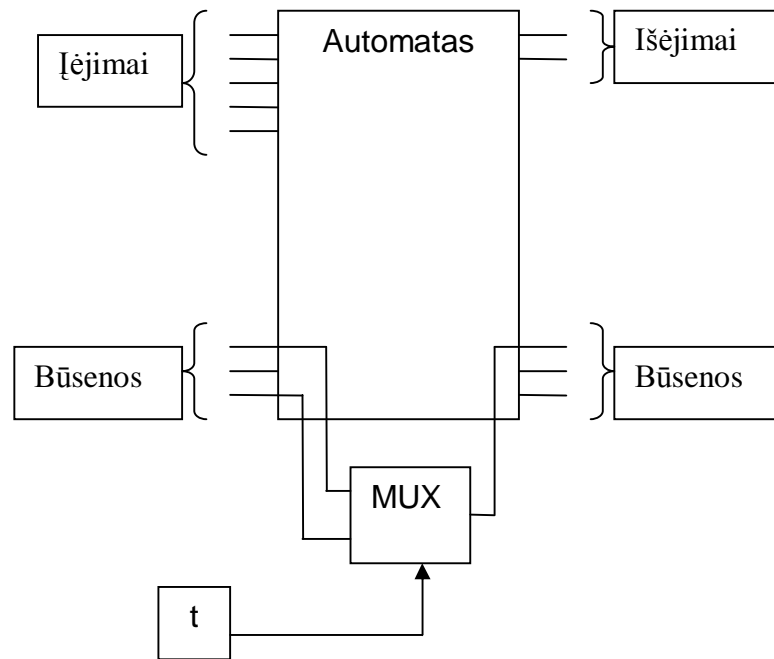
- 1) Į pirmą schemas kopiją yra paduodamos nulinės būsenos ir atsitiktinai sugeneruoti įėjimai.
- 2) Gautos pirmos kopijos būsenos perduodamos sekančioms kopijoms.
- 3) Invertuojamas vienas įėjimas ir su tomis pačiomis būsenomis paduodamas kopijai.
- 4) Lyginami gauti išėjimai (pradinio ir invertuoto varianto) ir išsaugomi skirtumai.
- 5) Kartojam 3) – 4) žingsnius kiekvienam įėjimui.
- 6) Kartojam 1) – 5) žingsnius N kartų.
- 7) Skaičiuojame kiek kiekvienas įėjimas paveikė išėjimų.
- 8) Skaičiuojame kiek kiekvienas įėjimas paveikė būsenų;
- 8) Kiekvienam įėjimui skaičiuojame SPF; skaičiuojamas stebimumas išėjimuose ir būsenose;
- 9) Apskaičiuojamas bendras automato stebimumas, kuris lygus visų įėjimų vidurkiui;

### **2.7.3 Stebimumo pagerinimas**

Atlikus automato stebimumo perdavimo faktorių apskaičiavimą turime visų įėjimo reikšmių stebimumo perdavimo faktorius. Tam, kad pagerinti tiriamo automato stebimumą, yra įvedamas



testinis schemos darbo režimas. Testiniu darbo režimu blogiausiai ir geriausiai stebimos būsenos yra sujungiamos per miltupleksorių. Miltupleksorius yra naudojamas tam, kad būseną būtų perduodama ne pastoviai, o priklausytų nuo atsitiktinio poveikio miltupleksoriaus įėjime,  $t$ . Kai  $t$  yra įjungtas, geriausiai stebimos būsenos reikšmė yra perduodama tiesiogiai į blogiausiai stebimą būseną; kai  $t$  yra išjungtas, įėjimo poveikis veikia ta grandinę, kuri yra numatyta pagal schemos veikimo specifikaciją.



### 2.7.3.1 pav Automato stebimumo pagerinimo schema

Tokiu atveju blogiausiai stebimo įėjimo pokytis turi tiesioginį poveikį būsenai, t.y. įėjimo pokytis visada yra matomas būsenoj, kas padidina schemos stebimumo lygi. Nustatytos būsenos yra pateikiamos vartotojui kaip stebimumo gerinimo pasiūlymas. Ištestavus ir nustačius stebimumo perdavimo koeficientus pakartotinai galima nustatyti, kokia nauda atneš naudojamo automato pokytis.

### 3. PROJEKTINĖ DALIS

#### 3.1 Sistemos paskirtis

Sistemos paskirtis edukacinė. Studentai individualiai mokysis įvertinti automato stebimumo koeficientą, padėti tyrinėtojiui padaryti specifines išvadas apie abstraktaus automato sudėtingumo lygį, jo testuojamumą. Dėstytojai naudosis sukurta įranga kaip vaizdine priemone paskaitose. Sistema bus skirta nustatyti abstraktaus automato stebimumo rodiklius, pagerinti automato stebimumo koeficientą bei pateikti vartotojui pasiūlymus dėl automato stebimumo pagerinimo. Sistemos paskirtis yra sudominti studentus testavimo tema ir vaizdžiai parodyti testavimo aspektus, skirtingų metodologijų darbo rezultatus, padėti aiškiau suprasti testavimo projektavimo vieta realiuose projektuose, bei viską tai pateikti aiškiai, paprastai ir vaizdžiai.

Sistemos paskirtis yra nustatyti schemas stebimumą bei pateikti pasiūlymus kaip būtų galima jį pagerinti. Stebimumo reikšmės yra pateikiamas kiekvienam schemas išėjimui bei vidinei būsenai. Taip pat yra pateikiamas kiekvieno išėjimo ir būsenos stebimumas kiekvienai iteracinio modelio kopijai t. y. galima nustatyti po kelių iteracijų galima gauti atitinkamą schemas taško valdomumą. Kaip bendras automato valdomumas yra pateikiamas ir visos schemas valdomumas, kuris yra apskaičiuojamas remiantis visų išėjimų ir būsenų valdomumo reikšmėmis.

Iš sistemos norima, kad vartotojas, turėdamas automato modelį, parašytą C/C++ kalba, iš bet kurios pasaulio vietos neturėdamas jokių papildomu darbo instrumentų, panaudojus tik internetinę naršyklę, galėtų patalpinti turimą automata į serverį, ištestuoti jį ir gauti jo skaitines (pagrindinė – stebimumas) ir grafines charakteristikas, bei gauti pasiūlymų, kaip modifikuoti schema, kad jos valdomumas padidėtų. O taip pat peržiūrėti ir susipažinti su jau esamais abstrakčiais automatais, kurie yra saugomi abstrakčių automatų archyve.

#### 3.2 Apribojimai sprendimui

##### Sąsajos kūrimo uždavinys.

Kadangi programa turi būti vykdoma darbo stotyje, tai vartotojo interfeisui buvo pasirinkta PHP kalba, kuri formuoja HTML puslapius. Kai kuriems grafiniams vaizdavimams yra naudojami Java apletai. Kad gautą automato kodą sukompiliuoti ir atlikti tyrimus yra naudojamas papildomas C++ kodas, kuris prisijungia atsiųstą automato kodą kaip biblioteką.

##### Sąsajos daugiakalbystės uždavinys

Kad realizuoti sąsajos daugiakalbystę buvo sukurtas atskiras modulis, kuriame saugomi visi sąsajoje rodomi žodžiai bei sakiniai, o formuojant HTML puslapius yra nustatoma kurios kalbos pageidauja vartotojas ir atitinkamai pagal tai iš kalbų modulio yra išrenkami sakiniai. Šio modulio pagalba, naujų

kalbų pridėjimas neužims daug laiko ir pastangų, be to nereikalaus iš vertėjo suprasti programos veikimo principus.

#### Matricos vizualizacijos uždavinys

Matricų vizualizacijos problema išspręsta keliais būdais. Matrica vaizduojama lentelės formatu HTML puslapyje, bet kai matricos būna didelės (100x100 ar net daugiau) tada iškyla problema- kaip pamatyti iškart visą matricą. Galimi du sprendimo būdai:

Vienas iš jų tai patalpinti matricą lentelės pavidalo į PDF tipo formatą, kuris leidžia suspausti lentelę į vieną puslapį.

Kitas iš jų yra atvaizduoti matricą paveikslu, kai atitinkama matricos reikšmė atvaizduojama atitinkamos spalvos tašku. Šiom problemom išspręsti buvo sukurti atitinkami moduliai.

#### Matricos charakteristikų vizualizacijos uždavinys

Grafiniam matricos charakteristikų generavimui bus naudojami JAVA apletai (Hall, 2002, p.1034-1286). Bus galima peržiūrėti matricos charakteristikas stulpeline arba sektorine diagramomis.

#### Automatų modelių saugojimo uždavinys

Kad vartotojas galėtų keisti savo įkeltas matricas buvo numatyta galimybė, kad įkeliant matricą tokiu pat pavadinimu (to paties vartotojo), naujoji būtų užrašoma ant senosios ir taip įvyktų atsinaujinimas. Kitu būdu matricų pakeisti negalima (kad vartotojai neiškraipytų programos gautų rezultatų).

Kad failus į darbo stotį galėtų atsiųsti tik autorizuoti vartotojai buvo sukurta vartotojų registravimo sistema(Aulbach, 2003, p.27-112), kurios pagalba vartotojai gali būti automatiškai įtraukiami į sąrašus, jei pamiršo slaptažodį, tada jis yra siunčiamas į vartotojo elektroninį paštą ir vartotojai gali būti registruojami tolimesniam darbui. Neužsiregistravę vartotojai gali dirbti tik vizualizacijos režime (peržiūrėti jau sugeneruotas automatų susietumų matricas tekstiniam arba grafiniam pavidale).

### **3.3 Diegimo aplinka**

Reikalavimai sisteminei programinei įrangai:

Operacinė sistema: Unix

Serverio valdymo įranga Apache

Serverio priežiūrai palengvinti sukurti tam tikra katalogų bei failų struktūra. Matricų kataloge, kiekviena matrica yra saugoma atskiruose kataloguose, kurių pavadinimai yra sudaryti iš automato pavadinimo ir vartotojo elektroninio pašto. Administratoriui reikia tik ištrinti atitinkamą katalogą ir matrica bus pašalinta iš serverio. Norint pašalinti vartotoją iš sistemos reikia ištrinti jo elektroninio pašto pavadinimu pavadintą failą iš katalogo users.

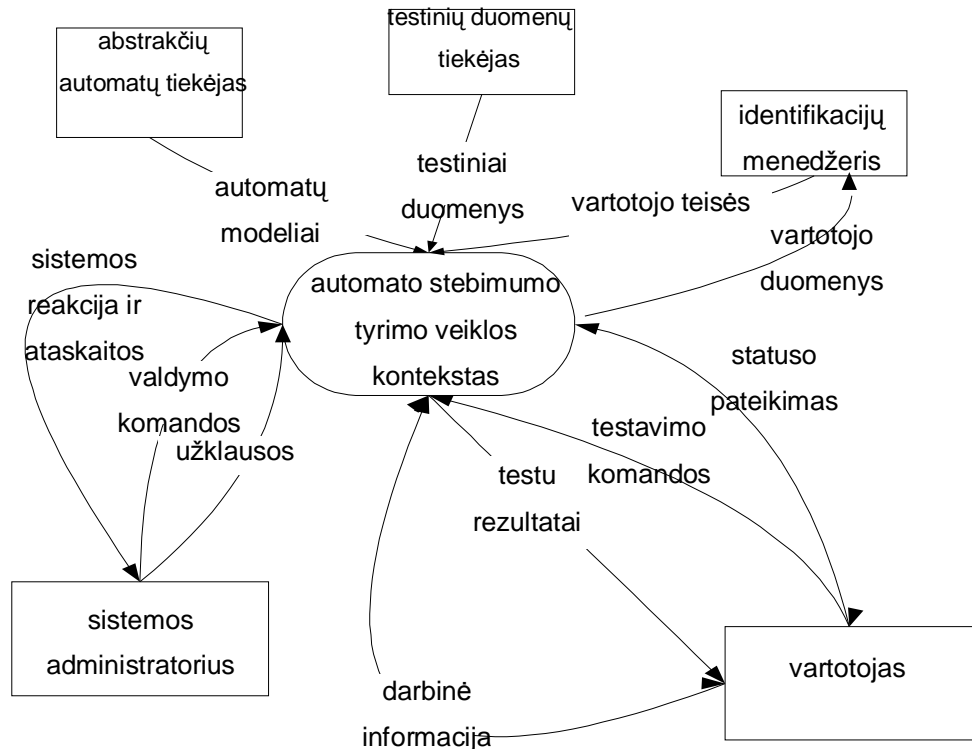
### 3.4 Bendradarbiaujančios sistemos

Kuriama sistema turės bendradarbiauti su anksčiau sukurtais projekto „REASON“ produktais. Kuriama sistema turės būti integruota su jau egzistuojančia abstrakčių automatų tyrimo elektroninę laboratorija, kuri šiuo metu funkcionuoja Internete adresu [www.soften.ktu.lt/~reason](http://www.soften.ktu.lt/~reason) .

Tai pat kuriama sistema turės bendradarbiauti su lygiagrečiai kuriamų produktų „Automato kontroliškumo tyrimo programinė įranga“. Abidvi sistemos turės dirbti vienoje darbo stotyje ir pagal vartotojo poreiki turi būti numatytas greitas perėjimas (perjungimas) iš vienos sistemos į kitą, o taip pat galimas sistemų darbo rezultatų palyginimas pagal atitinkamus kriterijus

### 3.5 Veiklos sfera

#### 3.5.1 Veiklos kontekstas



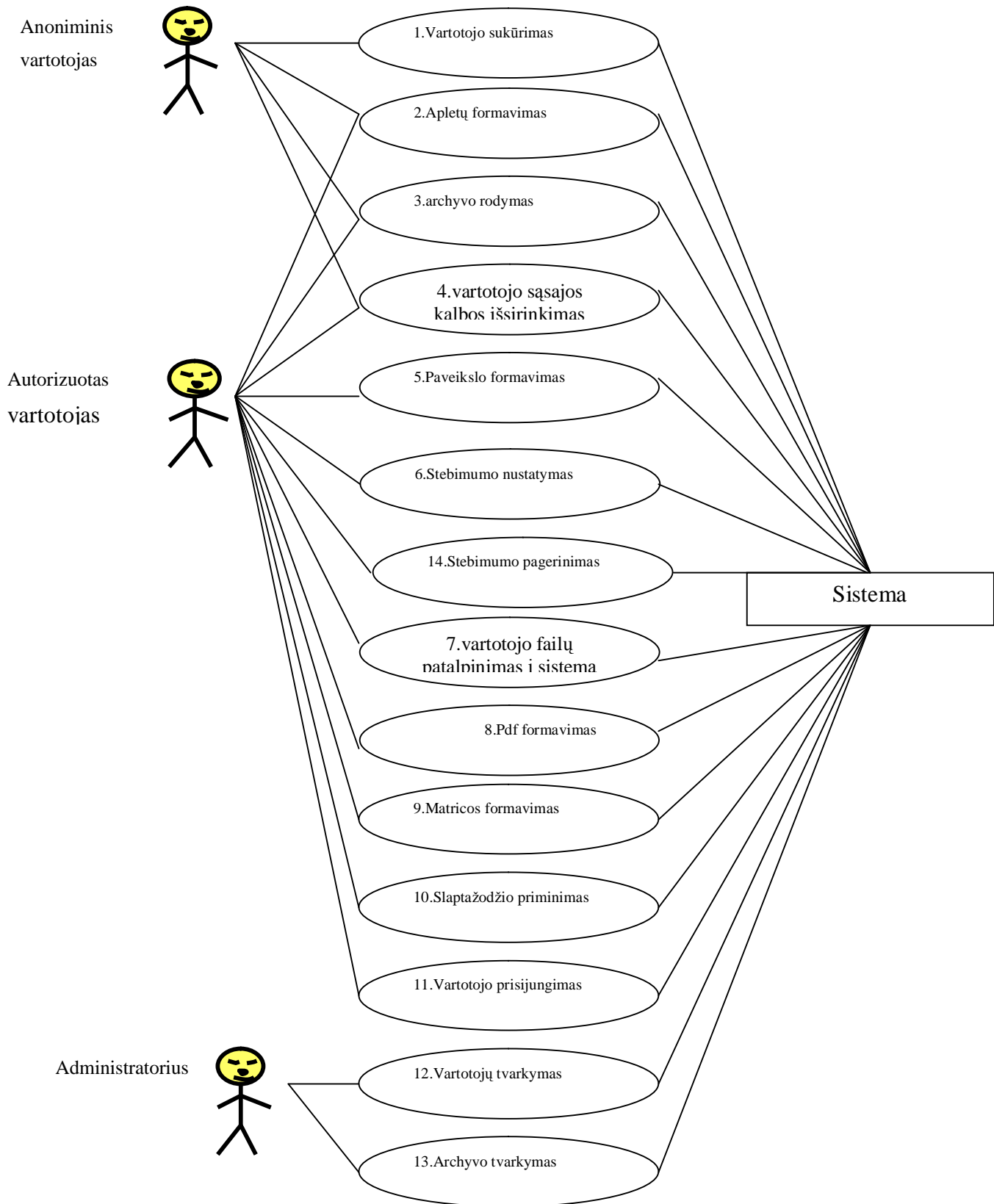
3.5.1.1 pav Konteksto diagrama

## 3.5.2 Veiklos pavadinimas

3.5.2.1 lentelė Veiklos įvykių sąrašas

Eil. Nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	abstrakčių automatų tiekėjas pateikia automatų modelius	automatų testiniai modeliai
2	testinių duomenų tiekėjas pateikia testinius duomenys	automatų testiniai duomenys
3	identifikacijos menedžeris gauna vartotojo informacija	vartotojo asmeninė informacija
4	identifikacijos menedžeris siunčia statusą	informacija su vartotojo teisėmis sistemoje
5	sistemos administratorius tvarko sistema	valdymo komandos
6	sistemos administratorius dirba su sistema	užklauso
7	reakcija į administratoriaus darbą	ataskaitos ir sistemos darbo rezultatas
8	vartotojo identifikacija	asmeninė pasijungimo informacija
9	testavimo atlikimas	testavimo komandos
10	vartotojas tvarko savo informacija	darbinė informacija
11	testų rezultatų gavimas	testų rezultatai

3.5.3 Panaudojimo atvejų diagrama

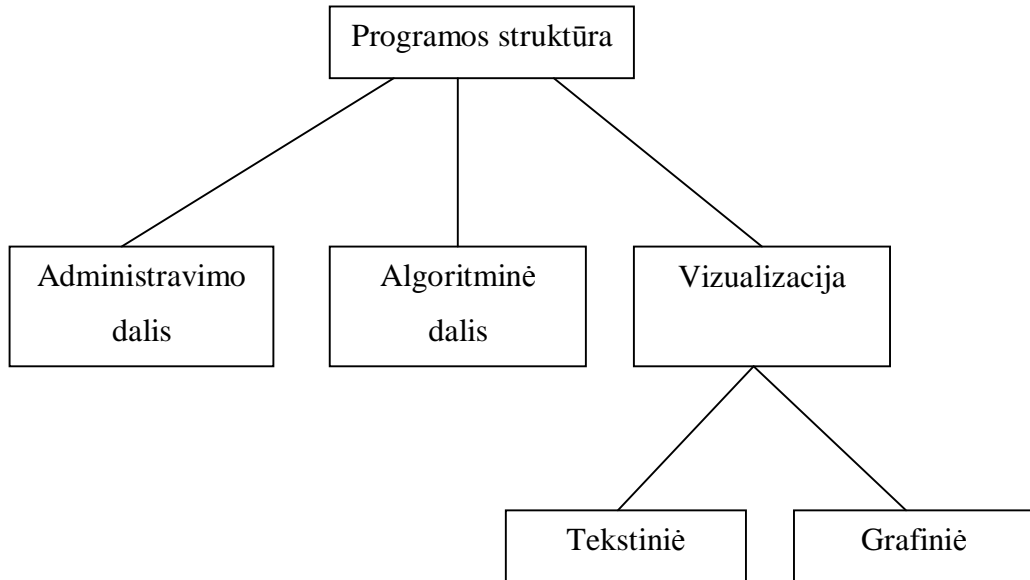


3.5.3.1 pav PA diagrama

### 3.6 Bendra sistemos architektūra

Kadangi sistema yra gana didelė ir sudėtinga todėl yra tikslinga ją suskaidyti atskiromis dalimis. Sistemos darbo režimus realizuoja atskiros programos dalys:

- 1) Vartotojo registracija, darbas su vartotojų profailais.
- 2) Esminiai sistemos skaičiavimai, pagrindiniai algoritmai.
- 3) Vizualizacija



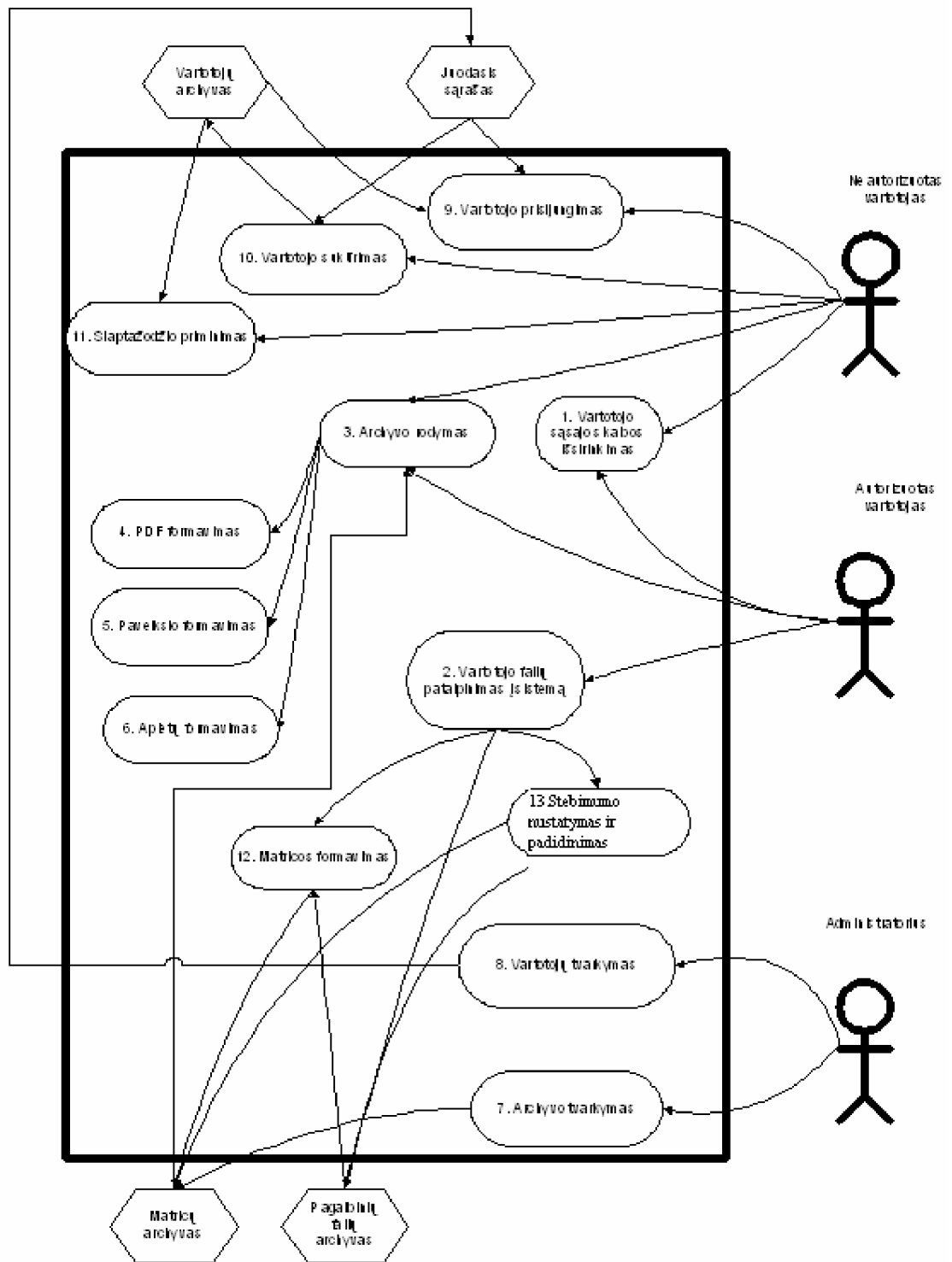
#### 3.6.1 pavSistemos struktūra

Vartotojo registracijos modulis skirtas realizuoti visoms funkcijoms kurios yra skirtos vartotojo registracijai bei autentifikavimui. Taip pat viena iš funkcijų yra slaptažodžio priminimas, bei „juodojo“ sąrašo valdymas (neleidžiama prisijungti vartotojams kurie yra atlikę neleistiną veiksmą sistemoje).

Skaičiavimų modulis skirtas naujo automato įkėlimo funkcijoms bei skaičiavimų atlikimui su sėkmingai įkeltu automatu. Šis modulis atsakingas už stebimumo apskaičiavimą bei stebimumo pagerinimą.

Vizualizacijos modulis skirtas rezultatų atvaizdavimui įvairiais būdais. Galimas tekstinis atvaizdavimas, naudojamas kai reikia žinoti tikslias rezultatų reikšmes. Grafinis režimas naudojamas kai norima susidaryti bendrą nuomonę apie schemas valdomumą.

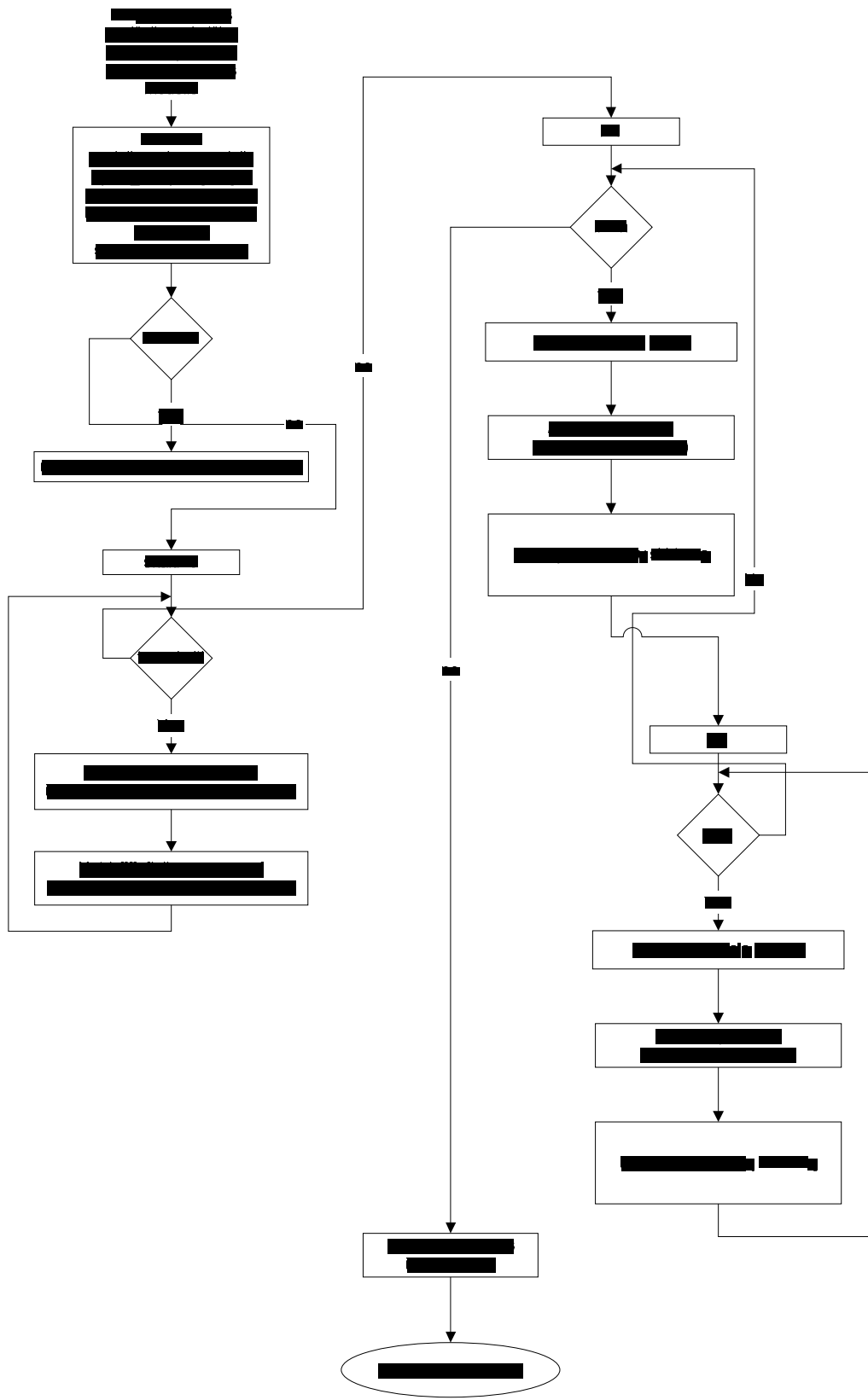
Detali sistemos architektūra pateikiama **3.6.2 pav.**



3.6.2 Sistemos architektūra



3.7 Stebimumo nustatymo detali architektūra



3.7.1 pav stebimumo nustatymo struktūrinė schema

Automato nustatyti stebimumo koeficientai priklauso nuo iteracijų skaičiaus, kurios nurodė vartotojas t.y. nuo automatų kopijų skaičiaus, per kuriuos pereis nustatymo algoritmas.

Atlikus skaičiavimus kiekvienam įėjimui yra nustatomas stebimumo perdavimo faktorius. Jis yra nustatomas pagal formulę:

$$SPF(I-O) = \frac{N(PDC:I-O)}{N(PDC:I-O) + N(NPDC:I-O)}$$

### 2.5.1 formule Stebimumo perdavimo faktorius

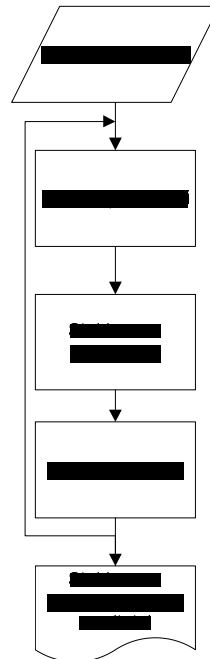
kur  $N(PDC : I - O)$  yra bendras skaičius skirtingų jautrių ryšių tarp vieno įėjimo ir visų išėjimų, o  $N(NPDC:I-O)$ - nejautrių ryšių skaičius.

## 3.8 Stebimumo padidinimo detali architektūra

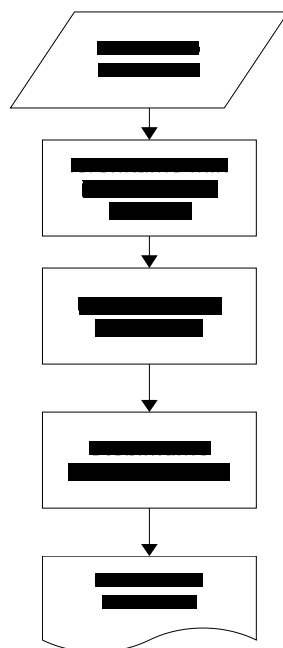
Komponento tikslas yra pagerinti vartotojo modelio stebimumo lygį ir pateikti vartotojui sprendimą, kaip pagerinimas yra pasiektas. Neaišku ar šis metodas duos teisingus rezultatus.

Po stebimumo apskaičiavimo kiekvienas būsenos įėjimas turi apskaičiuota stebimumo koeficientą, o taip kaip yra naudojamas iteracinis modelis galima teigti, kad išėjimo koeficientas atitinka įėjimo koeficientą.

Komponentas yra atsakingas už automato stebimumo padidinimą, remiantis anksčiau apskaičiuotais rezultatais. Priklausomai nuo ankstesnių rezultatų yra priimamas sprendimas kokius pakeitimus įnešti į modelį, kad jo stebimumas padidėtų. Yra įnešami pakeitimai, stebimumas apskaičiuojamas iš naujo. Radus geriausią stebimumo padidinimo variantą sprendimas yra pateikiamas vartotojui.



3.8.1 pav komponento darbo struktūra



3.8.2 skaičiavimų eiliškumas

## 4. EKSPERIMENTINĖ DALIS

### 4.1 Tyrimo objektas

Tyrimo objektas yra abstrakčių automatų stebimumo perdavimo faktorius, arba kitaip stebimumo koeficientas. Bus tiriamas stebimumo koeficientas baziniam bei pagerintam modeliui ir atliekamas tarpusavio palyginimas ir prie kokio iteracijų skaičiaus maksimalus stebimumo koeficientas yra pasiekiamas. Taip pat tyrimu objektu buvo nuspręsta pasirinkti sprendimo iteracinio modelio iteracijų skaičių, modelio atkartojimų skaičių, skaičiavimų tikslumo apibrėžimą bei stebimumo gaunamas reikšmes.

### 4.2 Terminų apibrėžimas

Tam, kad suprantamai ir aiškiai pateikti tyrimo rezultatus, reikia apibrėžti tyrime naudojamas sąvokas:

Iteracinis modelis - tai yra kombinacinių schemų, sujungtu tarpusavyje nuosekliai, grandinė. Atskiuro automato būsenos yra gaunamos iš prieš tai buvusio automato, gauto po darbo automato būsenos perduodamos sekančiam grandinėje automatui. Taip yra imituojamas schemas su atmintim darbas.

Iteracija – iteracinio modelio vieno automato įvykdymas.

Atkartojimas – viso iteracinio modelio įvykdymas dar kartą.

Procentinis nuokrypis – procentinis skirtumas tarp dviejų gretimų skaičiavimo rezultatų reikšmių

### 4.3 Tyrimo prielaidos ir metodika

Stebimumo apskaičiavimo algoritmas remiasi atsitiktinių duomenų generavimu. Būtų korektiška atlikinėti pilna duomenų, paduodamų automatams, perrinkimą, bet to buvo atsisakyta, nes turint didesnius ir sudėtingesnius automatus skaičiavimų sudėtingumas žymiai padidėtų, nes kiekvieno naujo įėjimo pridėjimas prie automato, padidina skaičiavimus eksponentiškai.

Turint atsitiktinį duomenų generavimą atsiras skaičiavimu paklaida. Mūso realizuoto modelio atveju ši paklaida priklausys nuo iteracijų skaičiaus ir nuo atkartojimų skaičiaus. Kuo didesnis bus iteracijų ir atkartojimų skaičius, tuo bus mažesnė paklaida. Bet iš kitos iteracijų ir atkartojimų skaičiaus didinimas apsunkina skaičiavimus. Kuo didesni iteracijų ir atkartojimų skaičių pateiksime sistemai, tuo lėčiau gausime laukiamus rezultatus.

Šioje vietoje sunku tiksliai pasakyti, koks iteracijų ir atkartojimų skaičius turi būti, kad gauti su dideliu tikslumu automato stebimumo reikšmę. Šie du parametrai yra skirtingi kiekvienai schemai. Pradinėje sistemoje šie du parametrai, iteracijų skaičius ir atkartojimų skaičius, turėjo įvertinti ir

pasirinkti vartotojas. Įnešus pakeitimus į sistemą, abudu šie koeficientai yra nustatomi dinamiškai. Jie yra didinami iki tol, kol įvyksta skaičiavimų „prisisotinimas“, t.y. darbo metu yra stebimi darbo rezultatai, ir schema testuojama iki tol, kol išėjimo reikšmių „bangavimas“ tampa mažesnis už pasirinktą konstantą.

Tyrimo darbo metu buvo nuspręsta ištirti šios du koeficientus atskirai kiekvienai schemai. Iteracijų skaičius ir atkartojimai buvo nustatomi dinamiškai. Taip pat bus pateikti iteracijų ir atkartojimų tarpusavio priklausomybės grafikai. Turint šios grafikus galima bus atsakyti į klausimą, koks iš šių parametru turi didesnę reikšmę mūsų modelyje.

Pakartojimų parametro prisisotinimas tikrinamas po kiekvieno iteracinio modelio įvykdymo. Žiūrima ar gautas bendras automato stebimumas nenukrypsta nuo prieš tai gautų penkiasdešimties rezultatų daugiau kaip dviem dešimtosiomis procento dalimis. Jei skiriasi daugiau nei 0,2% tada skaičiavimai tęsiami toliau. Jei per 50 paskutinių pakartojimų rezultatai nesvyravo virš šios ribos, tada galima teigti jog automato stebimumas, prie pasirinkto iteracijų skaičiaus, nusistovėjo ir netikslinga didinti pakartojimų skaičių, nes jie jau nebepakeis galutinio rezultato.

Iteracijų prisisotinimas nustatinėjamas kai jau yra nusistovėjęs pakartojimų skaičius. Pakeičiamas automatų iteracijų skaičius ir vėl skaičiuojamas jo bendras stebimumas. Gautas stebimumas yra lyginamas su prieš tai gautomis stebimumo reikšmėmis. Jos neturi svyruoti daugiau kaip dviem dešimtosiomis procento. Skaičiavimų pabaigos sąlyga ta pati kaip ir pakartojimų – jei per 50 paskutinių iteracijų pakeitimų stebimumas nesvyravo daugiau nei 0,2% tai reiškia kad iteracijų skaičiaus didinti nebereikia ir galima nutraukti skaičiavimus, nes rezultatai, gauti sekančių skaičiavimų metu, nebepakeis galutinio rezultato.

Taip pat palyginsime abstrakčių automatų stebimumo koeficientus baziniam ir pagerintam modeliams. Palyginimo reikšmes taip pat pateiksime grafiškai.

Tyrimo metu buvo iškeltos prielaidos:

1. Jei pakartojimų skaičius yra per mažas, tada gautas rezultatas bus netikslus, t. y. pakartotinai atlikus skaičiavimus su tuo pačiu automatu bus gautas kitoks rezultatas.
2. Jei iteracijų skaičius yra per mažas, tada gautas rezultatas taip pat gali būti netikslus, nes kai kuriems automatams kad pasiekti tam tikrą būseną reikia labai didelio iteracijų skaičiaus.

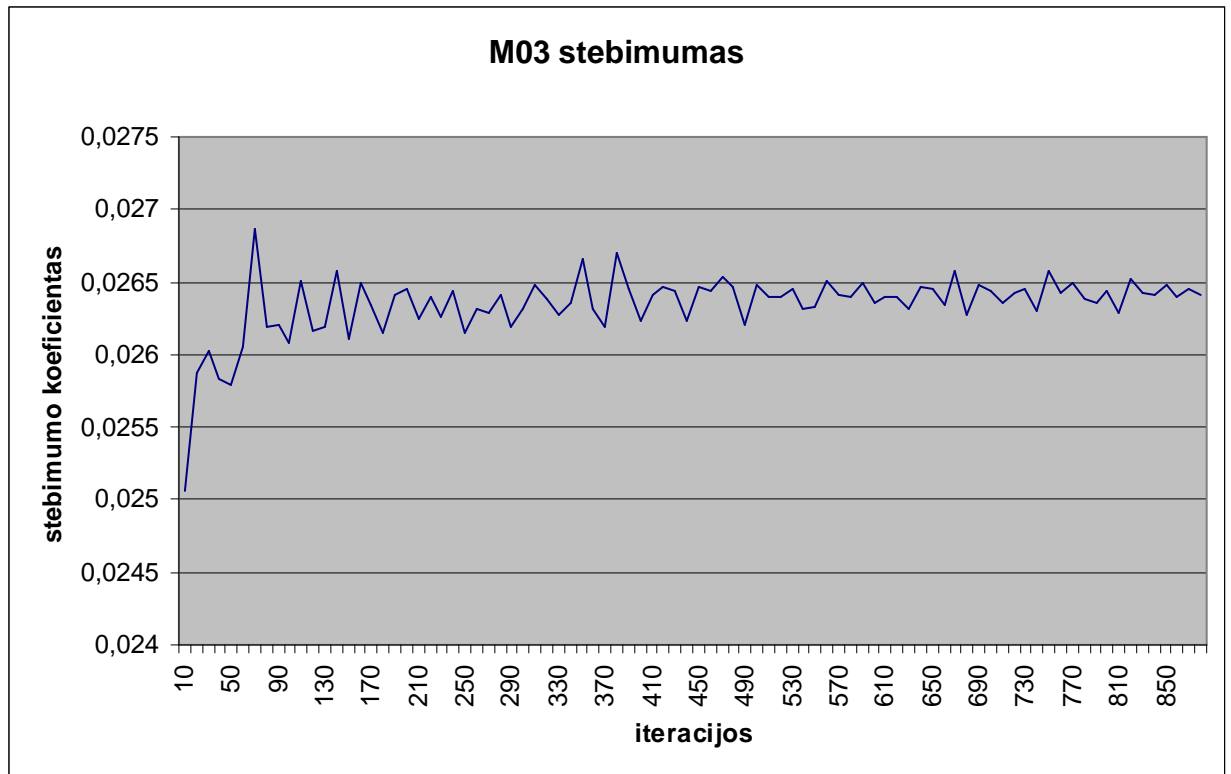
Pakartojimų skaičius turi priklausyti nuo iteracijų skaičiaus. Kuo didesnis iteracijų skaičius, tuo daugiau automato būsenų yra išnagrinėjama, tuo būdu yra gaunamas tikslesnis automato stebimumas. Kuo tikslesnis vieno pakartojimo vidurkis tuo mažesnis turi būti rezultatų svyravimas tarp pakartojimų.

Kad patvirtinti arba paneigti prielaidas buvo atlikti testai su abstrakčiais automatais. Buvo ištirti trys automatai. M03 ir M05 – automatai su atmintimi, ir C499 – automatas be atminties, todėl jo valdomumo pagerinimas nebus tiriamas.

### 4.3. Eksperimentų rezultatai

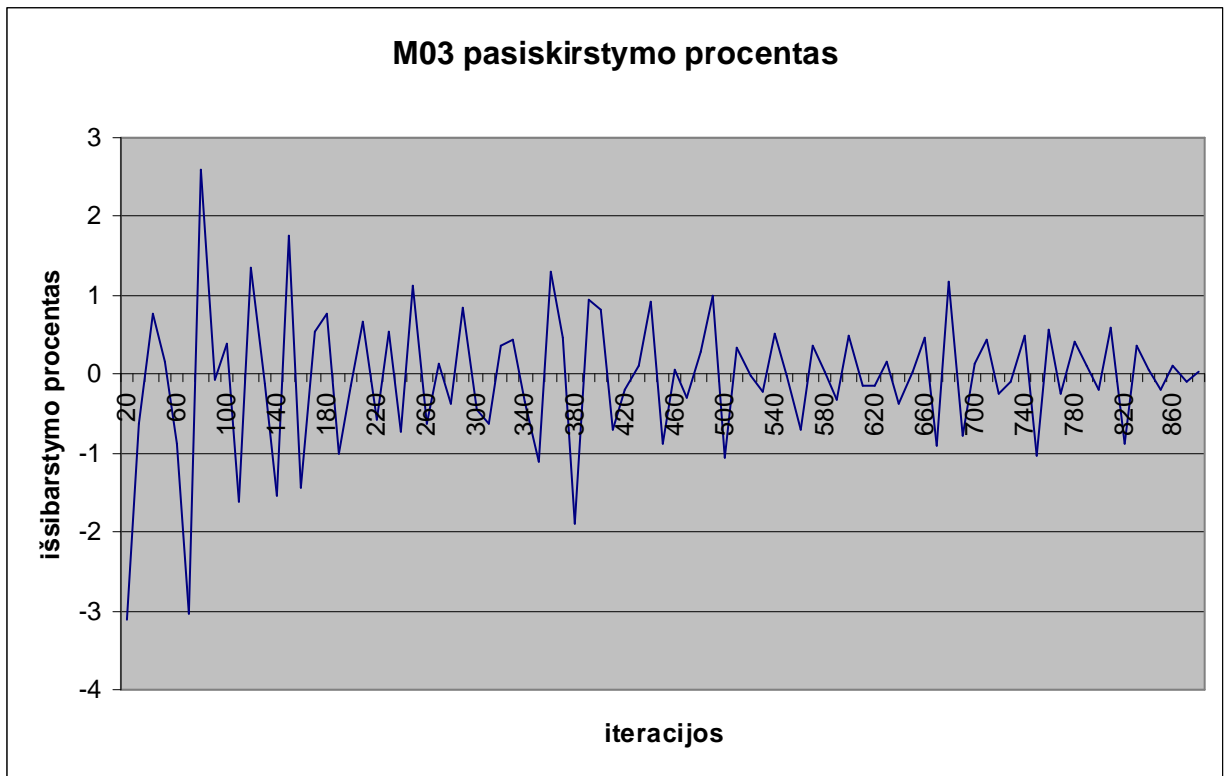
#### 4.3.1. Automato M03 tyrimas

##### Bazinio automato M03 modelio tyrimas



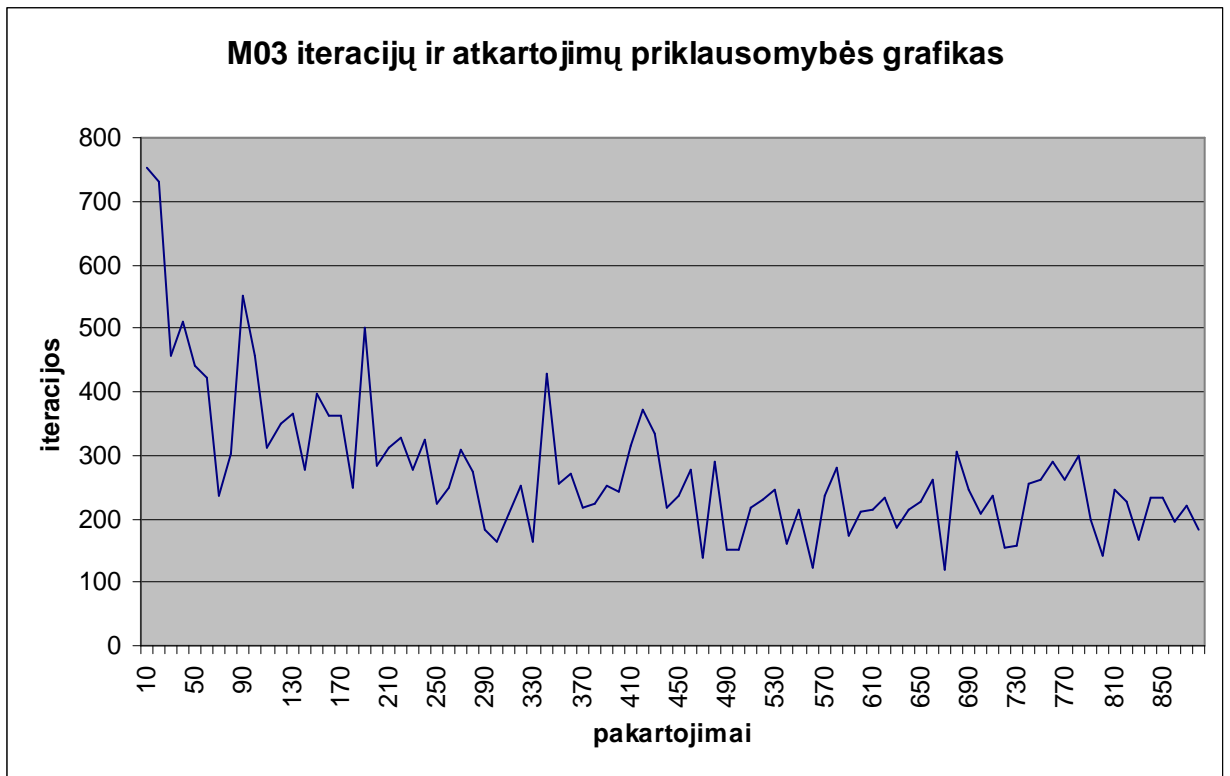
#### 4.3.1.1 pav M03 automato stebimumo priklausomybė nuo iteracijų skaičiaus

Iš gautų tyrimo rezultatų galima pamatyti, kad automato maksimalus stebimumas buvo pasiektas sudarius iteracinį modelį iš 70 iteracijų. Stebimumo reikšmė prie 70 iteracijų yra 0,026872.



#### 4.3.1.2 pav M03 automato tikslumo priklausomybė nuo iteracijų skaičiaus

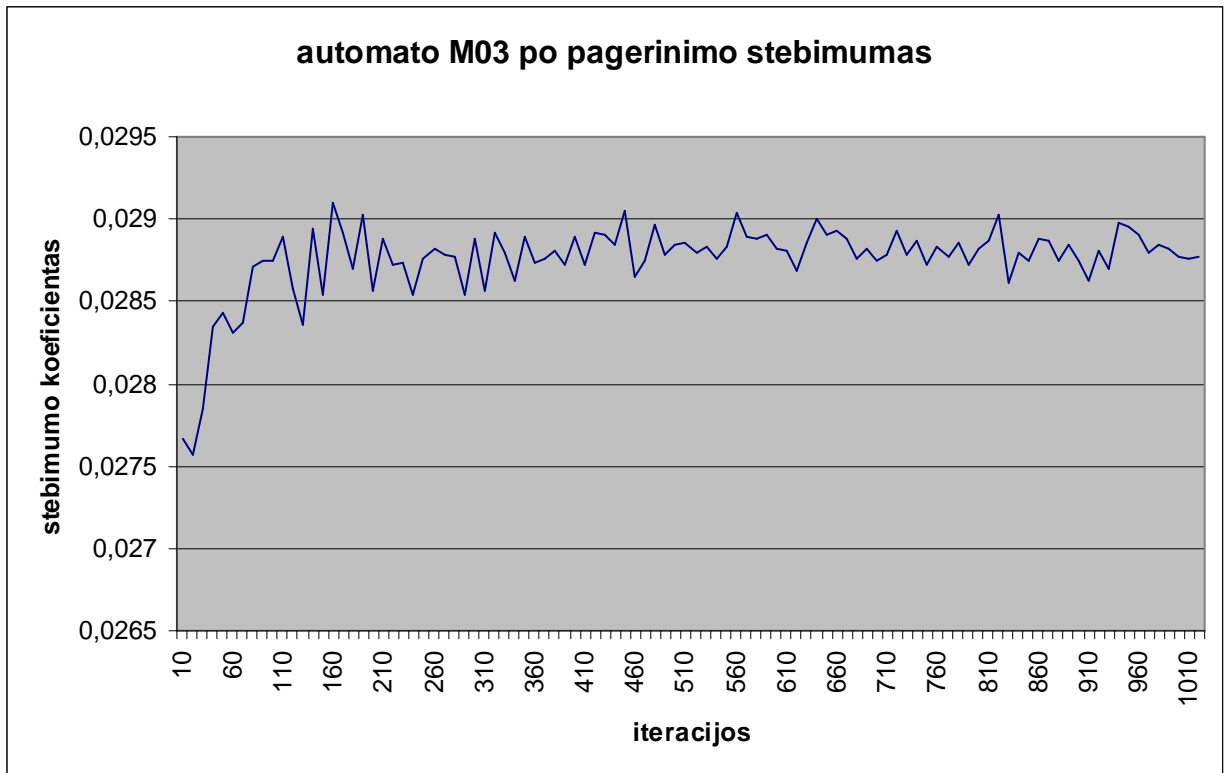
Iš pateikto grafiko matome kada įvyko skaičiavimų nutraukimas. Skaičiavimai buvo nutraukti, kai buvo pasiektas nusistovėjimas prie reikšmės 0,2% 50 kartų.



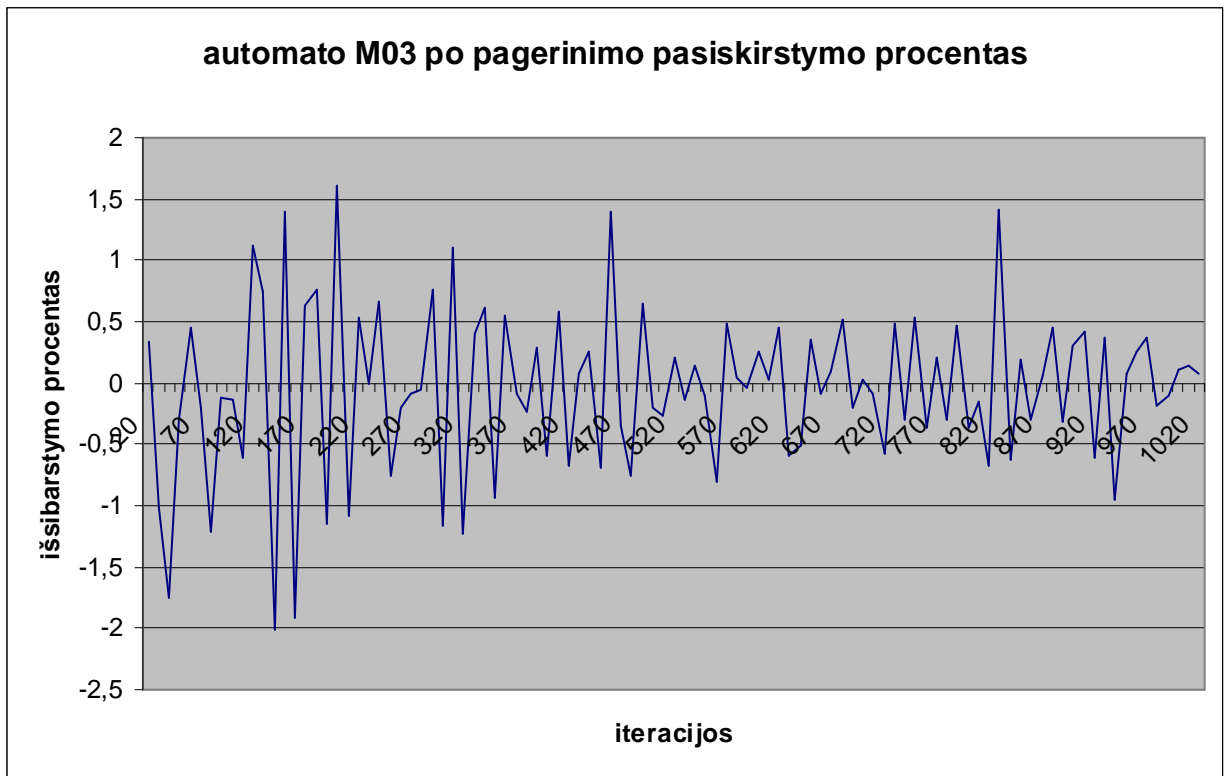
#### 4.3.1.3 pav M03 automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė

Iš šio grafiko matome, kaip modelio vykdymo metu keitėsi pakartojimų ir iteracijų skaičius. Tiriama šį grafiką galima padaryti išvadas, kad kuo daugiau kartų įvykdome iteracinio modelio atkartojimą, tuo mažiau iteracijų reikia nusistovėjimo nustatymui.



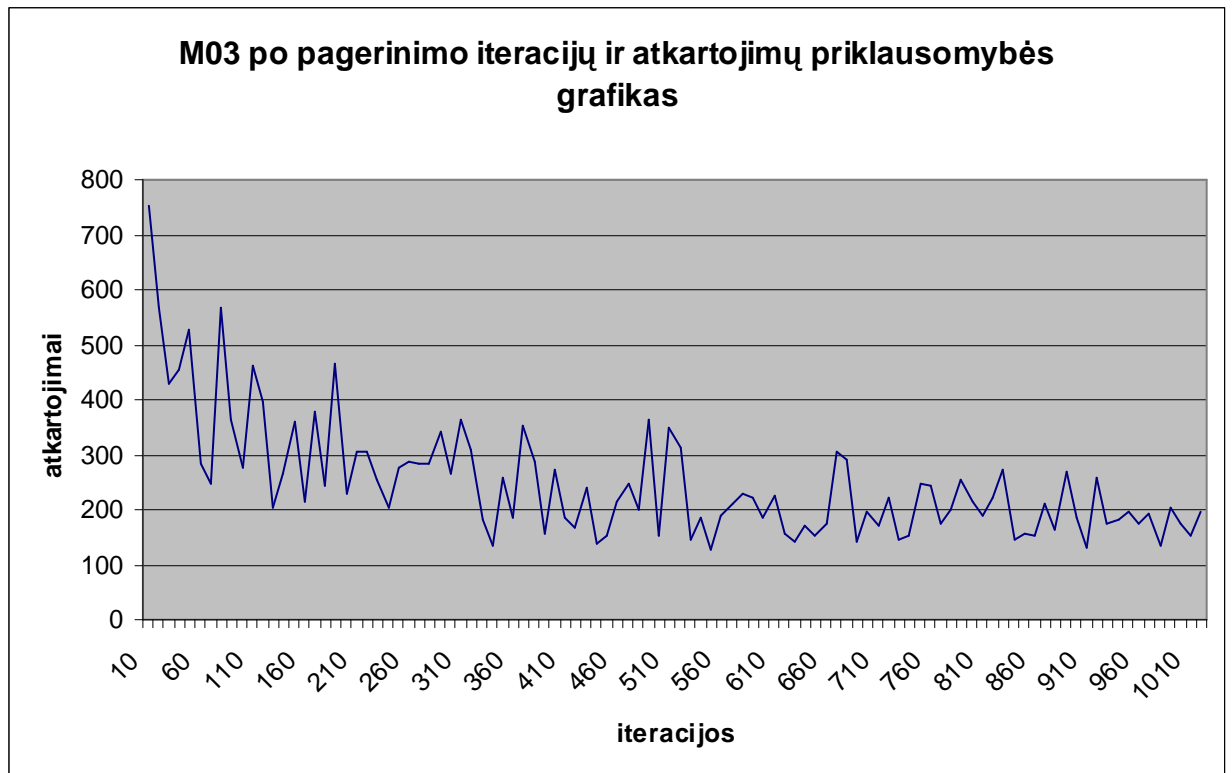
Pagerinto automato M03 modelio tyrimas**4.3.1.4 pav M03 pagerinto automato stebimumo priklausomybė nuo iteracijų skaičiaus**

Iš gautų tyrimo rezultatų galima pamatyti, kad automato maksimalus stebimumas buvo pasiektas sudarius iteracinį modelį iš 190 iteracijų. Stebimumo reikšmė prie 190 iteracijų yra 0,029903.



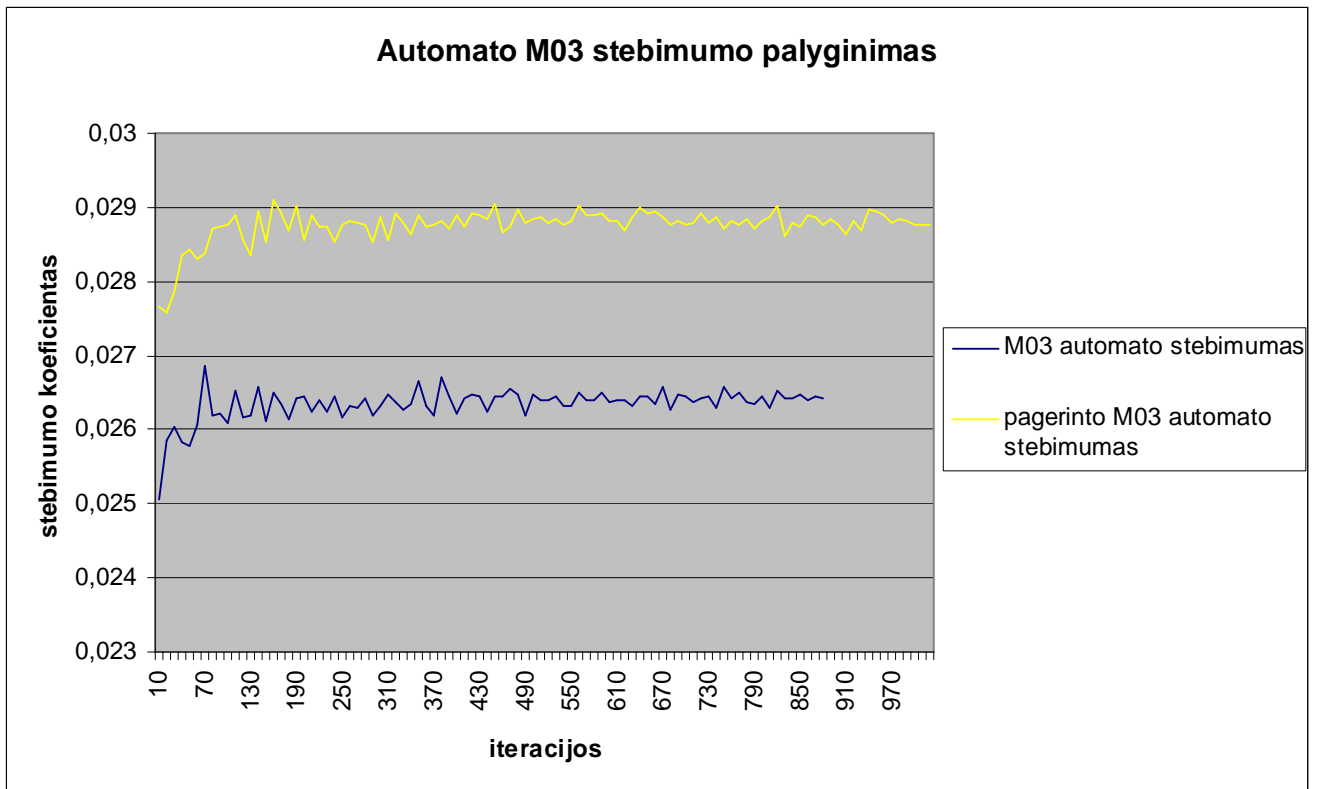
**4.3.1.5 pav M03 pagerinto automato tikslumo priklausomybė nuo iteracijų skaičiaus**

Iš pateikto grafiko matome kada įvyko skaičiavimų nutraukimas. Skaičiavimai buvo nutraukti, kai buvo pasiektas nusistovėjimas prie reikšmės 0,2% 50 kartų.

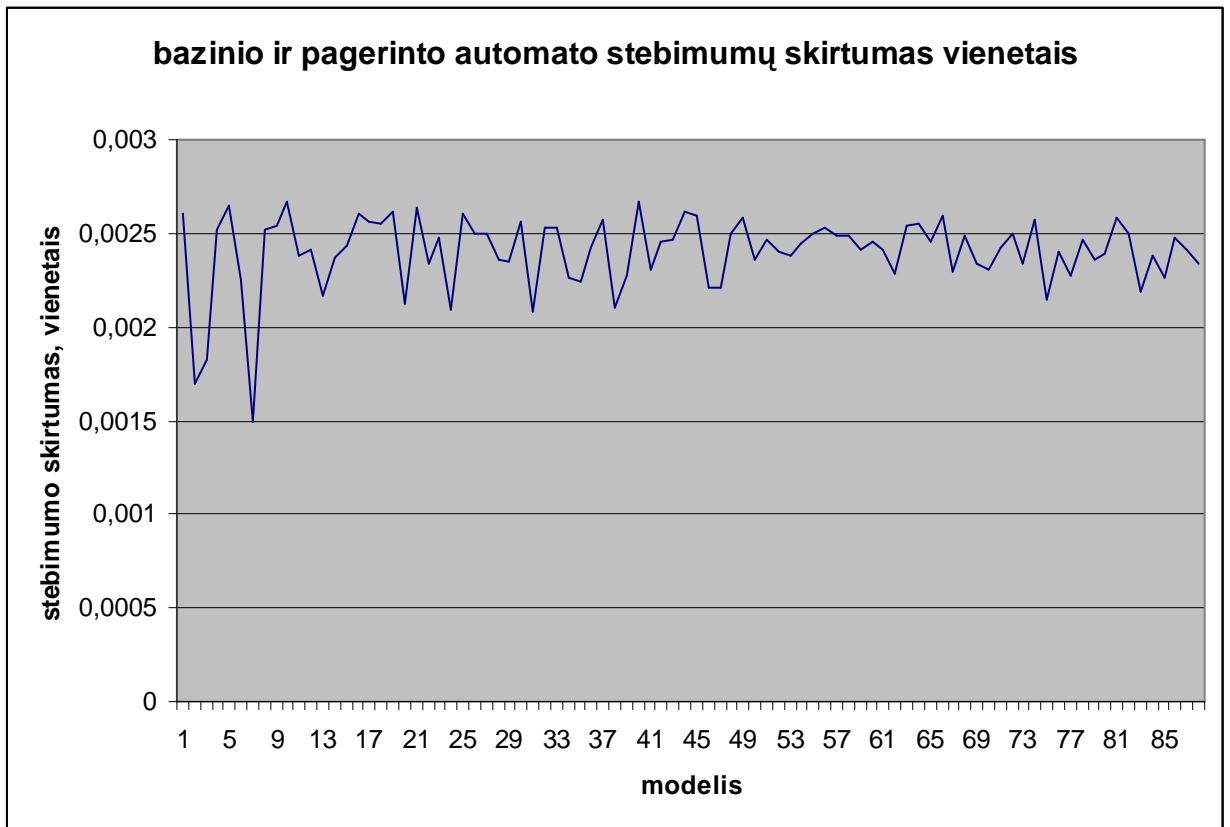


**4.3.1.6 pav M03 pagerinto automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė**

Iš šio grafiko matome, kaip modelio vykdymo metu keitėsi pakartojimų ir iteracijų skaičius. Tiriant šį grafiką galima padaryti išvadas, kad kuo daugiau kartų įvykdome iteracinio modelio atkartojimą, tuo mažiau iteracijų reikia nusistovėjimo nustatymui.

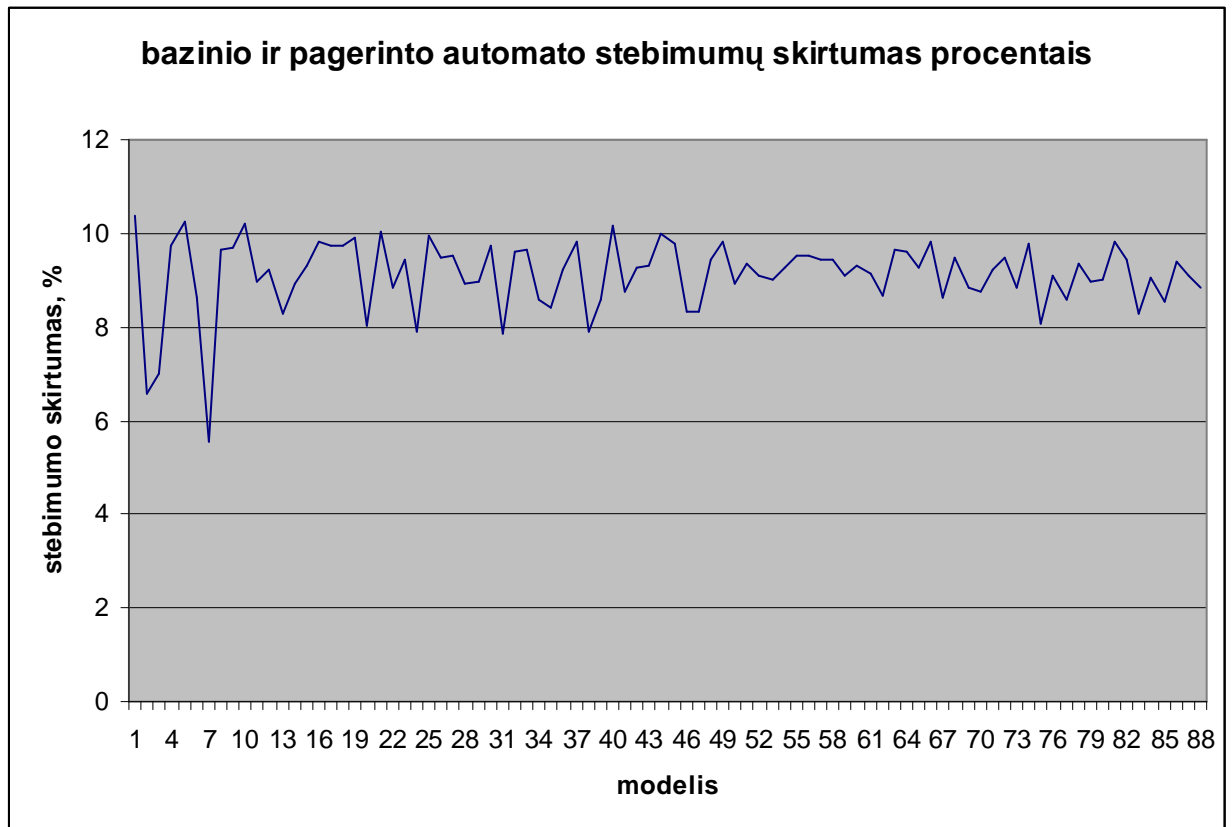
Bazinio ir pagerinto modelio palyginimas**4.3.1.7 pzv M03 automato bazinio ir pagerinto modelio stebimumo palyginimas**

Pateiktame grafike galima matyti, kiek padidėjo automato stebimumas po stebimumo pagerinimo. Vidutiniškai per visus modelius jis padidėjo 0,00240048 vienetais, arba 9,114% .



**4.3.1.8 pav M03 automato bazinio ir pagerinto automatų stebimumo skirtumas vienetais**

Pateiktame grafike atvaizduota kiek vienetų padidėjo automato stebimumas vienetais po kiekvieno modelio įvykdymo.



#### 4.3.1.9 pav M03 automato bazinio ir pagerinto automatų stebimumo skirtumas vienetais

Pateiktame grafike atvaizduota kiek vienetų padidėjo automato stebimumas vienetais po kiekvieno modelio įvykdymo.

##### M03 automato tyrimo apibendrinimas

Panaudojus sukurta programinę įrangą pavyko nustatyti automato stebimumą, bei atlikti automato stebimumo pagerinimą.

Automato stebimumas pagerėjo 0,00240048 vienetais, arba 9,114 procentais.

70 iteracijų yra 0,026872

Bazinio automato stebimumo maksimumas yra 0,026872 ir jis buvo pasiektas sudarius iteracinį modelį iš 70 automato kopijų(iteracijų). Iteracijų prisisotinimas įvyko vykdant modelį su 880 iteracijų.

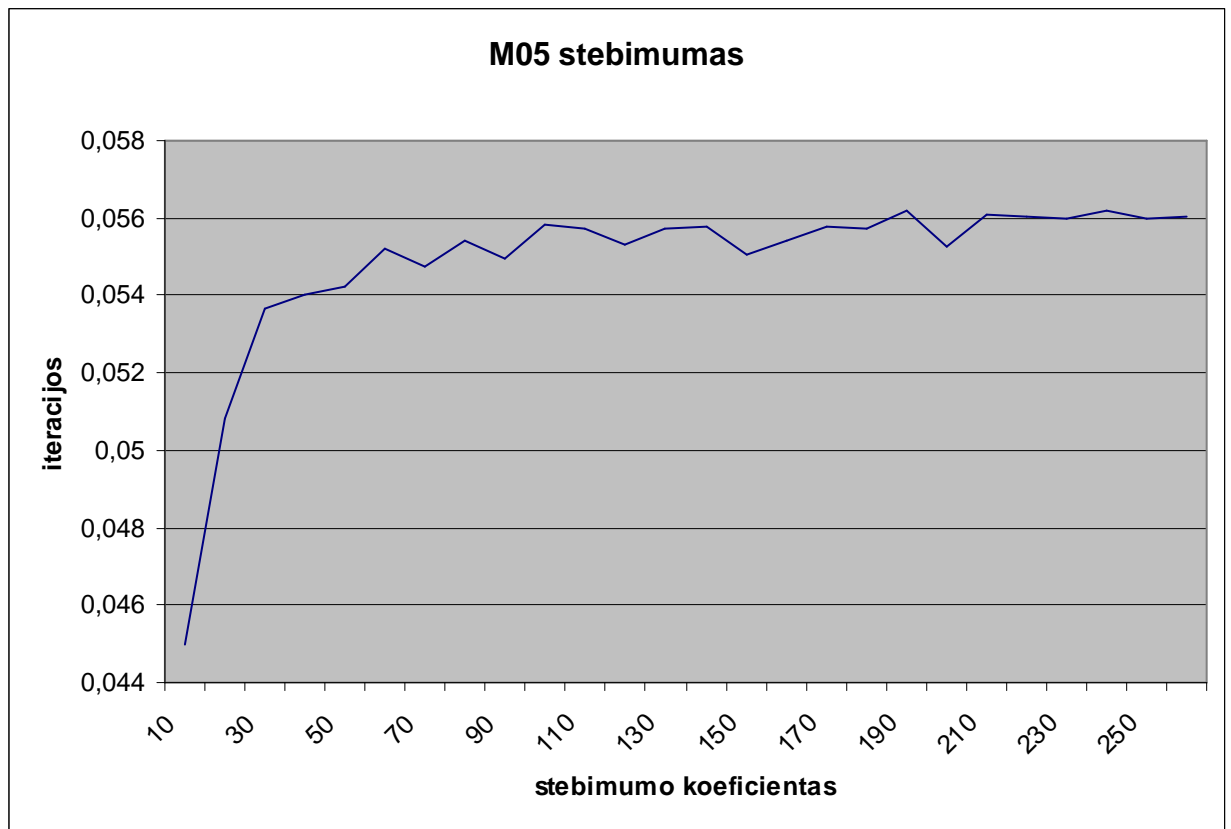
Pagerinto automato stebimumo maksimumas yra 0,029903 ir jis buvo pasiektas sudarius iteracinį modelį iš 190 automato kopijų(iteracijų). Iteracijų prisisotinimas įvyko vykdant modelį su 1020 iteracijų.

Iš to galima daryti išvada, kad šiam automatui pasiekti maksimalų stebimumo reikšmę bei pasiekti prisisotinimo būseną reikia atlikti daugiau iteracijų.

Iš iteracijų ir atkartojimų grafiko galima padaryti išvadą, kad kai yra mažas atkartojimų skaičius, reikalingas didelis iteracijų skaičius, bet didinant atkartojimų skaičių mažėja reikalingų iteracijų skaičius.

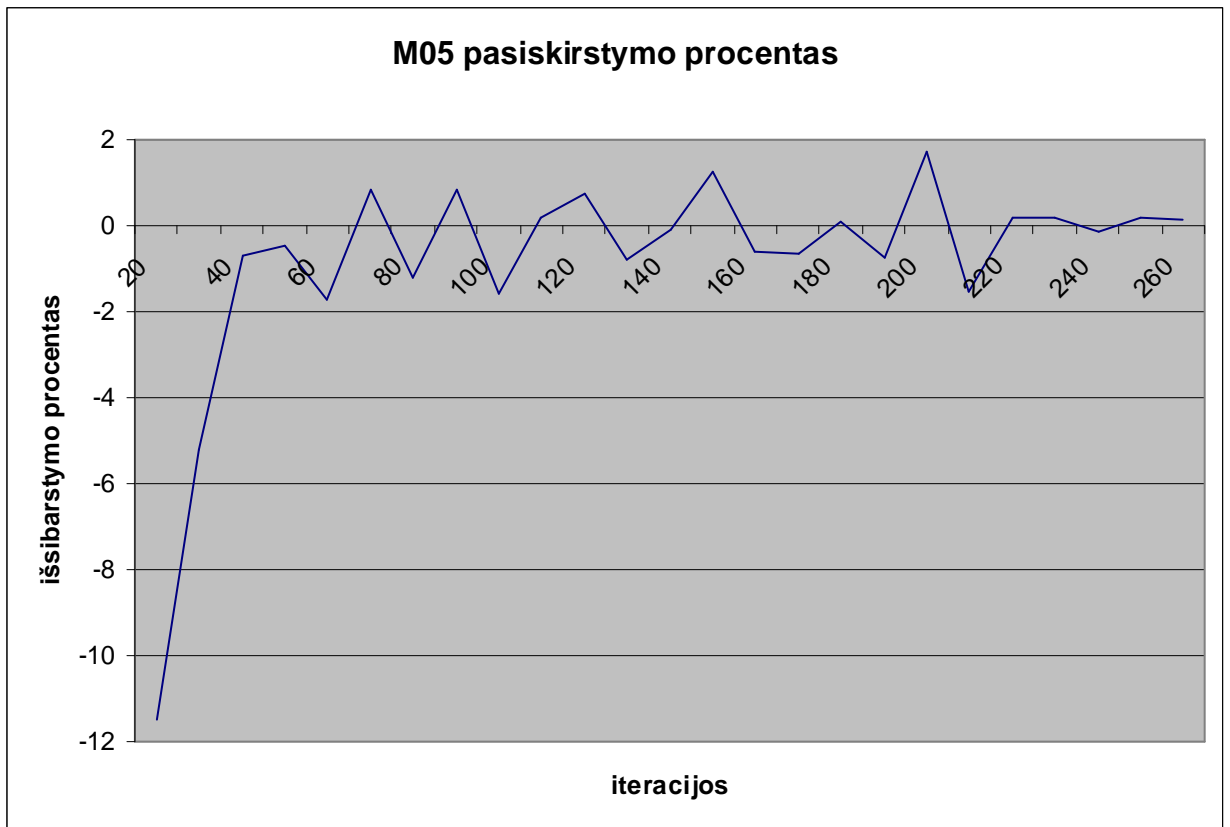
### 4.3.2. Automato M05 tyrimas

#### Bazinio automato M05 modelio tyrimas



#### 4.3.2.1 pav M05 automato Stebimumo priklausomybė nuo iteracijų skaičiaus

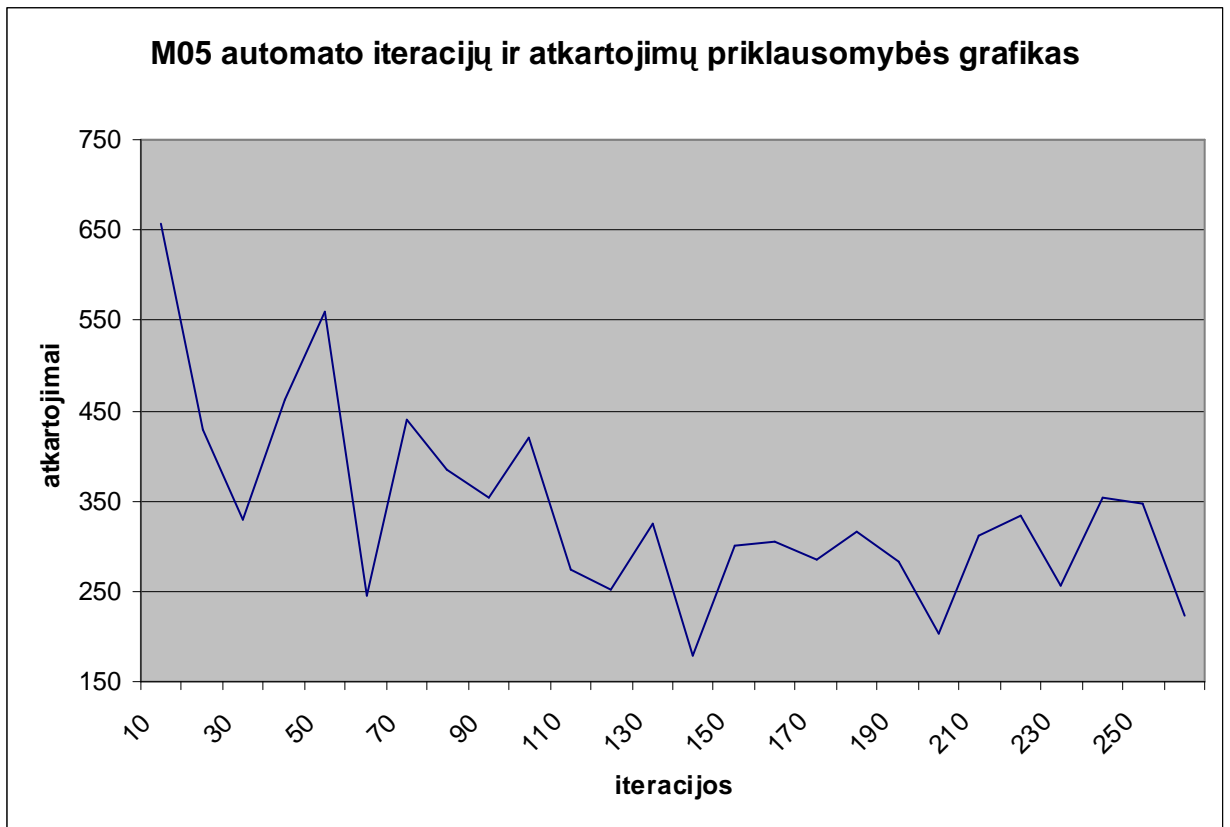
Iš gautų tyrimo rezultatų galima pamatyti, kad automato maksimalus stebimumas buvo pasiektas sudarius iteracinį modelį iš 190 iteracijų. Stebimumo reikšmė prie 190 iteracijų yra 0,056196.



#### 4.3.2.2 pav M05 automato tikslumo priklausomybė nuo iteracijų skaičiaus

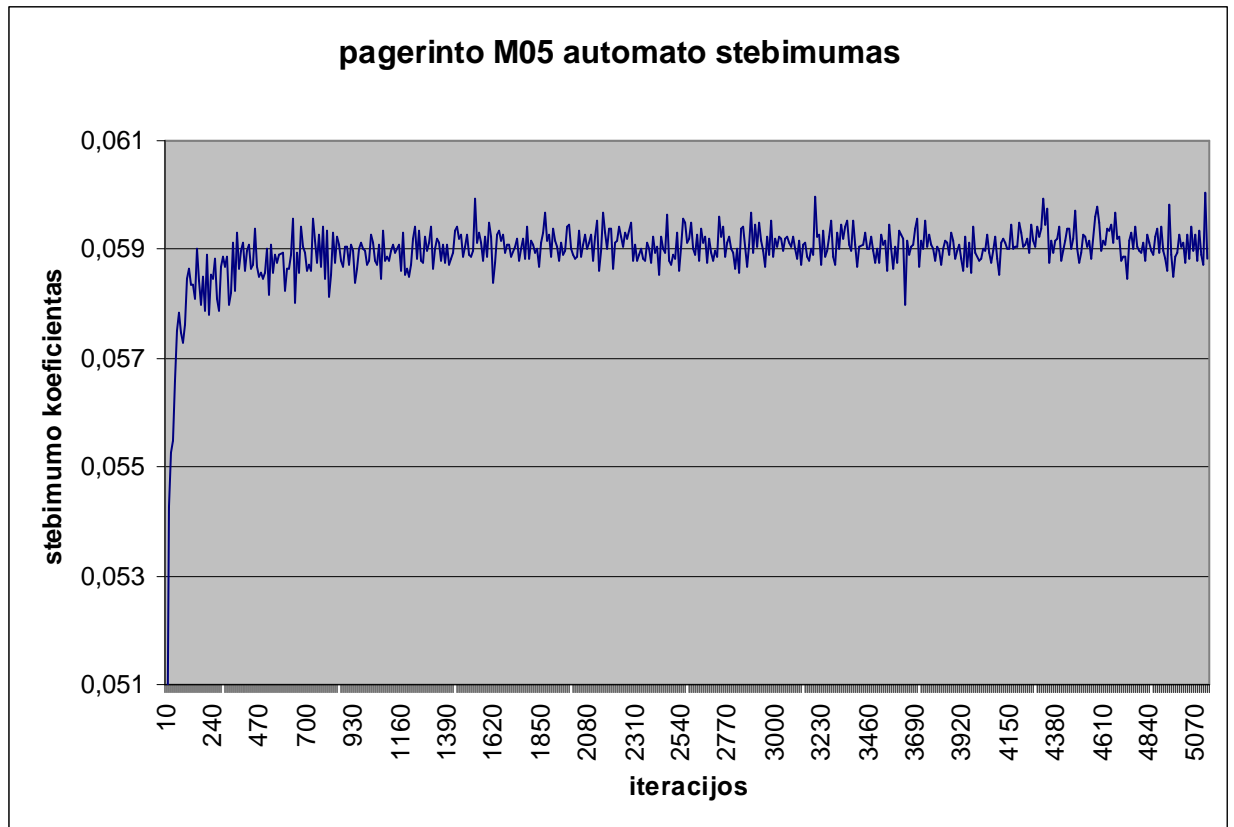
Iš pateikto grafiko matome kada įvyko skaičiavimų nutraukimas. Skaičiavimai buvo nutraukti, kai buvo pasiektas nusistovėjimas prie reikšmės 0,2% 50 kartų.



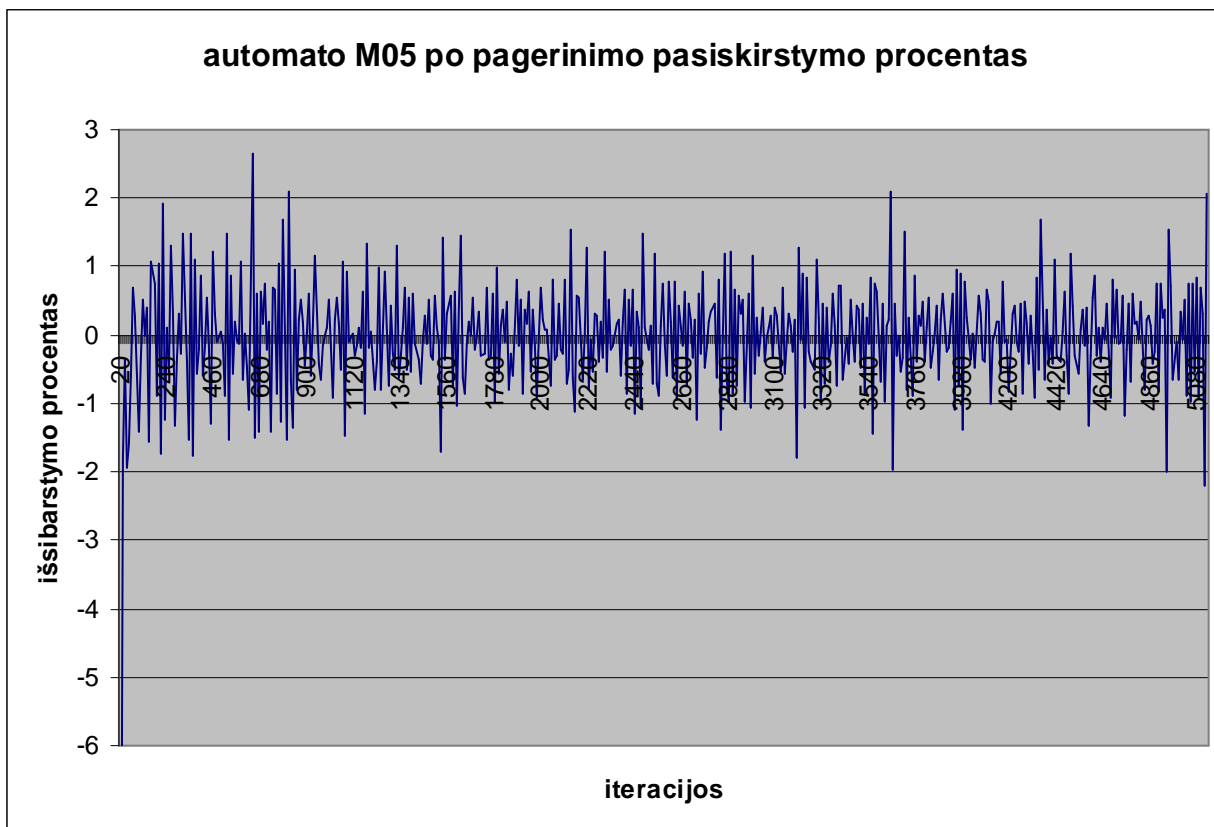


#### 4.3.1.3 pav M05 automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė

Iš šio grafiko matome, kaip modelio vykdymo metu keitėsi pakartojimų ir iteracijų skaičius. Tiriant šį grafiką galima padaryti išvadas, kad kuo daugiau kartų įvykdome iteracinio modelio atkartojimą, tuo mažiau iteracijų reikia nusistovėjimo nustatymui.

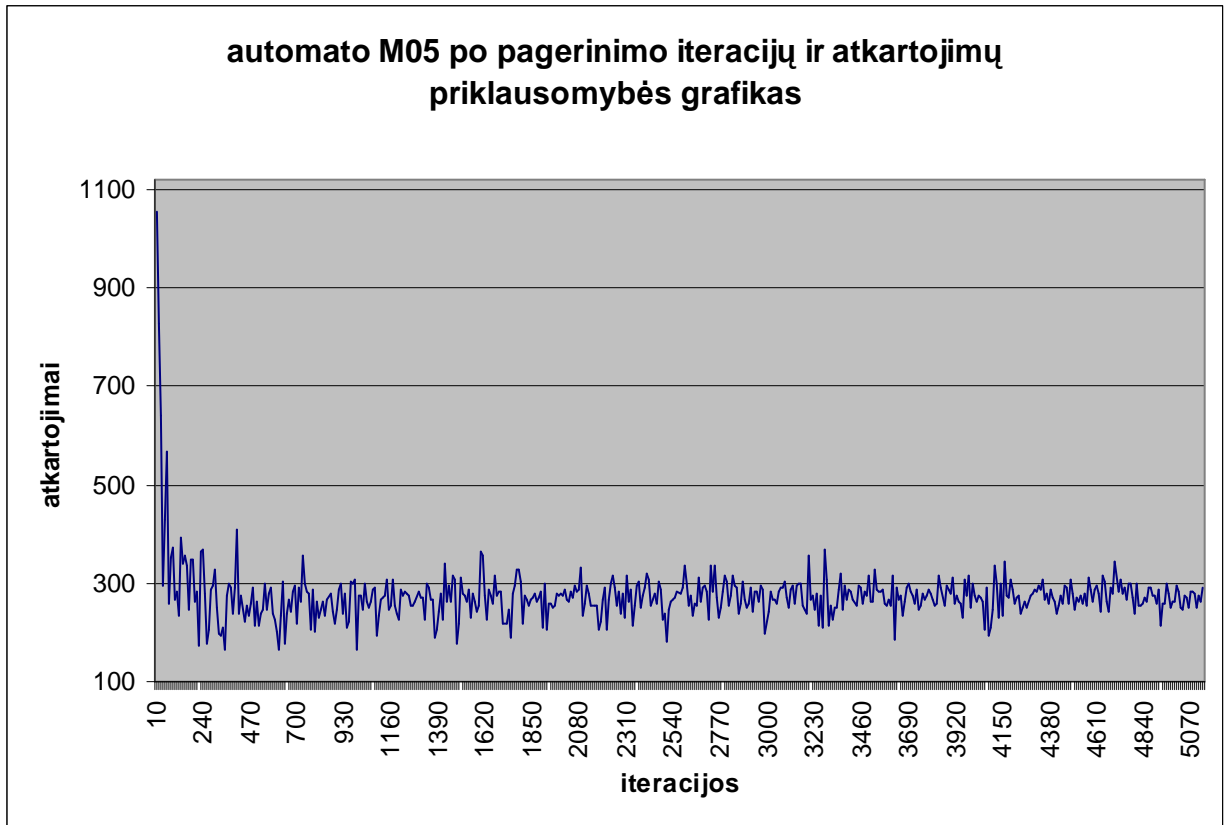
Pagerinto automato M05 modelio tyrimas**4.3.2.4 pav M05 pagerinto automato stebimumo priklausomybė nuo iteracijų skaičiaus**

Iš gautų tyrimo rezultatų galima pamatyti, kad automato maksimalus stebimumas buvo pasiektas sudarius iteracinį modelį iš 5120 iteracijų. Stebimumo reikšmė prie 5120 iteracijų yra 0,060048.



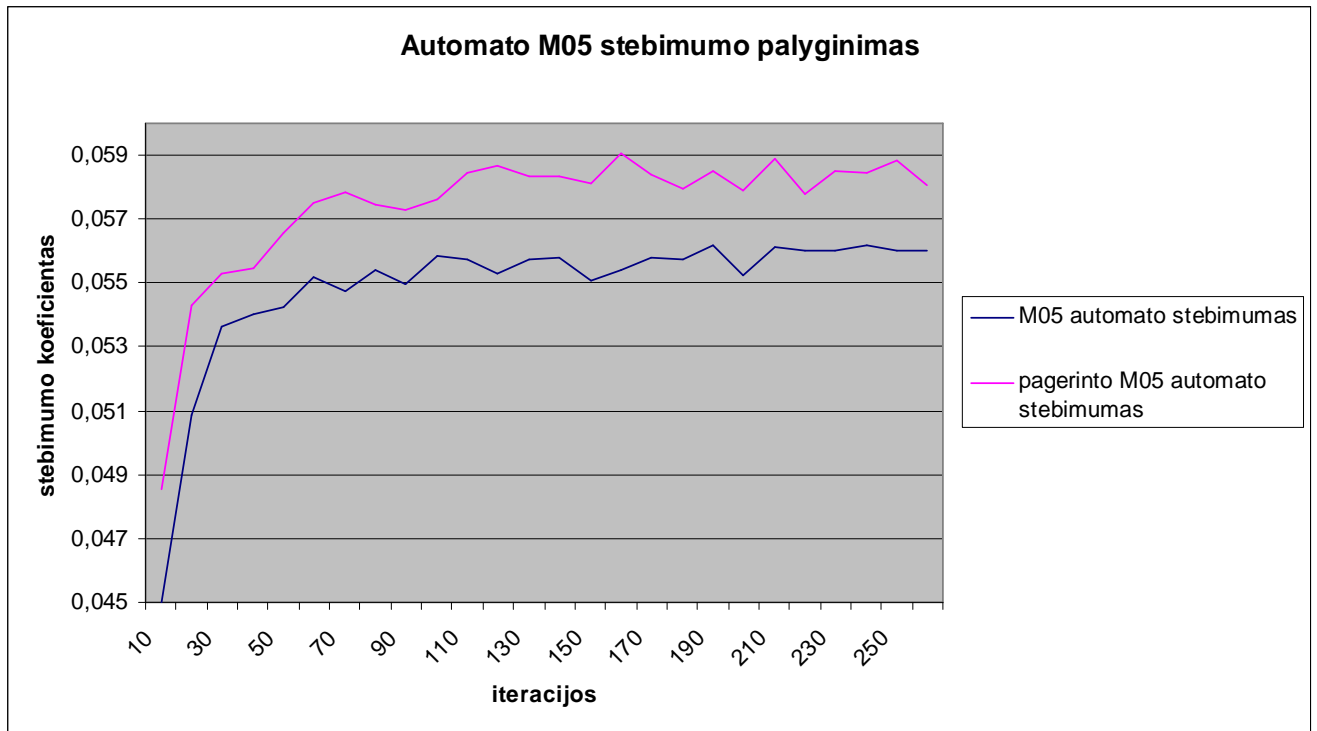
#### 4.3.2.5 pav M05 pagerinto automato tikslumo priklausomybė nuo iteracijų skaičiaus

Iš pateikto grafiko matome kada įvyko skaičiavimų nutraukimas. Skaičiavimai buvo nutraukti, kai buvo pasiektas nusistovėjimas prie reikšmės 0,2% 50 kartų.

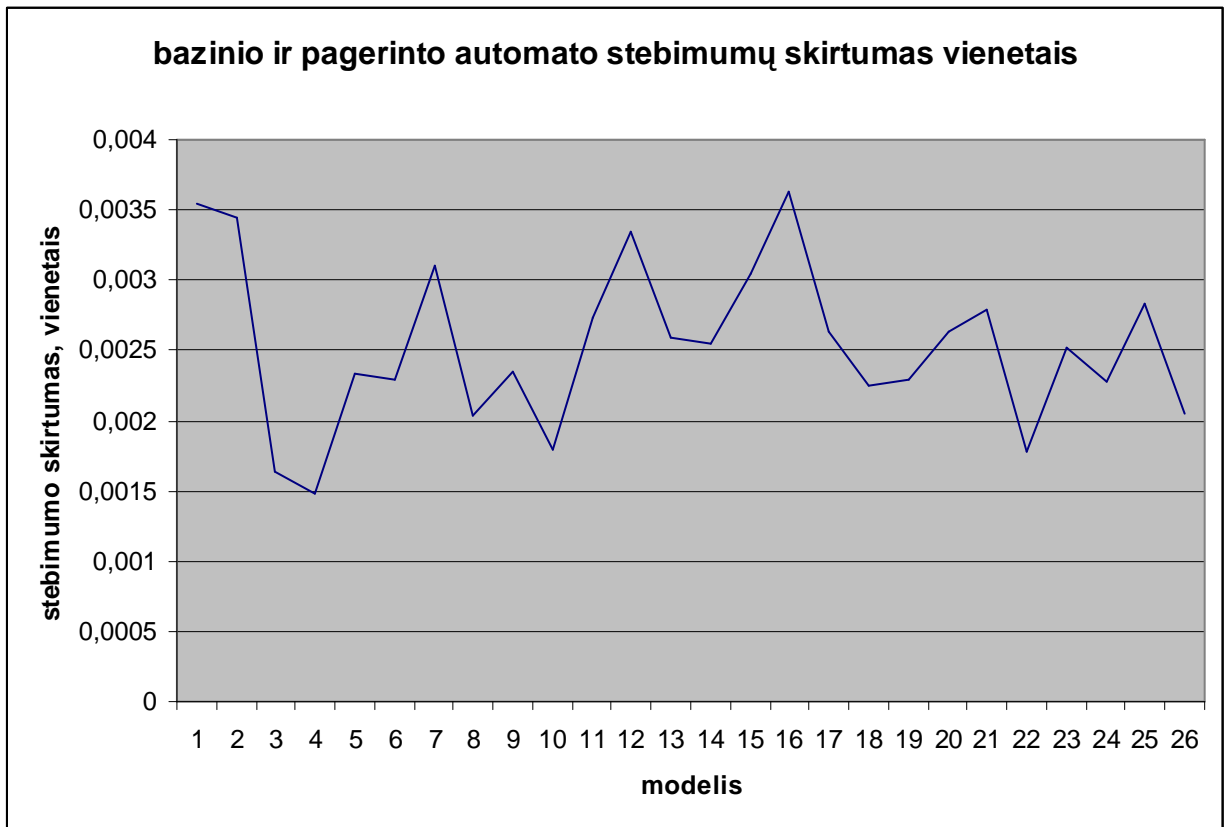


**4.3.2.6 pav M05 pagerinto automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė**

Iš šio grafiko matome, kaip modelio vykdymo metu keitėsi pakartojimų ir iteracijų skaičius. Tiriame šį grafiką galima padaryti išvadas, kad kuo daugiau kartų įvykdome iteracinio modelio atkartojimą, tuo mažiau iteracijų reikia nusistovėjimo nustatymui. Bet šiam automatui tai turi daugiau reikšmės prie didelio iteracijų skaičiaus bet mažo atkartojimų skaičiaus. Vėliau ši priklausomybė tampa beveik tiesinė.

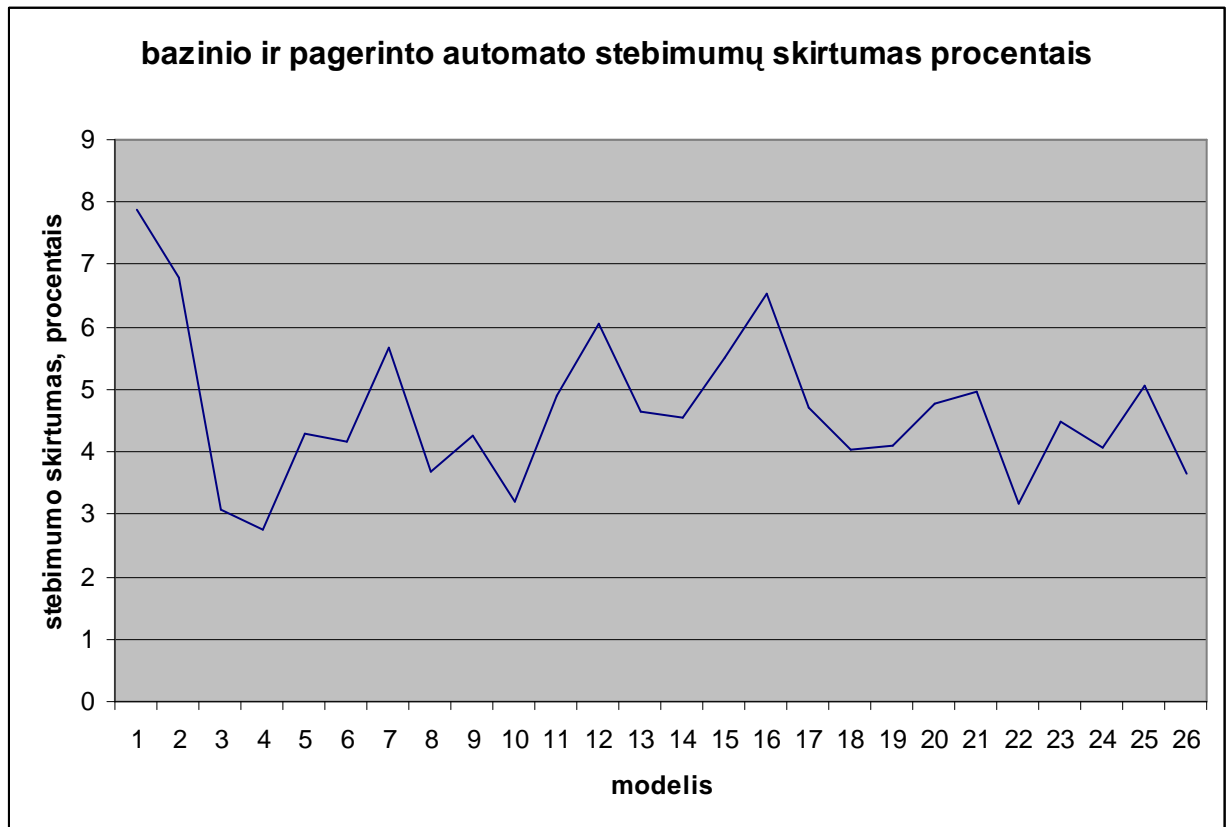
Bazinio ir pagerinto modelio palyginimas**4.3.2.7 pzv M05 automato bazinio ir pagerinto modelio stebimumo palyginimas**

Pateiktame grafike galima matyti, kiek padidėjo automato stebimumas po stebimumo pagerinimo. Vidutiniškai per visus modelius jis padidėjo 0,002538 vienetais, arba 4,655 % .



**4.3.2.8 pav M05 automato bazinio ir pagerinto automatų stebimumo skirtumas vienetais**

Pateiktame grafike atvaizduota kiek vienetų padidėjo automato stebimumas vienetais po kiekvieno modelio įvykdymo.



#### 4.3.2.9 pav Bazinio ir pagerinto automatų stebimumo skirtumas vienetais

Pateiktame grafike atvaizduota kiek vienetų padidėjo automato stebimumas vienetais po kiekvieno modelio įvykdymo.

##### M05 automato tyrimo apibendrinimas

Panaudojus sukurta programinę įrangą pavyko nustatyti automato stebimumą, bei atlikti automato stebimumo pagerinimą.

Automato stebimumas pagerėjo 0,002538 vienetais, arba 4,655 %.

Bazinio automato stebimumo maksimumas yra 0,056196 ir jis buvo pasiektas sudarius iteracinį modelį iš 190 automato kopijų(iteracijų). Iteracijų prisistotinimas įvyko vykdant modelį su 260 iteracijų.

Pagerinto automato stebimumo maksimumas yra 0,060048 ir jis buvo pasiektas sudarius iteracinį modelį iš 190 automato kopijų(iteracijų). Iteracijų prisistotinimas įvyko vykdant modelį su 5120 iteracijų.

Iš to galima daryti išvada, kad šiam automatui pasiekti maksimalų stebimumo reikšmę bei pasiekti prisistotinimo būseną reikia atlikti daugiau iteracijų.

Iš iteracijų ir atkartojimų grafiko galima padaryti išvadą, kad kai yra mažas atkartojimų skaičius, reikalingas didelis iteracijų skaičius, bet didinant atkartojimų skaičių mažėja reikalingų iteracijų skaičius.

M03 automato tyrimo apibendrinimas

Panaudojus sukurta programinę įrangą pavyko nustatyti automato stebimumą, bei atlikti automato stebimumo pagerinimą.

Automato stebimumas pagerėjo 0,00240048 vienetais, arba 9,114 procentais.

70 iteracijų yra 0,026872

Bazinio automato stebimumo maksimumas yra 0,026872 ir jis buvo pasiektas sudarius iteracinį modelį iš 70 automato kopijų(iteracijų). Iteracijų prisisotinimas įvyko vykdant modelį su 880 iteracijų.

Pagerinto automato stebimumo maksimumas yra 0,029903 ir jis buvo pasiektas sudarius iteracinį modelį iš 190 automato kopijų(iteracijų). Iteracijų prisisotinimas įvyko vykdant modelį su 1020 iteracijų.

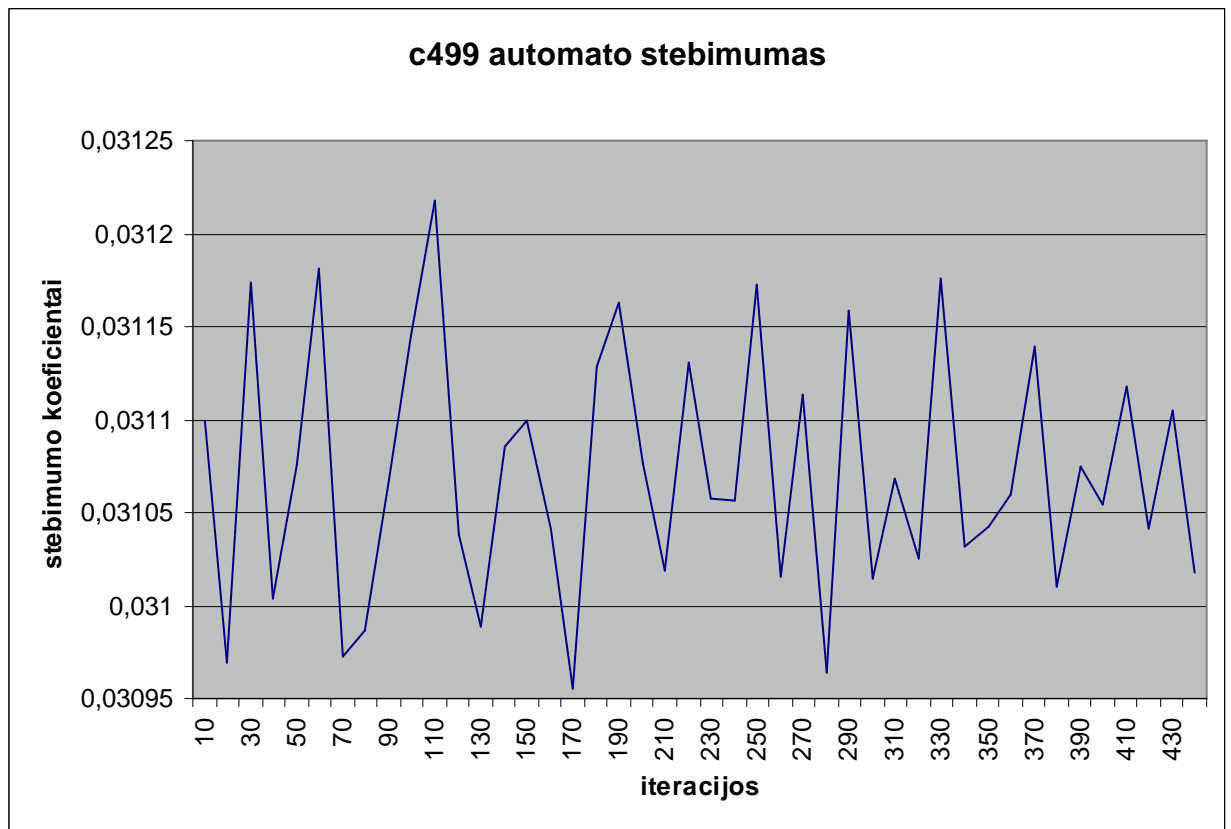
Iš to galima daryti išvada, kad šiam automatui pasiekti maksimalų stebimumo reikšmę bei pasiekti prisisotinimo būseną reikia atlikti daugiau iteracijų.

Iš iteracijų ir atkartojimų grafiko galima padaryti išvadą, kad kai yra mažas atkartojimų skaičius, reikalingas didelis iteracijų skaičius, bet didinant atkartojimų skaičių mažėja reikalingų iteracijų skaičius.



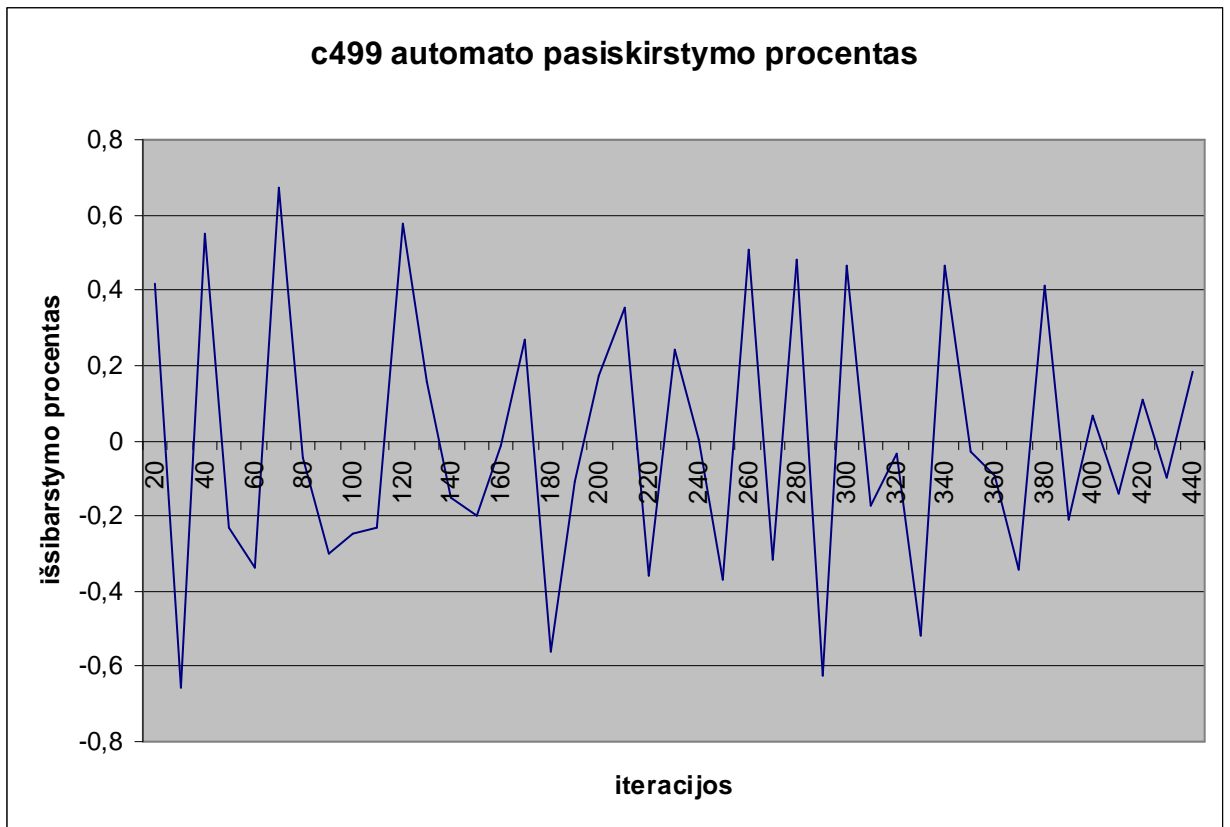
### 4.3.3. Automato c499 tyrimas

#### Bazinio automato c499(automatas be būsenų) modelio tyrimas



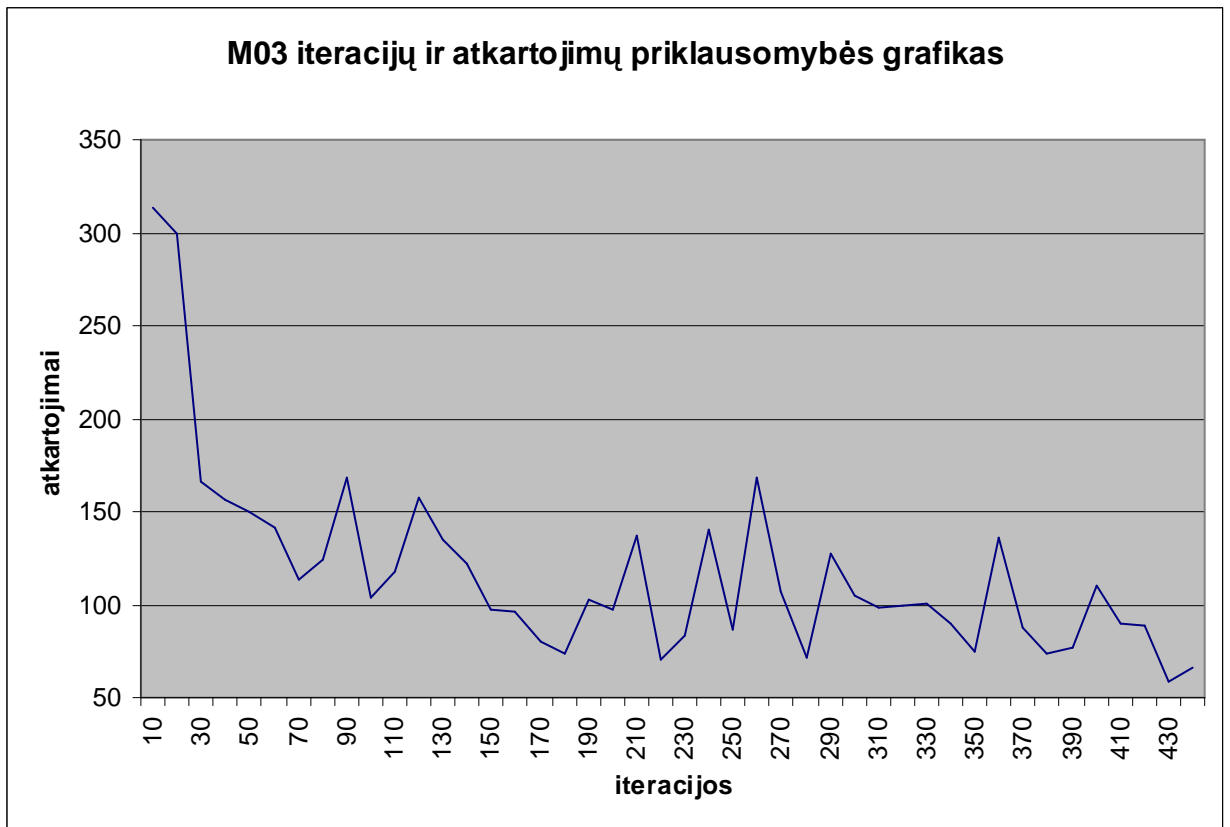
#### 4.3.3.1 pav c499 automato Stebimumo priklausomybė nuo iteracijų skaičiaus

Iš gautų tyrimo rezultatų galima pamatyti, kad automato maksimalus stebimumas buvo pasiektas sudarius iteracinį modelį iš 110 iteracijų. Stebimumo reikšmė prie 110 iteracijų yra 0,031218.



**4.3.3.2 pav c499 automato tikslumo priklausomybė nuo iteracijų skaičiaus**

Iš pateikto grafiko matome kada įvyko skaičiavimų nutraukimas. Skaičiavimai buvo nutraukti, kai buvo pasiektas nusistovėjimas prie reikšmės 0,2% 50 kartų.



#### 4.3.3 3 pav c499 automato pakartojimų ir iteracijų skaičiaus tarpusavio priklausomybė

Iš šio grafiko matome, kaip modelio vykdymo metu keitėsi pakartojimų ir iteracijų skaičius. Tiriant šį grafiką galima padaryti išvadas, kad kuo daugiau kartų įvykdome iteracinio modelio atkartojimą, tuo mažiau iteracijų reikia nusistovėjimo nustatymui.

#### C499 automato tyrimo apibendrinimas

Panaudojus sukurta programinę įrangą pavyko nustatyti automato stebimumą, bei nepavyko atlikti stebimumo pagerinimo, nes šis automatas neturi būsenų, todėl nėra įmanomą sudaryti iteracinio modelio pagal, reikalingo mūsų metodui.

Nustatytas automato stebimumo maksimumas yra 0,031218 ir jis buvo pasiektas sudarius iteracinį modelį iš 110 automato kopijų(iteracijų). Iteracijų prisisotinimas įvyko vykdant modelį su 440 iteracijų.

Iš iteracijų ir atkartojimų grafiko galima padaryti išvadą, kad kai yra mažas atkartojimų skaičius, reikalingas didelis iteracijų skaičius, bet didinant atkartojimų skaičių mažėja reikalingų iteracijų skaičius.

## 5. IŠVADOS

1. Pristatyta testuojamumo problema, išanalizuoti įvairūs stebimumo nustatymo metodai, pristatyta taikymo sritis, apibrėžta automato stebimumo sąvoka, bei automatų imitavimo priežastys. Pristatyti stebimumo padidinimo metodai.

2. „Baltų dėžių“ valdomumo nustatymo ir padidinimo metodai adaptuoti „juodoms dėžėms“. Sukurtas algoritmas atlikti tai programiškai.

3. Sukurta PĮ įvertinanti automatų stebimumą, bei pateikianti pasiūlymus kaip galima padidinti stebimumą.

4. PĮ buvo dokumentuojama kuriant reikalavimų, architektūros, detalios architektūros specifikacijas ir vartotojo vadovą.

5. Tyrimo prielaidos, dėl pakartojimų ir iteracijų skaičiaus nustatymo, bei pakartojimų mažėjimo didėjant iteracijų skaičiui, buvo patvirtintos eksperimentų metu.

6. Stebimumo pagerinimo funkcija pagerina automato valdomumą apie nuo 6 iki 10% priklausomai nuo automato modelio.

7. Dirbant su PĮ reikia įvertinti iteracijų ir atkartojimų skaičių. Tyrimai parodė, kad vykdant darbą su mažu atkartojimų skaičiumi yra pageidautinas didesnis iteracijų skaičius. Didinant atkartojimų skaičių galima mažinti iteracijų skaičių.

## 6. LITERATŪRA

- [1] Stephenson J. E., Grason J. *A testability measure for register transfer level digital circuits. Proc. Int. Symp. Fault tolerant Computing (FTCS)*, (Pittsburgh, PA), pp. 101-107, June 1976.
- [2] Grason J. *TMEAS-a testability measurement program. Proc. IEEE/ACM Design Automation Conf. (DAC)*, vol. 26, no. 9, pp. 156-161, 1979.
- [3] Goldstein L. H. *Controllability/observability analysis for digital circuits, IEEE Trans. Circuits and Systems*. vol. 26, no. 9, pp. 685-693, Sept. 1979.
- [4] Kovijanic P. G. *Testability analysis, in Proc. IEEE Semiconductor Test Conf.* pp. 310-316, 1979.
- [5] Kovijanic P. G. *Computer-aided testability analysis. in Proc. IEEE Autotestcon*, pp. 292-294, 1979.
- [6] Bennetts R. G., Maunder C. M., Robinson G. D. *CAMELOT:a computer-aided measure for logic testability. IEE Proc. Pt. E*, vol. 128, no. 5, pp. 177-189, 1981.
- [7] Bennetts R. G. *Design of Testable logic Circuits*. Reading, MA: Addison-Wesley, 1984.
- [8] Ratiu I. M., Sangiovanni-Vincentelli A., Pederson D. O. *VICTOR: a fast VLSI testability analysis program. Proc. Int. Test Conf. (ITC)*, (Philadelphia, PA), pp. 397-401, Nov. 1982.
- [9] Berg W. C., Hess R. C. *COMET : a testability analysis and design modification package. Proc. Int. Test Conf. (ITC)*, (Philadelphia, PA), pp. 364-378, Nov. 1982.
- [10] Gustavo R. C. Alves *Design for Test, Faculdade de Engenharia da Universidade do Porto / ISEP*, [žiūrėta 2004-11-12], prieiga per internetą: <[http://www.fe.up.pt/~allegro/wp3/d3x\\_wbm/MOD1/ControllabilityObservabilityTestability.html](http://www.fe.up.pt/~allegro/wp3/d3x_wbm/MOD1/ControllabilityObservabilityTestability.html)>
- [11] Jusas V., Šeinauskas R., Paulikas K. *[Testinių rinkinių generavimas aukštajame abstrakcijos lygyje]. Konferencijos pranešimų medžiaga*. Informacinės Technologijos, 2002, p.151-155.
- [12] *VLSI Testing and DTF Course*. [žiūrėta 2004-11-09], prieiga per internetą: <[larc.ee.nthu.edu.tw/~cww/n/625/6250/03.pdf](http://larc.ee.nthu.edu.tw/~cww/n/625/6250/03.pdf)>
- [13] Timothy Brian Wheeler, *Improving design observability and controllability for functional verification of FPGA-based circuits using design-level scan techniques*, [žiūrėta 2003-11-12], prieiga per internetą: <<http://splash.ee.byu.edu/docs/thesis-wheeler.pdf>>
- [14] Mani Somna, *Test generation*. [žiūrėta 2004-11-10], prieiga per internetą: <<http://larc.ee.nthu.edu.tw/~cww/n/625/6250/03.pdf>>
- [15] *Testability Measure*. [žiūrėta 2003-11-05], prieiga per internetą: <[www.te.tku.edu.tw/~chiang/courses/testing/Testing-2/tm.ppt](http://www.te.tku.edu.tw/~chiang/courses/testing/Testing-2/tm.ppt)>

- [16] Afonin A., Survila V. [Abstrakčių automatų stebimumo ir valdomumo lygio nustatymas] *Informacinė visuomenė ir universitetinės studijos. 9-ji magistrantų ir doktorantų konferencija*. VDU, 2004, p. 307-311
- [17] Abramovici M., Breuer M.A., Friedman A.D., *Digital Systems Testing and Testable Designs*. 1998.
- [18] Mourad S., Zorian Y. *Principles of Testing Electronic Systems*. John Wiley & Sons., Inc., New York, 2000
- [19] John L. Hennessy, David A. Patterson. *Computer Organization and Design*. San Francisco, California, 1997
- [20] Rajski J., Tyszer J. *Arithmetic Built-In Self-Test for embedded systems*. New Jersey, 1998
- [21] Bushnell M., Agrawal V. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Kluwer Academic Publishers. Norwell, USA, 2000
- [22] Ubar R., Orasson E., Wuttke H.-D. *Interactive Teaching Software "Introduction To Digital Test"*. 45th International Conference, Ilmenau, Germany, 2000, p.949-954, , [žiūrėta 2005-03-12], prieiga per Internetą: <[http://www.pld.ttu.ee/dildis/publications/iwk\\_2000.pdf](http://www.pld.ttu.ee/dildis/publications/iwk_2000.pdf)>
- [23] Ubar R., Wuttke H.-D. *Action Based Learning System for Teaching Digital Electronics and Test. Proc. of 3rd European Workshop on Microelectronics Education, Aix-en-Provence, France, 2000, p.65-66, [žiūrėta 2005-04-18], prieiga per internetą: <[http://www.pld.ttu.ee/dildis/publications/ewme\\_2000.pdf](http://www.pld.ttu.ee/dildis/publications/ewme_2000.pdf)>*
- [24] Ubar R., Wuttke H.-D. *The DILDIS-Project - Using Applets for More Demonstrative Lectures in Digital Systems Design and Test. Abstracts of 31st ASEE/IEEE Frontiers in Education Conference, Reno, NV, USA, 2001, p.83, [žiūrėta 2005-05-18], prieiga per internetą: <[http://www.pld.ttu.ee/dildis/publications/fie\\_2001.pdf](http://www.pld.ttu.ee/dildis/publications/fie_2001.pdf)>*
- [25] Ubar R., Orasson E., Evarson T. *Java Applet for Self-Learning of Digital Test Issues*. 13th EAEEIE Conference, York, Great Britain, 2002, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/york1.pdf>>
- [26] Ubar R., Orasson E., Evarson T. *Self-learning tool for digital test. Proceedings of 2nd Int. Conf. "Distance learning - educational sphere of the XXI century", Minsk, Belarus, 2002, p.36-38, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/minsk.pdf>>*
- [27] Jutman A., Kruus M., Sudnitson A., Ubar R. *Distance-Learning Tools for Digital Design and Test Issues,*" in Proc. 29th International Conference and Scientific Discussion Club Information Technologies in Science, Education, Telecommunications, Business (IT+SE'2002), Yalta-Gurzuf, Ukraine, May 20-30, 2002, pp. 269-272, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/IT%2BSE'2002.pdf>>

[28] Devadze S., Jutman A., Sudnitson A., Ubar R. *Web-based training system for teaching basics of RT-level Digital Design, Test, and Design for Test*. Proc. of 9th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES 2002), Wroclaw, Poland, June 20-22, 2002, pp. 699-704, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/RTapplet%20MIXDES'02.pdf>>

[29] Devadze S., Jutman A., Kruus M., Sudnitson A., Ubar R. *Web Based Tools for Synthesis and Testing of Digital Device*. Proc. International Conference on Computer Systems and Technologies (CompSysTech'2002), Sofia, Bulgaria, June 20-21, 2002, pp. I.91-I.96. (ISBN 954-9641-28-7), [žiūrėta 2005-05-10], prieiga per internetą: <[http://www.pld.ttu.ee/decomposition/publications/Sudnitson\\_Estonia\\_CST\\_paper.PDF](http://www.pld.ttu.ee/decomposition/publications/Sudnitson_Estonia_CST_paper.PDF)>

[30] Devadze S., Jutman A., Sudnitson A., Ubar R., Wuttke H-D. *Java Technology Based Training System for Teaching Digital Design and Test*. Proc. of 8th Baltic Electronics Conference (BEC 2002), Tallinn, Estonia, October 6-9, 2002, pp. 283-286, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/RTapplet%20-%20BEC'02.pdf>>

[31] Devadze S., Jutman A., Sudnitson A., Ubar R., Wuttke H-D. *Teaching Digital RT-Level Self-Test using a Java Applet*. 20th IEEE Conference NORCHIP'2002, Copenhagen, Denmark, November 11-12, 2002, pp.322-328, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/RTapplet%20norchip'02.pdf>>

[32] Ubar R., Orasson E. *E-Learning tool and Exercises for Teaching Digital Test*. Proc. of 2nd IEEE Conf. on Signals, Systems, Decision and Information Technology, Sousse, Tunisia, 2003, CIT-6, p.1-6, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/E-Learning%20tool%20and%20Exercices.pdf>>

[33] Jutman A., Sudnitson A., Ubar R. *Web-Based Training System for Teaching Principles of Boundary Scan Technique.*, Proc. 14th EAEEIE Conference on Innovation in Education for Electrical and Information Engineering, Gdansk, Poland, June 16-18, 2003, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/BSapplet-EAEEIE'03-final.pdf>>

[34] Jutman A., Sudnitson A., Ubar R. *Web-based Applet for Teaching Boundary Scan Standard IEEE 1149.1*, Proc. of 10th International Conference on Mixed Design of Integrated Circuits and Systems. (MIXDES'03), Lodz, Poland, June 26-28, 2003, pp. 584-589, [žiūrėta 2005-05-10], prieiga per Internetą: <<http://www.pld.ttu.ee/dildis/publications/BS-applet-MIXDES-2003.pdf>>

[35] A. Jutman, A. Sudnitson, Ubar R. *“Digital Design Learning System Based on Java Applets*. Proc. 4th Annual Conference of the LTSN Centre for Information and Computer Sciences, NUI Galway, Ireland, August 26-28, 2003, pp.183-187, [žiūrėta 2005-05-10], prieiga per Internetą: <<http://www.pld.ttu.ee/dildis/publications/Sudnitson%20LTSN-ICS%202003.pdf>>

[36] M.Aarna, E.Ivask, A.Jutman, E.Orasson, J.Raik, R.Ubar, V.Vislogubov, Wuttke H.-D. “*Turbo Tester - Diagnostic Package for Research and Training*”. Scientific-Technical Journal “Radioelectronics & Informatics”. KNURE. Vol. 3(24), 2003, pp.69-73, [žiūrėta 2005-05-10], prieiga per internetą: <<http://www.pld.ttu.ee/dildis/publications/TurboTester-EWDTC03.pdf>>

[37] Kusiak Andrew, Chung-Che Huang. *Design of Modular Digital Circuits for Testability*. [žiūrėta 2005-05-15], prieiga per internetą: <<http://www.icaen.uiowa.edu/~ankusiak/Journal-papers/Huang.pdf>>

[38] Hays J. P. *On modification logic networks to improve their diagnostability*. IEEE Trans. Comput., vol. C-23, pp. 56–62, 1974.

[39] Hays J. P., Friedman A. D. *Test point placement to simplify fault detection*. IEEE Trans. Comput., vol. C-23, pp. 727–735, 1974.

[40] Steward D. V. *Partition and testing systems of equations*. J. Soc. Ind. Appl. Math.: Numer. Anal., vol. 2, no. 2, pp. 345–365, 1965.

[41] Goldstein L. M., Thigen E. L. *SCOAP: Sandia controllability/observability analysis program*. Proc. 17th Design Automation, Conf., 1980, pp. 190–196.

[42] Chickermane V., Rudnick E. M., Banerjee P., Patel J. H. *Nonscan design-for-testability techniques for sequential circuits*. Proc. 30th Design Automation Conf., 1993, pp. 236–241.

[43] Goldstein L. H. *Controllability/observability analysis of digital circuits*. IEEE Trans. Circuits Syst., vol. CAS-26, pp. 685–693, 1979.

[44] K. Kim, D. S. Ha, and J. G. Tront, “*On using signature registers as pseudorandom pattern generators in built-in self-testing*,” IEEE Trans. Computer-Aided Design, vol. 7, pp. 919–928, 1988.

[45] Roth J. P., Bouricius, Schneider P. R. *Programmable algorithm to compute tests to detect distinguish between failures in logic circuits*. IEEE Trans. Electron. Comput., vol. EC-16, pp. 567–579, 1967.

[46] Lechner A., Richardson A., Hermes B. *Reconfiguration Based Built-In Self-Test for Analogue Front End Circuits*. [žiūrėta 2005-05-15], prieiga per internetą: <[http://www.comp.lancs.ac.uk/engineering/research/microsystems/publications/date99\\_al.pdf](http://www.comp.lancs.ac.uk/engineering/research/microsystems/publications/date99_al.pdf)>

[47] Lancaster University research group, *Mixed Signal Design-for-Testability*. [žiūrėta 2005-05-15], prieiga per internetą:

<<http://www.comp.lancs.ac.uk/engineering/research/microsystems/tutorials/msdft.html>>

[48] Lancaster University research group, *Integration of Design-for-Test (DfT) and Built-In Self-Test (BIST) into the Design Cycle*. [žiūrėta 2005-05-15], prieiga per internetą: <<http://www.comp.lancs.ac.uk/engineering/research/microsystems/tutorials/msdft.html>>



[49] Vaucher Christophe, *DFT: Design for Testability*. [žiūrėta 2005-05-15], prieiga per internetą:

<[http://www.circuitree.com/CDA/ArticleInformation/features/BNP\\_\\_Features\\_\\_Item/0,2133,15087,00.html](http://www.circuitree.com/CDA/ArticleInformation/features/BNP__Features__Item/0,2133,15087,00.html)>

[50] Olausson Mikael, Wiklund Daniel. *Introduction to Design for Testability*. [žiūrėta 2005-05-15], prieiga per internetą: <<http://www.ida.liu.se/~zebpe/teaching/test/lec6.pdf>>

[51] Salahuddin A. Almajdoub. *A Design Methodology for Physical Design for Testability*. [žiūrėta 2005-05-15], prieiga per internetą: <<http://scholar.lib.vt.edu/theses/available/etd-522014589642481/unrestricted/almajdoub.pdf>>

[52] Zebo Peng. *Design and Synthesis for Test*. [žiūrėta 2005-05-15], prieiga per internetą: <<http://ece.concordia.ca/~tahar/coen7501/notes/testing-notes.pdf>>

## 7. SANTRUMPŲ IR TERMINŲ ŽODYNAS

**CGI-** (Common Gateway Interface) kliento serverio interfeisas

**PHP-** proramavimo kalba skirta dinaminiam interneto puslapių kūrimui

**HTML-** (Hypertext markup language) interneto puslapių kūrimo kalba

**FTP-** (File Transfer Protocol) Failų perdavimo protokolas

**PDF-** (Portable Document Format) dokumento formatas skirtas įvairiom sistemom

**gcc-** UNIX sistemos C\C++ kalbos kompiliatorius

**Serveris** (darbo stotis)- pagrindinis tinklo kompiuteris (tarnybinė stotis), kuriame patalpinta programinė įranga.

**Susietumų matrica-** matrica charakterizuojanti automato išėjimų-būsenų priklausomybę nuo įėjimų-būsenų.

**Matricos charakteristikos-** tai yra matricos reikšmių kiekiai, nurodantys kaip išėjimai-būsenos priklauso nuo įėjimų-būsenų.

**DFT** – Testuojamumo projektavimas (Design for testability)

**8. PRIEDAI**