



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

MATEMATINIO MODELIAVIMO KATEDRA

Mikalojus Ramanauskas

STAČIAKAMPIO LAŠTELINIO AUTOMATO BRANDUOLIO STRUKTŪRA
KERNEL STRUCTURE OF RECTANGULAR CELLULAR AUTOMATON

Baigiamasis magistro darbas

Technomatematikos studijų programa, valstybinis kodas **612G16001**

Matematinio modeliavimo specializacija

Matematikos studijų kryptis

Vilnius, 2015

Vilniaus Gedimino technikos universitetas Fundamentinių mokslų fakultetas Matematinio modeliavimo katedra		ISBN ISSN Egz. sk. Data-.....-.....	
Antrosios pakopos studijų Technomatematikos programos magistro baigiamasis darbas Pavadinimas Stačiakampio ląstelinio automato branduolio struktūra Autorius Mikalojus Ramanauskas Vadovas prof. dr. Aleksandras Krylovas			
Kalba: lietuvių			
Anotacija Baigiamajame magistro darbe analizuojamas tam tikras ląstelinis automatas. Matematinėje literatūroje dominuoja tie automatai, kurių ląstelės įgyja reikšmes iš liekanų klasių pirminių modulių p lauko. Šio darbo tikslas yra tirti stačiakampius tinklinius ląstelinus automatus, kurių ląstelių būsenos yra realieji skaičiai. Taikant studijų metu įgytas žinias, buvo sudarytas automato matematinis modelis, atlikta sprendinių analizė, suformuluotos bei įrodytos teoremos apie sprendinių egzistavimą bei struktūrą. Darbą sudaro 7 dalys: įvadas, ląsteliniai automatai, tinklinio ląstelinio automato matematinis modelis, spartesnis algoritmas, branduolio struktūra, išvados ir literatūros sąrašas. Darbo apimtis - 30 p. teksto be priedų, 17 iliustr., 4 lent., 24 bibliografiniai šaltiniai. Atskirai pridedami darbo priedai.			
Prasminiai žodžiai: ląstelinis automatas, konfigūracija, perėjimo operatorius, branduolys			

Vilnius Gediminas Technical University Faculty of Fundamental Sciences Department of Mathematical Modelling	ISBN ISSN Copies No. Date
Master Degree Studies Technomathematics study programme Master Graduation Thesis Title Kernel Structure of Rectangular Cellular Automaton Author Mikalojus Ramanauskas Academic supervisor Prof Dr Aleksandras Krylovas	
Thesis language: Lithuanian	
Annotation The thesis analyzes certain cellular automaton. Mathematical literature is dominated by these machines, which cells acquire the values of the remains of the original classes module p field. The aim of this is to investigate the cellular rectangular mesh machines whose cell states are real numbers. Applying the knowledge acquired during their studies, it was concluded machine mathematical model, carried out an analysis of the solutions are formulated and proved theorems about the solutions of the existence and structure. The work consists of 7 parts: introduction, cellular automata, volleyball cellular automaton mathematical model, a faster algorithm, kernel structure, conclusions and references. Work size - 30 p. text without extras, 17 pictures., 4 tables., 24 bibliographical sources. Appendixes.	
Keywords: cellular automaton, configuration, transition operator, kernel	

TURINYS

Iliustracijų ir lentelių sąrašas	8
Įvadas	9
1 Ląsteliniai automatai	10
1.1 Ląstelių automatų istorija	10
1.2 Ląstelinio automato sąvoka	11
1.3 Pagrindiniai ląstelių automatų tipai	12
1.4 Žinomiausi ląsteliniai automatai	15
1.4.1 Žaidimas „Gyvenimas“	15
1.4.2 Žaidimas „Užgesink šviesas“	16
2 Tinklinio ląstelinio automato matematinis modelis	19
2.1 Matematinio modelio sudarymas	19
2.2 Nedidelių lentelių tyrimas	21
3 Spartesnis algoritmas	22
3.1 Čebyšovo daugianariai	23
3.2 Matricos C determinanto analizė	26
4 Branduolio struktūra	31
5 Išvados	35
Literatūros sąrašas	36
Priedas A	37
Priedas B	38
Priedas C	39
Priedas D	39

Iliustracijų sąrašas

1 pav.	Ląstelinio automato konfigūracijos keitimas.	12
2 pav.	Tinklinio grafo vizualizavimas lentele.	12
3 pav.	Toras.	13
4 pav.	Uždaras ląstelinis automatas iš šešiakampių ir penkiakampių.	14
5 pav.	Spalvoto ląstelinio automato konfigūracija.	14
6 pav.	„V“ formos konfigūracijos ciklai.	15
7 pav.	Pirmieji „mažosios r“ formos konfigūracijos ciklai.	16
8 pav.	Žaidimas Lights Out.	17
9 pav.	Žaidimo Lights Out principas.	17
10 pav.	Pirma lentelės gesinimo stadija.	19
11 pav.	Antra lentelės gesinimo stadija.	19
12 pav.	Pertvarkyta matrica C	24
13 pav.	Trig. diof. lygties sprendimas su „WolframAlpha“.	30
14 pav.	Gardelės 9x14 branduolio išraiška.	31
15 pav.	Gardelės 9x14 kanoninė branduolių erdvės bazė.	31

Lentelių sąrašas

1 lent.	Užgesinimo kombinacijos.	18
2 lent.	Branduolių dimensijos.	33

Ivadas

Gamtoje vyksta daugybė įvairiausių procesų, kurie daugeliu atvejų aprašomi ir analizuojami matematiniais modeliais: diferencialinėmis lygtimis, tikimybiniais metodais, diskrečiosiomis struktūromis bei kitais žinomais metodais. Tarp daugelio matematinių modelių yra žinomi ir aktyviai tiriami ląsteliniai automatai. Matematinėje literatūroje dominuoja tie automatai, kurių ląstelės įgyja reikšmes iš liekanų klasių pirminiu modulių p lauko. Šio darbo tikslas yra tirti stačiakampius tinklinius ląstelinius automatus, kurių ląstelių būsenos yra realieji skaičiai. Šiuo darbu tęsiame bakalauro darbe pradėtą kvadratinų tinklinių automatų analizę.

Teorinėje dalyje daugiausia dėmesio skiriama susipažinimui su ląsteliniais automatais. Pirmiausia trumpai pristatoma jų atsiradimo istorija, pateikiamas ląstelių automatų sąvokos apibrėžimas, taikymo sritys bei žinomiausių automatų pavyzdžiai.

Praktinėje darbo dalyje, sudarant įvairius algoritmus, analizuojami automatai virš nulinės charakteristikos lauko, nors mokslinėje literatūroje dominuoja tyrimai virš baigtinių laukų. Atliekama stačiakampio ląstelinio automato apgrežiamumo analizė, o neapgrežiamais atvejais tiriama ląstelinio automato branduolio struktūra.

1. Ląsteliniai automatai

1.1. Ląstelių automatų istorija

Džonas von Neimanas (John von Neumann), dirbdamas JAV Los Alamos nacionalinėje laboratorijoje 1940 metais, tyrinėjo savarankiškai atsinaujinančių (angl: self reproducing) sistemų problemas. Tuo pačiu metu, Stanislavas Ulamas (Stanislaw Ulam) JAV Los Alamoje tyrinėjo kristalų augimą. Modeliams sudaryti jie naudojo tinklinį grafą ir pirmieji panaudojo ląstelinio automato sąvoką [24]. Tačiau jų darbai buvo itin akademinio pobūdžio, todėl didelio susidomėjimo ląsteliniiais automatais nekilo.

Viskas radikaliai pasikeitė 1970 metais, kai Martynas Gardneris (Martin Gardner) žurnale „Scientific American“ [8] paskelbė straipsnį apie dabar jau visame pasaulyje žinomą, garsiojo J. Conway sukurtą žaidimą „Gyvenimas“ (angl. Game of Life) [25]. Žaidimo taisyklės yra gana paprastos. Kiekviena ląstelė gali turėti tik dvi būsenas – aktyvuotą arba neaktyvuotą. Kiekvienos ląstelės nauja būsena priklauso nuo 8 jos kaimynių ankstesnių būsenų. Įvairiai parinkus pradines būsenas buvo tiriama, kaip kinta būsenos iš kartos į kartą.

Klausas Satneris (Klaus Sutner) taip pat labai aktyviai tiria ląstelinius automatus. Jis išsamiai nagrinėjo automatus, kurių principu buvo sukonstruoti „Lights out“ tipo žaidimai. Jis įrodė, kad jei būsenos įgyja reikšmes iš liekanų klasių modulių 2, tuomet bet kokių matmenų stačiakampė gardelė, kurios visų ląstelių būsenos yra lygios 1, turi pirmavaizdį, tai yra užgesinama [19]. Taip pat jis įrodė teorema, kuri, panaudojant Čebyševio daugianarius, leidžia nustatyti ląstelinio automato branduolio dimensiją [22].

Laikui bėgant buvo sukurta ir aktyviai tiriama daug kitų ląstelių automatų. Kiekvienas automatas turėdavo vis kitokias galimas būsenas ir jų keitimo operatorius. Tokiu būdu ląsteliniai automatai suteikdavo galimybes modeliuoti procesus, kuriuose esamos ląstelės būsena priklauso nuo kaimyninių ląstelių būsenų. Mums gerai žinomas pavyzdys yra temperatūros pasiskirstymo uždavinys, kur kiekviename vidiniame langelyje esanti temperatūra yra lygi kaimyninių langelių temperatūrų vidurkiui.

1.2. Ląstelinio automato sąvoka

Suformuluosime Klausio Satnerio (Klaus Sutner) pateiktą adityvaus ląstelinio automato apibrėžimą [21]. Tegul $G = (V, E)$ yra neorientuotas baigtinis grafas, kur V – šio grafo viršūnių aibė, o E – grafo briaunų aibė. Grafo viršūnių būsenų aibė F yra aibė su komutatyvia operacija ir neutraliuoju elementu, tai yra Abelio monoidas. Aibė F yra vadinama ląstelinio automato alfabetu.

Apibrėžimas. Grafo $G = (V, E)$ konfigūracija yra vadinama funkcija $X: V \rightarrow F$.

Paprastai sakant, konfigūracija kiekvienai grafo viršūnei priskiria elementą iš monoido F . Šiame darbe tiriamie ląstelinius automatus, kurių grafo viršūnės $v \in V$ aplinka $N(v)$ yra aibė, kurią sudaro pati viršūnė v ir grafo viršūnės, kurios turi briauną su viršūne v . Tai yra

$$N(v) := \{u \in V : (u, v) \in E\} \cup \{v\}$$

Kitaip tariant, viršūnės v aplinką sudaro visos kaimyninės viršūnės (viršūnės turinčios briauną su v) bei pati viršūnė v . Grafo G konfigūracijų aibę žymėsime C_G , kuri, jei F yra laukas, yra tiesinė erdvė ir jos dimensija lygi grafo viršūnių skaičiui.

K. Satneris apibrėžė perėjimo nuo vienos konfigūracijos prie kitos operatorių taisyklę:

$$\sigma_G: C_G \rightarrow C_G,$$

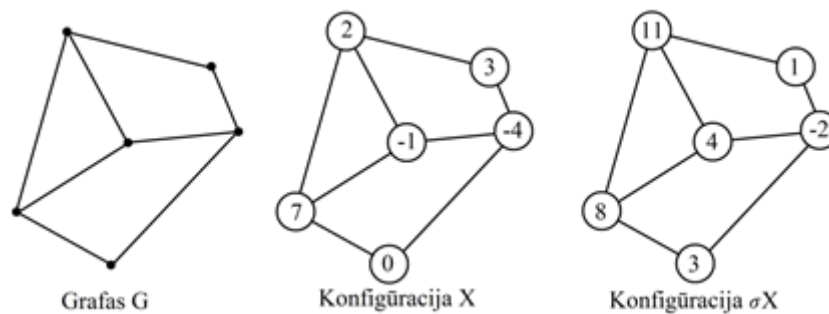
kur

$$\sigma_G(X)(v) = \sum_{u \in N(v)} X(u).$$

Perėjimo operatorius kiekvienoje grafo viršūnėje esančią reikšmę pakeičia reikšme, lygia jos aplinkoje esančių viršūnių būsenų sumai. Tokiu būdu perėjimo operatorius yra tiesinis operatorius erdvėje C_G .

Apibrėžimas. Ląstelinis automatas yra grafas su nurodytu jo būsenų keitimo tiesiniu operatoriumi.

Kitaip tariant, ląstelinis automatas yra grafas, kurio viršūnių pradinės būsenos parenkamos iš Abelio monoido, o naujos konfigūracijos viršūnių būsenos yra gaunamos panaudojant perėjimo operatorių σ_G .

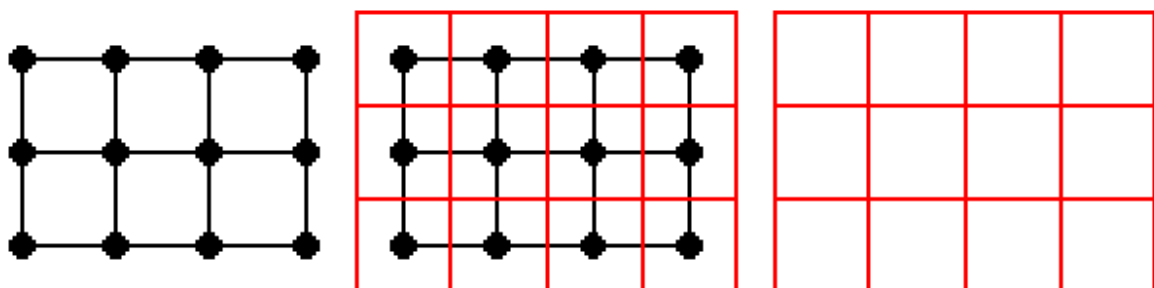


1 pav. Ląstelinio automato konfigūracijos keitimas.

Vienas iš svarbiausių šio tipo automatų teorijos uždavinių yra nustatyti ar kiekviena konfigūracija turi pirmvaizdį. Tiesinės algebros terminais šis klausimas ekvivalentus klausimui – ar būsenos keitimo operatorius yra siurjekcija? Žinome, kad baigtinės dimencijos tiesinės erdvės atveju tiesinis operatorius yra siurjekcija tada ir tik tada kai jis yra injekcija. O tai ekvivalentu tiesinio operatoriaus apgręžiamumui. Taigi, jei operatoriaus matricos determinantas nelygus nuliui, tai kiekviena konfigūracija turi pirmvaizdį. O jei operatoriaus matricos determinantas lygus nuliui, šis operatorius turi nenulinį branduolį ir tada operatorius nėra siurjekcija. Perėjimo operatoriaus branduolio dimensijos nustatymo ir jo struktūros tyrimas taip pat yra vienas iš svarbiausių ląstelinio automatų teorijos uždavinių.

1.3. Pagrindiniai ląstelinio automatų tipai

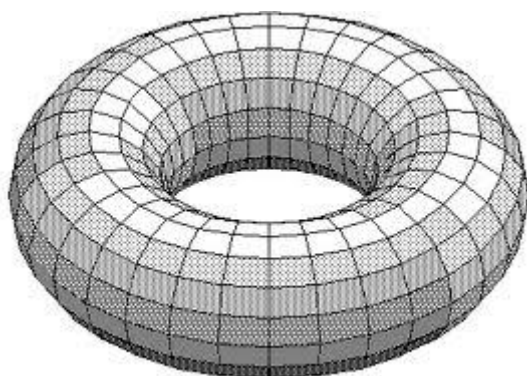
Nors apibrėžime ląstelinis automatas traktuojamas kaip grafas, tačiau tyrimuose dažniausiai naudojami stačiakampiai tinkliniai grafai, kuriuos visada patogu pertvarkyti į dvimates lenteles. Kiekvienai grafo viršūnei priskiriamas vienas langelis, o grafo briaunos pažymimos kaip lentelėje esančių langelių briaunos (2 pav.).



2 pav. Tinklinio grafo vizualizavimas lentele

Modeliuojant realius procesus, ląstelinio automato tinklas gali būti ir begalinis, tačiau dažniausiai tiriama trijų tipų grafai:

1. Stačiakampis plokščiasis tinklas.
2. Cilindrinis grafas.
3. Toras.



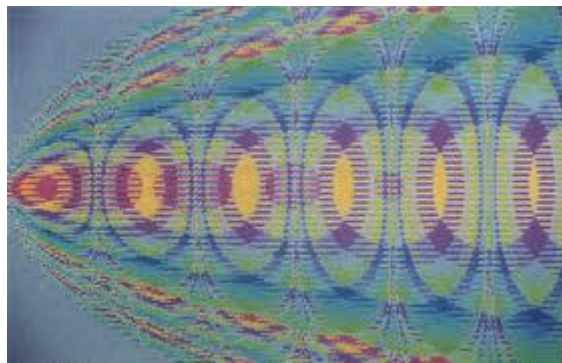
3 pav. Toras

Kitas svarbus žingsnis yra parinkti ląstelės vaizdavimo formą. Dažniausiai naudojamos trikampės, keturkampės ir šešiakampės ląstelės. Iš jų populiariausios yra keturkampės. Jų populiarumo paslaptis yra ta, kad iš keturkampių ląstelių sudarytus automatus yra patogiau programuoti, nes visą ląstelinį automatą galime traktuoti kaip matricą, tai yra dvimatį masyvą. Naudojant trikampus arba šešiakampius reikėtų naudotis kitokia duomenų struktūra. Tačiau taikant keturkampus, kyla klausimas kuriuos langelius laikyti kaimynais: turinčius tik bendrą kraštinę ar ir tuos kurie turi bendrą viršūnę. Šešiakampio atveju, ląstelė su visais kaimynais turės bendrą kraštinę ir papildomų klausimų nekils. Tačiau neapsiribojama tik šiomis trimis formomis. Ląsteliniam automatui galima pritaikyti bet kokius daugiakampius ir nebūtinai jie visi turi būti vienodi. Futbolo kamuolys yra puikus pavyzdys, kaip galima sukurti uždarą ląstelinį automatą iš skirtingų daugiakampių, tai yra penkiakampių ir šešiakampių (4 pav.).



4 pav. Uždaras ląstelinis automatas iš šešiakampių ir penkiakampių

Dažniausiai ląstelės būsenoms priskiriamos dvi galimos reikšmės: 1 (aktyvuota) ir 0 (neaktyvuota). Taip pat populiari imti reikšmės iš liekanų klasių pirminiu moduliu p lauko F_p . Jei $p=2$, kaip tik ir gauname pirmąjį paminėtą atvejį. Kai galimų būsenų skaičius yra didelis, pasidaro gana sunku stebėti visus pokyčius, todėl tokiais atvejais vietoje skaitinių reikšmių būna imamos atitinkamos spalvos (5 pav.). Stebint spalvinį vaizdą yra kur kas lengviau viską analizuoti.



5 pav. Spalvoto ląstelinio automato konfigūracija

Stebint spalvinį vaizdą yra kur kas lengviau viską analizuoti. Tokie metodai dažnai taikomi tiriant šiluminius procesus, įvairių ligų plitimą, miško gaisrus ar paukščių migraciją.

1.4. Žinomiausi ląsteliniai automatai

Per pusę amžiaus yra sukurta ir analizuojama daug įvairiausių ląstelinių automatų, tačiau daugumos analizė yra sunkiai suprantama ir nevaizdinga, todėl visuomenėje neišpopuliarėjo. Tačiau yra ir tokių, kurie pasiekė didžiulės sėkmės komerciniuose projektuose bei sulaukė didžiulio matematikų dėmesio. Jie bus aptarti 1.4.1 ir 1.4.2. poskyriuose.

1.4.1 Žaidimas „Gyvenimas“

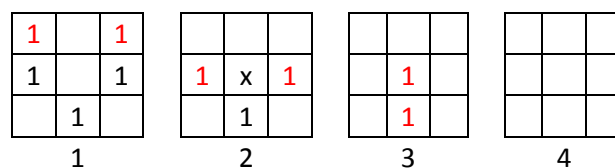
„Gyvenimas“ (Game of Life) [25] yra pirmasis itin populiarius ląstelinis automatas. Jį sukūrė garsus matematikas Džonas Hortonas Konvėjus (John Horton Conway) 1970 m.

Jis apibrėžiamas visoje plokštumoje, ją padalinus į kvadratinės formos ląsteles. Kaimyninėmis ląstelėmis laikomos tos, kurios turi bendrą kraštinę arba bendrą viršūnę, neįtraukiant pačios ląstelės. Kiekviena ląstelė turi dvi galimas būsenas: 1 (aktyvuota) arba 0 (neaktyvuota). Dažniausiai pradinė konfigūracija sudaroma tik iš kelių aktyvuotų ląstelių, o visos kitos laikomos neaktyviomis. Kiekviename cikle išnyksta arba atsiranda naujos ląstelės pagal šias taisykles:

1. Jeigu neaktyvuota ląstelė turi lygiai trys aktyvuotas kaimynines ląsteles, tuomet ji aktyvuojasi.
2. Jeigu aktyvuota ląstelė turi mažiau nei dvi aktyvuotas kaimynines ląsteles, tuomet ji tampa neaktyvuota.
3. Jeigu aktyvuota ląstelė turi daugiau nei tris aktyvuotas kaimynines ląsteles, tuomet ji tampa neaktyvuota.
4. Jeigu aktyvuota ląstelė turi dvi arba tris aktyvuotas kaimynines ląsteles, tuomet ji išlieka aktyvuota.

Jeigu pradinę konfigūraciją imtumė kaip „V“ raidę, visas gyvenimas baigtųsi labai greitai.

Užtektų keturių ciklų, kad visos ląstelės taptų neaktyvuotomis (6 pav.).



6 pav. „V“ formos konfigūracijos ciklai

Raudona spalva rodo, kad ląstelė sekančioje konfigūracijoje taps neaktyvuota, o x parodo kurioje vietoje sekančioje kartoje ląstelės aktyvuosis.

Imkime dar vieną pradinę konfigūraciją, kuri vadinama „mažąja r“ (7 pav.).

Figure 1 shows four 5x5 grids illustrating the step-by-step construction of a Latin square. The grids are labeled 1, 2, 3, and 4 below them.

- Grid 1: Row 1: 1, 1, x, 1, x; Row 2: empty; Row 3: empty; Row 4: empty; Row 5: empty.
- Grid 2: Row 1: 1, 1, x, 1, x; Row 2: x, 1, 1, x, 1; Row 3: empty; Row 4: empty; Row 5: empty.
- Grid 3: Row 1: 1, 1, x, 1, x; Row 2: x, 1, 1, x, 1; Row 3: 1, x, 1, 1, x; Row 4: empty; Row 5: empty.
- Grid 4: Row 1: 1, 1, x, 1, x; Row 2: x, 1, 1, x, 1; Row 3: 1, x, 1, 1, x; Row 4: x, 1, 1, x, 1; Row 5: empty.

7 pav. Pirmieji „mažosios r“ formas konfigurācijas cikli

Ši kartą pradinę konfiguraciją taip pat sudaro penkios aktyvuotos ląstelės, tačiau jos ilgiau išlieka aktyvios ir plečiasi į dar didesnes „kolonijas“. Pasitelkus į pagalbą kompiuterį, buvo apskaičiuota, kad tik 1105 cikle ląstelės įgyja stabilią būseną ir jų skaičius kas du ciklus pradeda periodiškai kartotis (1 priedas).

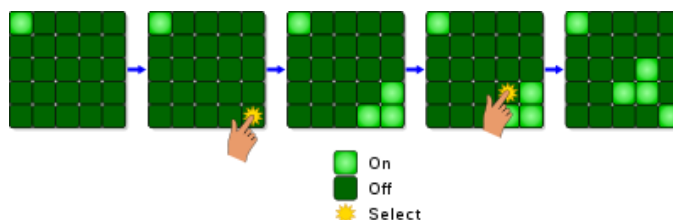
Nors viskas turi vykti begaliniame lauke, tačiau kompiuteriui tai yra neįveikiama užduotis, todėl pradžioje užtenka imti pakankamai didelio dydžio lauką, kuriame telpa pradinė konfigūracija. Vėliau, kiekvienoje naujoje konfigūracijoje, atlikti patikrinimą, ar nėra naujų aktyvuotų ląstelių ant kraštinės. Atsiradus aktyvuotai ląstelei ant kraštinės, tereikia pridėti dar vieną papildomą eilutę arba stulpelį.

1.4.2 Žaidimas „Užgesink šviesas“ – Lights Out

Žaidimas „Užgesink šviesas“ (Lights Out) yra 1995 metais patentuotas „Tiger Electronics“ firmos elektroninis žaidimas (8 pav.). Žaidimą sudaro 5x5 lentelė padalinta į 25 langelius, kurie gali įgyti dvi būsenas: šviečia arba ne. Pradinėje būsenoje dalis langelių šviečia, likusieji – nešviečia. Paspaudus, kurį nors mygtuką, jis ir jo kaimynai pakeičia būseną: jei buvo užgesintas – užsidega, jei degė – užgęsta. Šiame žaidime kaimynais laikomi tik tie langeliai, kurie turi bendrą kraštinę. Žaidimo tikslas yra užgesinti visus langelius.



8 pav. Žaidimas „Lights out“



9 pav. Žaidimo „Lights Out“ principas

Žaidimas išpopuliarėjo ne tik tarp loginių galvosūkių mėgėjų, tačiau ir tarp rimtų matematikų. Labiausiai visus domino, ar įmanoma užgesinti kiekvieną konfigūraciją ir, jei tai įmanoma, kiek mažiausiai kartų reikia spustelti langelius. Šiais klausimais itin aktyviai domėjosi Klausas Satneris (Klaus Suttner), kuris analizavo ne tik 5x5 kvadratą, bet ir kitų dydžių kvadratus bei stačiakampius [19]. Taip pat jis įrodė (bet kokių matmenų lentelei), kad jei visi langeliai šviečia, tuomet visada įmanoma užgesinti visą lentelę.

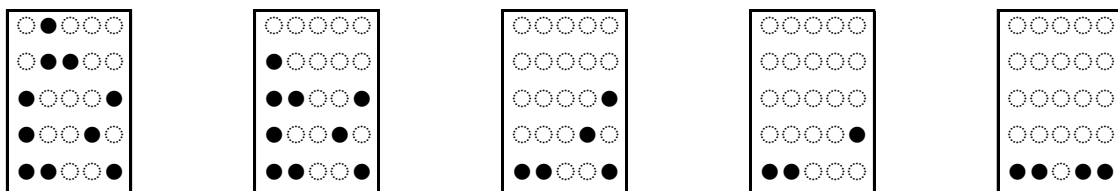
Šiam žaidimui buvo sukurta eilė sprendimo algoritmų, tačiau nei vienas jų neapsieina be tiesinės algebros žinių, nes reikia spręsti lygčių sistemas. Pateiksime vieną iš algoritmų, kuriame nereikės atlikti jokių skaičiavimų, bet reikės išiminti keletą kombinacijų, kurios gaunamos išsprendus tiesinių lygčių sistemą. Pirmasis algoritmo žingsnis yra užgesinti visas viršutinėje eilutėje esančias lemputes, pasinaudojant žemiau esančia eilute. Užgesinus visą pirmą eilutę, paeiliui analogiškai užgesiname antrą, trečią bei ketvirtą eilutes. Toks procesas vadinamas bėgimu žemyn (angl. chasing down) Jeigu tai atlikus apatinėje eilutėje visos lemputės išjungtos, vadinasi tikslą pasiekėme. Jeigu bent viena lemputė liko degti, tuomet reikės priklausomai nuo apatinėje eilutėje esančios kombinacijos, viršutinėje eilutėje aktyvuoti atitinkamas lemputes. Kombinacijų pavyzdžiai pateikti 1 lentelėje:

apatinė eilutė	viršutinė eilutė
●○○○●	●●○○○
○●○○○	●○○●○
●●●○○	○○○○○
○○●●●	○○○●○
●○○●○	○○○○●
○●●○○	●○○○○
●●○○●	○○●○○

1 lentelė. Užgesinimo kombinacijos

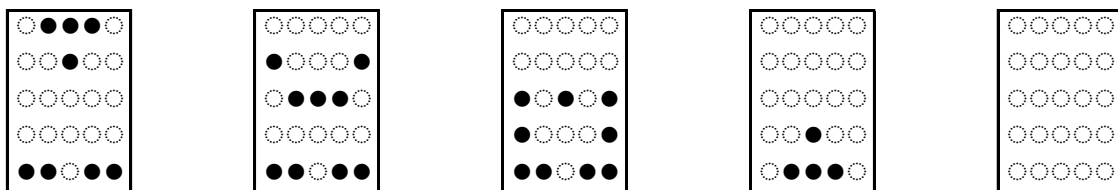
Viršutinėje eilutėje įvedus atitinkamą kombinaciją, vėl paeiliui užgesinama kiekviena eilutė ir kai galiausiai bus užgesinta ketvirta eilutė, paskutinėje eilutėje taip pat nebeliks nė vienos degančios lemputės, tad tikslas bus pasiektas.

8 paveiksluke pateiktas pavyzdys, kaip paeiliui išjungiamos visos kiekvienos eilutės lemputės, kol lieka tik apatinė eilutė.



10 pav. Pirma lentelės gesinimo stadija

Apatinės eilutės kombinacija yra 1 lentelėje, todėl reikės viršutinėje eilutėje aktyvuoti vidurinę lemputę:



11 pav. Antra lentelės gesinimo stadija

Lentelėje neliko nė vienos degančios lemputės, tad tikslą pasiekėme. Kad jį pasiektume, nereikėjo spręsti jokių lygčių ar atlikinėti įvairių skaičiavimų, užteko prisiminti tik kombinacijų lentelę.

2. Tinklinio ląstelinio automato matematinis modelis

2.1 Matematinio modelio sudarymas

Imkime stačiakampę $m \times n$ matmenų lentelę ir kiekvienam jos langeliui priskirkime po realią reikšmę:

$$X = \begin{array}{|c|c|c|c|} \hline x_1 & x_2 & \dots & x_n \\ \hline x_{n+1} & x_{n+2} & \dots & x_{2n} \\ \hline \dots & \dots & \dots & \dots \\ \hline x_{(m-1)n+1} & x_{(m-1)n+2} & \dots & x_{mn} \\ \hline \end{array}$$

Lentelėje esančius realius skaičius numeruojame nuo pirmos eilutės kairiojo langelio iki eilutės galo. Tuomet pereiname į sekančią eilutę ir tęsiame numeraciją. Indeksus galima numeruoti ir kaip matricose su dviem indeksais (x_{ij} , i-eilutė, j-stulpelis). Šį kartą tokio numeravimo atsisakoma, nes vėliau konfigūraciją X reikia vaizduoti ne matrica, o vektoriumi.

Norėdami pereiti nuo konfigūracijos X prie konfigūracijos $\sigma X =$ (pagal Satnerio apibrėžimą), prie kiekviename langelyje esančios reikšmės, pridedame kaimyninių langelių reikšmes:

$$\sigma X = \begin{array}{|c|c|c|c|} \hline x_1 + x_2 + x_{n+1} & x_1 + x_2 + x_3 + x_{n+2} & \dots & x_{n-1} + x_n + x_{2n} \\ \hline x_1 + x_{n+1} + x_{n+2} + x_{2n+1} & x_2 + x_{n+1} + x_{n+2} + x_{n+3} + x_{2n+2} & \dots & x_n + x_{2n-1} + x_{2n} + x_{3n} \\ \hline \dots & \dots & \dots & \dots \\ \hline x_{(m-2)n+1} + x_{(m-1)n+1} + x_{(m-1)n+2} & x_{(m-2)n+2} + x_{(m-1)n+1} + x_{(m-1)n+2} + x_{(m-1)n+3} & \dots & x_{(m-1)n} + x_{(m-1)n+n-1} + x_{mn} \\ \hline \end{array}$$

Pažymėjus $\sigma X = Y$, konfigūracijos Y būsenos išreiškiamos lygybėmis:

$$\begin{aligned}x_1 + x_2 + x_{n+1} &= y_1 \\x_1 + x_2 + x_3 + x_{n+2} &= y_2 \\&\dots\end{aligned}$$

Šios išraiškos leidžia gauti perėjimo operatoriaus σ matricą C . Matricos C išraiškoje naudojamos dvi kvadratinės n eilės matricos:

$$B_n = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ 0 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}, \quad I_n = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Matrica B_n yra trijstrižinė matrica. Ši matrica yra sudaryta iš kiekvieno langelio ir jo kaimynų, esančių kairiau arba dešiniau, koeficientų. Matrica I_n yra vienetinė matrica. Ji sudaryta iš žemiau arba aukščiau esančių kaimynų koeficientų. Panaudodami šias matricas ir nulinę matricą O , gauname blokinę matricą C , kuri susideda iš m blokų kiekviename stulpelyje ir eilutėje. Matrica $C_{mn} - I_{mn}$ yra stačiakampio tinklinio grafo gretimumo matrica.

$$C_{mn} = \begin{pmatrix} B & I & O & \dots & O \\ I & B & I & \dots & O \\ O & I & B & \dots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & \dots & B \end{pmatrix}.$$

Taigi perėjimo operatoriaus σ matricinė išraiška yra:

$$CX = Y.$$

Čia

C – operatoriaus $mn \times mn$ matmenų matrica,

X – pradinė konfigūracija,

Y – gauta konfigūracija.

Kaip jau minėta, klausimas apie perėjimo operatoriaus siurjekciją yra vienas iš svarbiausių ląstelinių automatų teorijos klausimų. Norint atsakyti į šį klausimą, reikia nustatyti ar tiesinių lygčių sistema su duotąja konfigūracija Y turi sprendinį. Iš tiesinės algebros kurso žinome, kad jeigu $\det C \neq 0$, tuomet bet kuri konfigūracija Y turės pirmvaizdį $X = C^{-1} \cdot Y$. Tačiau jeigu $\det C = 0$, tuomet tiesinis operatorius σ nėra siurjekcija ir apgręžiamos bus tik tos konfigūracijos, kurios priklauso operatoriaus σ vaizdui $\sigma(C_G)$. Gerai žinome, kad tiesinio operatoriaus vaizdo $\sigma(C_G)$ dimensija yra lygi $mn-k$, kur k yra operatoriaus σ branduolio dimensija.

2.2 Nedidelių lentelių tyrimas

Nagrinėkime atvejį 3×4 . Pirmiausia lentelės langeliams priskirkime pradinę konfigūraciją Y :

y_1	y_2	y_3	y_4
y_5	y_6	y_7	y_8
y_9	y_{10}	y_{11}	y_{12}

Norėdami nustatyti, ar ši konfigūracija turi pirmvaizdį, kiekvienam langeliui sudarome po lygtį ir lygčių sistemą užrašome matriciniu pavidalu:

$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	x_1	y_1
			x_2	y_2
			x_3	y_3
			x_4	y_4
$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	x_5	y_5
			x_6	y_6
			x_7	y_7
			x_8	y_8
			x_9	y_9
$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$	x_{10}	y_{10}
			x_{11}	y_{11}
			x_{12}	y_{12}

$$=$$

Pasinaudojus matematiniu paketu „Matlab“ buvo apskaičiuotas šios dvyliktos eilės matricos determinantas, kuris yra lygus -9. Determinantas nelygus nuliui, todėl 3×4 lentelėje bet kuri konfigūracija turės pirmvaizdį $X = C^{-1} \cdot Y$.

Toliau nagrinėkime atvejį, kai lentelės matmenys 2×3 . Analogiškai sprendami gauname tokią lygčių sistemą:

$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	x_1	y_1
		x_2	y_2
		x_3	y_3
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	x_4	y_4
		x_5	y_5
		x_6	y_6

$$=$$

Šios matricos determinantas yra lygus 0, todėl šį kartą ne kiekviena konfigūracija Y turės pirmvaizdį. Imkime tokią konfigūraciją:

$$Y = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 3 & 1 \end{bmatrix}$$

Gauname, kad sistemos matricos rangas yra 5, o išplėstinės matricos rangas lygus 6, vadinasi lygčių sistema yra nesuderinta ir konfigūracija Y pirmvaizdžio neturi.

3. Spartesnis algoritmas

Sprendžiant tiesinių lygčių sistemą, dažniausia naudojamas Gauso metodas. Tačiau mūsų atveju, norint ištirti $m \times n$ gardele, reikėtų spręsti lygčių sistemą iš $m \cdot n$ lygčių ir $m \cdot n$ nežinomųjų. Gauso metodo sudėtingumas yra trečios eilės, todėl sprendžiant tokią sistemą reikėtų atlikti apie $m^3 \cdot n^3$ veiksmų. Tačiau net ir paėmus santykinai nedidelių matmenų lentelę $100 \cdot 300$, reikės atlikti apie $2,7 \cdot 10^{13}$ veiksmų, o tai yra sunkiai išsprendžiama užduotis net ir šiuolaikiniams kompiuteriams. Todėl norint testuoti didesnių matmenų matricų analizę, pateiksime spartesnį sprendimo algoritmą.

Bendru atveju lygčių sistema jau yra sudaryta ir jos matrica yra C :

$$\begin{matrix} B & I & O & \cdots & O \\ I & B & I & \cdots & O \\ O & I & B & \cdots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & \cdots & B \end{matrix}.$$

Pritaikę Gauso metodą, galėtume matricą pertvarkyti į laiptuotą trikampę formą, tačiau šį kartą tokia strategija nėra tinkama. Norint eliminuoti antroje eilutėje esantį pirmą bloką I , iš šios eilutės reikėtų atimti pirmą eilutę padaugintą iš B^{-1} , tačiau matrica B ne visada turės atvirkštinę. Kai $n=2$, matrica B bus:

$$B = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Akivaizdu, kad šios matricos determinantas lygus nuliui, todėl atvirkštinė neegzistuoja. Taigi taikydami Gauso metodą, negalėsime matricos pertvarkyti į laiptuotą formą išlaikant matricą B jos blokiniu pavidalu.

3.1 Čebyševio daugianariai

Rekurentiniu sąryšiu $U_n = x \cdot U_{n-1} - U_{n-2}$ apibrėžti daugianariai $U_n x$, su sąlygomis $U_0 = 1$, $U_1 = x$, yra vadinami antro tipo normalizuotais Čebyševio daugianariais [9]. Šie Čebyševio daugianariai yra plačiai išnagrinėti ir turi labai daug taikymų. Visos n -ojo daugianario šaknys yra žinomos ir jų absoliutinis dydis visada mažesnis už 2. Čebyševio daugianarių $U_n x$ šaknų išraiška:

$$x_i = 2 \cdot \cos \frac{i \cdot \pi}{n+1}, i = 1, 2, \dots, n.$$

Panaudodami šiuos daugianarius, sudarysime spartesnį algoritmą matricos C analizei. Norėdami supaprastinti matricą C , naudosime metodą, kuriame neprireiks atvirkštinių matricų. Pažymėkime kiekvieną matricos C blokų eilutę L_i :

$$\begin{array}{cccccc} L_1 & B & I & O & \dots & O \\ L_2 & I & B & I & \dots & O \\ L_3 & O & I & B & \dots & O \\ \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ L_m & O & O & O & \dots & B \end{array}.$$

Matricą C pertvarkysime rekurentiniu būdu. Tegul L_0 yra nulinė blokų eilutė. Naujas eilutes konstruojame rekurentiškai:

$$\begin{aligned} L'_0 &= L_0, \\ L'_1 &= L_1, \\ L'_{i+2} &= -B \cdot L'_{i+1} - L'_i, \quad 0 \leq i \leq m-2. \end{aligned}$$

Šis rekurentinis sąryšis labai panašus į antro tipo Čebyševio daugianarius generuojantį sąryšį. Skiriasi tik daugianarių su lyginiais laipsniais ženklai.

Pritaikę šią strategiją, blokinę matricą C pertvarkome į 2×2 blokų matricą:

$$\begin{pmatrix} P & I \\ M & O \end{pmatrix},$$

kur

I yra $m-1 \times n$ eilės vienetinė matrica,

$$M_n = -1^{m-1} \cdot U_m B,$$

O – nulinė matrica.

P yra blokų stulpelis:

$$P = \begin{pmatrix} B \\ I - B^2 \\ B^3 - 2B \\ \dots \end{pmatrix},$$

Pastebėsime, kad šiame stulpelyje gauname matricas išreikštas per matricą B ir Čebyševio daugianarį:

$$P_{m-1, n, n} = \begin{pmatrix} U_1 B \\ -U_2 B \\ U_3 B \\ \dots \\ -1^{m-1} \cdot U_{m-1} B \end{pmatrix}.$$

Iliustruokime matricos C pertvarkymą atveju 3×4 :

$$C = \begin{pmatrix} B_4 & I_4 & 0 \\ I_4 & B_4 & I_4 \\ 0 & I_4 & B_4 \end{pmatrix}.$$

Iš antros blokų eilutės atimame pirmą blokų eilutę padaugintą iš B_4 :

$$\begin{pmatrix} B_4 & I_4 & 0 \\ I_4 - B_4^2 & 0 & I_4 \\ 0 & I_4 & B_4 \end{pmatrix}.$$

Iš trečios blokų eilutės atimame pirmą blokų eilutę:

$$\begin{pmatrix} B_4 & I_4 & 0 \\ I_4 - B_4^2 & 0 & I_4 \\ -B_4 & 0 & B_4 \end{pmatrix}.$$

Iš trečios blokų eilutės atimame antrą blokų eilutę padaugintą iš B_4 :

$$\begin{pmatrix} B_4 & I_4 & 0 \\ I_4 - B_4^2 & 0 & I_4 \\ B_4^3 - 2B_4 & 0 & 0 \end{pmatrix}.$$

Gavome matricą iš keturių blokų, tarp kurių du yra kvadratinio tipo:

1	1	0	0	1	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
-1	-2	-1	0	0	0	0	0	1	0	0	0
-2	-2	-2	-1	0	0	0	0	0	1	0	0
-1	-2	-2	-2	0	0	0	0	0	0	1	0
0	-1	-2	-1	0	0	0	0	0	0	0	1
2	3	3	1	0	0	0	0	0	0	0	0
3	5	4	3	0	0	0	0	0	0	0	0
3	4	5	3	0	0	0	0	0	0	0	0
1	3	3	2	0	0	0	0	0	0	0	0

12 pav. Pertvarkyta matrica C

Teorema: $\det C = \det U_m B_n$

[rodymas: Iš blokinių matricų teorijos [14] žinome, kad blokinės matricos

$$\begin{pmatrix} X & Y_k \\ Z_l & O \end{pmatrix}$$

determinantas yra lygus:

$$(-1)^{k \cdot l} \cdot \det(Y) \cdot \det(Z).$$

Mūsų atveju Y_k yra vienetinė matrica ir $k = m - 1$, o matrica Z_l yra $(-1)^{m-1} \cdot U_m B_n$ ir $l = n$. Todėl determinanto išraiška yra:

$$\begin{aligned} (-1)^{m-1 \cdot n^2} \cdot \det (-1)^{m-1} \cdot U_m B_n &= (-1)^{m-1 \cdot n^2 + m-1 \cdot n} \cdot \det U_m B_n = \\ (-1)^{m-1 \cdot n + m-1 \cdot n} \cdot \det U_m B_n &= (-1)^{2 \cdot m-1 \cdot n} \cdot \det U_m B_n = \det U_m B_n. \quad \square \end{aligned}$$

Teorema: Matricos C branduolio dimensija yra lygi matricos $U_m B_n$ branduolio dimensijai.

[rodymas: Iš tiesinės algebros kurso žinome, kad diagonalinės matricos branduolio dimensija yra lygi jos įstrižainėje esančių lygių nuliui elementų skaičiui. Sukeiskime, modifikuotoje matricoje C , visus stulpelius, kurie priklauso vienetiniai matricai I , taip, kad visi vienetai atsidurtų matricos I' šalutinėje įstrižainėje:

$$\begin{pmatrix} P & I \\ M & O \end{pmatrix} \Rightarrow \begin{pmatrix} P & I' \\ M & O \end{pmatrix}.$$

Panaudodami vienetinės matricos elementus, galime atlikti elementariusius pertvarkius, po kurių matrica P taps nulių matrica:

$$\begin{pmatrix} P & I' \\ M & O \end{pmatrix} \Rightarrow \begin{pmatrix} O & I' \\ M & O \end{pmatrix}.$$

Gavome matricą, kurios šalutinėje įstrižainėje yra vienetinė matrica ir matrica M . Todėl šios matricos branduolio dimensija lygi blokinės matricos M branduolio dimensijai. $\square$

3.2 Matricos C determinanto analizė

Kadangi žinome visas Čebyševio daugianarių šaknis, todėl šiuos daugianarius galime išskaidyti pirmo laipsnio daugikliais:

$$U_n(x) = (x - x_1) \cdot (x - x_2) \cdot \dots \cdot (x - x_n).$$

Kadangi matricų sandaugos determinantui galioja savybė:

$$\det A \cdot B = \det A \cdot \det B,$$

todėl matricos C_{mn} determinantas yra lygus:

$$\det C_{mn} = \prod_{i=1}^m \det B_n - 2 \cdot \cos \frac{i \cdot \pi}{m+1} \cdot I_n.$$

Prisiminkime, kad vienas pagrindinių šio darbo tikslų yra rasti stačiakampių grafų matmenis m ir n prie kurių perėjimo operatoriaus matricos determinantas yra lygus nuliui. Pasinaudojant šia formule matome, kad determinantas bus lygus nuliui tada ir tik tada, kai bent vienas daugiklis bus lygus nuliui. Todėl užteks spręsti lygtį:

$$\det B_n - x_i \cdot I_n = 0,$$

kur

$$x_i = 2 \cdot \cos \frac{i \cdot \pi}{m+1}, i = 1, 2, \dots, m.$$

Pažymėkime $x = x_i$ ir užrašykime determinantą $B_n - x_i \cdot I_n$ matriciniu pavidalu:

$$\det(B_n - x_i \cdot I_n) = \begin{vmatrix} 1-x & 1 & 0 & \dots & 0 \\ 1 & 1-x & 1 & \dots & 0 \\ 0 & 1 & 1-x & \dots & 0 \\ \dots & \dots & \dots & \dots & 1 \\ 0 & 0 & 0 & 1 & 1-x \end{vmatrix}$$

Skaičiuodami determinantą, pirmiausia jį skleisime pagal pirmą stulpelį:

$$(1-x) \begin{vmatrix} 1-x & 1 & \dots & 0 \\ 1 & 1-x & \dots & 0 \\ \dots & \dots & \dots & 1 \\ 0 & 0 & 1 & 1-x \end{vmatrix} - \begin{vmatrix} 1 & 0 & \dots & 0 \\ 1 & 1-x & \dots & 0 \\ \dots & \dots & \dots & 1 \\ 0 & 0 & 1 & 1-x \end{vmatrix}$$

Sekančiame žingsnyje dešinės matricos determinantą skleisime pagal pirmą eilutę:

$$(1-x) \begin{vmatrix} 1-x & 1 & \dots & 0 \\ 1 & 1-x & \dots & 0 \\ \dots & \dots & \dots & 1 \\ 0 & 0 & 1 & 1-x \end{vmatrix} - \begin{vmatrix} 1-x & \dots & 0 \\ \dots & \dots & 1 \\ 0 & 1 & 1-x \end{vmatrix}$$

Taip skleidami, gavome rekurentinį sąryšį matricos $B_n - x_i \cdot I_n$ determinantui D_n skaičiuoti:

$$D_n = (1 - x) \cdot D_{n-1} - D_{n-2}, \quad D_0 = 1, \quad D_1 = 1 - x.$$

Išspręskime šią homogeninę rekurentinę lygtį. Pirmiausia užrašykime charakteringąją lygtį:

$$q^2 - (1 - x) \cdot q + 1 = 0.$$

Išsprendę šią lygtį, gauname charakteringasias šaknis:

$$q_{1,2} = \frac{1 - x \pm \sqrt{x + 1} \sqrt{x - 3}}{2}.$$

Prisiminkime, kad x yra Čebyševio daugianario šaknis, kurios absoliutinė reikšmė neviršija 2. Todėl reikės nagrinėti du atvejus:

- kai $x = -1$ ($q_1 = q_2 = 1$)
- kai $x \neq -1$ ($q_1 \neq q_2$)

Pirmu atveju, kai šaknys sutampa, bendrasis homogeninės lygties sprendinys atrodys taip:

$$D_n = c_1 + c_2 \cdot n.$$

Panaudokime pradines sąlygas ir raskime konstantų c_1, c_2 reikšmes:

$$\begin{matrix} c_1 = 1 \\ c_1 + c_2 = 2 \end{matrix} \Rightarrow \begin{matrix} c_1 = 1 \\ c_2 = 1 \end{matrix}$$

Gauname atskirą sprendinį:

$$D_n = 1 + n.$$

Determinantas bus lygus nuliui tik tada, kai $n = -1$, tačiau n yra vienas iš lentelės matmenų ir negali įgyti neigiamos reikšmės, todėl šis atvejis mums neįdomus. Nagrinėkime antrą atvejį, kai $x \neq -1$. Šį kartą homogeninės lygties bendrasis sprendinys atrodys taip:

$$D_n = c_1 \cdot q_1^n + c_2 \cdot q_2^n.$$

Panaudodami pradines sąlygas raskime konstantų c_1, c_2 reikšmes:

$$\begin{matrix} c_1 + c_2 = 1 \\ c_1 \cdot \frac{1 - x + \sqrt{x + 1} \sqrt{x - 3}}{2} + c_2 \cdot \frac{1 - x - \sqrt{x + 1} \sqrt{x - 3}}{2} = 1 - x \end{matrix}$$

Pertvarkykime antrąją lygtį:

$$(c_1 + c_2) \cdot \frac{1 - x}{2} + (c_1 - c_2) \cdot \frac{\sqrt{x + 1} \sqrt{x - 3}}{2} = 1 - x,$$

Kadangi žinome kam lygi c_1, c_2 suma, todėl galime dar labiau supaprastinti lygtį:

$$(c_1 - c_2) \cdot \sqrt{x + 1} \sqrt{x - 3} = 1 - x,$$

$$c_1 - c_2 = \frac{1-x}{x+1 \quad x-3}.$$

Gauname tokią lygčių sistemą:

$$\begin{aligned} c_1 + c_2 &= 1 \\ c_1 - c_2 &= \frac{1-x}{x+1 \quad x-3}. \end{aligned}$$

Jas abi sudėję, gausime c_1 :

$$c_1 = \frac{1-x + \frac{x+1 \quad x-3}{2}}{x+1 \quad x-3}.$$

Turėdami c_1 išraišką, nesunkiai randame c_2 išraišką:

$$c_2 = -\frac{1-x - \frac{x+1 \quad x-3}{2}}{x+1 \quad x-3}.$$

Išstatome šias išraiškas į bendrąjį sprendinį ir gauname atskirą rekurentinės lygties sprendinį:

$$D_n = \frac{\frac{1-x + \frac{x+1 \quad x-3}{2}}{x+1 \quad x-3}^{n+1} - \frac{1-x - \frac{x+1 \quad x-3}{2}}{x+1 \quad x-3}^{n+1}}{x+1 \quad x-3}.$$

Išspręskime lygtį:

$$D_n = 0.$$

Skaitiklį prilyginame nuliui ir padaliname iš antrojo skaitiklio nario, kuris niekada nelygus nuliui:

$$\frac{1-x + \frac{x+1 \quad x-3}{2}}{1-x - \frac{x+1 \quad x-3}{2}}^{n+1} - 1 = 0.$$

Panaikiname iracionalumą vardiklyje :

$$\frac{1-x + \frac{x+1 \quad x-3}{2}}{1-x^2 - \frac{x+1 \quad x-3}{2}} = \frac{1-x + \frac{x+1 \quad x-3}{2}}{2}.$$

Įvedame keitinį:

$$t = 1-x.$$

Pasinaudoję keitiniu, gauname tokią lygtį:

$$\frac{t + \frac{t^2-4}{2}}{2}^{2n+2} = 1.$$

Metodo, kuris išspręstų šią lygtį, rasti nepavyko. Tačiau panaudojus skaitinius metodus buvo rasti realūs sprendiniai prie įvairių n reikšmių. Taip pat buvo pastebėta, kad visi realūs

sprendiniai yra tokie patys kaip Čebyševio daugianarių šaknys prijungus dvi naujas reikšmes -2 ir 2:

$$t = 2 \cdot \cos \frac{j \cdot \pi}{n+1}, j = 0, 1, \dots, n+1.$$

Patikrinkime šiuos sprendinius įstatę juos į lygtį:

$$\frac{2 \cdot \cos \frac{j \cdot \pi}{n+1} + \sqrt{4 \cdot \cos^2 \frac{j \cdot \pi}{n+1} - 4}}{2} = \cos \frac{j \cdot \pi}{n+1} + \sqrt{\cos^2 \frac{j \cdot \pi}{n+1} - 1} =$$

$$= \cos \frac{j \cdot \pi}{n+1} + i \cdot \sin \frac{j \cdot \pi}{n+1}.$$

Pasinaudoje kompleksinio skaičiaus kėlimo laipsniu formule gauname:

$$\cos j \cdot 2 \cdot \pi + i \cdot \sin j \cdot 2 \cdot \pi = \cos j \cdot 2 \cdot \pi = 1.$$

Taigi šie sprendiniai tenkina lygtį, tačiau ar neegzistuoja kitų realių sprendinių pasakyti negalime.

Prisiminkime, kad $x = x_m$, kai:

$$x_i = 2 \cdot \cos \frac{i \cdot \pi}{m+1}, i = 1, 2, \dots, m.$$

Grįžkime prie keitinio $t = 1 - x$ ir sudarykime lygtį:

$$\cos \frac{i \cdot \pi}{n+1} + \cos \frac{j \cdot \pi}{m+1} = \frac{1}{2},$$

kur:

$$i = 0, 1, \dots, n,$$

$$j = 0, 1, \dots, m.$$

Teorema: Jeigu trigonometrinė diofantinė lygtis turi bent vieną sprendinį (i, j) , tuomet

$$\det C_{mn} = 0.$$

Spręsdami šią lygtį pasinaudosime internete esančiu matematiniu paketu „WolframAlpha“, kuriame yra realizuotas algoritmas šiai lygčiai spręsti. Šio paketo pagalba išspręsim lygtį, kai

$$m = 9, n = 14.$$



computational...
knowledge engine

$\cos(x\pi/10) + \cos(y\pi/15) = 1/2, 0 < x < 10, 0 < y < 15$

Examples Random

Assuming "0" is a decimal number | Use as [referring to math](#) instead

Input:

$$\left\{ \cos\left(x \times \frac{\pi}{10}\right) + \cos\left(y \times \frac{\pi}{15}\right) = \frac{1}{2}, 0 < x < 10, 0 < y < 15 \right\}$$

Exact result:

$$\left\{ \cos\left(\frac{\pi x}{10}\right) + \cos\left(\frac{\pi y}{15}\right) = \frac{1}{2}, 0 < x < 10, 0 < y < 15 \right\}$$

Integer solutions:

$x = 2,$	$y = 9$
$x = 5,$	$y = 5$
$x = 6,$	$y = 3$

13 pav. Trig. diof. lygties sprendimas su „WolframAlpha”

Bendroji diofantinių lygčių sprendimo metodika nėra sukurta ir daugeliu atveju jos yra itin sudėtingos. Yra sukurti keli metodai labai siauros klasės trigonometrinėms diofantinėms lygtims [5], tačiau mūsų sudarytai lygtčiai sprendimo metodo rasti nepavyko.

4. Branduolio struktūra

Prisimename, kad kiekviena konfigūracija turi pirmavaizdį tada ir tik tada, kai matricos C determinantas nelygus nuliui. Tačiau egzistuoja be galo daug stačiakampių gardelių $m \times n$, kuriose perėjimo operatoriaus determinantas lygus nuliui. Tokiu atveju perėjimo operatorius turės nenulinį branduolį, kurio dimensijos bei struktūros nustatymas yra vienas svarbiausių šio darbo uždavinių. Panaudojus matematinį paketą „Matlab“, buvo išspręstos nedidelių matmenų stačiakampių sistemos, kai determinantas yra lygus nuliui. Pastebėta, kad visų lentelių branduoliai sudaryti iš dviejų branduolių:

x	-x
---	----

x	y	-y	-x
-x-y	x	-x	x+y
x+y	-x	x	-x-y
-x	-y	y	x

Kanoninę šių branduolių erdvės bazę sudaro šie sprendiniai:

1	-1
---	----

1	0	0	-1
-1	1	-1	1
1	-1	1	-1
-1	0	0	1

0	1	-1	0
-1	0	0	1
1	0	0	-1
0	-1	1	0

Nesunku pastebėti, kad branduolių bazės sudarytos iš domino kauliukų. Įrodyti, kad nebus kitų bazinių branduolių yra gana sudėtinga. Su VGTU klasteriu „Vilkas“ buvo patikrintos visos gardelės, kurių matmenys kinta nuo 1 iki 50000, tačiau naujų branduolių nebuvo rasta. Sprendžiant didesnių matmenų gardeles, buvo pastebėta, kad jų branduoliai susideda iš mažesniųjų gardelių branduolių. Pavaizduokime 9×14 gardelės branduolį, kurio dimensija lygi trim, bei kanoninę branduolių erdvės bazę:

x+z	y-z	-y	-x+z	-z	x	y+z	-y-z	-x	z	x-z	y	-y+z	-z
-x-y	x	-x	x+y	0	-x-y	x	-x	x+y	0	-x-y	x	-x	x+y
x+y-z	-x+z	x	-x-y-z	z	x+y	-x-z	x+z	-x-y	-z	x+y+z	-x	x-z	-x-y+z
-x	-y	y	x	0	-x	-y	y	x	0	-x	-y	y	x
z	-z	0	z	-z	0	z	-z	0	z	-z	0	z	-z
x	y	-y	-x	0	x	y	-y	-x	0	x	y	-y	-x
-x-y-z	x+z	-x	x+y-z	z	-x-y	x-z	-x+z	x+y	-z	-x-y+z	x	-x-z	x+y+z
x+y	-x	x	-x-y	0	x+y	-x	x	-x-y	0	x+y	-x	x	-x-y
-x+z	-y-z	y	x+z	-z	-x	-y+z	y-z	x	z	-x-z	-y	y+z	x-z

14 pav. Gardelės 9×14 branduolio išraiška

1	0	0	-1	0	1	0	0	-1	0	1	0	0	-1
-1	1	-1	1	0	-1	1	-1	1	0	-1	1	-1	1
1	-1	1	-1	0	1	-1	1	-1	0	1	-1	1	-1
-1	0	0	1	0	-1	0	0	1	0	-1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	-1	0	1	0	0	-1	0	1	0	0	-1
-1	1	-1	1	0	-1	1	-1	1	0	-1	1	-1	1
1	-1	1	-1	0	1	-1	1	-1	0	1	-1	1	-1
-1	0	0	1	0	-1	0	0	1	0	-1	0	0	1

0	1	-1	0	0	0	1	-1	0	0	0	1	-1	0
-1	0	0	1	0	-1	0	0	1	0	-1	0	0	1
1	0	0	-1	0	1	0	0	-1	0	1	0	0	-1
0	-1	1	0	0	0	-1	1	0	0	0	-1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	-1	0	0	0	1	-1	0	0	0	1	-1	0
-1	0	0	1	0	-1	0	0	1	0	-1	0	0	1
1	0	0	-1	0	1	0	0	-1	0	1	0	0	-1
0	-1	1	0	0	0	-1	1	0	0	0	-1	1	0

1	-1	0	1	-1	0	1	-1	0	1	-1	0	1	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	1	0	-1	1	0	-1	1	0	-1	1	0	-1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-1	0	1	-1	0	1	-1	0	1	-1	0	1	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	1	0	-1	1	0	-1	1	0	-1	1	0	-1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-1	0	1	-1	0	1	-1	0	1	-1	0	1	-1

15 pav. Gardelės 9x14 kanoninė branduolių erdvės bazė

Nesunku pastebėti, kad 9x14 gardelės branduolio bazė sudaryta iš bazinių branduolių, atskirtų nulių stulpeliais ir eilutėmis. Tokiu būdu, nenaudojant jokių kompiuterinių technologijų, galima generuoti ir didesnes lenteles. Pasinaudojant šiuo konstravimo būdu, buvo sudaryta pakankama sąlyga nenulinio branduolio egzistavimui:

- 1 matmenis padidiname vienetu: $(m+1, n+1)$.
- 2 jeigu ši skaičių pora yra lygi bent vienai iš šių skaičių porų $(2a, 3b)$, $(3a, 2b)$, $(5a, 5b)$, kai $a, b \in \mathbb{N}$, tuomet $m \times n$ gardelė turės nenulinį branduolį.

Hipotezė: Stačiakampės gardelės $m \times n$ branduolio dimensija yra lygi trigonometrinės diofantinės lygties sprendinių skaičiui.

Ši hipotezė buvo patikrinta visoms gardelėms, kurių matmenys kinta nuo 1 iki 100, tačiau bendro įrodymo sudaryti nepavyko.

K. Satneris (K. Sutner) įrodė teoremą [22], kad jei būsenos įgyja reikšmes iš liekanų klasių modulių 2, tuomet branduolio dimensija yra lygi:

$$\deg BDD(U_m, x, U_n, x-1) = \dots$$

Kitaip tariant, stačiakampės gardelės branduolio dimensija yra lygi daugianarių U_m, x ir $U_n, x-1$ bendrų šaknų skaičiui. Darbe [9] yra teigiama, kad ši teorema yra teisinga visose liekanų klasėse pirminiu modulių p , tačiau įrodymas nepateiktas. Nagrinėjant nedidelių matmenų lenteles buvo pastebėta, kad ši teorema galioja ir imant realias būsenų reikšmes, tačiau įrodyti, kad taip bus su visų matmenų lentelėmis nepavyko. Imkime stačiakampę gardelę 9×14 , kurios branduolį jau buvome suskaičiavę. Užrašykime daugianarius U_9, x ir $U_{14}, x-1$:

$$U_9, x = x^4 - x^2 + x - 1, \quad x^2 - x - 1, \quad x^4 - 5x^2 + 5,$$

$$U_{14}, x-1 = x^4 - 3x^3 - x^2 + 3x + 1, \quad x^2 - x - 1, \quad x^4 - 5x^3 - 5x^2 + 5x - 5.$$

Surenkame bendrus daugiklius:

$$x^2 - x - 1.$$

Kadangi gavome trečios eilės daugianarį, todėl gardelės 9×14 branduolio dimensija yra lygi 3. Išnagrinėjus įvairių dydžių lenteles, buvo pastebėta, kad bendrų daugiklių skaičius neviršija 4, kai imant reikšmes iš liekanų klasės modulių 2, bendrų daugiklių skaičius yra begalinis. Bendri daugikliai virš nulinės charakteristikos lauko yra:

$$x,$$

$$x - 1,$$

$$x^2 - x - 1.$$

Kaip žinome, daugianario $x^2 - x - 1$ šaknys yra garsusis aukso pjūvis $\varphi = \frac{1+\sqrt{5}}{2}$ ir $1 - \varphi = \frac{1-\sqrt{5}}{2}$.

Su matematiniu paketu „Matlab“ buvo apskaičiuotos visų stačiakampių gardelių branduolių dimensijos, su visais $1 \leq m, n \leq 30$, kurios pateiktos lentelėje:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
3	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	0	0	0	2	0	0	0	0	2	0	0	0	0	2	0	0	0	0	2	0	0	0	0	2	0	0	0	0	2	0
5	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
9	0	1	0	2	1	0	0	1	2	0	1	0	0	3	0	0	1	0	2	1	0	0	1	2	0	1	0	0	3	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	0	1	2	1	0	1	0	3	0	1	0	1	2	1	0	1	0	3	0	1	0	1	2	1	0	1	0	3	0
15	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	1	0	2	1	0	0	1	2	0	1	0	0	3	0	0	1	0	2	1	0	0	1	2	0	1	0	0	3	0
20	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
21	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0	1	1	1	0	2	0
24	0	0	0	2	0	0	0	0	2	0	0	0	0	2	0	0	0	0	2	0	0	0	0	2	0	0	0	0	2	0
25	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
26	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
27	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	1	1	1	2	2	0	1	1	3	0	2	0	1	3	1	0	2	0	3	1	1	0	2	2	1	1	1	0	4	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2 lent. Branduolių dimensijos

Skaiciuojant didesnių lentelių branduolių dimensijas buvo pastebėta, kad dimensijos kartojasi su periodu 30 abiejų matmenų atžvilgiu. Tai buvo labai netikėta, nes literatūroje tiriamuose modeliuose modulių 2 periodiškumo nėra. Tačiau fiksavus vieną matmenį, pavyzdžiui m , branduolių dimensijos kinta su tam tikru periodu, kurio ilgis yra skaičiaus 30 dalikliai:

$$3 - 2m + 1 \pmod{2} \cdot 2 - m + 1 \pmod{3} \cdot 5 - 4m + 1 \pmod{5}.$$

Taip pat pastebėta, kad jeigu $n+1$ arba $m+1$ yra pirminis skaičius didesnis už 5, tuomet nepaisant to, kokia yra kito matmens reikšmė, branduolio dimensija yra lygi nuliui, tai yra bet kokia tokių gardelių konfigūracija turės pirmavaizdį.

5. IŠVADOS

1. Sudarytas ląstelinio automato matematinis modelis, kai jo būsenos įgyja realias reikšmes, o perėjimo operatorius yra K. Satnerio apibrėžtas būsenų keitimo operatorius σ .
2. Nagrinėtame ląsteliniam automate branduolio dimensija yra aprėžta ir kinta nuo 0 iki 4, kai K. Satnerio darbuose, kuriuose būsenos įgyja reikšmes 0 arba 1, branduolio dimensijos yra neprognozuojamos ir auga nenusipėjamai.
3. Pateiktas rekurentinis metodas, kuriuo matrica C_{mn} pertvarkoma į kvadratinę n eilės matricą $U_m(B_n)$ taip, kad branduolio dimensija ir determinantas išlieka nepakitę. Šis metodas ne tik sumažina matricos matmenis $\max(m,n)$ kartų, bet ir palengvino matricos analizę.
4. Jeigu egzistuoja bent vienas trigonometrinės diofantinės lygties sprendinys, tuomet egzistuoja nenulinis branduolys. Sudaryta hipotezė, kad trigonometrinės diofantinės lygties sprendinių skaičius yra lygus branduolio dimensijai.
5. Visų stačiakampių gardelių, kurių branduolio dimensija yra nelygi nuliui, branduolių erdvės bazės sudarytos iš bazinių 1×2 ir 4×4 branduolių sujungtų nulių stulpeliais arba eilutėmis. Pasinaudojus šia savybe, išvestos pakankamos sąlygos, leidžiančios nustatyti ar stačiakampė lentelė turės nenulinį branduolį. Su VGTU klasteriu „Vilkas“ buvo patikrintos visos lentelės, kurių matmenys kinta nuo 1 iki 50000, tačiau naujų bazinių sprendinių nebuvo rasta.

Literatūros sąrašas

- [1] Čiegis, R. *Lygiagretieji algoritmai*. Vilnius; Technika, 2001.
- [2] Čiegis, R.; Būda, V. *Skaičiuojamoji matematika*. Vilnius; TEV, 1997.
- [3] Krylovas, A. *Diskrečioji matematika*. Vilnius: Technika, 2008.
- [4] Barua, R.; Ramakrishnan, S. σ -game, σ^+ -game and two-dimensional additive cellular automata. Theoret. Comput. Sci., 1996, vol 154, p. 349–366.
- [5] Conway, J. H.; Jones, A. J. *Trigonometric diophantine equations (On vanishing sums of roots of unity)*. Acta Arith., 1976, vol 30, p. 229–240.
- [6] Das, S., Formenti, E., Kari, J. *Non-uniform Cellular Automata*. Theor. Comput. Sci., 2014, vol 559, p 1-2.
- [7] Durand-Lose, J. *The signal point of view: from cellular automata to signal machines*. JAC, 2008, p 238-249.
- [8] Gardner, M. *Mathematical Games – The fantastic combinations of John Conway's new solitaire game „life”*. Scientific American, 1970, vol 223, p 120-123.
- [9] Hunziker, M.; Machiavelo, A.; Park, J. *Chebyshev polynomials over finite fields and reversibility of σ -automata on square grids*. Theoret. Comput. Sci., 2004, vol 320, p. 465–483.
- [10] Kari, J. *A Universal Cellular Automaton Without Sensitive Subsystems*. Automata, 2014, p 44-55.
- [11] Kurka, P. *Topological Dynamics of Cellular Automata*. Encyclopedia of Complexity and System Science, 2008, vol 40.
- [12] Sakarovitch, J. *Elements of Automata Theory*. Cambridge University Press 2009.
- [13] Sarkar, P. σ^+ -automata on square grids. Complex Systems, 1996, vol 10, p 121–141.
- [14] Sylvester, J. R. *Determinants of block matrices*. Maths Gazette, 2000, vol 84, p 460–467.
- [15] Sutner, K. *Cellular automata, decidability and phasespace*. Fundamenta Informaticae, 2010, vol 104(1), p 141-160.
- [16] Sutner, K. *Computational classification of cellular automata*. International Journal of General Systems, 2012, vol 41(6), p 595-607.
- [17] Sutner, K. *De Bruijn Graphs and Linear Cellular Automata*. Complex Systems, 1991, vol 5(1), p 19-30.
- [18] Sutner, K. *Linear cellular automata and Fischer automata*. Parallel Comput, 1997, vol 23, p 1613-1634.
- [19] Sutner, K. *Linear cellular automata and the Garden-of-Eden*. Math. Intelligencer, 1989, vol 11, p 49–53.
- [20] Sutner, K. *Model Checking One-Dimensional Cellular Automata*. J. Cellular Automata, 2009, vol 4(3), p 213-224.
- [21] Sutner, K. *The σ -game and cellular automata*. Amer. Math. Monthly, 1990, vol 94, p. 24–34.
- [22] Sutner, K. *σ -Automata and Chebyshev-polynomials*. Theoret. Comput. Sci., 2000, vol 230, p. 49–73.
- [23] Yaku, T. *The Constructibility of a Configuration in a Cellular Automaton*. Journal of Computer and Systems Sciences, 1973, vol 7, p 481-496.
- [24] Wolfram, S. „A New Kind of Science”.
[žiūrėta 2012m spalio 21 d.]. Prieiga per internetą:
< <https://www.wolframscience.com/reference/notes/876b> >.
- [25] Rendell, P. „This Is a Turing Machine Implemented in Conway's Game of Life.”.
[žiūrėta 2014m sausio 28 d.]. Prieiga per internetą:
< <http://www.rendell.uk.co/gol/tm.htm> >.

Priedas A

```
#include <iostream>
#include <cmath>
#include <math.h>
#include <cstdlib>
#include <string.h>
#include <climits>
using namespace std;
    double a[2001][2001];
    double k[2001][2001];
int main()
{
    long i,j,gime,mire,liko,v,metai;
    metai=1;
    for(int i=1;i<2000;i++)
    {
        for(int j=1;j<2000;j++)
        {
            a[i][j]= 0;
            a[1000][1001]=1;
            a[1001][1001]=1;
            a[1002][1001]=1;
            a[1000][1002]=1;
            a[1001][1000]=1;
        }
    }
    while(metai<100)
    {
        gime=0;
        mire=0;
        liko=0;
        v=0;
        for(int i=2;i<1999;i++)
        {
            for(int j=2;j<1999;j++)
            {
                k[i][j]=a[i-1][j-1]+a[i-1][j]+a[i-1][j+1]+a[i][j-1]+a[i][j+1]+a[i+1][j-1]+a[i+1][j]+a[i+1][j+1];
            }
        }
        for(int i=2;i<1999;i++)
        {
            for(int j=2;j<1999;j++)
            {
                if ((a[i][j]==0)&&(k[i][j]==3))
                {
                    a[i][j]=1;
                    gime=gime+1;
                }
            }
        }
        metai++;
    }
}
```

```

    }
    if (a[i][j]==1)
        v=v+1;
    if ((a[i][j]==1) && ((k[i][j]==2)||((k[i][j]==3))))
    {
        a[i][j]=1;
        liko=liko+1;
    }
    else
    {
        a[i][j]=0;
        mire=mire+1;
    }
}
cout<<"gime "<<gime<<"liko "<<liko-gime<<" mire "<<v-liko<<" metai
"<<metai<<endl;
metai=metai+1;
}
system("pause");
return 0;
}

```

Priedas B

```

clear A;
clear b;
clear D;
clear C;
n=3;
m=4;
for i=1:n*m
    for j=1:n*m
        A(i,j)=0;
        if i == j
            A(i,j) = 1;
        end;
        if i-j == 1
            if mod(i,n)>0
                A(i,j)=0;
            end;
            if mod(j,n)>0
                A(i,j)=1;
            end;
        end;
        if j-i == 1
            if mod(j,n)>0
                A(i,j)=0;
            end;
            if mod(i,n)>0
                A(i,j)=1;
            end;
        end;
        if abs(i-j)==n
            A(i,j)=1;
        end;
    end;
end;

```

```

        b(i)=1;
    end;
end;
sprend = [A transpose(b)];
format long;
C=rref(sprend);
for i=1:n*m
    D(i)=C(i,n*m+1);
end;
rangas=rank(A)
determinantas=det(A)

```

C priedas

x+z+v	y-v	-y-z	-x+v	z-v	x	y-z+v	-y-v	-x+z	v	x-z-v	y	-y+z+v	-x-v	z	x+v	y+z-v	-y	-x-z+v	-v	x+z	y+v	-y-z-v	-x	z+v	x-v	y-z	-y+v	-x-z+z-v
-x-y-z	-x	x+z	x+y	-z	-x-y	-x+z	x	x+y-z	0	-x-y+z	-x	x-z	x+y	-z	-x-y	-x-z	x	x+y+z	0	-x-y-z	-x	x+z	x+y	-z	-x-y	-x+z	x	x+y-z
x+y-v	x+v	-x	-x-y-v	v	x+y	x-v	-x+v	-x-y	-v	x+y+v	x	-x-v	-x-y+v	0	x+y-v	x+v	-x	-x-y-v	v	x+y	x-v	-x+v	-x-y	-v	x+y+v	x	-x-v	-x-y+v
-x+z	-y	y-z	x	z	-x	-y-z	y	x+z	0	-x-z	-y	y+z	x	-z	-x	-y+z	y	x-z	0	-x+z	-y	y-z	x	-z	-x	-y-z	y	x+z
-z+v	-v	z	v	-z-v	0	z+v	-v	-z	v	z-v	0	-z+v	-v	z	v	-z-v	0	z+v	-v	-z	v	z-v	0	-z+v	-v	z	v	-z-v
x	y	-y	-x	0	x	y	-y	-x	0	x	y	-y	-x	0	x	y	-y	-x	0	x	y	-y	-x	0	x	y	-y	-x
-x-y+z-v	-x+v	x-z	x+y-v	z+v	-x-y	-x-z-v	x+v	x+y+z	-v	-x-y-z+v	-x	x+z-v	x+y+v	-z	-x-y-v	-x+z+v	x	x+y-z-v	v	-x-y+z	-x-v	x-z+v	x+y	z-v	-x-y+v	-x-z	x-v	x+y+z+v
x+y-z	x	-x+z	-x-y	-z	x+y	x+z	-x	-x-y-z	0	x+y+z	x	-x-z	-x-y	z	x+y	x-z	-x	-x-y+z	0	x+y-z	x	-x+z	-x-y	-z	x+y	x+z	-x	-x-y-z
-x+v	-y-v	y	x+v	-v	-x	-y+v	y-v	x	v	-x-v	-y	y+v	x-v	0	-x+v	-y-v	y	x+v	-v	-x	-y+v	y-v	x	v	-x-v	-y	y+v	x-v
-z	0	z	0	-z	0	z	0	-z	0	z	0	-z	0	-z	0	-z	0	z	0	-z	0	z	0	z	0	z	0	-z
x-z-v	y+v	-y+z	-x-v	-z+v	x	y+z-v	-y+v	-x-z	-v	x+z+v	y	-y-z-v	-x+v	z	x-v	y-z+v	-y	-x+z-v	v	x-z	y-v	-y+z+v	-x	-z-v	x+v	y+z	-y-v	-x-z+v
-x-y	-x	x	x+y	0	-x-y	-x	x	x+y	0	-x-y	-x	x	x+y	0	-x-y	-x	x	x+y	0	-x-y	-x	x	x+y	0	-x-y	-x	x	x+y
x+y+z+v	x-v	-x-z	-x-y+v	z-v	x+y	x-z+v	-x-v	-x-y+z	v	x+y-z-v	x	-x+z+v	-x-y-v	-z	x+y+v	x+z-v	-x	-x-y-z+v	-v	x+y+z	x+v	-x-z-v	-x-y	z+v	x+y-v	x-z	-x+v	-x-y+z-v
-x-z	-y	y+z	x	-z	-x	-y+z	y	x-z	0	-x+z	-y	y-z	x	z	-x	-y-z	y	x+z	0	-x-z	-y	y+z	x	-z	-x	-y+z	y	x-z
-v	v	0	-v	v	0	-v	v	0	-v	v	0	-v	v	0	-v	v	0	-v	v	0	-v	v	0	-v	v	0	-v	v
x+z	y	-y-z	-x	z	x	y-z	-y	-x+z	0	x-z	y	-y+z	-x	-z	x	y+z	-y	-x-z	0	x+z	y	-y-z	-x	-z	x	y-z	-y	-x+z
-x-y-z+v	-x-v	x+z	x+y+v	-z-v	-x-y	-x+z+v	x-v	x+y-z	v	-x-y+z-v	-x	x-z+v	x+y-v	z	-x-y+v	-x-z-v	x	x+y+z+v	-v	-x-y-z	-x+v	x+z-v	x+y	-z+v	-x-y-v	-x+z	x+v	x+y-z-v
x+y	x	-x	-x-y	0	x+y	x	-x	-x-y	0	x+y	x	-x	-x-y	0	x+y	x	-x	-x-y	0	x+y	x	-x	-x-y	0	x+y	x	-x	-x-y
-x+z-v	-y+v	y-z	x-v	z+v	-x	-y-z-v	y+v	x+z	-v	-x-z+v	-y	y+z-v	x+v	-z	-x-v	-y+z+v	y	x-z-v	v	-x+z	-y-v	y-z+v	x	z-v	-x+v	-y-z	y-v	x+z+v
-z	0	z	0	-z	0	z	0	-z	0	z	0	-z	0	z	0	-z	0	z	0	-z	0	z	0	-z	0	z	0	-z
x+v	y-v	-y	-x+v	-v	x	y+v	-y-v	-x	v	x-v	y	-y+v	-x-v	0	x+v	y-v	-y	-x+v	-v	x	y+v	-y-v	-x	v	x-v	y	-y+v	-x-v
-x-y+z	-x	x-z	x+y	-z	-x-y	-x-z	x	x+y+z	0	-x-y-z	-x	x+z	x+y	-z	-x-y	-x+z	x	x+y-z	0	-x-y+z	-x	x-z	x+y	z	-x-y	-x-z	x	x+y+z
x+y-z-v	x+v	-x+z	-x-y-v	-z+v	x+y	x+z-v	-x+v	-x-y-z	-v	x+y+z+v	x	-x-z-v	-x-y+v	z	x+y-v	x-z+v	-x	-x-y+z-v	v	x+y-z	x-v	-x+z+v	-x-y	-z-v	x+y+v	x+z	-x-v	-x-y+z+v
-x	-y	y	x	0	-x	-y	y	x	0	-x	-y	y	x	0	-x	-y	y	x	0	-x	-y	y	x	0	-x	-y	y	x
z+v	-v	-z	v	z-v	0	-z+v	-v	z	v	-z-v	0	z+v	-v	-z	v	z-v	0	-z+v	-v	z	v	-z-v	0	z+v	-v	-z	v	z-v
x-z	y	-y+z	-x	-z	x	y+z	-y	-x-z	0	x+z	y	-y-z	-x	z	x	y-z	-y	-x+z	0	x-z	y	-y+z	-x	-z	x	y+z	-y	-x-z
-x-y-v	-x+v	x	x+y-v	v	-x-y	x+v	x+y	-v	-x-y+v	-x	x-v	x+y+v	0	-x-y-v	-x+v	x	x+y-v	v	-x-y	-x-v	x+v	x+y	-v	-x-y+v	-x	x-v	x+y+v	
x+y+z	x	-x-z	-x-y	z	x+y	x-z	-x	-x-y+z	0	x+y-z	x	-x+z	-x-y	-z	x+y	x+z	-x	-x-y-z	0	x+y+z	x	-x-z	-x-y	-z	x+y	x-z	-x	-x-y+z
-x-z+v	-y-v	y+z	x+v	-z-v	-x	-y+z+v	y-v	x-z	v	-x+z-v	-y	y-z+v	x-v	z	-x+v	-y-z-v	y	x+z+v	-v	-x-z	-y+v	y+z-v	x	-z+v	-x-v	-y+z	y+v	x-z-v

Priedas D

```

#include<stdio.h>
#include<stdlib.h>
int R,C;
int i, j;
int mat[100][100];
void display( int, int);
void input( int, int);
int Rangas(int , int);
void swap(int, int, int);
void swap( int r1,int r2, int col)
{
    for( i = 0; i < col; i++)
    {
        int k = mat[r1][i];
        mat[r1][i] = mat[r2][i];

```

```

        mat[r2][i] = k;
    }
}
int Rangas(int r1, int c1)
{
    int r, c;
    for(r = 0; r < c1; r++)
    {
        display(R,C);
        if( mat[r][r] )
            for(c = 0; c < r1; c++)
                if(c != r)
                {
                    float ratio = mat[c][r] / mat[r][r];
                    for( i = 0; i < c1; i++)
                        mat[c][i] -= ratio * mat[r][i];
                }
            else
                1;
        else
        {
            for(c = r+1 ; c < r1; c++)
                if (mat[c][r])
                {
                    swap(r,c,c1);
                    break ;
                }
            if(c == r1)
            {
                -- c1;
                for(c = 0; c < r1; c++)
                    mat[c][r] = mat[c][c1];
            }
            --r;
        }
    }
    return c1;
}
void display( int row, int col)
{
    for(i = 0; i < row; i++)
    {
        for(j = 0; j < col; j++)
        {
            1;
        }
        1;
    }
}
void input( int row, int col)
{
    int value,saknis;
    saknis=4;
    for(i = 0 ; i< row; i++)
    {
        for(j = 0 ; j<col; j++)
        {
            if ((i==j) || (abs(i-j)==1) || (abs(i-j)==saknis))
            {
                mat[i][j] = 1;
            }
        }
    }
}

```

```

    if ((abs(i-j)==1)&&((i+j+1)%saknis==0)&&((i*j)%saknis==0))
        {
            mat[i][j] = 0;
        }
    }
}
void main()
{
    int rank;
    printf("\n Lenteles matmenys :");
    scanf("%d", &R);
    input(R, R);
    display(R,R);
    rank = Rangas(R, R);
    printf("\n Matricos rangas yra : %d", rank);
    scanf("%d", &R);
}

```