

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики



Дослідження стохастичної поведінки клітинних автоматів

**Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки» – 122**

Керівник курсової роботи
доцент Чорней Р.К

_____ (підпис)
“ ____ ” _____ 2021 р.

Виконав студент
Войцеховський Є.О
“ ____ ” _____ 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики,
Доцент., к. ф.-м. н. С.С.Гороховський
(підпис)

„____” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Войцеховському Євгенію Олександровичу
факультету інформатики 4 курсу бакалаврської програми

ТЕМА: Дослідження стохастичної поведінки клітинних автоматів

Зміст ТЧ до курсової роботи:

1. Індивідуальне завдання
2. Календарний план
3. Анотація
4. Вступ
5. Теоретична частина
6. Практична частина
7. Висновки
8. Список літератури

Дата видачі „____” _____ 2020 р. Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Дослідження стохастичної поведінки клітинних автоматів

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	15.10.2020	
2.	Огляд літератури за темою роботи.	12.01.2021	
3.	Проведення досліджень	28.01.2021	
3.	Опис результатів дослідження	10.02.2021	
4.	Аналіз отриманих результатів з керівником	08.03.2021	
6.	Корегування роботи	28.03.2021	
7.	Створення презентації та написання доповіді.	30.03.2021	
8.	Остаточне оформлення пояснювальної роботи та слайдів.	11.04.2021	
9.	Захист курсової роботи	18.04.2021	

Студент _____
 Керівник _____
 “ ”

ЗМІСТ

Анотація	5
ВСТУП.....	6
Розділ 1. Теоретична частина	7
1.1 Загальні поняття	7
1.2 Різновиди клітинних автоматів	8
1.3 Гра «Життя».....	12
1.4 Висновки до розділу 1	15
Розділ 2. Практична частина	16
2.1 Створення програмного додатку	16
2.2 Тестування програмного додатку.....	20
2.3 Висновки до розділу 2	28
ВИСНОВОК	29
СПИСОК ЛІТЕРАТУРИ	30
ДОДАТКИ.....	32

Анотація

У цій роботі розглянуто поняття клітинний автомат та його різновиди. Створений додаток на мові програмування Java для представлення та керування роботою клітинних автоматів. Проведений аналіз клітинних автоматів з використанням створеного додатку.

ВСТУП

З розвитком технологій все більше зростає потреба в методах та алгоритмах ефективної обробки даних. Особливо гострою ця проблема стає в областях, де потрібно якомога швидше та простіше обробити великі масиви даних або змодельовати процеси, у яких беруть участь тисячі об'єктів.

Останнім часом значного прогресу у виконанні таких завдань досягли клітинні автомати, що поступово проникають в усі сфери життя. Уже зараз клітинні автомати використовуються у теоретичній біології, криптографії [14], соціальних науках, хімії, моделюванні фізичних явищ та мікроструктур тощо. К роботі [1] клітинні автомати використовувалися для пошуку контурів зображень. У праці [2] був розроблений клітинний автомат для копіювання кольорових зображень.

Вигляд набору клітинних автоматів у вигляді решітки робить можливим їх обробку методами паралельного програмування та розподілених обчислень, що зменшує навантаження на обчислювальний пристрій.

Можливість одночасної зміни станів групи клітинних автоматів робить їх ефективним засобом для моделювання перебігу фізичних процесів [15]. Клітинні автомати використовують для створення моделей поведінки рідин та газів у різних середовищах, для моделювання систем, що складаються з дуже великої кількості частинок зі складними взаємодіями між собою[3].

Мета цієї курсової – розглянути та проаналізувати різні види клітинних автоматів. Для цієї цілі було створене відповідне програмне забезпечення.

Текстова частина цієї курсової роботи складається з двох розділів.

У першому розділі описується теоретична інформація про клітинні автомати та розглядаються їх різновиди.

У другому розділі розглянутий програмний додаток для побудови та взаємодії з клітинними автоматами та проведені різні експерименти з ним.

РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА

1.1 Загальні поняття

Клітинний автомат (КА) – це дискретна математична модель, що визначається набором однорідних об’єктів-клітинок та правил, за якими ці об’єкти взаємодіють між собою. У загальному КА можна представити наступною формулою[4] $KA = \langle Z^n, (N_1, \dots, N_n), A, Y, \sigma \rangle$.

Z^n – це n -вимірний простір цілочисельних клітинок решітки, (N_1, \dots, N_n) – вектор розміру решітки, A – скінченна кількість станів кожної клітинки, Y – околиці клітинки та σ – функція переходу. Далі розглянемо кожну компоненту більш детально.

В багатьох задачах переважно розглядаються не поодинокі КА, а наявна сітка певної розмірності, в якій кожна клітина є КА. Найчастіше розглядаються одновимірні або двовимірні сітки, бо їх найпростіше представити у графічному вигляді. Але ніщо не обмежує використання в задачах вищих розмірностей.

Кількість КА та розмір решітки обмежується лише можливостями обчислювального пристрою. Причиною цього є те, що зі зростанням розміру сітки зростає і час, необхідний для повного проходу сітки та переходу на наступний етап. Це завдання певною мірою вирішується розпаралелюванням обчислень, але і в цього методу є свої недоліки, зокрема збільшується час на спілкування між процесорами [9].

Кожний КА має скінченний набір символів, які визначають його поточний стан. Переважно використовуються цілі невід’ємні числа. У найпростіших автоматів є лише два стани, які можна представити «одиницею» та «нулем».

З наявності сітки клітин впливає те, що кожен КА має певну кількість сусідів з набору цієї сітки, з якими він взаємодіє в ході своєї роботи. Кількість сусідів залежить від розмірності сітки та форми клітин. Двовимірні

сітка дозволяє просто представити клітини у вигляді квадратів або інших правильних фігур на зразок шестикутників чи рівносторонніх трикутників.

В процесі обчислень КА змінює свій стан на основі певного набору правил, що є спільними для всіх КА з певного набору. Саме ці правила визначають, який стан КА матиме на наступному кроці. Правила базуються на порівнянні стану автомату з його сусідами й на основі цього визначається новий стан КА.

1.2 Різновиди клітинних автоматів

Найпоширенішою класифікацією КА є класифікація автоматів за їх поведінкою. У 2002 році у своїй книзі «A New Kind of Science» [8] Стівен Вольфрам запропонував класифікацію КА на чотири класи:

- Клас 1. Автомати цього класу характеризуються тим, що всі їх клітини швидко сходяться до стабільного стану, що не міняється з ходом часу.
- Клас 2. Автомати цього класу швидко стабілізуються, але можуть утворювати періодичні коливання.
- Клас 3. До цього класу відносять автомати, що утворюють хаотичні структури, що не мають періоду повторення. Автомати цього класу дуже чутливі до збурень, що сильно впливають на їх поведінку в майбутньому.
- Клас 4. Цей клас об'єднує найскладніші автомати, в результаті роботи яких виникають складні структури, що впродовж тривалого часу не стабілізуються.

Ця класифікація не є чітко визначеною, оскільки деякі автомати за різними критеріями можна віднести до одного або іншого класу. В роботі [5] розглядається схожа класифікація КА за їх поведінкою, що ділить автомати на наступні шість класів:

- Клас 1. Система переходить в один з кількох можливих кінцевих незмінних станів.

- Клас 2. Система переходить в стабільний стан та може переміщуватися по полі.
- Клас 3. Система зберігає свій стан та дозволяє поширення й просту взаємодію інформації.
- Клас 4. Інформація необмежено поширюється та взаємодіє між собою, утворюючи прості структури.
- Клас 5. Інформація необмежено поширюється та має складну взаємодію, що виглядає хаотичною.
- Клас 6. Інформація поширюється шляхом утворення, переміщення та взаємодії локальних структур.

Окрім класифікації за поведінкою виділяють й інші класифікації на основі певних характеристик КА. Клітинні автомати можна класифікувати за:

- Розмірністю сітки: одномірні, двомірні, тривимірні тощо.
- Кількістю станів: бінарні, тернарні та інші.
- Типом виконання: синхронні та асинхронні.
- Видом околиць.

Варто звернути увагу на останню класифікацію за видом околиць. Існує два головні види околиць: **околиця фон Неймана** та **околиця Мура**. На рисунку 1.1 показані ці два види околиць.

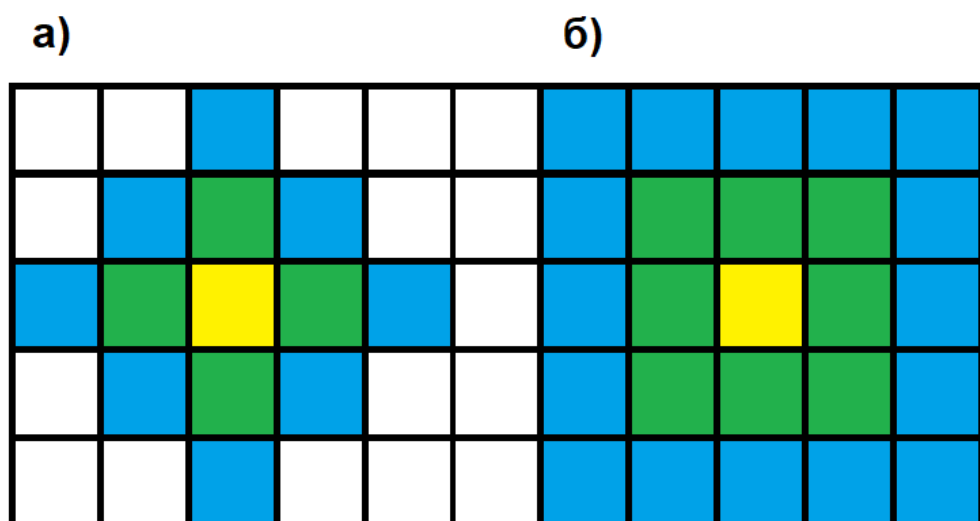


Рисунок 1.1 Види околиць

Рисунок 1.1 а) зображує околицю фон Неймана навколо жовтої клітинки. Околиці можуть мати ранг, що показує, на скільки клітин продовжується околиця. Зелені клітини на рисунку зображують околицю фон Неймана першого рангу, а сині разом з зеленими – околицю другого рангу. До цієї околиці належать клітини, що мають спільне ребро з початковою клітиною або ребро з клітинами, що є сусідами цієї клітини (зокрема це зображено синіми клітинами другого рангу, що мають спільні ребра зеленими клітинами, що є прямими сусідами жовтої клітини).

Рисунок 1.1 б) зображає околицю Мура навколо жовтої клітини. Колірні позначення тут аналогічні до першого малюнку. Ця околиця також може мати свій ранг. В околицю Мура входять клітини, що мають спільну вершину з початковою клітиною або її сусідами.

Ці околиці можуть використовуватися в сітках з будь-якими формами клітинок. В одномірній сітці можна визначити лише околицю фон Неймана, а у двовимірній сітці визначається околиця фон Неймана та околиця Мура. Окрім цих околиць можна задати інші принципи визначення сусідів клітини. Наприклад, у роботі [6] вводиться поняття «ефір», куди можуть посилати сигнали всі автомати та отримувати сигнали від інших автоматів, що дозволяє автомату керувати станом будь-якого автомату на відстані від себе.

Поведінка КА залежить від станів його сусідів та правил переходу, що дозволило називати окремі автомати. У 1983 році Стівен Вольфрам у своїй праці [7] запропонував систему іменування одновимірних клітинних автоматів в залежності від правил та станів сусідів клітини й самої клітини. Зараз ця система називається **кодами Вольфрама**.

В основі ідеї коду Вольфрама лежить те, що таблицю, яка визначає новий стан кожного КА, можна представити у вигляді числа з k цифр у S -арній позиційній системі. S – це кількість станів, в яких може знаходитися кожний КА, а $k = S^{2n+1}$ – це кількість можливих станів околиці клітини, де n – радіус околиці. Щоб визначити назву автомата за цією системою необхідно:

- Визначити всі можливі стани околиці даної клітинки.
- Розглянути кожну околицю як число та відсортувати ці числа за спаданням.
- Для кожного такого набору визначити стан, в який перейде дана клітинка.
- Помістити в ряд всі кінцеві стани з попереднього кроку та перетворити цей ряд у десяткове число. Це число і буде кодом Вольфрама.

Коди Вольфрама є простим методом іменування КА, що найчастіше використовується для одновимірних автоматів. На рисунку 1.2 представлені всі можливі комбінації одновимірних автоматів з двома станами, один з станів позначимо синім кольором, а інший – зеленим.

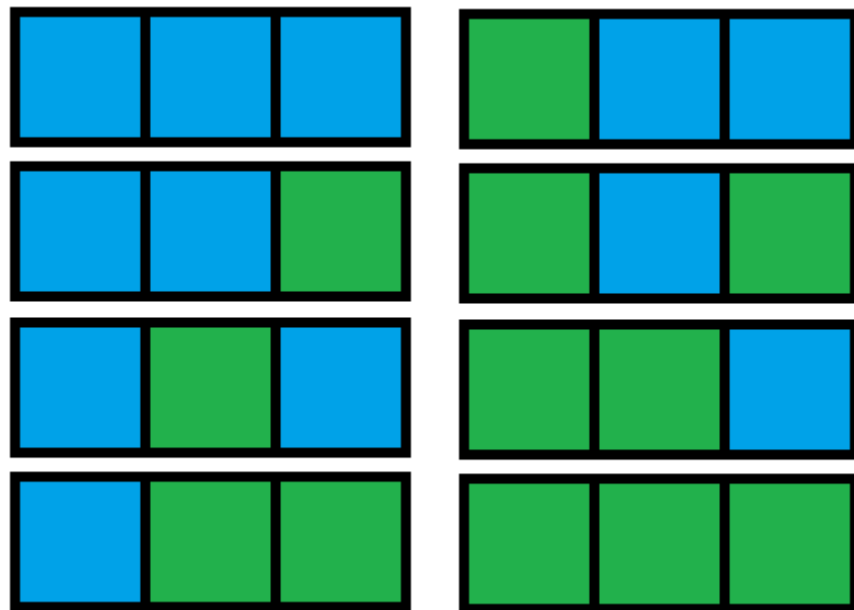


Рисунок 1.2 – Можливі околиці одновимірного КА

Як бачимо, КА такого типу може мати лише 8 комбінацій (кожна клітина має лише 2 стани і множення станів клітини та її двох сусідів дасть 8). Щоб знайти загальну кількість можливих КА необхідно кількість станів піднести до степені кількості комбінацій, що у цьому випадку перетвориться у формулу 2^8 , що в результат дасть 256 автоматів. Оскільки це число невелике, його зручно записувати кодами Вольфрама, якими називають

одновимірні КА відповідним числом від 0 до 255. На двохвимірній сітці з околицями фон Неймана або Мура кількість можливих автоматів дуже швидко зростає, що ускладнює їх іменування, хоч це і не заважає називати їх таким чином. У цій роботі я використовую коди Вольфрама для задавання різних одновимірних та двовимірних автоматів.

Продемонструю приклад утворення коду Вольфрама для 123 автомату за цією класифікацією. Нам потрібно перевести число 123 у двійкову систему, і ми отримаємо число 01111011. Заносимо ці дані до таблиці 1.

111	110	101	100	011	010	001	000
0	1	1	1	1	0	1	1

Таблиця 1 – формування коду Вольфрама

У нас утворився набір правил, що однозначно ідентифікує певний клітинний автомат. Для прикладу, якщо цей автомат матиме стан 1, і його сусіди перебуватимуть у стані 1, то на наступному кроці цей автомат перейде у стан 0, що відповідає першому стовпчику таблиці 1.

Спробуємо за прикладом одновимірних автоматів обрахувати кількість кодів Вольфрама для двовимірних КА та двох вже згаданих околиць. Околиця фон Неймана дасть 32 варіантів клітини та її чотирьох сусідів, що створить 2^{32} або ж 4294967296 комбінацій. Околиця Муна створює 512 варіантів клітин, що перетвориться в 2^{512} кількість автоматів. Отже, околиці фон Неймана та Мура дають надто велику кількість автоматів, щоб їх можна було проіменувати кодами Вольфрама.

1.3 Гра «Життя»

Спробою назвати КА з околиць Мура стало створення певного типу КА, що у 1970 році запропонував Джон Конвей. Цей автомат називається «гра життя». Це двовимірний автомат з двома станами, що діє за певними правилами, за якими його і називають. Клітини в цьому КА можуть бути живими (активними) або мертвими (неактивними). Кольори можуть

відрізнятися, але в цій роботі я позначатиму живі клітини чорним, а мертві – білим кольором. «Гра життя» базується на взаємодії живих клітин з мертвими.

Стандартна «гра життя» має наступні правила:

- Якщо в живої клітини є два або три живих сусіди, то вона продовжує жити.
- Якщо в живої клітини один або нуль сусідів, то вона помирає від «самотності».
- Якщо в живої клітини чотири або більше сусідів, то вона помирає від «перенаселення».
- Якщо у мертвої клітини є рівно три живих сусіда, то вона оживає.

Існують інші КА типу «гри життя», що будуються за подібними принципами, але вони відрізняються кількістю клітин, що помирають, оживають та продовжують жити.

Цей КА розглядався в багатьох роботах [9, 10, 11, 13]. Через це багато фігур, які можуть утворитися в ході роботи цього КА, отримали власні назви та їх поділили на категорії:

- Стійкі фігури.
- Періодичні фігури.
- Рухомі фігури.
- Гармати.
- Паротяги.
- Пожирачі.
- Довгожителі



Рисунок 1.3 – Гармата планерів

На рисунку 1.3 зображена гармата планерів, що через сталі проміжки часу утворює рухомі об'єкти планери. Завдяки існуванню таких об'єктів КА «гра життя» показує спроможність виконувати алгоритмічні дії. Це дозволило Полю Ренделю [12] у 2000 році створити на основі цього КА такий набір, що імітував роботу машини Тюрінга. Цей набір автоматів мав три можливих стани та міг оброблювати на стрічці три різних символи. Ця структура показана на рисунку 1.4.

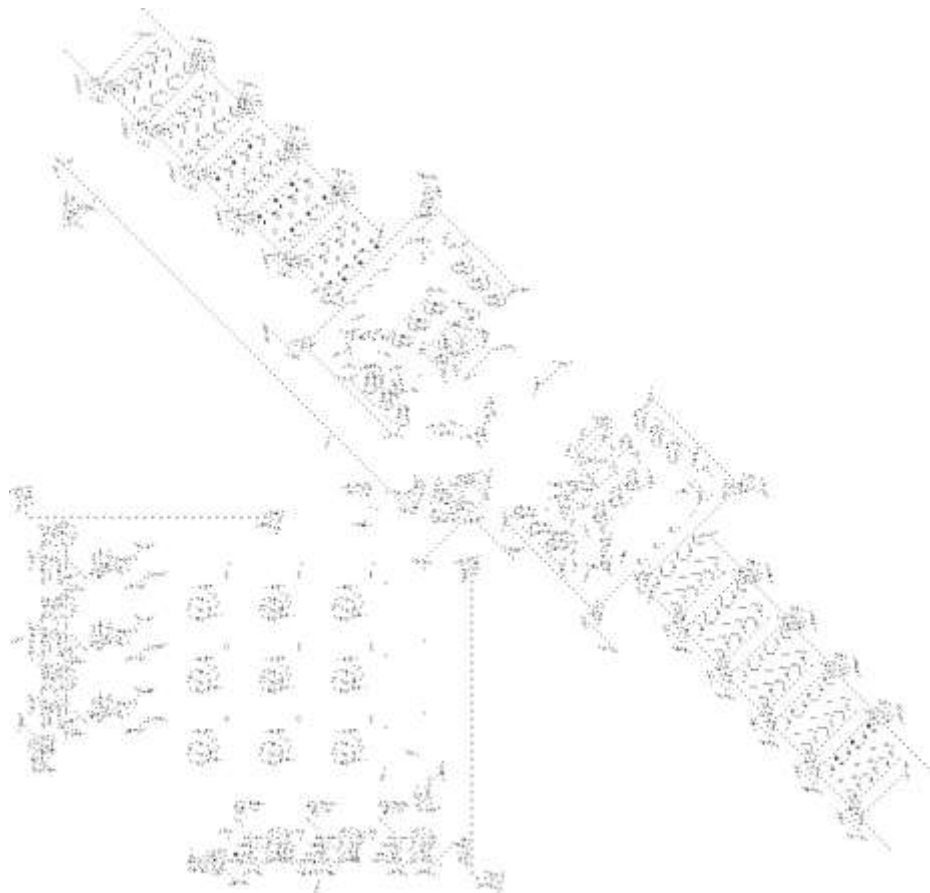


Рисунок 1.4 – Перша реалізація машини Тюрінга

1.4 Висновки до розділу 1

У цьому розділі було розглянуто основні поняття про КА та їх різні класифікації. Також тут було коротке введення в клітинний автомат «гра життя» та вказані його можливості.

РОЗДІЛ 2. ПРАКТИЧНА ЧАСТИНА

2.1 Створення програмного додатку

В якості індивідуального завдання до цієї роботи я реалізував програму, що дозволяє запускати різні КА та аналізувати результати. Для створення програми була використана мова програмування Java та засіб графічного інтерфейсу Swing. Програма була зібрана у файл з типом jar, який можна запустити програмними засобами мови Java.

Ця програма дозволяє зчитувати дані про автомати з текстових файлів розширення txt та запускати їх роботу з різними користувацькими налаштуваннями. Також користувач може створити власні правила для автомату «гра життя» та запускати його у програмі.

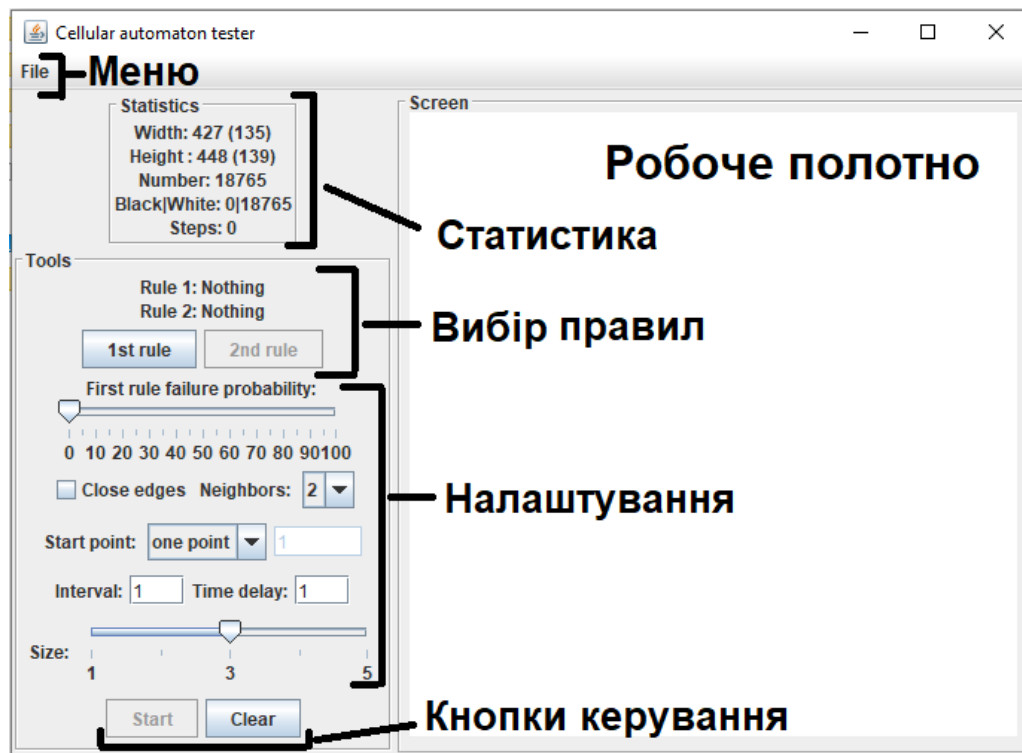


Рисунок 2.1 – Графічний інтерфейс програмного застосунку
Робоче вікно складається з трьох головних компонентів:

- робоче полотно;
- статистика;
- панель інструментів.

Робоче полотно. Цей елемент використовується для графічного представлення покрокової роботи різних КА з плином часу. Користувач може змінити розміри вікна програми, що змінить розміри робочого полотна.

Статистика. Цей елемент показує поточну статистику про робоче полотно. Зокрема тут показана ширина та висота робочого полотна в пікселях, а потім в дужках після цих чисел показана кількість автоматів, що вміститься у відповідну ширину або довжину. Далі на основі кількості автоматів по ширині та висоті робочого полотна обраховується загальна кількість КА. Коли запускається автомат, то відображається та змінюється кількість чорних та білих клітинок, а також показується на якому кроці зараз КА.

Панель інструментів. Цей елемент дозволяє користувачу взаємодіяти з програмою та налаштовувати КА. Ця панель ділиться на три змістовні частини:

- вибір правил;
- налаштування;
- кнопки керування.

У частині **вибір правил** користувач повинен обрати хоча б одне правило, за яким повинні працювати клітинні автомати, що покривають квадратною сіткою область робочого полотна. При виборі першого правила зміниться напис «Rule 1: Nothing» на «Rule 1: File_Name.txt» та користувач зможе запустити роботу КА. Після обрання першого правила користувач може обрати друге правило, після чого подібним чином зміниться заголовок «Rule 2: Nothing», що покаже ім'я файлу. На рисунку 2.2 показане вікно вибору файлу.

Ця програма працює з текстовими файлами, в яких міститься текст у форматі [кількість_сусідів][код Вольфрама]. Приклад для файлу з 4 сусідами:

```
410001100110000110001110010011000
```

Зауваження. Якщо користувач обрав два файли, то вони повинні описувати правила однакових околиць. Тобто, у двох файлах повинно бути

однакове число сусідів. Якщо один файл описує правила для двох сусідів, а інший для чотирьох, то при спробі запустити програму кнопкою Start програма закриється.

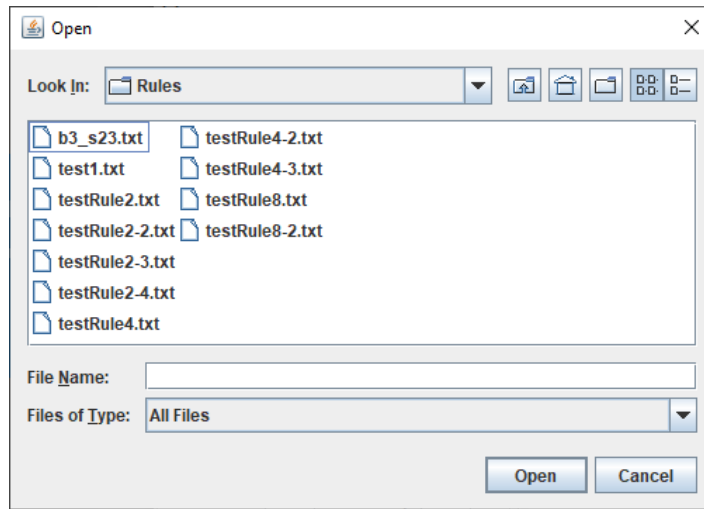


Рисунок 2.2 – Меню вибору файлу

Налаштування. При роботі КА користувач має змогу налаштувати багато параметрів за допомогою цього елемента. Першим йде елемент вибору ймовірності спрацювання правила. За замовчуванням після обрання першого правила воно спрацює з ймовірністю 100%. Користувач може обрати ймовірність з кроком у 5%. Якщо користувач обрав друге правило, то цей елемент керування починає відповідати за ймовірність спрацювання другого правила. Це дозволяє змішувати між собою два набори правил з певними ймовірностями спрацювання кожного з них.

Наступний елемент вибору «Close edges» дозволяє користувачу закрити межі робочого полотна. За замовчуванням робоче полотно зациклене (правий край полотна пов'язаний з лівим, а верхній з нижнім, згорнувши полотно у формі фігури тор). Якщо ж обрати цей елемент, то межі полотна закриваються, та під час роботи КА межі діятимуть таким чином, ніби за межами лежать білі клітинки, що не можуть змінювати свій стан, але на їх основі змінюють свій стан граничні КА.

Елемент «Neighbors» показує кількість сусідів у клітинки. Можливі варіанти: два, чотири або вісім. Взаємодія користувача з цим елементом

нічого не дасть. Цей елемент використовується для графічного представлення того, зі скількома сусідами зараз працює алгоритм. Програма автоматично зчитує кількість сусідів з файлів з правилами та відображає це число у цьому елементі.

Елемент «Start Point» дозволяє користувачу обрати кількість початкових точок, з яких почне роботу алгоритм. Є варіант одної точки, що з'явиться по центру робочого полотна, та варіант випадкового розміщення кількості точок, що вкаже користувач у текстовому полі праворуч. Це дозволяє користувачу підбирати кількість точок для різних результатів роботи КА.

Елемент «Interval» дозволяє користувачу вказати, з яким інтервалом він хоче бачити зміну робочого полотна. За замовчуванням користувач бачить кожний хід КА, але може змінити це значення на щось інше. Елемент «Time delay» після попереднього елемента дозволяє вказати затримку в часі між кроками КА.

Елемент «Size» дозволяє користувачу обрати розмір кожного КА від 1 до 5 пікселів. Зі зменшенням розміру КА збільшується кількість автоматів та зростає час, необхідний для їх обробки та переходу на наступний крок.

Кнопки керування. Користувач може двома кнопками керувати роботою КА. Кнопка «Start» запускає роботу алгоритму. Ця кнопка є неактивною, поки користувач не обрав хоча б одне правило роботи КА. Після натиснення почнеться обрахування станів КА на робочому полотні. Щоб зупинити обрахунок, користувач повинен натиснути на цю ж кнопку, що змінить свій напис на «Stop». Кнопка «Clear» використовується для очищення робочого полотна та обраних правил.

Окрім цих елементів у програмі є наступне меню, що зображене на рисунку 2.3 а). Це меню містить у собі кнопку «Close» для закриття програми та кнопку «Create», що відкріє додаткове вікно, що показане на рисунку 2.3 б). Це вікно дозволяє користувачу створити правило для клітинного автомату «гра життя». Для цього користувач повинен відмітити пташками умови, при

яких клітина народиться та коли вона залишиться живою, а потім натиснути на кнопку «Create rules». Потім створене правило можна обрати в програмі та запустити.

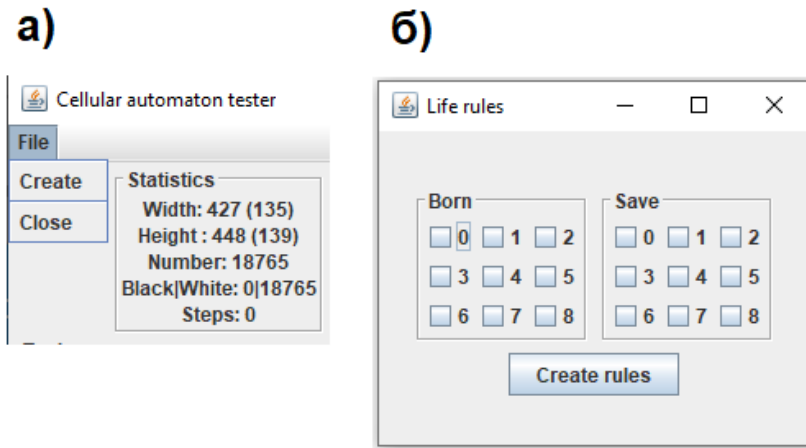


Рисунок 2.3 – Меню та створення правила

2.2 Тестування програмного додатку

Для початку протестуємо кілька правил одномірних КА. Автомати такого типу в моїй програмі будуються наступним чином. Ці автомати створюються в найнижчому рядку робочого полотна, а потім ця рядки поступово піднімаються вище по полотні, поки не доходять до верхньої межі та не зникають.

Моя програма працює на файлах, в яких за допомогою коду Вольфрама записаний відповідний автомат. Наприклад протестуємо автомат 3 з кодом Вольфрама 00000011.

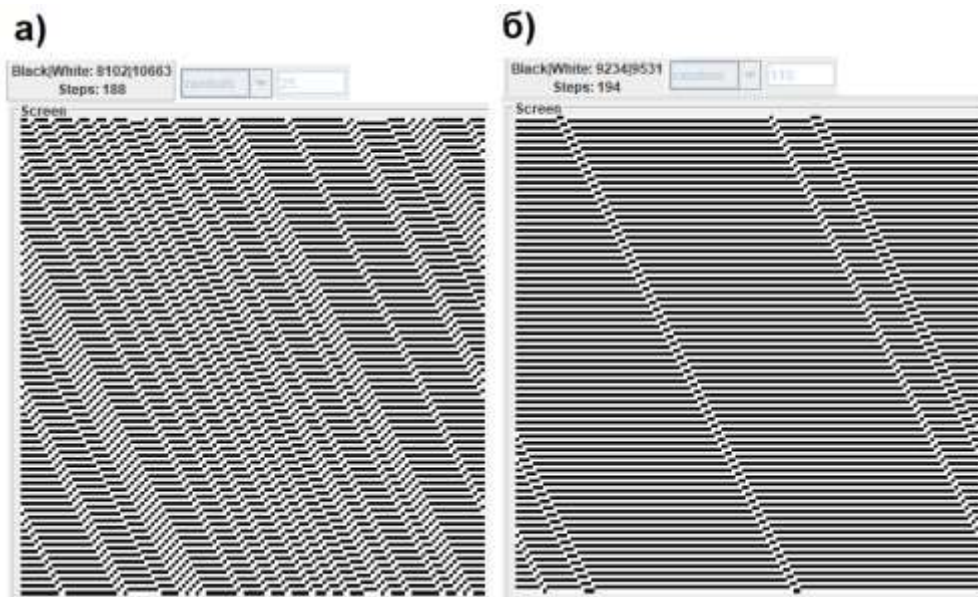


Рисунок 2.4 – Правило 3, зміна кількості початкових точок

Цей КА можна віднести до другого класу автоматів, оскільки він має сталий візерунок, але рухається по екрану. На рисунку 2.4 а) показаний автомат з меншою кількістю випадкових точок, а саме було взято 25 точок при ширині нижнього рядка в 135 точок, або нижній рядок було зайнято 18,5% випадковими точками. Бачимо, що при такій кількості точок відношення чорних клітинок до білих становить 8:10.

Але збільшення кількості початкових крапок не завжди впливає позитивно на результат або збурення автомату. З наближенням кількості випадкових початкових точок до ширини рядка на полотні з'являється все менше діагональних ліній, що і видно на рисунку 2.4 б). Тут поставлено 110 початкових точок або ж 81,5% від ширини рядка. При подальшому збільшенні кількості точок на полотні з'являється все менше і менше діагональних ліній, поки і зовсім їх не стає.

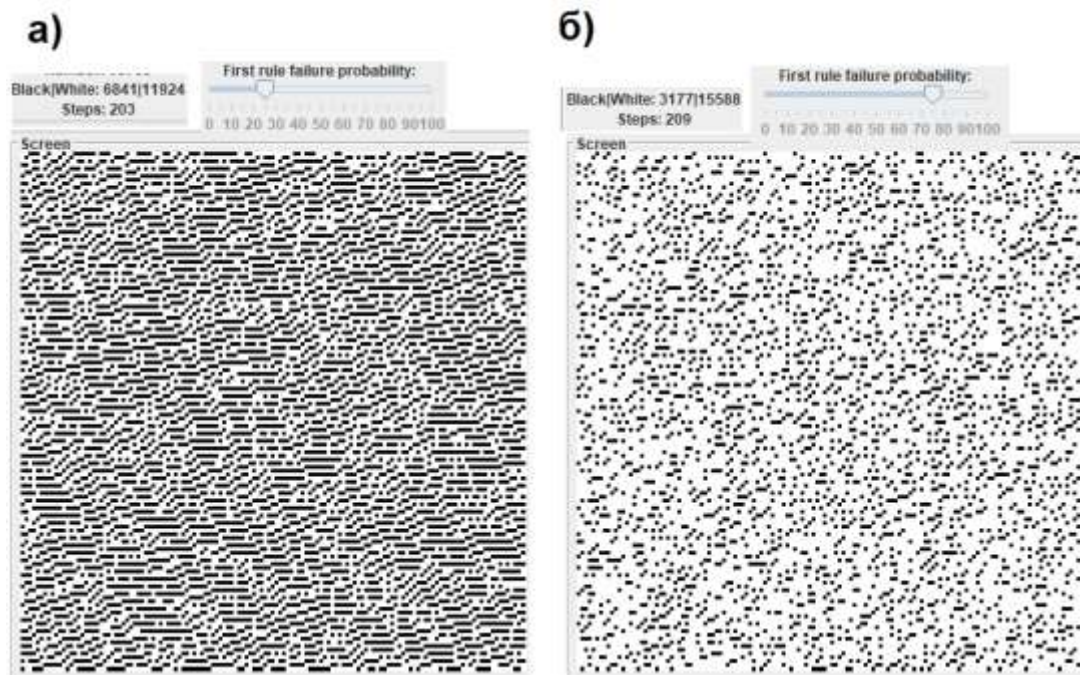


Рисунок 2.5 – Правило 3, зміна ймовірності

Спробуємо протестувати це ж правило, але змінивши ймовірність його спрацювання. На рисунку 2.5 а) та 2.5 б) визначено 25 початкових випадкових точок. На першому рисунку поставлена ймовірність спрацювання правил автомата 3 у 25%, а на другому – у 75%. Просто помітити, що при зменшенні ймовірності успішного спрацювання правила кількість чорних точок також зменшується, а малюнок більше не має чіткого узору, як було на рисунку 2.4. Якщо рисунок 2.4 можна було передбачити та обрахувати вручну, то на рисунку 2.5 використаний генератор випадкових чисел, що робить результат непередбачуваним.

Тепер протестуємо закриття границь полотна на іншому правилі. Візьмемо правило 22 з кодом Вольфрама 00010110. На рисунку 2.6 обрано дві випадкові точки. Це правило дуже чутливе до збурень. В залежності від початкового положення випадкових точок візерунок або залишався більш-менш правильним, або перетворювався в хаос, причому ймовірність хаотичного розподілу була вищою. На рисунку 2.6 а) видно, що ці правильні трикутники продовжуються з лівого краю полотна на правий і навпаки. Тоді як на 2.6 б) можна помітити, що на межах полотна візерунок обривається та його поширення зупиняється.

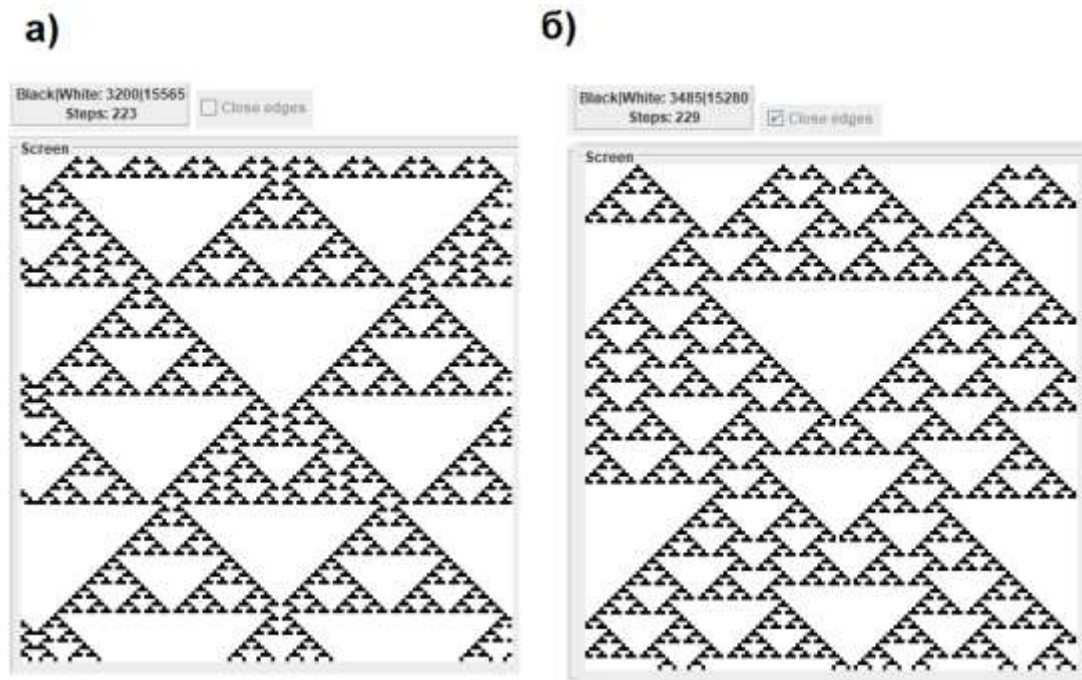


Рисунок 2.6 – Правило 22, закриття границь

Спробуємо поєднати правило 22 з правилом 3. Для цього ми спершу обираємо один файл з правилами, а потім обираємо інший файл в якості другого файлу. Як бачимо, напис над шкалою для обирання ймовірності змінився на «Chance of second rule». На рисунку 2.7 а) ймовірність спрацювання правила 3 склала 25%, а на рисунку 2.7 б) – 75%. У правилі 22 будувалися трикутники, і ми бачимо, що на першому з цих двох рисунків переважають трикутники, хоч і неправильної форми. Коли ж ймовірність появи трикутників зменшилася до 25%, як на другому рисунку, то трикутників вже майже немає, а переважають лінії з правила 3. Така суміш двох правил на додачу до ймовірності дуже зменшує можливість передбачити результат та дозволяє з передбачуваних правил зробити хаотичні. Кожен з цих рисунків також починався з одної випадкової точки, але додавання правила 3 значно збільшила початкову кількість точок, оскільки це правило будує горизонтальні лінії. У Додатку 1 показана суміш правила 110 та 30 за подібним принципом.

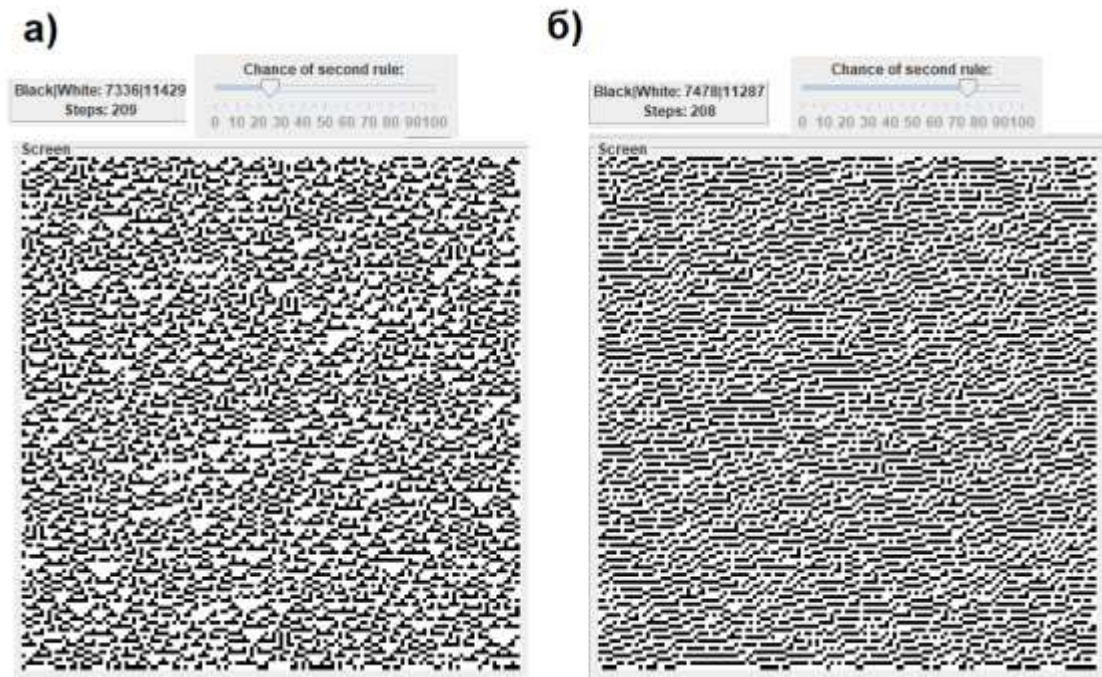


Рисунок 2.7 – Суміш правила 22 та 3

Протестуємо правило, що оглядає не двох сусідів, а чотирьох. Тобто, правило, що сусідами якого є клітини з околиці фон Неймана. Якщо це правило у формі коду Вольфрама, а потім перевести у десяткову систему, то вийде число 2361597081. Це правило малює білими пікселями по чорному фоні, як бачимо на малюнку 2.8 а). Але в таких алгоритмах, що малюють білим по чорному є один недолік. Якщо не закривати границі полотна, то КА поступово заповнює все поле, на що потрібно приблизно 250 кроків. Але спробуємо закрити границі і результат бачимо на рисунку 2.8 б). Причина такого явища полягає в тому, що в мене граничні КА розглядають клітинки за межами полотна в якості білих, через це на повне заповнення малюнку до хаотичного рівня потрібно на сто кроків менше. Вихід з такої ситуації – додати користувачу можливість обирати, який колір бачитимуть КА за межами границі полотна, але в цій програмі це прописано напяму в коді. В обох рисунках користувач обирає лише одну початкову точку посередині екрана, а вийшло, що в другому рисунку до початкових точок потрапили і всі КА з нижньої, лівої та правої границі.

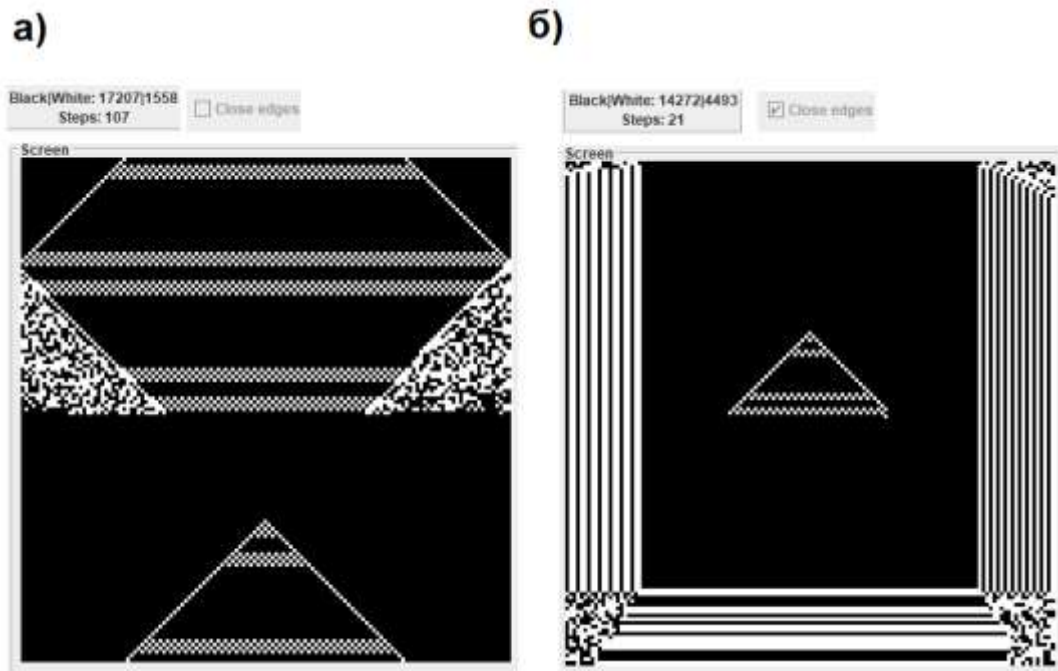


Рисунок 2.8 – Правило 2361597081 з закриванням границь

Додамо ще одне правило з чотирма сусідами 2361597080. Це правило відрізняється від попереднього лише на останній біт. Але по рисунку 2.9 видно, наскільки ці правила відрізняються по зовнішньому вигляду. Для кожного правила було взято 1500 початкових випадкових точок та результат після 207 кроків видно на рисунках. А всьому причиною те, що був змінений останній біт запису клітинного автомату у коді Вольфрама. Останній біт відповідає за те, який колір буде переважати на малюнку: чорний або білий. Якщо цей біт дорівнює одиниці, як у правилі 2361597081, то виходить білий малюнок на чорному фоні. Якщо ж цей біт дорівнює нулю, як у правилі 2361597080, то виходить чорний малюнок на білому фоні. У такій друкованій роботі не можна показати це, але на рисунку 2.9 а) складається ілюзія руху клітинок вгору, а на рисунку 2.9 б) все рухається праворуч.

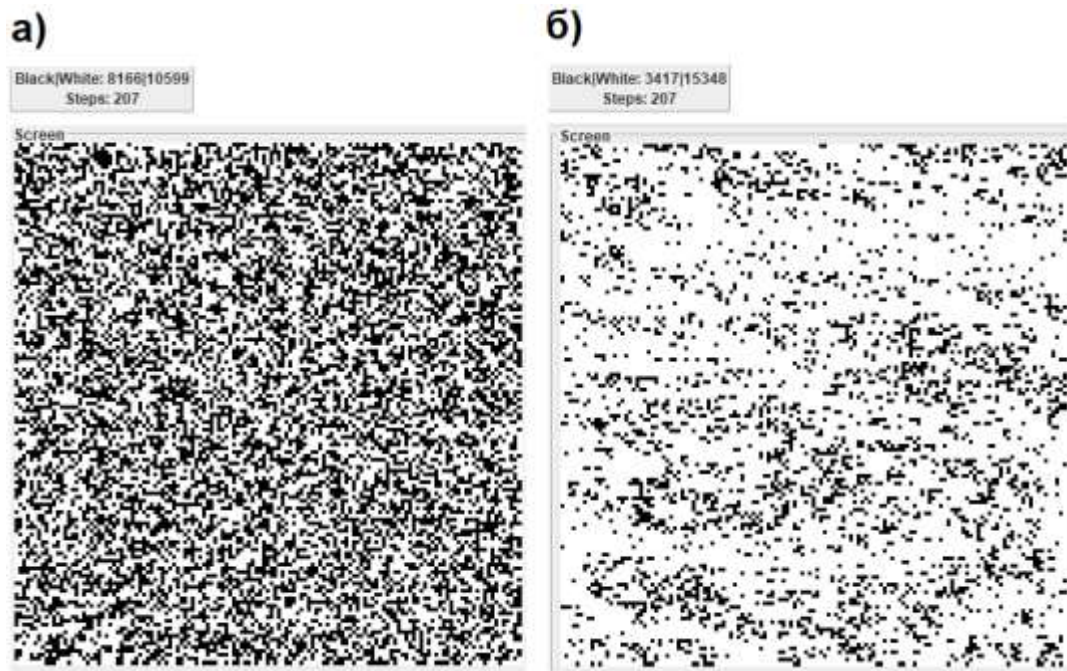


Рисунок 2.9 – Правило 2361597081 та 2361597080

Клітинні автомати з околицями Мура не дуже відрізняються по зовнішньому вигляду від КА з околицями фон Неймана. Приклад порівняння кількох автоматів з околицями Мура та їх суміш представлена у Додатку 2.

Більш цікавим автоматом з околицями Мура є «гра життя». Створена в межах цієї роботи програма надає зручний інтерфейс для створення та тестування різних варіацій цього автомату. Завдяки інтерфейсу користувач сам обирає, яким буде його автомат такого типу, і це не прописане десь в коді, що ніяк не зміниш. Це надає широкі можливості тестування таких автоматів. У роботі [10] автори дають перелік різних варіацій КА за принципами «гри життя». Автомати такого типу мають спеціальні правила називання. Так звичайна гра життя за цими правилами буде називатися B3/S23, де B – народжений (born), а S – продовжує жити (save). Вважається, що у всі інші випадки, що не описані назвою автомата, клітини помирають.

Розглянемо автомат B3/S45678 під назвою «Корал» та автомат B35678/S5678 під назвою «Діадема». При малому відсотку живих клітин автомат Корал швидко згасає. Для його стабільної роботи необхідно як мінімум 20% живих клітин від загальної кількості КА. При відсотку 30% живих клітин на початку цей автомат починає досить стабільно рости, як

показано на рисунку 2.10 а). Автомат Діадема є більш вимогливим до кількості клітин. Він ледь починає формуватися при 30% живих клітин, а на 40% більш широко поширюється по робочому полотні, як видно на рисунку 2.10 б). І лише на 50% живих клітин цьому автомату вдається захопити більше половини робочого полотна.



Рисунок 2.10 – Правило Корал та Діадема

У Додатку 3 показана спроба об'єднати ці правила. Для цього прикладу взято 37% живих клітин. При такій кількості якщо ймовірність Діадеми значно більша за Корал, то автомат швидко зникає. Але якщо у цій суміші більша ймовірність появи правил Коралу (тести проводилися на 20% появи правил Діадеми), то будуються стійкі видовжені нарости, що формою нагадують структури Діадеми. При цьому зникають всі білі клітинки всередині коралів, які можна побачити на рисунку 2.10 а).

Розглянемо стандартне правило для «гри життя» B3/S23. Це правило дуже просто формує стійкі структури. Для початку формування таких структур достатньо лише 1,5% живих клітин на початку, як показано на рисунку 2.11 а). При 10% живих клітин утворюється набагато більше стійких структур (рисунок 2.11 б), кількість яких не набагато збільшується зі збільшенням кількості живих клітин на початку.

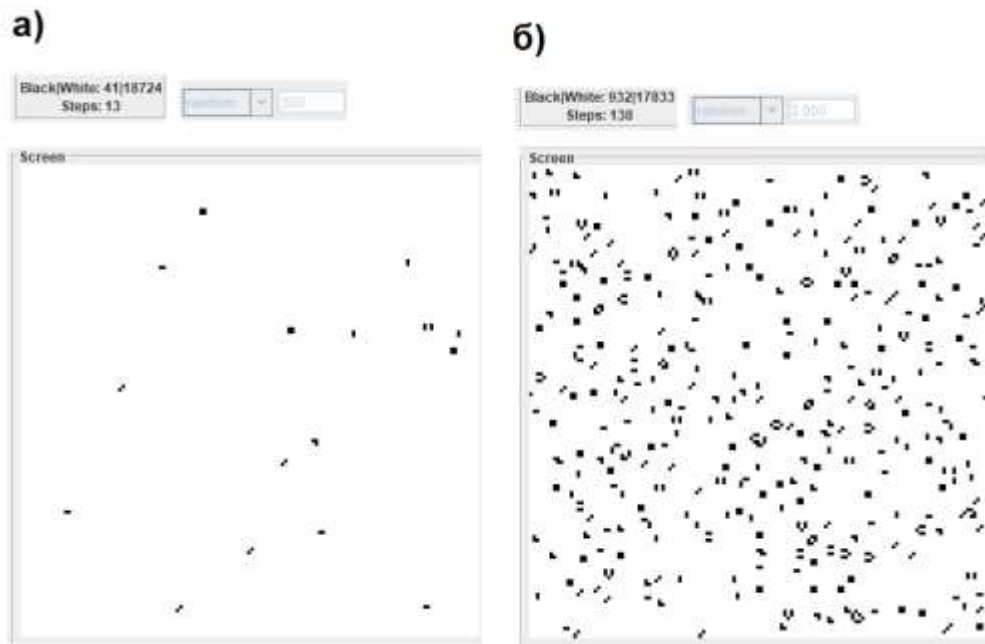


Рисунок 2.11 – Гра «Життя»

І наостанок порівняємо ці три клітинні автомати на стійкість до збурень. В якості збурень виступатиме 5% ймовірність невдалого виконання правила. Автомат з рисунку 2.11 б) повністю зникає за 80-90 ходів. Корал також не витримує це збурення. Навіть якщо на початку живі клітини займають 100% від усіх клітин, то Корал повністю зникає вже за 60-70 ходів. Але Діадема показує себе дуже стійким утворенням. Якщо вона займає так само 100% від всіх клітин, то її повне зникнення займає приблизно 1100 ходів.

2.3 Висновки до розділу 2

У цьому розділі була розглянута програма для побудови клітинних автоматів та розглянуті різні властивості цих автоматів. Були досліджені клітинні автомати та їх реакції на різні зміни.

Висновок

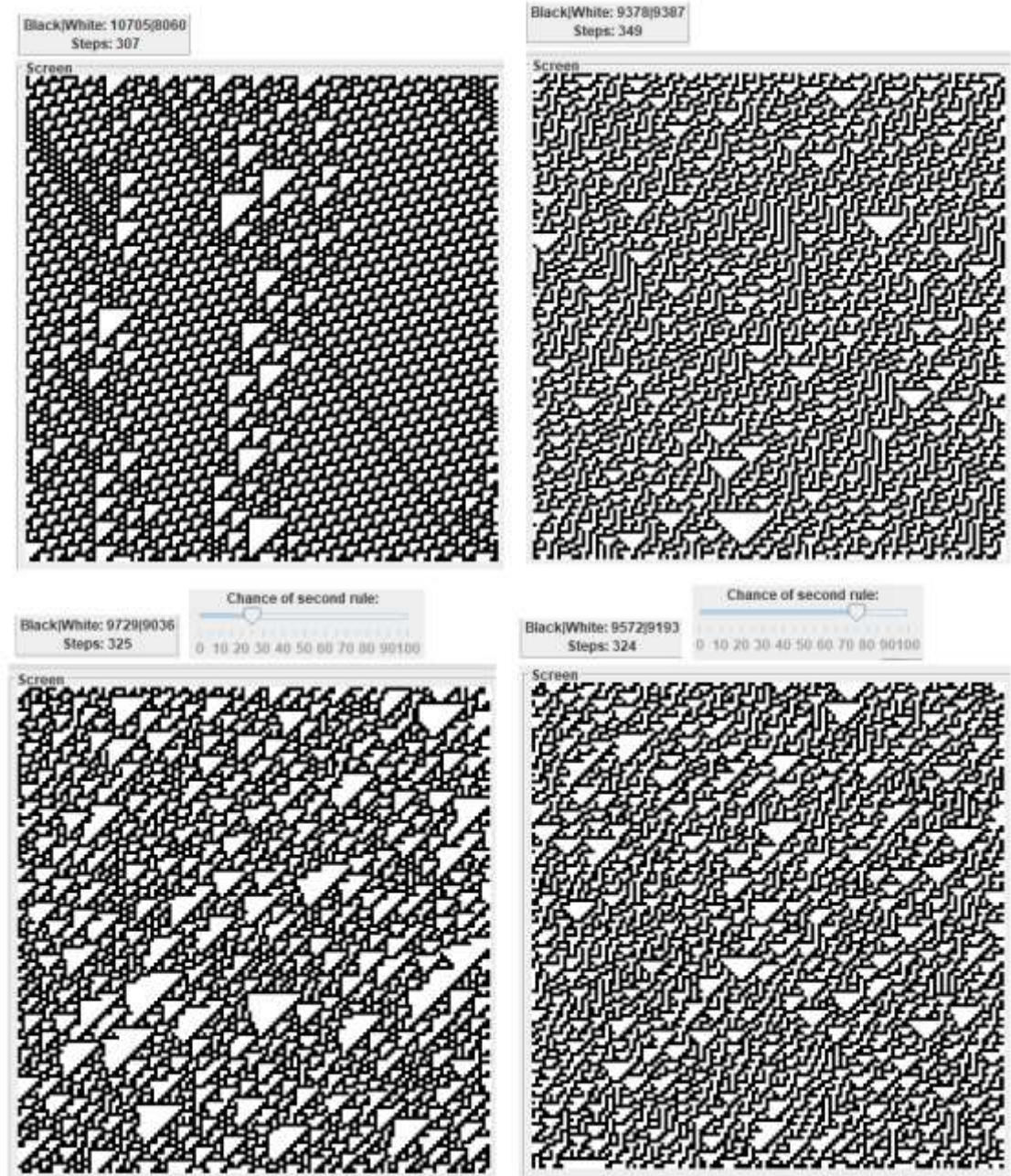
У цій роботі були успішно розглянуті класифікації клітинних автоматів за їх поведінкою та типом околиць. Була створена програма керування клітинними автоматами, що дозволяє користувачу самому створювати правила для «гри життя» та запускати безліч автоматів.

Список літератури

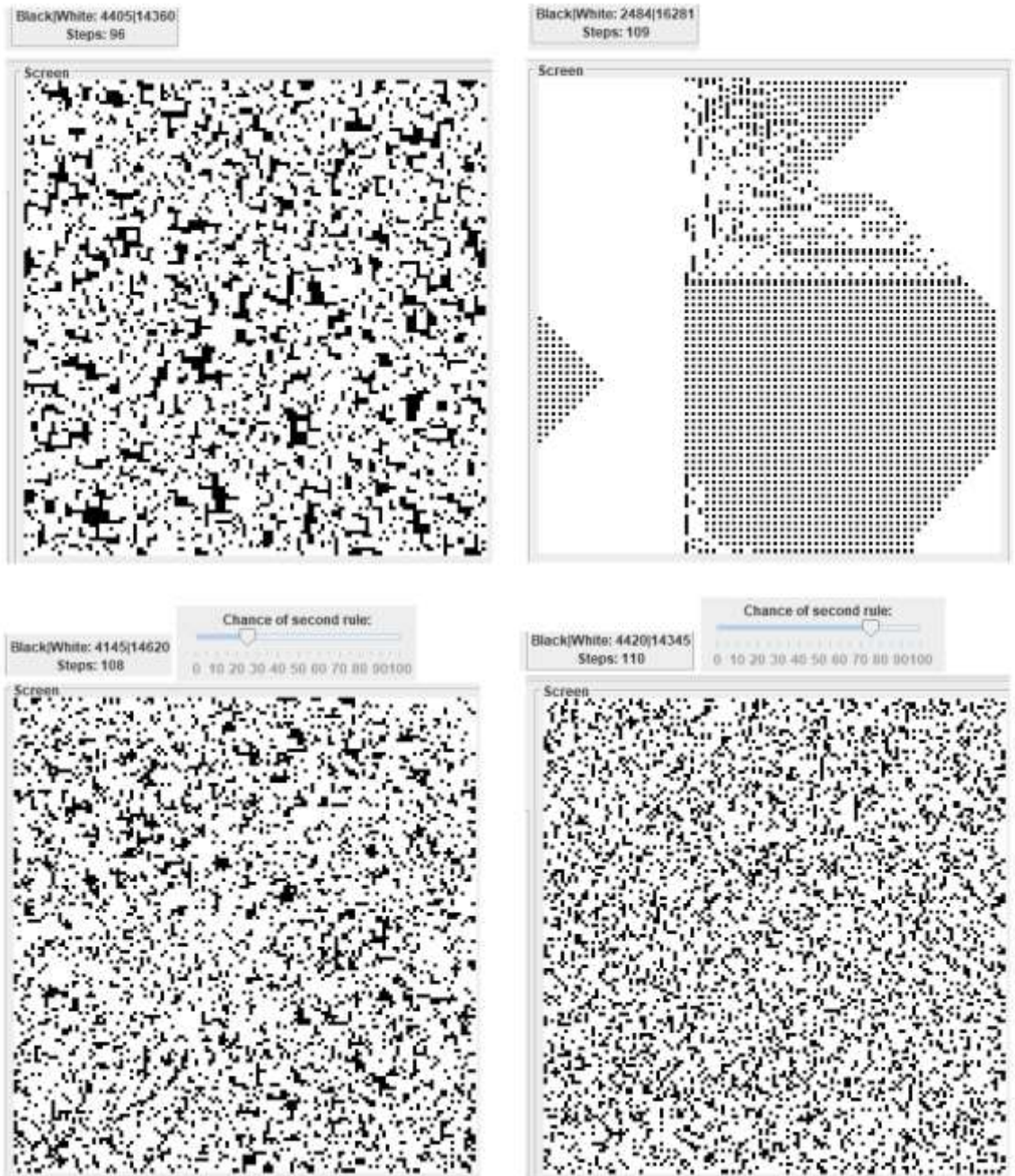
1. Светлова О. К. Застосування клітинних автоматів для обробки зображень / О. К. Светлова // Міжнародний науковий журнал [Електронний ресурс]. - 2016. - № 6(1). - С. 96-98. - Режим доступу: [http://nbuv.gov.ua/UJRN/mnj_2016_6\(1\)__26](http://nbuv.gov.ua/UJRN/mnj_2016_6(1)__26).
2. "Цветные" клеточные автоматы, или Клонирование Моны Лизы [Текст] / Лев Наумов, Алексей Шалыто // Мир ПК. - 2004. - N 5. - С. 64-71. - Библиогр.: с. 71 (8 назв.). - 0; Клонирование Моны Лизы. - RUMARS-mir_04_000_005_0064_1
3. Шмидт, Ю. Д. О некоторых подходах к моделированию пространственной диффузии инноваций / Ю. Д. Шмидт, О. Н. Лободина // Пространственная экономика. - 2015. - № 2. - С. 103-115.
4. Евсютин, О. О. (2012). Разработка и тестирование вычислительного метода построения базисов декоррелирующих преобразований с использованием клеточных автоматов на разбиении. Труды СПИИРАН, 4(23), 324-342. <https://doi.org/10.15622/sp.23.16>
5. Скаков П.С. Классификация поведения одномерных клеточных автоматов [Електронний ресурс]. - Режим доступу: http://is.ifmo.ru/papers/_skakov_master.pdf, своб.
6. Э. Э. Гасанов, Клеточные автоматы с локаторами, Интеллектуальные системы. Теория и приложения, 2020, том 24, выпуск 2, 119–132
7. Eppstein D. Classification of Cellular Automata. [Електронний ресурс]. - Режим доступу <http://www.ics.uci.edu/~eppstein/ca/wolfram.html>
8. Wolfram S. A New Kind of Science. Wolfram Media, Inc., 2002.
9. Лекция 2. Клеточные автоматы [Електронний ресурс]. - Режим доступу: <https://docplayer.ru/126307193-Lekciya-2-kletochnye-avtomaty.html>

10. Лобанов А.И. Модели клеточных автоматов // Компьютерные исследования и моделирование, 2010, т. 2, № 3, с. 273-293
11. Матюшкин И.В., Заплетина М.А. Обзор по тематике клеточных автоматов на базе современных отечественных публикаций // Компьютерные исследования и моделирование, 2019, т. 11, № 1, с. 9-57
12. This is a Turing Machine implemented in Conway's Game of Life [Электронный ресурс]. - Режим доступа: <http://rendell-attic.org/gol/tm.htm>
13. Тоффли Т., Марголюс Н. Т63. Машины клеточных автоматов: Пер. с англ. - М.: Мир,. 1991. - 280 с
14. А. А. Ефремова, А. Н. Гамова, Самопрограммируемые клеточные автоматы в криптографии, ПДМ. Приложение, 2017, выпуск 10, 76–81
15. Голубчиков М. А., Колнооченко А. В., Лебедев И. В., Меньшутин Н. В. Моделирование процесса мономолекулярной адсорбции активных веществ в пористых телах клеточным автоматом Марголюса // Успехи в химии и химической технологии. — 2014. — Т. XXVIII, № 1. — С. 38–42.

ДОДАТКИ



Додаток 1. Суміш правила 110 та 30



Додаток 2. Суміш двох правил з околицями Мура



Додаток 3. Суміш Коралу та Діадеми