ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ

Навчально-науковий інститут денної освіти Форма навчання денна Кафедра комп'ютерних наук та інформаційних технологій

> Допускається до захисту Завідувач кафедри Олена ОЛЬХОВСЬКА (підпис) «___»____2024 p.

КВАЛІФІКАЦІЙНА РОБОТА на тему МІНІМІЗАЦІЯ СКІНЧЕННИХ АВТОМАТІВ: ДОСЛІДЖЕННЯ МЕТОДІВ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПРАКТИЧНОГО НАВЧАННЯ ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «ДИСКРЕТНА МАТЕМАТИКА»

зі спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня магістра

Виконавець роботи Тертишник Єгор Ігорович <u>(*niдnuc*)</u> «____»____ 2024 р.

Науковий керівник к. ф.-м. н., доцент, Парфьонова Тетяна Олександрівна <u>_____«___»</u>___ 2024 р.

Рецензент

ПОЛТАВА 2024

3MICT

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧІ	ЕНЬ,
ТЕРМІНІВ	3
ВСТУП	4
1. ПОСТАНОВКА ЗАВДАННЯ	7
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	8
2.1 Актуальність обраної теми	8
2.1 Огляд основних понять з теми та методів мінімізації скінченого	
автомата	9
2.3. Огляд існуючих подібних програм	12
3. ТЕОРЕТИЧНА ЧАСТИНА	17
3.1. Алгоритм реалізації розділу системи практичного навчання	
«Побудова скороченого скінченого автомату»	17
3.2. Алгоритм реалізації розділу системи практичного навчання	
«Мінімальна форма скінченого автомата»	21
3.3. Опис засобів програмного забезпечення	29
4. ПРАКТИЧНА ЧАСТИНА	32
4.1 Блок-схема алгоритму роботи тренажера	32
4.2 Опис програмної реалізації	35
4.3. Необхідна користувачу інструкція	47
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58
ДОДАТОК А. КОД ПРОГРАМИ	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи,	Пояснення умовних позначень,	
скорочення, терміни	скорочень, символів	
CA	Скінчений автомат	
Явно еквівалентні стани СА	– це такі еквівалентні стани <i>a_i</i> , <i>a_j</i> , для	
	яких рядки в таблиці переходів і	
	таблиці виходів однакові, або стають	
	однаковими при заміні a _i на a _j .	
Явно відмінні стани СА	це такі стани $a_i, a_j,$ для яких	
	відмінними є відповідні їм рядки в	
	таблиці виходів.	

ВСТУП

Підвищення якості освіти є одним із пріоритетних завдань для держави, освітніх установ, педагогічних працівників та суспільства в цілому. Якісна освіта являється базою розвитку країни, створення активного громадянського суспільства та забезпечує конкурентоспроможність на світовому рівні. З метою досягнення цього завдання є потреба у впровадженні сучасних педагогічних підходів, забезпеченні доступності освіти для всіх верств населення, підтриманні високої кваліфікації освітян та розвитку системи оцінювання якості освіти. Реалізація даних заходів сприятиме підвищенню якості освіти, підготовці майбутніх поколінь та розвитку суспільства в цілому.

Крім цього, також важливо активно використовувати різні інструменти системи практичного навчання, до яких можна віднести:

1. Стажування, де у студентів є можливість працювати в реальному робочому середовищі під керівництвом досвідчених фахівців.

2. Професійна практика, на якій студенти набувають практичного досвіду, здійснюючи роботу над реальними проектами або в компаніях, що спеціалізуються у їхній галузі.

3. Симуляційні тренажери – це програмні або апаратні пристрої, що дозволяють студентам імітувати реальні об'єкти, процеси або ситуації, що дозволяє отримати практичний досвід та вміння у відповідній галузі.

4. Лабораторні роботи, де студенти проводять практичні експерименти та дослідження, щоб набути практичних навичок у своїй галузі.

5. Проектна робота – це коли студенти працюють над конкретними проектами, що дозволяє їм застосовувати свої знання та вміння на практиці.

6. Інтерактивні тренажери – це комп'ютерні програми або пристрої, які надають студентам можливість виконувати практичні завдання та отримувати миттєвий зворотний зв'язок. Такі тренажери можуть бути використані для навчання різних предметів, від математики та науки до мови та мистецтва.

Завдяки інтерактивності студенти можуть проводити експерименти чи вирішувати задачі різної складності у безпечному середовищі.

Використання навчальних тренажерів, як елементу системи практичного навчання, допомагає забезпечити ефективне і практичне навчання, тому навчальні тренажери залишаються важливими та актуальними інструментами в освітній сфері.

Мета роботи – побудова алгоритму та програмна реалізація системи практичного навчання на основі аналізу методів мінімізації скінчених автоматів для дистанційного курсу «Дискретна математика».

Методи дослідження – методи мінімізації скінчених автоматів, зокрема, пошук скороченого СА та знаходження мінімальної форми СА.

Для програмної реалізації було обрано мову програмування Java з використанням JPanel у бібліотеці Java Swing.

Об'єкт розробки – дослідження методів та програмна реалізація системи практичного навчання дистанційного навчального курсу "Дискретна математика".

Предмет розробки – система практичного навчання дистанційного навчального курсу "Дискретна математика", яка досліджує методи мінімізації скінченних автоматів, та її програмна реалізація.

Основні завдання розробки –

- огляд необхідного теоретичного матеріалу та наявного подібного програмного продукту;

- вивчення методичних рекомендацій щодо написання магістерської кваліфікаційної роботи;

- проектування інтерфейсу програми та контенту, який планується розмістити у тренажері;

- обрано засоби програмування для реалізації;

- реалізація основних функцій тренажеру шляхом побудови алгоритму його роботи та створення його програмної реалізації; - проведення тестування програми для виявлення та виправлення помилок та недоліків;

- впровадження створеної програми в навчальний процес з теми «Мінімізація скінчених автоматів» дистанційного навчального курсу «Дискретна математика».

Структура кваліфікаційної роботи. Робота складається із чотирьох розділів, що розкривають тему. У першому описується постановка задачі, що включають головні вимоги до програмної реалізації. Другий розділ, інформаційний огляд, присвячено опису актуальності теми, огляду теоретичного матеріалу та існуючих подібних програмних продуктів. У третьому розділі, теоретичній частині, покроковий опис алгоритму реалізації системи практичного навчання та блок-схема його роботи. Четвертий розділ, практична частина, включає блок-схему алгоритму роботи програми, опис програмної реалізації та необхідна користувачу інструкція

Розроблено функціонуючий навчальний тренажер, як елемент системи практичного навчання на тему «Мінімізація скінченого автомата» дистанційного навчального курсу «Дискретна математика». Подібної програми з даної теми ще не було, тому являється необхідною складовою у навальному процесі.

Особистий внесок – сформульовано питання згідно теми та здійснено вибір завдань для реалізації тренінгу, побудовано алгоритм та виконано його програмну реалізацію.

Обсяг пояснювальної записки: 74 стор., в т.ч. основна частина - 59 стор., джерела - 11 назв.

1. ПОСТАНОВКА ЗАВДАННЯ

На першому етапі необхідно сформулювати вимоги до тренажеру:

На стартовій сторінці має бути

- тема тренажеру «Мінімізація скінчених автоматів»;

- інформація про автора (прізвище та ініціали виконавця, університет, прізвище та ініціали керівника);

- перед початком тренінгу передбачити можливість повторити теоретичний матеріал;

- два посилання (дві кнопки чи вкладки) «Побудова скороченого скінченого автомату» та «Мінімальна форма скінченого автомату» (чи одна, наприклад, «Початок тренінгу», яка переводить на іншу сторінку із цими кнопками чи вкладками). Важливо врахувати, що проходження тренінгу щодо побудови мінімальної форми було доступним тільки після завершення тренінгу з побудови скороченого скінченого автомату».

Якщо завдання виконується неправильно, має з'являтися повідомлення про помилку, яке містить правильну відповідь та пояснення. Це передбачити на кожному кроці алгоритму. В деяких випадках можна не одразу надавати правильну відповідь, а лише рекомендації та надати можливість користувачу самостійно отримати правильні результати.

В ході реалізації частини «Побудова скороченого скінченого автомату» необхідно передбачити ряд питань, які стосуються основних понять, таких, як явно еквівалентні стани, явно відмінні стани та ін.

Для реалізації розділу «Мінімальна форма скінченого автомату» передбачити наступне завдання: «Скінчений автомат задано загальною таблицею переходів. Знайти мінімальну форму скінченого автомата.».

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Актуальність обраної теми

Дискретна математика є важливим розділом математики, який включає в себе різні розділи (такі як теорія множин та відношень, булеві функції, елементи комбінаторики, теорія графів, елементи математичної логіки та інші). Ця галузь математики є основою для підготовки фахівців у галузі комп'ютерних наук.

Навчальні тренажери з дискретної математики можуть бути цінними для студентів, які вивчають цей предмет, оскільки вони дозволяють практично застосовувати теоретичні знання у розв'язанні реальних завдань. Такі тренажери можуть містити вправи на розв'язання задач з різних тем, інтерактивні практичні завдання, а також можливість отримувати зворотний зв'язок та рекомендації для подальшого вдосконалення.

Зважаючи на зростання значущості дискретної математики в сучасному інформаційному світі, застосування навчальних тренажерів у цій галузі може сприяти покращенню розуміння основних концепцій і отриманню необхідних практичних навичок. Їх використання стануть студентам в нагоді для покращення своїх вмінь у розв'язуванні задач, а також розвитку абстрактного та критичного мислення. Разом з цим важливо, щоб під час розробки таких програм враховувались сучасні методології та підходи до навчання для сприяння якісного засвоєння дискретної математики.

Зокрема, актуальною серед всіх інших є тема щодо скінчених автоматів.

Скінчені автомати є основною моделлю, що дозволяє досліджувати різні аспекти автоматизованих систем, таких як комп'ютерні програми, електронні пристрої та інші. Завдяки навчальним тренажерам, студенти можуть активно взаємодіяти з матеріалом, вирішувати завдання, демонструвати різноманітні види скінчених автоматів та їх можливості. Також актуальним є вивчення методів мінімізації скінчених автоматів в сучасній інформаційній технології. Мінімізація СА є фундаментом в теорії автоматів і важливою складовою у розробці електронних пристроїв, програмного забезпечення та програмування. Навчальні тренажери з мінімізації скінчених автоматів допомагають студентам вирішувати завдання, пов'язані з мінімізацією, застосовуючи алгоритми та техніки, які основані на теоретичних знаннях. Це допоможе збільшити доступність навчання в цій галузі і дозволить краще розуміти складні аспекти та методи мінімізації автоматів.

2.2. Огляд основних понять з теми та методів мінімізації скінченого автомата

Скінченним автоматом (СА) називають $(A, X, Y, \delta, \lambda)$, де: A, X, Y скінчені множини, а саме: A – множина внутрішніх станів (множина станів), X – множина вхідних сигналів (вхідний алфавіт); Y – множина вихідних сигналів (вихідний алфавіт); а δ та λ – функції: $\delta: A \times X \to A$ – функція переходів; $\delta(t+1) = \delta(a(t), x(t))$. $\lambda: A \times X \to Y$ $\lambda: A \times X \to Y$ – функція виходів: $y(t) = \lambda(a(t), x(t))$.

Еквівалентними називаються стани a та b двох автоматів зі спільними вхідним і вихідним алфавітом, якщо однаковими є автоматні відображення $\varphi_a = \varphi_b$, де $\varphi_a \varphi_a^{-1} - \alpha_a = 0$ ссавовалентних станів – $a \sim b$.

Еквівалентні автомати A_1 та A_2 — це якщо для довільного стану *а* автомата A_1 існує стан *b* автомата A_2 , тобто $a \sim b$.

Зведений автомат А₀ – це якщо усі його стани попарно не еквівалентні.

Розглянемо задачу мінімізації кількості станів автомата, яка полягає у побудові такого скінченого автомата по заданому СА, який має мінімально можливу кількість станів і має ту ж відповідність між входом і виходом, яка присутня в заданому СА.

Нехай СА – автомат 1 роду скрізь визначений та початковий стан a_0 – фіксований. Задача мінімізації кількості станів: Знайти зведений СА, який є еквівалентним даному.

Метод Ауфенкампфа-Хона. Згідно методу виконується послідовне спеціальне розбивання множини станів заданого СА.

На *i*-му (i = 1, 2, ...) кроці алгоритму розбивають множини станів та об'єднують в один клас стани a_j СА, які визначають відображення $\varphi_a: X^* \to Y^*$, які збігаються на всіх словах довжиною не більше номеру кроку алгоритму *i*. Після скінченої кількості кроків алгоритму така послідовність розбивань припиняється на певному розбитті та отримується зведений автомат, який еквівалентний вихідному.

Явно еквівалентні стани – це такі еквівалентні стани a_i, a_j , для яких рядки в таблиці переходів і таблиці виходів однакові, або стають однаковими при заміні a_i на a_j .

Явно відмінні стани СА – це такі стани a_i, a_j , для яких відмінними є відповідні їм рядки в таблиці виходів.

Алгоритм побудови автомату, еквівалентного заданому СА, що має менше станів, ніж A₀ – скороченого СА.

Спершу за таблицею виходів чи загальною таблицею переходів необхідно знайти явно відмінні стани тобто підмножини *M_i* множини станів такі, що стан з однієї з них явно відмінний від стану інших.

Далі серед елементів однієї і тієї ж множини M_i , знаходяться еквівалентні стани шляхом виявлення явно еквівалентних станів. У випадку їх існування здійснюється перехід на наступний крок. Інакше – скорочений скінчений автомат побудовано. На наступному етапі потрібно об'єднати явно еквівалентні стани згідно загальної таблиці переходів, викреслюючи рядок з a_i та заміняючи скрізь в чисельниках a_i на a_j . Знову повторюється пошук явно еквівалентних станів до тих пір доки таких не виявиться. З метою знаходження всіх пар еквівалентних станів процес знаходження явно еквівалентних станів не застосовний, так як не всі пари знаходяться.

Коли відомі всі пари еквівалентних станів, то на множині всіх станів Aвизначено відношення еквівалентності, якому відповідає розбиття на класи еквівалентності $A = \{S_0, S_1, ..., S_k\}$. Якщо a'_i представники класів еквівалентності $S_i, i = 1, 2, ..., k$, а $A' = \{a'_1, a'_2, ..., a'_k\}$, то можна довести, що автомат $A_0 = \{A, X, Y, \delta, \lambda\}$, та автомат $A_0 = \{A', X, Y, \delta, \lambda\}$ еквівалентні, причому A_1 має мінімально можливу (серед еквівалентних автоматів) кількість |A'| = kстанів.

Розіб'ємо множину станів A на класи еквівалентності. Дане розбиття ґрунтується на почерговому переборі пар станів і видалення тих з них, що не є еквівалентними. Пари (a_i, a_i) є еквівалентні і, отже, не розглядаються. На основі загальної таблиці переходів складається таблиця, яку називають таблицею пар станів.

Явно відмінні стани не являються еквівалентними. Отже, в таблицю пар вони не включаються. Кожній парі станів відповідає рядок, кожному входу – стовпець. На перетині рядка пари станів i, j та стовпця x ставиться пара станів, у які переходить автомат з пари станів i, j при вхідному впливі x, не враховуючи при цьому порядок станів. Пари, що видаляються, позначаються (наприклад, V).

Видалення пар враховує наступне: якщо два стани a_i та a_j – еквівалентні, то еквівалентними будуть і стани $a_{i+1,}a_{j+1,}$ де $a_{i+1,} = \delta(a_i,x)$, $a_{j+1,} = \delta(a_j,x)$, в які автомат переходить відповідно зі станів a_i та a_j під впливом довільного входу x. Тоді спершу потрібно позначити всі ті пари, що хоч при одному вхідному впливі x_i переходять у пари, що складаються з різних станів, а також відсутні у першому стовпці, тобто стовпці пар таблиці. Позначимо їх w.

Так як пари, що позначені знаком V_1 , не можуть бути еквівалентними, то на наступному етапі необхідно позначити V_2 всі ті пари, які хоч при одному вхідному впливі x_i переходять у пари, що позначені V_1 і т. д.

Еквівалентними є невідмічені пари. В кожній такій парі елементи еквівалентні. Далі необхідно згрупувати по класам еквівалентності S_i , до кожного з яких помістити еквівалентні стани. Можуть бути і одноелементні класи еквівалентності, коли елементи не увійшли до інших класів. Нехай a_i – представник класу S_i . Тоді після позначення кожного представника класу еквівалентності як a_i отримаємо мінімальну (за кількістю станів) форму представлення автомата.

2.3. Огляд існуючих подібних програм

Розвиток IT технологій та доступність інтернету сприяли появі навчальних тренажерів, зробили їх популярними засобами навчання. Такі інструменти навчання можуть бути в різних форматах, таких як веб-додатки, мобільні додатки, віртуальна реальність тощо. Дані тренажери допомагають студентам і учням отримати пракЗичні навички та знання в інтерактивному форматі, що робить навчання більш ефективним та цікавим.

На сьогоднішній день створено багато тренажерів та ігрових платформ, які розроблені спеціально для дітей. Ці тренажери та ігри часто мають навчальну та розвивальну спрямованість, допомагаючи дітям розвивати різні навички та вміння. До таких можна віднести:

"LeapFrog LeapPad": це планшетна система, спеціально розроблена для дітей від 3 до 9 років, має широкий вибір навчальних програм та ігор, які допомагають дітям вивчати літери, математику, науку та багато іншого.

"Osmo": дана ігрова платформа використовує додатки для смартфонів або планшетів, пропонуючи ігри, які сприяють розвитку логічного мислення, математичних навичок та творчості.

"Khan Academy Kids" – безкоштовна платформа, що надає доступ до великої кількості навчальних матеріалів та ігор для дітей від 2 до 7 років. Вона розвиває навички читання, математики, наукового мислення та багато іншого. "Lego Education": дана платформа використовує конструктори Lego для навчання різних предметів, таких як наука, технологія, інженерія та математика, допомагаючи розвивати творчість, просторове мислення та проблемне мислення у дітей.

Для прикладу, розглянемо один із ресурсів, де зібрано різні тренажери для учнів 1-8 класів (рис.2.1). Зокрема, тут реалізовані:

- Математичні тренажери для 1 4 класів та 5 6 класів, математика.
- Математичні тренажери для 7 8 класів, алгебра та геометрія.
- Математична академія Робочий зошит з математики: вчимося додавати у межах 20, усі уроки математики 5 та 6 клас.
- Базові математичні навички: Математичні настільні ігри Вчимося рахувати від 1 до 100 Таблицю множення Таблицю ділення.

Крім запропонованих завдань для виконання, також передбачені самостійні та контрольні роботи по темам, що надає можливість оцінити засвоєння матеріалу.





Рисунок 2.1 – Приклад тренажера для школярів

Студенти спеціальності "Комп'ютерні науки" ПУЕТ розробили багато різних тренажерів у рамках виконання кваліфікаційних робіт. В доступі є багато робіт, що присвячені створенню таких програм.

Розглянемо деякі з них: тренажери для дистанційного навчального курсу «Дискретна математика» на тему «Відношення. Область визначення, область значень, граф, матриця відповідності, переріз за елементами», автор Борута І.В. (рис.2.2) [] і «Решітки та булеві алгебри», автор Хорольський В.О. (рис.2.5) [].

Для створення програми Борутою I.В. було обрано мову програмування C++ та середовище Borland Builder. Тренажер складається з трьох тем: 1) «Область визначення та область значень відношень», 2) «Перерізи відношень», 3) «Способи представлення відношення» .Перша тема містить два приклади; друга – 3 приклади, третя – один приклад.



Рисунок 2.2 – Тренажер Борути І.В.

В програмі реалізовані питання різної форми, з вибором варіанту, в можливістю введення відповіді у комірку тощо. На рисунку 2.3 показано деякі фрагменти реалізованого алгоритму, які це демонструють.

Для виявлення помилок програмою передбачено, що мають з'являтись повідомлення з роз'ясненнями та вказівками в разі допущення помилок.



Рисунок 2.3 – Фрагменти тренажеру Борути I.В.

Тренажер, реалізований Хорольським В.О., складається із двох основних частин. В першій реалізовано основні поняття теми, друга містить завдання на визначення, чи являється задана частково впорядкована множина А решіткою.

Для програмної реалізації тренажеру використовувалась мова C++ у середовищі Borland Builder.

📗 Решітки та булеві алгебри 📃 🗔 🗙
Тренажер з теми
"Решітки та булеві алгебри"
ДК "Дискретна математика"
Теми тренінгу : 1) Деякі поняття, необхідні для означення решітки. 2) Решітки.
Розпочати
Розробник: Хорольский В.О., гр. КН-41.
Каф. ММСІ, ПУЕТ, 2021 р.

Рисунок 2.4 – Тренажер Хорольського В.О.

Виконавши огляд функціонуючих навчальних симуляторів, можна відмітити, що застосування навчальних тренажерів має кілька переваг для студентів і школярів. Серед них:

1. Залучення: Навчальні тренажери створюють інтерактивне навчання, що залучає учнів до процесу. Вони можуть бути використані для створення цікавих завдань, ігор та симуляцій, що спонукають учнів активно досліджувати та вирішувати проблеми.

2. Практичні навички: Навчальні тренажери дозволяють учням отримати практичні навички, які вони можуть застосовувати у реальному житті. Вони можуть навчитися вирішувати математичні задачі, виконувати хірургічні операції, програмувати комп'ютери та багато іншого.

3. Індивідуалізоване навчання: Навчальні тренажери можуть бути налаштовані під потреби кожного учня. Вони дозволяють учням вчитися у своєму власному темпі і зосереджуватися на конкретних аспектах навчання, які є для них найважливішими або найскладнішими.

4. Миттєва зворотній зв'язок: Багато навчальних тренажерів надають миттєвий зворотній зв'язок, що допомагає учням зрозуміти свої помилки та

вдосконалити свої навички. Вони можуть отримувати негайне повідомлення про правильність або неправильність своїх відповідей і вчитися на основі цього.

5. Доступність: Завдяки технологічному розвитку, навчальні тренажери стають все більш доступними. Вони можуть бути доступні як онлайн-ресурси, мобільні додатки або навчальні програми, що дозволяють учням мати доступ до навчання в будь-який зручний для них час і місце.

Використання навчальних тренажерів може значно покращити якість навчання і сприяти успіху студентів.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Алгоритм реалізації розділу системи практичного навчання «Побудова скороченого скінченого автомату»

Крок 1. Користувачу відображається наступна інформація: «Згадаємо означення скінченого автомату.

Скінченним автоматом (СА) називають п'ятірку $\langle A, X, Y, \delta, \lambda \rangle$, де A – множина *внутрішніх станів*, X – множина *вхідних сигналів* (*вхідний алфавіт*); Y – множина *вихідних сигналів* (*вихідний алфавіт*); а δ та λ – функції переходів та виходів.

Дайте відповіді на наступні запитання:

1. Скороченим СА називається

А) еквівалентний до заданого СА із більшою кількістю станів, і який забезпечує ту ж відповідність між входом і виходом, що і в заданому СА

<u>Б)</u> еквівалентний до заданого СА із меншою кількістю станів, і який забезпечує ту ж відповідність між входом і виходом, що і в заданому СА

 В) такий скінчений автомат, у якого кількість вихідних сигналів мінімальна

Г) еквівалентний до заданого CA із меншою кількістю станів та виходів, і який забезпечує ту ж відповідність між входом і виходом, що і в заданому CA

2. Явно еквівалентними станами називаються

<u>А)</u> такі еквівалентні стани a_i , a_j , для яких рядки в таблиці переходів і таблиці виходів однакові, або стають однаковими при заміні a_i , на a_j

Б) такі еквівалентні стани a_i , a_j , для яких рядки в таблиці переходів і таблиці виходів однакові

В) такі еквівалентні стани a_i , a_j , для яких рядки в таблиці виходів однакові або стають однаковими при заміні a_i , на a_j

Г) такі еквівалентні стани a_i , a_j , для яких рядки в таблиці переходів однакові

3. Явно відмінними станами називаються

А) такі стани a_i , a_j , для яких рядки в таблиці переходів і таблиці виходів однакові, або стають однаковими при заміні a_i , на a_j

Б) такі стани a_i , a_j , для яких рядки в таблиці переходів і таблиці виходів різні

<u>В)</u> такі стани a_i , a_j , для яких рядки в таблиці виходів відрізняються

Г) такі стани a_i , a_j , для яких рядки в таблиці переходів відрізняються

Користувач має обрати один варіант правильної відповіді.

У випадку правильного вибору – здійснюється перехід на наступне питання. В іншому – користувачу надається пояснення у повідомленні про помилку. Після надання відповідей на всі питання здійснюється перехід до наступного кроку із наступним завданням.

Тексти повідомлення про помилку:

1. «Вибір неправильний. *Скороченим СА називається* еквівалентний до заданого СА із меншою кількістю станів, і який забезпечує ту ж відповідність між входом і виходом, що і в заданому СА».

2. «Вибір неправильний. **Явно еквівалентними станами** *називаються* такі еквівалентні стани a_i , a_j , для яких рядки в таблиці переходів і таблиці виходів однакові, або стають однаковими при заміні a_i , на a_j ».

3. «Вибір неправильний. *Явно відмінними станами називаються* такі стани a_i , a_j , для яких рядки в таблиці виходів відрізняються». Перехід на крок 2.

Крок 2. На екрані висвічується наступне: «Задано СА (див. табл. 3.1).

Вибрати пари явно еквівалентних станів.

0 i 1	0 i 4	1 i 3	2 i 3	3 i 4
0 i 2	0 i 5	1 i 4	2 i 4	3 i 5

0i3 1i2 1i5 2i5 4i5

Користувачу надати можливість вибору явно еквівалентних станів (може бути декілька).

Стани а _і	Входи х _і		
	0	1	2
0	5/1	2/1	3/0
1	2/0	1/0	4/1
2	5/1	2/1	3/0
3	2/0	3/0	4/1
4	4/0	0/0	2/1
5	0/1	1/1	3/0

Таблиця 3.1 – Загальна таблиця переходів СА до завдання

Після вибору в разі присутності помилкових варіантів відповіді, запропонувати підказку-означення з текстом:

«Явно еквівалентними станами називаються такі еквівалентні стани a_i , a_j , для яких рядки в таблиці переходів і таблиці виходів однакові, або стають однаковими при заміні a_i , на a_j ».

Після цього є можливість зробити виправлення. Якщо ж знову допущено помилки, то з'являється наступне повідомлення:

«Рядки виходів однакові для станів 0,2,5. Це рядок 1 1 0. А також для станів 1, 3, 4. Це рядок 0 0 1. Явно еквівалентними є пари 0 і 2 (однакові) та 1 і 3 (стають однаковими, якщо замінити стан 1 на 3)». Перехід на крок 3.

Крок 3. На екрані залишається таблиця 3.1 та з'являється наступне завдання: Вибрати пари явно відмінних станів.

0 i 1	0 i 4	1 i 3	2 i 3	3 i 4
0 i 2	0 i 5	1 i 4	2 i 4	3 i 5
0 i 3	1 i 2	1 i 5	2 i 5	4 i 5

Після вибору в разі присутності помилкових варіантів відповіді, запропонувати підказку-означення з текстом:

«*Явно відмінними станами називаються* такі стани a_i , a_j , для яких рядки в таблиці виходів відрізняються».

Після цього є можливість зробити виправлення. Якщо ж знову допущено помилки, то з'являється наступне повідомлення:

«Рядки виходів однакові для станів 0,2,5. Це рядок 1 1 0. А також для станів 1, 3, 4. Це рядок 0 0 1. Явно відмінними є ті, які належать групам з різними рядками виходів. Тобто це 0 і 1, 0 і 3, 0 і 4, 2 і 1, 2 і 3, 2 і 4, 1 і 5, 3 і 5, 4 і 5». Перехід на крок 4.

Крок 4. З'являється запитання: «На скільки рядків буде скорочена наступна загальна таблиця переходів?

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

Правильна відповідь – 2. Якщо обрано інше, то з'являється пояснення:

«Помилка. Наступна загальна таблиця переходів буде скорочена на два рядки, так як маємо дві пари явно еквівалентних станів. Тобто рядки, що відповідають станам 2 і 3 зникнуть, а в інших рядках скрізь треба замінити 2 на 0 і 3 на 1.». Перехід на крок 5.

Крок 5. З'являється завдання: «Заповнити наступну таблицю.»

Для заповнення нової таблиці передбачити наявність на екрані загальної таблиці переходів заданого СА (табл.3.2).

Стани а _і	Входи <i>x_i</i>			
	0	1	2	
0 (2)	/1	/1	/0	
1 (3)	/0	/0	/1	
4	/0	/0	/1	
5	/1	/1	/0	

Таблиця 3.2 – Таблиця для заповнення на кроці 5

Повідомлення в разі неправильного заповнення: «Значення дублювати із таблиці 1, але крім тих клітинок, в яких містяться елементи 2 і 3, скрізь потрібно замінити 2 на 0 і 3 на 1.». Після заповнення таблиця має вигляд:

Отримано скорочену форму заданого СА.

Роботу алгоритму завершено.

Стани а _і	Входи х _і			
	0	1	2	
0 (2)	5/1	0/1	1/0	
1 (3)	0/0	1/0	4/1	
4	4/0	0/0	0/1	
5	0/1	1/1	1/0	

Таблиця 3.3 – Скорочена таблиця СА на завершальному кроці

3.2. Алгоритм реалізації розділу системи практичного навчання «Мінімальна форма скінченого автомата»

Крок 1. На екрані завдання:

Скінчений автомат задано загальною таблицею переходів (див.табл.1). Знайти мінімальну форму скінченого автомата.

Таблиця 3.4 – Загаль	на таблиця пер	еходів СА до	завдання
----------------------	----------------	--------------	----------

Стани <i>а_і</i>	Входи х _і		
	0	1	2
0	1/1	1/0	4/0
1	0/0	3/1	3/1
2	1/1	1/0	4/0
3	2/0	1/1	1/1
4	5/1	3/0	2/0
5	7/0	8/1	5/1
6	5/1	1/0	7/0
7	3/1	3/0	6/0
8	6/0	8/1	6/1

Вибрати серед заданих рядків виходів ті, які є у заданій таблиці

-	100	-	532
-	1 1 4	-	211
-	011	-	686
-	1 1 0	-	336

Правильний вибір – 1 0 0 та 0 1 1.

Умова з таблицею має бути увесь час видимою чи в доступі. Перехід на наступний крок.

Крок 2. Виписати множини станів, у яких однакові рядки виходів (через кому, без пропусків у порядку зростання):

$$S' = \{ \}, S'' = \{ \}$$

Передбачити в програмі можливість введення правильної відповіді:

$$S' = \{0,2,4,6,7\}, S'' = \{1,3,5,8\}$$
 abo $S' = \{1,3,5,8\}, S'' = \{0,2,4,6,7\}.$

Якщо користувач помилився, то з'являється повідомлення про помилку: «Маємо два рядки виходів: 1 0 0 та 0 1 1. Відповідно рядок 1 0 0 мають стани 0,2,4,6,7, а 0 1 1 – стани 1,3,5,8.

На екрані з'являється правильна відповідь: $S' = \{0,2,4,6,7\}, S'' = \{1,3,5,8\}.$ Перехід на наступний крок.

Крок 3. На екрані завдання:

Записати у таблиці (див. табл. 3.4) в стовпці «Пари» ті пари станів, які не є явно еквівалентними. (Значення вводити по зростанню через кому, без пропусків. Необхідно врахувати також, що в разі введення пари, наприклад 1,3 – пару 3,1 не вводимо.)

На даному етапі мають бути активними для введення клітинки стовпця «Пари». В разі помилкового введення у першу комірку з'являється повідомлення:

«Явно відмінними станами називаються такі стани a_i , a_j , для яких рядки в таблиці виходів відрізняються, тобто пари, що не є явно відмінними маємо формувати із елементів кожної із множин *S' ma S''*.».

Надається повторна спроба введення.

Виконайте перші позначення згідно алгоритму для пар елементів зі стовпця «Пари». Крім цього зробіть позначки біля тих пар для відповідного входження x_i , які обираємо згідно алгоритму.

	Пари	Пари Входи x_i					
позначення		0		1		2	

Таблиця 3.4 – Таблиця для введення до кроку 3

Таблиця 3.5 – Таблиця після введення пар на кроці 3

	Пари	Входи <i>х_і</i>					
позначення		0		1		2	
	0,2						
	0,4						
	0,6						
	0,7						
	1,3						
	1,5						
	1,8						
	2,4						
	2,6						
	2,7						
	3,5						
	3,8						
	4,6						
	4,7						
	5,8						
	6,7						

Пари	Вхо ди <i>x</i> _i

позначення		0	позначення	1	позначення	2	позначення
	0,2	1,1		1,1		4,4	
	0,4	1,5		1,3		4,2	
	0,6	1,5		1,1		4,7	
	0,7	1,3		1,3		4,6	
	1,3	0,2		3,1		3,1	
	1,5	0,7		3,8		3,5	
	1,8	0,6		3,8		3,6	
	2,4	1,5		1,3		4,2	
	2,6	1,5		1,1		4,7	
	2,7	1,3		1,3		4,6	
	3,5	2,7		1,8		1,5	
	3,8	2,6		1,8		1,6	
	4,6	5,5		3,1		2,7	
	4,7	5,3		3,3		2,6	
	5,8	7,6		8,8		5,6	
	6,7	5,3		1,3		7,6	

Таблиця 3.6 – Таблиця після введення пар на кроці 3

Таблиця 3.7 – Таблиця для введення позначень на кроці 3

	Пари	Вхо ди <i>x_i</i>					
позначення		0	позначення	1	позначення	2	позначення
•	0,2	1,1		1,1		4,4	
•	0,4	1,5		1,3		4,2	
•	0,6	1,5		1,1		4,7	
•	0,7	1,3		1,3		4,6	
•	1,3	0,2		3,1		3,1	
•	1,5	0,7		3,8		3,5	
•	1,8	0,6		3,8		3,6	
•	2,4	1,5		1,3		4,2	
•	2,6	1,5		1,1		4,7	
•	2,7	1,3		1,3		4,6	
•	3,5	2,7		1,8		1,5	
•	3,8	2,6		1,8		1,6	
•	4,6	5,5		3,1		2,7	
•	4,7	5,3		3,3		2,6	
•	5,8	7,6		8,8		5,6	
•	6,7	5,3		1,3		7,6	

При натисненні позначок на цьому етапі зі стовпця «Пари» біля них з'являється V₁. А біля пари для входження x_i позначка w.

Якщо вибір здійснено правильно, то перехід на наступний крок, інакше – повідомлення-підказка: «Необхідно позначити ті пари, що хоч при одному вхідному впливі x_i переходять у пари, які складаються з різних станів і відсутні у стовпці пар таблиці. Ці пари для відповідного входження також позначити».

Надається повторна спроба.

Якщо знову помилковий вибір, то повідомлення: «Позначити на даному етапі позначкою V_1 потрібно пари {1,8}, {3,8}, {5,8}, так як вони під дією входження $x_i = 2$ переходять у пари {3,6}, {1,6}, {5,6}, відповідно яких немає у стовпці «Пари». При цьому враховуємо, що 7,6=6,7. Пари {3,6}, {1,6}, {5,6} позначаємо *w*».

Після ознайомлення з цим повідомленням в таблиці з'являються автоматично названі позначки. Перехід на наступний крок.

	Пари	Входи х _і					
позначення		позначення	0	позначення	1	позначення	2
•	0,2	•	1,1	•	1,1	•	4,4
•	0,4	•	1,5	•	1,3	•	4,2
•	0,6	•	1,5	•	1,1	•	4,7
•	0,7	•	1,3	•	1,3	•	4,6
•	1,3	•	0,2	•	3,1	•	3,1
•	1,5	•	0,7	•	3,8	•	3,5
• V ₁	1,8	•	0,6	•	3,8	• W	3,6
•	2,4	•	1,5	•	1,3	•	4,2
•	2,6	•	1,5	•	1,1	•	4,7
•	2,7	•	1,3	•	1,3	•	4,6
•	3,5	•	2,7	•	1,8	•	1,5
• V ₁	3,8	•	2,6	•	1,8	• W	1,6
•	4,6	•	5,5	•	3,1	•	2,7
•	4,7	•	5,3	•	3,3	•	2,6
• V ₁	5,8	•	7,6	•	8,8	• W	5,6
•	6,7	•	5,3	•	1,3	•	7,6

Таблиця 3.8 – Таблиця після введення позначень V₁на кроці 3

Виконайте наступні позначення згідно алгоритму для пар елементів зі стовпця «Пари». Крім цього зробіть позначки біля тих пар для відповідного входження *x_i*, які обираємо згідно алгоритму.

При натисненні позначок на цьому етапі зі стовпця «Пари» біля них з'являється V₂. А біля пари для входження x_i позначка w.

Якщо вибір здійснено правильно, то перехід на наступний крок, інакше – повідомлення-підказка: «Необхідно позначити ті пари, що хоч при одному вхідному впливі x_i переходять у пари, які позначені V_1 . Ці пари для відповідного входження також позначити». Надається повторна спроба.

Якщо знову помилковий вибір, то повідомлення: «Позначити на даному етапі позначкою V_2 потрібно пари {1,5}, {3,5}, так як вони під дією входження

 $x_i = 1$ переходять відповідно у пари {3,8}, {1,8}, які позначені на першому кроці V_1 у стовпці «Пари». Пари {3,8}, {1,8} позначаємо w».

Після ознайомлення з цим повідомленням в таблиці з'являються автоматично названі позначки. Перехід на наступний крок.

	Пари	Входи <i>x_i</i>						
позначення		позначення	0	позначення	1	позначення	2	
•	0,2	•	1,1	•	1,1	•	4,4	
•	0,4	•	1,5	•	1,3	•	4,2	
•	0,6	•	1,5	•	1,1	•	4,7	
•	0,7	•	1,3	•	1,3	•	4,6	
•	1,3	•	0,2	•	3,1	•	3,1	
• V ₂	1,5	•	0,7	• W	3,8	•	3,5	
• <i>V</i> ₁	1,8	•	0,6	•	3,8	• W	3,6	
•	2,4	•	1,5	•	1,3	•	4,2	
•	2,6	•	1,5	•	1,1	•	4,7	
•	2,7	•	1,3	•	1,3	•	4,6	
• V ₂	3,5	•	2,7	• W	1,8	•	1,5	
• <i>V</i> ₁	3,8	•	2,6	•	1,8	• W	1,6	
•	4,6	•	5,5	•	3,1	•	2,7	
•	4,7	•	5,3	•	3,3	•	2,6	
• V ₁	5,8	•	7,6	•	8,8	• W	5,6	
•	6,7	•	5,3	•	1,3	•	7,6	

Таблиця 3.9 – Таблиця після повторних позначень V₂ на кроці 3

Виконайте наступні позначення згідно алгоритму для пар елементів зі стовпця «Пари». Крім цього зробіть позначки біля тих пар для відповідного входження *x_i*, які обираємо згідно алгоритму.

При натисненні позначок на цьому етапі зі стовпця «Пари» біля них з'являється V₃. А біля пари для входження x_i позначка w.

Якщо вибір здійснено правильно, то перехід на наступний крок, інакше – повідомлення-підказка: «Необхідно позначити ті пари, що хоч при одному вхідному впливі x_i переходять у пари, які позначені V_2 . Ці пари для відповідного входження також позначити». Надається повторна спроба.

Якщо знову помилковий вибір, то повідомлення: «Позначити на даному етапі позначкою V_3 потрібно пари {0,4}, {0,6}, {2,4}, {2,6}, {4,7}, {6,7}, так як вони під дією входження x_i переходять відповідно у пари, які позначені V_2 у стовпці «Пари». А саме: пари {0,4}, {0,6}, {2,4}, {2,6} під дією входження $x_i = 0$ переходять у пару {1,5}, а пари {4,7}, {6,7} під дією входження $x_i = 0$ переходять у пару {5,3}. При цьому 3,5=5,3.

Пари {0,4}, {0,6}, {2,4}, {2,6}, {4,7}, {6,7} позначаємо w».

Після ознайомлення з цим повідомленням в таблиці з'являються автоматично названі позначки. Перехід на наступний крок.

Процес позначення завершується, так як серед пар, що залишились непозначеними немає таких, що переходять в пари, позначені V₃.

Виберіть пари еквівалентних станів

Якщо вибір здійснено правильно, то перехід на наступний крок, інакше – повідомлення: «Еквівалентними будуть ті пари, які залишились невідмченими. Тобто пари {0,2}, {0,7}, {1,3}, {2,7}, {4,6}.».

Розбийте множину станів на класи еквівалентності. Введіть без пропусків у порядку зростання елементи відповідних множин-класів еквівалентності:

$$S_0 = \{ \}, S_1 = \{ \},$$

 $S_2 = \{ \}, S_3 = \{ \}, S_4 = \{ \}$

	Пари	Входи х _і						
позначення		позначення	0	позначення	1	позначення	2	
•	0,2	•	1,1	•	1,1	•	4,4	
• V ₃	0,4	• W	1,5	•	1,3	•	4,2	
• V ₃	0,6	• W	1,5	•	1,1	•	4,7	
•	0,7	•	1,3	•	1,3	•	4,6	
•	1,3	•	0,2	•	3,1	•	3,1	
• V ₂	1,5	•	0,7	• W	3,8	•	3,5	
• V ₁	1,8	•	0,6	•	3,8	• W	3,6	
• V ₃	2,4	• w	1,5	•	1,3	•	4,2	
• V ₃	2,6	• W	1,5	•	1,1	•	4,7	
•	2,7	•	1,3	•	1,3	•	4,6	
• V ₂	3,5	•	2,7	• W	1,8	•	1,5	
• V ₁	3,8	•	2,6	•	1,8	• W	1,6	
•	4,6	•	5,5	•	3,1	•	2,7	
• V ₃	4,7	• w	5,3	•	3,3	•	2,6	
• V ₁	5,8	•	7,6	•	8,8	• W	5,6	
• V ₃	6,7	• W	5,3	•	1,3	•	7,6	

Таблиця 3.10 – Таблиця після повторних позначень V₃ на кроці 3

Якщо введення неправильне, то з'являється повідомлення: «Так як пари $\{0,2\}, \{0,7\}, \{2,7\}$ – пари еквівалентних станів (тобто 0~2, 0~7, 2~7), то стани 0, 2, 7 можна віднести до одного класу еквівалентності. Аналогічно для пар $\{1,3\}, \{4,6\}$ – 1~3, 4~6. Всі стани з умови, що не увійшли до еквівалентних пар утворюють одноелементні класи еквівалентності. Це стани 5 і 8. Тоді маємо

 $S_0 = \{0,2,7\}, S_1 = \{1,3\}, S_2 = \{4,6\}, S_3 = \{5\}, S_4 = \{8\}.$

Позначивши *a_i* представника класу еквівалентності *S_i* одержимо мінімальну форму скінченого автомата, що задана таблицею:

Стани а _і	Входи х _і						
	0	1	2				
a_0	$a_1/1$	$a_1/0$	$a_2/0$				
<i>a</i> ₁	$a_0/0$	$a_1/1$	$a_1/1$				
a ₂	<i>a</i> ₃ /1	$a_1/0$	$a_0/0$				
a ₃	<i>a</i> ₀ /0	$a_4/1$	<i>a</i> ₃ /1				
a4	$a_2/1$	$a_4/0$	$a_2/0$				

Таблиця 3.11 – Мінімальна форма скінченого автомата

3.3. Опис засобів програмного забезпечення

Вибір інструментів для написання програм залежить від різних факторів, включаючи ваші потреби, досвід, тип проекту та мову програмування. Нижче наведено кілька популярних інструментів та мов програмування, які можна використовувати для написання програм, разом з їхніми перевагами:

- 1. Мови програмування:
- Руthon: Легка для вивчення, високорівнева мова, часто використовується для розробки веб-додатків, аналізу даних, машинного навчання і багатьох інших завдань.
- Java: Крос-платформена мова програмування, популярна для розробки великих програм та веб-додатків.
- JavaScript: Мова для розробки веб-додатків, динамічних веб-сайтів і багатьох інших клієнтських та серверних додатків.
- C++: Ефективна мова програмування, яка часто використовується для системного програмування, розробки ігор і вбудованих систем.
- 2. Інтегровані середовища розробки (IDE):
 - Visual Studio Code (VS Code): Безкоштовний та легкий у використанні редактор коду з великою кількістю розширень для різних мов програмування.
 - о IntelliJ IDEA: Популярна IDE для розробки на Java, Python та інших мовах.
 - PyCharm: Спеціалізована IDE для розробки на Python.

- Eclipse: Інша популярна IDE для Java та інших мов.
- NetBeans: Спеціалізується на розробці програм на різних мовах програмування, зокрема на Java, PHP, C/C++, та інших.
- 3. Фреймворки та бібліотеки:
 - React.js та Angular.js: Фреймворки для розробки веб-додатків на JavaScript.
 - Spring Framework: Фреймворк для розробки веб-додатків на Java.
 - о Django: Фреймворк для розробки веб-додатків на Python.

Для реалізації тренажера було обрано мову програмування Java, особливо набір бібліотек і компонентів Java Swing, та інтегроване середовище розробки NetBeans.

Вибір мови програмування Java може бути обґрунтованим з багатьма причинами, оскільки Java є потужною та широко використовуваною мовою з багатьма перевагами.

NetBeans є потужним та розширюваним інструментом розробки з великою кількістю переваг, які роблять його привабливим для розробників різних спеціалізацій та різних видів проектів.

NetBeans є вільною та відкритою для розробників, що означає, що ви можете використовувати його безкоштовно і навіть змінювати вихідний код, якщо це необхідно для ваших потреб.

NetBeans має велику кількість функціональних можливостей, включаючи підсвічування синтаксису, автоматичне завершення коду, інтегровану систему керування версіями (включаючи Git), візуальні редактори для різних мов і багато інших.

NetBeans має візуальні редактори та дизайнери для створення графічних інтерфейсів користувача для веб-додатків та десктоп-програм.

Java Swing - це набір бібліотек і компонентів, які використовуються для створення графічного інтерфейсу користувача (GUI) в програмах на мові програмування Java. Java Swing надає розробникам можливість створювати різноманітні віконні додатки, включаючи десктопні програми, ігри, редактори, віджети та інше. Ось кілька ключових аспектів використання Java Swing при створенні програм:

- Створення графічного інтерфейсу: Java Swing надає розробникам можливість створювати різні елементи графічного інтерфейсу, такі як вікна, кнопки, поля введення, списки, таблиці, панелі і багато інших. Ці компоненти можуть бути розміщені та налаштовані на вікні додатку для забезпечення бажаного функціоналу та вигляду.
- Можливості налаштування: Java Swing дозволяє розробникам налаштовувати зовнішній вигляд та поведінку компонентів GUI. Ви можете визначити кольори, шрифти, розміри, макети та інші атрибути для створення вигляду, який відповідає вашим потребам.
- Подійна модель: В Java Swing використовується подійна модель, яка дозволяє обробляти події, такі як кліки мишею, натискання клавіш, ресайз вікна тощо. Ви можете додавати обробники подій для реагування на дії користувача.
- Можливості міжпроцесорної інтеграції: Ви можете інтегрувати Java Swing з іншими технологіями та бібліотеками Java для створення повноцінних багатоплатформових додатків, включаючи серверну частину.
- 5. Підтримка різних макетів: Java Swing має різні менеджери розміщення (layout managers), такі як BorderLayout, FlowLayout, GridLayout тощо, що дозволяє докладно керувати розміщенням компонентів на вікні.
- 6. Мале споживання ресурсів: Java Swing є відомою своєю здатністю працювати ефективно та витрачати мало системних ресурсів, що робить його відмінним вибором для розробки легковагових додатків.
- 7. Підтримка мультиплатформеності: Відтворюючи компоненти Swing на різних платформах, ви можете створювати додатки, які працюють на різних операційних системах без змін вихідного коду.

Загалом, Java Swing є потужним інструментом для розробки графічних додатків на Java, і він підходить для створення різних видів програм, від

десктопних додатків до ігор і веб-проектів, де важливо мати користувацький інтерфейс.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Блок-схема алгоритму роботи тренажера

Блок-схема дозволяє візуально представити алгоритм роботи програми. Це робить процес розробки та розуміння програми більш зрозумілим для розробників та інших зацікавлених осіб.

Блок-схеми можуть служити частиною документації для програмного проекту. Вони допомагають зберігати інформацію про логіку програми, яка може бути корисною для майбутніх розробників чи технічних спеціалістів.

Блок-схема алгоритму реалізації елемента тренажеру «Побудова скороченого скінченого автомату» має такий вигляд:



Рисунок 4.1 – Блок-схема алгоритму роботи тренажера



Рисунок 4.2 – Продовження блок-схеми алгоритму «Побудова скороченого

скінченого автомату»



Рисунок 4.3 – Продовження блок-схеми алгоритму «Побудова скороченого

скінченого автомату»



Рисунок 4.4 – Блок-схема алгоритму перевірки відповіді

4.2. Опис програмної реалізації

Створення програми з використанням JPanel у бібліотеці Java Swing передбачає створення графічного інтерфейсу користувача (GUI) за допомогою різних компонентів, включаючи JFrame i JPanel.

Після створення JPanel (рис.4.5) можна створювати і додавати різні компоненти, такі як кнопки, текстові поля, мітки тощо, щоб створити бажаний інтерфейс користувача.



Рисунок 4.5 – Створення JPanel

Також є змога налаштовувати розмір, колір, шрифт, текст і інші властивості компонентів, щоб вони відповідали потребам.

Щоб визначити, як компоненти розташовані на JPanel, використовується менеджер розміщення (Layout Manager), такий як FlowLayout, BorderLayout, GridLayout або інші (рис.4.6). В даному випадку використано Free Design.

Після створення дизайну головної сторінки (рис. 4.7) можна переходити до обробки подій. Якщо програма має взаємодію з користувачем (наприклад, обробка натискань кнопок), то обов'язково потрібно додати обробники подій для відповідних компонентів. ${\it Q}$ The Tools>Palette> Swing/AWT Components menu item allows you to modify the content of the Palette.







Рисунок 4.7 – Вибір розміщення

При натисканні кнопки «Повторити теоретичний матеріал» відбувається подія void theoryBtnMouseClicked, що відкриває нове вікно, де міститься теоретичний матеріал.

```
private void theoryBtnMouseClicked(java.awt.event.MouseEvent evt) {
    //this.setVisible(false);
    TheoryFrame theoryFrame = new TheoryFrame();
    theoryFrame.setVisible(true);
}
```

x

Аналогічним чином реалізована подія void treningBtnMouseClicked для кнопки «Почати тренінг».

```
private void treningBtnMouseClicked(java.awt.event.MouseEvent evt) {
    //this.setVisible(false);
    TreningFrame treningFrame = new TreningFrame();
    treningFrame.setVisible(true);
}
```

Оскільки теоретичний матеріал і тренінг знаходяться в нових вікнах, то для цього створено дві панелі TheoryFrame і TreningFrame. Для них також розроблено свій дизайн і функціонал.

Sext Page × Виригалие зон × В Тикор frame.jon × X (Cuestion.jon × E) Taki,jon ×

To change layout manager of a container use Set Layout submenu from its centert menu.

x

TEOPETI/IHINЙ MATEPIAN

Sakpurm

Так TheoryFrame має такий вигляд (рис.4.8):

Рисунок 4.8 – Панель TheoryFrame

Тут наявна лише одна кнопка, що закриває дане вікно.

private void closeBtnMouseClicked(java.awt.event.MouseEvent evt) {
 this.dispose();
}

,

Панель TreningFrame має вже більш складну структуру і весь функціонал тренінгу.



Рисунок 4.9 – Структура TreningFrame

Першим кроком було додано необхідні імпорти для пакетів Swing і AWT.

import java.awt.CardLayout; import java.awt.Color; import java.awt.Component; import java.awt.Font; import java.util.Arrays; import java.util.List; import javax.swing.JCheckBox; import javax.swing.JLabel; import javax.swing.JTable; import javax.swing.table.DefaultTableCellRenderer; import javax.swing.table.DefaultTableModel;

Було створено два класи Question i Task. Перший буде представляти окреме питання. У цьому класі знаходяться поля, такі як текст питання, список варіантів відповідей, повідомлення про помилку.

private String text; private List<String> answerOptions; private String wrong; public Question(String text, List<String> answerOptions, String wrong) { this.text = "<html>"+text+"</html>"; this.answerOptions = answerOptions; this.wrong = "<html>"+wrong+"</html>";

```
}
```

Другий представляє завдання з полями:

- текст завдання;
- текст підказки;
- текст повідомлення про помилку.

private String text; private String hint; private String wrong; private String[][] answer;

```
public Task(String text, String hint, String wrong, String[][] answer) {
    this.text = "<html>"+text+"</html>";
    this.hint = "<html>"+hint+"</html>";
    this.wrong = "<html>"+wrong+"</html>";
    this.answer = answer;
}
```

В кожному класі прописані геттери (getter) - це методи класу, які дозволяють звертатися до атрибутів класу (поля). Геттери використовуються для отримання значення атрибуту. Вони дозволяють контролювати доступ до атрибутів класу.

Для Question:

```
public String getText() {
    return text;
    }

public List<String> getAnswerOptions() {
    return answerOptions;
    }

public String getWrong() {
    return wrong;
    }

Для Task:
    public String getText() {
    return text;
    }

    public String getHint() {
```

```
return hint;
}
public String getWrong() {
  return wrong;
}
public String[][] getAnswer() {
  return answer;
}
```

Наступник кроком оголошено деякі змінні, а саме:

- CardLayout cl менеджерів розміщення;
- int step = $1 \kappa p \circ \kappa u$ тренінгу;
- boolean hint = true позначка для виведення спочатку підказки;
- String[][] rightAnsw таблиця з правильними відповідями для деяких завдань;
- MyTableModel tableModel модель таблиці;
- DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer() – рендерер комірок таблиці.

Також введено тексти питань завдань і правильні відповіді для таблиці (див. Додаток А).

Question $q^2 = new$ Question("Явно еквівалентними станами $\langle i \rangle$ називаються $\langle /i \rangle$ ", Arrays.asList("<html>такі еквівалентні стани <i>a_i</i>, <i>a_j</i>, для яких рядки в таблиці переходів і таблиці виходів однакові, або стають однаковими при заміні <i>a_i</i>, <i>a_j</i>", "<html>такі еквівалентні на стани <i>a_i</i>, <i>a_j</i>, для яких рядки в таблиці переходів і таблиці однакові", "<html>такі еквівалентні <i>a_i</i>, виходів стани <i>a_j</i>, для яких рядки в таблиці виходів однакові або стають однаковими при заміні <i>a_i</i>, на <i>a_j</i>","<html>такі еквівалентні стани <i>a_i</i>, <i>a_j</i>, для яких рядки в таблиці переходів однакові"), style=\"color:red\">Вибір неправильний. "<span <i>Явно еквівалентними станами називаються</i> <i>a_i</i>. такі еквівалентні стани <i>a_j</i>, для яких рядки в таблиці переходів і таблиці виходів однакові, або стають однаковими при заміні <i>a_i</i>, на <math><i>a_j</i>);

Task t1 = new Task("Задано СА. Вибрати пари явно еквівалентних станів",

"<i>Явно еквівалентними станами називаються</i> такі еквівалентні стани <i>a_i</i>, <i>a_j</i>, для яких рядки в таблиці переходів і таблиці виходів однакові, або стають однаковими при заміні <i>a_i</i>, на <i>a_j</i>, "Рядки виходів однакові для станів 0, 2, 5. Це рядок 1 1 0. Крім того, для станів 0 і 2 рядки станів однакові (5 2 3). Тому <i>b>0 і 2 – явно еквівалентні стани

станів 1 і 3 рядки станів стають однаковими, якщо замінити стан 1 на 3 (2 1 4 на 2 3 4). Тому <i>1 і 3 – явно еквівалентні стани/i>");

•••

Створено клас DefaultTableCellRenderer для налаштування вигляду клітинок таблиці (JTable) у програмі. Конструктор приймає об'єкт JTable і зберігає його в полі table для подальшого використання.

```
public CustomTableCellRenderer(JTable table) {
    this.table = table;
}
```

Метод getTableCellRendererComponent визначає, як буде виглядати компонент клітинки. Якщо дані у клітинці невірні (за допомогою методу isDataIncorrect), текст у цій клітинці встановлюється червоним кольором. В іншому випадку використовується колір тексту таблиці. Також встановлюється горизонтальне вирівнювання тексту по центру.

@Override

public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column) {

Component cellComponent = super.getTableCellRendererComponent(table, value, isSelected, hasFocus, row, column);

```
if (isDataIncorrect(row, column) && row > 0) {
    cellComponent.setForeground(Color.RED);
    } else {
    cellComponent.setForeground(table.getForeground());
    }
    setHorizontalAlignment(JLabel.CENTER);
    return cellComponent;
}
```

Метод isDataIncorrect визначає, чи дані у конкретній клітинці таблиці є невірними. Він порівнює значення у клітинці з правильними даними, які, очевидно, знаходяться у rightAnsw. Якщо дані невірні, метод повертає true, інакше - false.

```
private boolean isDataIncorrect(int row, int column) {
    if (row > 0) {
        String cell = (String) table.getValueAt(row, column);
        return !rightAnsw[row-1][column].equals(cell);
    } else {
        return false;
    }
```

Клас MyTableModel може бути використаний для створення моделі таблиці з додатковою логікою управління доступністю редагування комірок у конкретних стовпцях.

Конструктор приймає дані для заповнення таблиці (data), імена стовпців (columnNames) та масив логічних значень canEdit, який вказує, чи можна редагувати комірки у відповідних стовпцях.

```
public MyTableModel(Object[][] data, String[] columnNames, boolean[] canEdit) {
    super(data, columnNames);
    this.canEdit = canEdit;
}
```

Метод getColumnClass визначає клас для стовпця з вказаним індексом. У цьому випадку встановлено, що всі значення у таблиці будуть представлені як рядки (String).

```
@Override
public Class getColumnClass(int columnIndex) {
   return java.lang.String.class;
}
```

}

Метод isCellEditable визначає, чи можна редагувати комірку у вказаному рядку та стовпці. Зазначено, що комірки можна редагувати тільки у рядках з індексом більше 0 (рядки поза заголовком) та у відповідних стовпцях, для яких canEdit дорівнює true.

```
@Override
public boolean isCellEditable(int rowIndex, int columnIndex) {
  return (rowIndex > 0 && canEdit[columnIndex]);
}
```

Метод дозволяє динамічно змінювати доступність редагування для конкретного стовпця. Встановлює новий статус редагування для стовпця та викликає fireTableStructureChanged(), щоб оновити вигляд таблиці.

```
public void setCellEditable(int columnIndex, boolean editable) {
    canEdit[columnIndex] = editable;
    fireTableStructureChanged(); // Оновлення таблиці
}
```

Для виведення наступного питання peaniзовано функцію void nextQuestion(Question question) з аргументом типу Question. Спочатку очищується вибір у RadioButton, потім встановлюється текст у відповідні елементи.

```
private void nextQuestion(Question question) {
    buttonGroup1.clearSelection();
    qText.setText(question.getText());
    List<String> answerOptions = question.getAnswerOptions();
    qRb1.setText(answerOptions.get(0));
    qRb2.setText(answerOptions.get(1));
    qRb3.setText(answerOptions.get(2));
    qRb4.setText(answerOptions.get(3));
    qWrong.setText(question.getWrong());
    qWrong.setVisible(false);
}
```

Функція void nextTask(Task task) виконує схожі дії для Task.

Присвоюється змінній hint значення true і елементам встановлюється текст.

```
private void nextTask(Task task) {
    hint = true;
    tText.setText(task.getText());
    tWrong.setText(task.getHint());
    if(task.getHint().equals("<html></html>")) {
        tWrong.setText(task.getWrong());
    } else {
        tWrong.setText(task.getHint());
    }
    tWrong.setVisible(false);
}
```

Для завдань алгоритму «Мінімальна форма скінченого автомата»

створена власна функція.

```
private void nextMinTask(Task task) {
    hint = true;
    t2Text.setText(task.getText());
    if(task.getHint().equals("")) {
        t2Wrong.setText(task.getHint());
    } else {
        t2Wrong.setText(task.getWrong());
    }
    t2Wrong.setVisible(false);
}
```

При відкритті вікна виконується подія void formWindowOpened, що обробляє такі дії:

• відключення вкладки у вкладених панелях;

- налаштування шрифту для заголовку таблиці;
- центрування тексту у комірках таблиці;
- створення та налаштування моделі для jTable2;
- налаштування рендерера та центрування для кожного стовпця в jTable1, jTable2, jTable3, jTable5;
- приховання компонентів qWrong та tWrong;
- ініціалізація змінної step значенням 1.

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
           jTabbedPane1.setEnabledAt(1, false);
           jTable1.getTableHeader().setFont(new Font("Segoe UI", Font.PLAIN, 14));
           centerRenderer.setHorizontalAlignment(JLabel.CENTER);
           for (int i=0; i < jTable1.getColumnCount(); i++)
              jTable1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
           tableModel = new MyTableModel(
              new Object [][] {
                {null, "0", "1", "2"},
                {"0 (2)", "", "", ""},
                {"1 (3)", "", "", ""},
                {"4", "", "", ""},
                {"5", "", "", ""}
              },
              new String [] {
                "<html><b>Стани a<sub>i</sub></b>", "",
                                                                             "<html><b>Входи
x<sub>i</sub></b>", ""
              },
              new boolean [] {
                false, true, true, true
              }
           ):
           jTable2.setModel(tableModel);
           for (int i=0; i < jTable2.getColumnCount(); i++)
              jTable2.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
           for (int i=0; i < jTable3.getColumnCount(); i++)
              jTable3.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
           for (int i=0; i < jTable5.getColumnCount(); i++)
              jTable5.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
           qWrong.setVisible(false);
           tWrong.setVisible(false);
           step = 1;
         }
```

Кнопка «Продовжити» опрацьовує процес проходження тренінгу. Згідно поточному кроку здійснюється перевірка відповіді (див. Додаток А). Якщо відповідь правильна, то виводиться наступне питання чи завдання згідно алгоритму. Якщо ні – відображається підказка або повідомлення про помилку.

```
private void contBtnMouseClicked(java.awt.event.MouseEvent evt) {
    switch(step) {
        case 1 -> {
            if(qRb2.isSelected()) {
                nextQuestion(q2);
                step++;
            } else {
                qWrong.setVisible(true);
            }
        }
}
```

jRadioButton i jCheckBox перевіряються завдяки isSelected(), jTextField – за допомогою .getText().equals(), заповнення таблиці через:

```
boolean arrCheck = true;
String cell;
rightAnsw = t4.getAnswer();
for (int i = 1; i<jTable2.getRowCount(); i++) {
  for (int j = 1; j<jTable2.getColumnCount(); j++) {
     cell = (String)jTable2.getValueAt(i,j);
     if(!rightAnsw[i-1][j].equals(cell)) {
        arrCheck = false;
        break;
     }
  }
}
```

При відкритті нової вкладки у jTabbedPanel виконується кілька дій:

- включення та встановлення активної вкладки;
- приховання компоненту t2Wrong;
- ініціалізація нової моделі для jTable4;
- налаштування рендерера та центрування для кожного стовпця в jTable4;
- зміна тексту кнопки contBtn та інкрементування step.

jTabbedPane1.setEnabledAt(1, true); jTabbedPane1.setSelectedIndex(1); t2Wrong.setVisible(false);

```
tableModel = new MyTableModel(
                  new Object [][] {
                     {"позначення", "", "позначення", "0", "позначення", "1", "позначення",
"2"},
                  },
                  new String [] {
                     "", "<html><b>Пари</b>", "", "<html><b>Входи x<sub>i</sub></b>",
"", "", ""
                  },
                  new boolean [] {
                     false, true, false, false, false, false, false
                  }
                );
                jTable4.setModel(tableModel);
                for (int i=0; i < jTable4.getColumnCount(); i++)
                  jTable4.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
                contBtn.setText("Продовжити");
                step++;
```

На останньому кроці вікно закривається.

```
case 18 -> {
    this.dispose();
```

Користувачу потрібно взаємодіяти з jTable певним чином лише на певних кроках (step) та у певних стовпцях. Тому розроблено подію клікання мишею (MouseClicked) для jTable4.

В залежності від значення step та стовпця, в якому був клік, виконуються відповідні дії. Наприклад:

• Якщо step piвне 13 і клікнуто по стовпцю 0 (перший стовпець), змінюється значення на "V1" або "*", в залежності від поточного значення комірки.

• Аналогічно для step 14 та 15, а також для стовпців 2, 4, та 6 (кроки для введення "w").

```
private void jTable4MouseClicked(java.awt.event.MouseEvent evt) {
    if (step >= 13 && step <=15) {
        int row = jTable4.rowAtPoint(evt.getPoint());
        int col = jTable4.columnAtPoint(evt.getPoint());
        String cell = (String)jTable4.getValueAt(row,col);</pre>
```

```
if (col == 0) {
       switch(step) {
          case 13 -> {
            if(cell.equals("*")) {
               jTable4.setValueAt("V1", row, col);
             } else {
               jTable4.setValueAt("*", row, col);
             }
          }
          . . .
       }
     else if (col == 2 || col == 4 || col == 6) 
       if(cell.equals("*")) {
          jTable4.setValueAt("w", row, col);
       } else {
          jTable4.setValueAt("*", row, col);
       }
     }
  }
}
```

4.3. Необхідна користувачу інструкція

Після відкриття тренажера відображається головна сторінка (рис.4.10):

• тема тренажеру «Мінімізація скінчених автоматів»;

 інформація про автора (прізвище та ініціали виконавця, університет, прізвище та ініціали керівника);

- кнопка для переходу до теоретичного матеріалу;
- кнопка для переходу до тренінгу.



Рисунок 4.10 – Головна сторінка

Щоб перейти до теорії потрібно натиснути «Повторити теоретичний матеріал». Відкриється відповідне вікно (рис. 4.11).

ТЕОРЕТИЧНИИ МАТЕРІАЛ
 1. Властивості автоматних відображень та мінімізація СА
Автоматні відображення мають властивість зберігати довжину слів
та початкові відрізки. Тобто, якщо X^{*} та Y^{*} – множини слів
алфавітів X і Y , то $\mathit{sidoofpa}$ ження $\varphi_a:X^* \to Y^*$ є автоматним тоді і
тільки тоді, коли довжина слова $arphi_a(ho)$ дорівнює довжині слова $ ho$, а
також $arphi_a(ho)$ є початковим відрізком слова $arphi_a(ho q) \;\; orall ho, q \in X^*.$
Стани а та b двох автоматів зі спільними вхідним і вихідним
алфавітом називаються еквівалентними, якщо однаковими є автоматні
відображення $\varphi_a=\varphi_b$, де φ_a – для одного СА, φ_b – для іншого СА.
(Зокрема це означення дає також еквівалентні стани одного і того ж автомата). Позначають еквівалентні стани так: <i>a</i> ~ <i>b</i> .
Автомати A_1 та A_2 називаються еквівалентними, якщо для будь-
якого стану a автомата A_i існує стан b автомата $A_2: a \sim b$.
Автомат $A_{\mathfrak{g}}$ називається зведеним, якщо всі його стани попарно не
еквівалентні.
Можна ставити задачі мінімізації скінченного автомата за тим чи
іншим критерієм.
гозплянемо минимізацію кількості станів автомата. Задача
полягае в пооудовг по заданому скинченному автомату такого СА, що

Рисунок 4.11 – Теоретичний матеріал

Для відкриття тренінгу слід натиснути «Початок тренінгу». Початок тренінгу передбачає спочатку відповіді на питання (рис. 4.12).

A		
	ТРЕНІНГ	
Побудова скороченого скінченого автомату	Мінімальна форма скінченого автомату	
Скінченним автоматом (СА) називають п'ятірку , де л (вхідний алфавіт); Y – множина вихідних сигналів (в	А — множина внутрішніх станів, X — множина вхідних suxідний aлфasim); а δ та λ — функції переходів та вихо	с <i>сигналів</i> одів.
1. Скороченим СА називається		
А) еквівалентний до заданого СА із більшою кіли виходом, що і в заданому СА	ькістю станів, і який забезпечує ту ж відповідність .	між входом і
 Б) еквівалентний до заданого СА із меншою кіль виходом, що і в заданому СА 	кістю станів, і який забезпечує ту ж вібповібність і	игж входом і
 В) такий скінчений автомат, у якого кількість 	вихідних сигналів мінімальна	
Г) еквівалентний до заданого СА із меншою кіль між входом і виходом, що і в заданому СА	кістю станів та виходів, і який забезпечує ту ж відг	повідність
	Продовжити	

Рисунок 4.12 – Теоретичні питання

Якщо відповідь була вибрана неправильно, то відображається повідомлення про помилку (рис. 4.13).



Рисунок 4.13 – Повідомлення про неправильний вибір

Після правильної відповіді виводиться наступне питання. Відповідаємо на три питання і відображається завдання (рис. 4.14). В даному випадку вибираються декілька варіантів.

<u>ی</u>									
ТРЕНІНГ									
Побудова скороченого скінченого автомат	у Мінімальна фо	рма скінченого автомап	ny						
Задано СА. Вибрати пари явно еквівалентних станів									
	Загал	ьна таблиця переходів СА	4						
	Стани а _і	Входи х _і							
	0	0 1 5/1 2/1	3/0						
	2	2/0 1/0 5/1 2/1	4/1 3/0						
	3 4	2/0 3/0 4/0 0/0	4/1 2/1						
	5	0/1 1/1	3/0						
			314 0 415						
	0i3 = 1i4	215	515						
)i4 🗌 1i5								
	015								
_									
		Продовжити							

Рисунок 4.14 – Перше завдання

Якщо перший раз відповісти неправильно, то виводиться спочатку підказка (рис. 4.15), а потім повідомлення про помилку (рис. 4.16).

ТРЕНІНГ									
Побудова скороченого скінченого автомату	Мінімальна	форма скінчена	го автомату	/					
Задано СА. Вибрати пари явно еквівалентних стані	в								
	3a.	гальна таблиця	переходів СА						
	Стани а _і	Вх	оди х _і						
-	0	5/1	2/1	2 3/0					
-	1	2/0	1/0	4/1					
	3	2/0	3/0	4/1					
	4 5	4/0 0/1	0/0	2/1 3/0					
	i1 🗌 1	2 21	з 🗌 з і	4 🗌 4 i 5					
	i 2 🔲 1	i 3 🗌 2 i -	4 🗌 3 i	5					
. 0	i3 🗌 1	i4 🗌 2 i :	5						
	i4 🗌 1	5							
	i 5								
Явно еквівалентними станами називаються т	такі еквівалентн	і стани а _ї , а _л для :	яких рядки в та	аблиці переходів і	таблиці виходів однакові, або стають				
однаковими при заміні <i>а</i> , на <i>а</i> ,									
- ,									
		Продовя	кити						

Рисунок 4.15 – Підказка



Рисунок 4.16 – Повідомлення про помилку

Останнє завдання побудови скороченого скінченого автомату передбачає собою заповнення таблиці (рис. 4.17).

▲	1.1.2.2	-	and the second second			_ C X		
ТРЕНІНГ								
Побудова скороченого скінченого автомату Мінімальна форма скінченого автомату								
Задано СА. Заповнити наступну таблицю								
	Загал	њна табл	иця переходів С	CA				
(Стани а _і		Входи х _і					
		0	1	2				
-	0	5/1	2/1	3/0				
	2	5/1	2/1	3/0				
-	3	2/0	3/0	4/1				
	5	0/1	1/1	3/0				
C	тани а		Входи х					
		0	1	2				
_	0(2)							
	4							
	5							
		Прод	овжити					

Рисунок 4.17 – Заповнення таблиці побудови скороченого СА

Якщо хоч в одній комірці неправильна відповідь, то виведеться відповідне повідомлення і всі неправильні значення будуть позначені червоним

кольором (рис. 4.18). При наступній помилці таблиця буде автоматично заповнена вірними значеннями (рис. 4.19).



Рисунок 4.18 – Виділення неправильних значень

		ТР	ЕНІНГ		
будова скороченого скінченого автомат	у Мінімально	а форма скін	ченого автома	my	
но СА. Заповнити наступну таблицю					
	3	агальна таб	лиця переходів С	A	
	Стани а _і		Входи х _і		
		0	1	2	
	0	5/1	2/1	3/0	-
	2	5/1	2/1	3/0	
	3	2/0	3/0	4/1	
	5	0/1	1/1	3/0	
	Стани а _і 0 (2) 1 (3) 4 5	0 5/1 0/0 4/0 0/1	Входи х, 1 0/1 1/0 0/0 1/1	2 1/0 4/1 0/1 1/0	
Помилка. Значення дублювати із таб	лиці 1, але крім	тих клітинок,	в яких містяться	елементи 2 і 3,	скрізь потрібно замінити 2 на 0 і 3 на

Рисунок 4.19 – Автоматичне заповнення таблиці значеннями

Після правильної відповіді закінчується проходження побудови скороченого скінченого автомату (рис. 4.20).

						- C - X -				
ТРЕНІНГ										
Побудова скороченого скінченого автомату	Побудова скороченого скінченого автомату Мінімальна форма скінченого автомату									
Задано СА. Заповнити наступну таблицю										
	Загал	њна табл	иця переходів (CA						
c	Стани а _ј		Входи х _і							
		0	1	2						
-	0	5/1	2/1	3/0						
	2	5/1	2/1	3/0						
-	3	2/0	3/0	4/1						
	5	0/1	1/1	3/0						
C	тани а		Входи х							
		0	1	2						
-	0 (2)	5/1	0/1	1/0						
	4	4/0	0/0	0/1						
	5	0/1	1/1	1/0						
	Отриманс	скорочен	у форму задан	oro CA.						
	Роботу ал	горитму з	авершено							
	Мінімальна	а форма	скінченого а	автомату						
				_						

Рисунок 4.20 – Результат побудови скороченого СА

Щоб продовжити тренажер зі знаходження мінімальної форми скінченого автомата потрібно натиснути «Мінімальна форма скінченого автомата». Відкриється перше завдання цього алгоритму (рис. 4.21).

<u>*</u>		-				_	-			
	ТРЕНІНГ									
Побудо	Побудова скороченого скінченого автомату Мінімальна форма скінченого автомату									
			Скінчений	автомат задан	но загальною табл	ицею переході	в. Знайти мініма	альну форму скінченого авто	мата.	
				Вибрати се	еред заданих рядкі	в виходів ті, які	є у заданій таб.	лиці		
Загальн	а табли	ця перех	одів СА				00	5 3 2		
Стани а _і	0	Входи х _і 1	2				114	211		
0	1/1	1/0	4/0				011	686		
2	1/1 2/0	1/0	4/0				110	336		
4	5/1 7/0	3/0 8/1	2/0							
6	5/1 3/1	1/0 3/0	7/0 6/0							
8	6/0	8/1	6/1							
	Продовжити									

Рисунок 4.21 – Перше завдання знаходження мінімальної форми СА

Подальше проходження аналогічне попереднім етапам, розглянутим вище. Виділимо лише деякі кроки, що відрізняються.

На другому завданні необхідно ввести відповіді у поля (рис. 4.22).

4	£									
	ТРЕНІНГ									
	Побудова скороченого скінченого автомату Мінімальна форма скінченого автомату									
	Скінчений автомат задано загальною таблицею переходів. Знайти мінімальну форму скінченого автомата.									
i	Випи Загальна таблиця переходів СА поря, Стани в. Входи х.	ати множини станів, у яких однакові рядки виходів (введіть значення у комірки через кому, без пропусків у ку зростання)								
	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$S' = \{ 0.2, 6 \},$ $S'' = \{ 1.3, 8 \}$								
	3 2/0 1/1 1/1 4 5/1 3/0 2/0 5 7/0 8/1 5/1 6 5/1 1/0 7/0 7 3/1 3/0 6/0									
	8 6/0 8/1 6/1									
L		Продовжити								

Рисунок 4.22 – Введення значень у поля

Якщо відповідь неправильна виводиться повідомлення і в полях значення заміняються на правильні (рис. 4.23).

٤									
ТРЕНІНГ									
Побудова скороченого скінченого	Побудова скороченого скінченого автомату Мінімальна форма скінченого автомату								
Скінчений	автомат задано загальною таблицею переходів. Знайти мінімальну форму скінченого автомата.								
Загальна таблиця переходів СА Станя а, Входи х, 0 1 2 0 1/1 1/0 4/0 1 0/0 3/1 3/1 2 1/1 1/0 4/0 3 2/0 1/1 1/1 4 5/1 3/0 2/0 5 7/0 8/1 5/1 6 5/1 1/0 7/0 7 3/1 3/0 6/0 8 6/0 8/1 6/1	Виписати множини станів, у яких однакові рядки виходів (введіть значення у комірки через кому, без пропусків у порядку зростання) <i>S'</i> = { 0,2,4,6,7 }, <i>S''</i> = { 1,3,5,8 }								
Продовжити									

Рисунок 4.23 – Заміна значень на правильні

Даний алгоритм потребує поетапного заповнення таблиці. Так спочатку буде доступна для редагування лише колонка «Пари» (рис. 4.24). Наступне завдання закриє доступ до цієї колонки і відкриє для «Входи *x_i*».



Рисунок 4.24 – Редагування колонки «Пари»

Щоб виконати позначення згідно алгоритму потрібно натиснути на символ «*» в таблиці. При повторному натисненні позначення відміняється (рис. 4.25).



Рисунок 4.25 – Позначення згідно алгоритму

На наступних кроках попередні позначення вже не змінюються. Необхідно виставити далі для V₂ і V₃ (рис. 4.26).

	-			-	-								
	ТРЕНІНГ												
Побудое	ва скоро	ченого	скінченог	о авт	омату	Мінім	альна форм	ла скінченого а	втомату				
<i>Загально</i> Стани а _і	а табли 0 1/1	ця перех Входи х _і 1 1/0	Скінчениі годів СА	і автог Ви ти	мат задан іконайте іх пар для позна	ю загал перші п і відпов	ьною таблиц юзначення з ідного входя Пари	цею переходів. З гідно алгоритму кення х _і , які оби позначення	найти мініма у для пар елег раємо згідно 0	альну форму скі ментів зі стовпц алгоритму. Входи х _і позначення	нченого авт я «Пари». К	омата. рім цього зробіт позначення	ь позначки біл 2
1 2 3 4 5 6 7 8	0/0 1/1 2/0 5/1 3/1 6/0	3/1 1/0 1/1 3/0 8/1 1/0 3/0 8/1	3/1 4/0 1/1 2/0 5/1 7/0 6/0 6/1			* * * * * * * * * * * * * * * * * * *	0.2 0.4 0.6 0.7 1.3 1.5 1.8 2.4 2.6 2.7 3.5 3.8 4.6 4.7 5.8 6.7	A R	$\begin{array}{c} 1.1 \\ 1.5 \\ 1.5 \\ 0.2 \\ 0.7 \\ 0.6 \\ 1.5 \\ 1.5 \\ 1.3 \\ 2.7 \\ 2.6 \\ 5.5 \\ 5.3 \\ 7.6 \\ 5.3 \\ 5.3 \\ \end{array}$	A X X X X X X X X X X X X X	$\begin{array}{c} 1.1 \\ 1.3 \\ 1.1 \\ 3.1 \\ 3.8 \\ 3.8 \\ 1.3 \\ 1.1 \\ 1.3 \\ 1.8 \\ 1.8 \\ 1.8 \\ 3.1 \\ 3.3 \\ 8.8 \\ 1.3 \\ 1.3 \\ \end{array}$	* * * * * * * * * * * * * * * * * * *	$\begin{array}{r} 4.4\\ 4.2\\ 4.7\\ 4.6\\ 3.1\\ 3.5\\ 3.6\\ 4.2\\ 4.7\\ 4.6\\ 1.5\\ 1.6\\ 2.7\\ 2.6\\ 5.6\\ 7.6\\ \end{array}$
Продовжити													

Рисунок 4.26 – Позначення V_2 і V_3 згідно алгоритму

На останньому кроці виводиться результат знаходження мінімальної форми СА (рис. 4.27). Щоб закрити тренінг необхідно натиснути «Закрити».

ТРЕНІНГ											
Побудова скороченого скінченого автомату Мінімальна форма скінченого автомату											
Скінчений автомат зада	Скінчений автомат задано загальною таблицею переходів. Знайти мінімальну форму скінченого автомата.										
Позначиви загальна таблиця переходів СА <u>Стани а Вкоди к 1</u> <u>0 1/1 1/0 4/0</u> <u>1 0/0 3/1 3/1 2/0</u> <u>3 2/0 1/1 1/1 4/0</u> <u>3 2/0 1/1 1/1 4/0</u> <u>5 7/0 8/1 5/1 6/0</u> <u>8 6/0 8/1 6/1</u>	и a_i представника класу еквівалентності S_i одер	завершено.									
	Закрити										
L											

Рисунок 4.27 – Результат знаходження мінімальної форми СА

ВИСНОВКИ

Головним результатом кваліфікаційної роботи є виконання всіх поставлених завдань. Зокрема,

- здійснено огляд необхідного теоретичного матеріалу та наявного подібного програмного продукту;

- опрацьовано методичні рекомендації щодо написання магістерської кваліфікаційної роботи;

- спроектовано та реалізовано інтерфейс програми, підібрано запитання та завдання для наповнення навчального інструменту;

- обрано засоби програмування для реалізації;

- побудовано алгоритм та створено програмну реалізацію основних функцій тренажеру;

- проведено перевірку коректності роботи програми з метою виявлення та виправлення помилок та недоліків;

Створена програма цілком відповідає темі та може бути впроваджена у навчальний процес для вивчення теми «Мінімізація скінчених автоматів» дистанційного навчального курсу «Дискретна математика».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гаркуша С. В. Методичні рекомендації щодо виконання кваліфікаційної роботи студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня магістра / С. В. Гаркуша, О. В. Ольховська, О. О. Черненко. – Полтава: ПУЕТ, 2023. – 68 с.

2. Ємець О. О. Дискретна математика: навчальний посібник для самостійного вивчення навчальної дисципліни студентами денної форми навчання спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра / О. О. Ємець, Т. О. Парфьонова. – Вид 3-тє, допов. і перероб. – Полтава: ПУЕТ, 2023. – 282 с. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/12869

3. Борута I. В. Тренажер «Відношення. Область визначення, область значень, граф, матриця відповідності, переріз за елементами» дистанційного навчального курсу «Дискретна математика» / І. В. Борута, Т. О. Парфьонова // Комп'ютерні науки і прикладна математика (КНіПМ-2021): матеріали наук.-практ. семінару. Випуск 6. / За ред. Ємця О.О. – Полтава: Кафедра ММСІ ПУЕТ, 2021 – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/10408

4. Бурко А. О. Створення тренажеру дистанційного навчального курсу «Дискретна математика» з теми «Алгебра Жегалкіна, способи побудови поліномів Жегалкіна» / А. О. Бурко // Комп'ютерні науки і прикладна математика (КНіПМ-2021): матеріали наук.-практ. семінару. Випуск 6. / За ред. Ємця О.О. – Полтава: Кафедра ММСІ ПУЕТ, 2021. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/10409

5. Ліхоузова Т. А. Дискретна математика. Практикум [Електронний pecypc]: навч. посібник для студ. спеціальностей 121 «Інженерія програмного забезпечення», 126 «Інформаційні системи та технології» / Т. А. Ліхоузова. – Київ: КПІ ім. Ігоря Сікорського, 2020. – 62 с. – Режим доступу: file:///C:/Users/owner/Downloads/DM_praktykum%20(4).pdf

Копитко М.Ф. Основи програмування мовою Java: Тексти лекцій / М.Ф. Копитко, К.С. Іванків. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002.– 83 с.

7. Тарнавський Ю. А. Java-програмування: комп'ютерний практикум [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки», освітньо-професійної програми «Комп'ютерний моніторинг та геометричне моделювання процесів і систем» / КПІ ім. Ігоря Сікорського; уклад.: Ю. А. Тарнавський. – Електронні текстові дані (1 файл: 686 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 95 с.

8. Matific Play для учнів: <u>https://www.matific.com/ua/uk/home/matific-</u> <u>play/#playnow</u>

9. Новатіка. Навчання із захопленням: Збірник онлайн тренажерів з математики, алгебри та геометрії: <u>https://novatika.org/uk/clas-1/vidnimannya-1-klas/</u>

10. Шабоян А. Т. Тренажер «Матриці суміжності для неорієнтованих графів без петель» / А. Т.Шабоян, Є. М. Ємець, Ол-ра. О. Ємець // Комп'ютерні науки і прикладна математика (КНіПМ-2020): матеріали наук.- практ. семінару. Випуск 5. / За ред. Ємця О.О. – Полтава: Кафедра ММСІ ПУЕТ, 2020. – С. 17-21. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/8269

11. Шабоян А. Т. Тренажер «Матриці суміжності для орієнтованих графів без петель» / А. Т.Шабоян, Є. М. Ємець, Ол-ра. О. Ємець // Комп'ютерні науки і прикладна математика (КНіПМ-2020): матеріали наук.- практ. семінару. Випуск 5. / За ред. Ємця О.О. – Полтава: Кафедра ММСІ ПУЕТ, 2020. – С. 52-55. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/8905

ДОДАТОК А. КОД ПРОГРАМИ