МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів аналізу скінченних темпоральних автоматів

(тема)

Виконав:
Студент 2 курсу, групи ІПЗм-20-1
Горбенко В.М.
(прізвище, ініціали)

Спеціальність 121 Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник проф. Четвериков Г.Г.
(посада, прізвище)

Допускається до захисту

Зав. кафедри _____ З.В. Дудар
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121 Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав.кафедри _____
(підпис)
« ____ » _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Горбенко Владиславу Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів аналізу скінченних темпоральних автоматів

затверджена наказом університету від 24.03.2022 р. № 412 Ст

2. Термін подання роботи до екзаменаційної комісії 15 05 2022р.

3. Вихідні дані до роботи *теорія автоматів, теорія темпоральних, логік, середовище програмування*

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі і постановка задачі, опис запропонованих варіантів оптимізації, використовувані методи та алгоритми, опис розробленої програмної системи, опис застосованих програмних рішень, аналіз можливих застосувань*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	31 березня 2022	*виконано*
2.	Огляд існуючих методів	10 квітня 2022.	*виконано*
3.	Розробка алгоритмів, проектування та розробка ПЗ	15 квітня 2022	*виконано*
4.	Підготовка пояснювальної записки	20 квітня 2022	*виконано*
5.	Спецчастина	28 квітня 2022	*виконано*
6.	Підготовка презентації та доповіді	03 травня 2022	*виконано*
7.	Попередній захист	05 травня 2022	*виконано*
8.	Нормоконтроль, рецензування	07 травня 2022.	*виконано*
9.	Занесення роботи в електронний архів	11 травня 2022.	*виконано*
10.	Допуск до захисту в зав. кафедри	12 травня 2022	*виконано*

Дата видачі завдання 28 березня 2022р.

Студент _____

(підпис)

Керівник роботи _____ проф. Четвериков Г.Г.

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 96 с., 34 рис., 5 табл., 32 джерела.

СКІНЧЕННИЙ АВТОМАТ, ТЕСТУВАННЯ СКІНЧЕННИХ АВТОМАТІВ, КОНФОРМНІСТЬ, ЧАСОВІ СКІНЧЕННІ АВТОМАТИ.

Об'єктом дослідження є методи представлення програмних реактивних систем кінцевими автоматами.

Метою роботи є розробка методів оптимізації часових автоматів і їх композицій і синтезу тестів з гарантованою повнотою для контролю виконання функціональних і нефункціональних вимог до систем, дію яких описано такою моделлю.

Методи розробки – методи темпоральної логіки і темпоральних автоматів.

У результаті реалізовано алгоритм оптимізації довжини тестів, що перевіряють, з гарантованою повнотою для повністю визначених недетермінованих часових автоматів.

FINITE AUTOMATIC MACHINE, TESTING OF FINITE AUTOMATIC MACHINES, NEURAL NETWORK, TEMPORARY FINITE MACHINES.

The object of research is the methods of representation of software jet systems by finite state machines.

The aim of the work is to develop methods for optimizing time machines and their compositions and synthesis of tests with guaranteed completeness to monitor the implementation of functional and non-functional requirements for systems, the operation of which is described in this model.

Methods of development - methods of temporal logic and temporal automata.

As a result, an algorithm for optimizing the length of testing tests with guaranteed completeness for fully defined nondeterministic time machines is implemented.

Умови публікації пояснювальної записки

Я, _____Горбенко Владислав Миколайович_____
(прізвище, ім'я, по батькові)

студент групи ___ІПЗм-20-1_ здобувач вищої освіти на другому (магістерському) рівні

кафедра програмної інженерії,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему ___ Дослідження методів аналізу скінченних темпоральних автоматів ___,
(назва роботи)

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ ……..……..8
1 Аналіз методів побудови і аналізу скінченних автоматів ……..……...10
1.1 Реактивні скінченні автомати ……10
1.2 Часові автомати ……..……12
1.3 Методи конформності автоматів ……..…17
1.4 Моделі несправності, що перевіряють тести ……..……...19
1.5 Обґрунтування цілей дослідження ……..……..22
2 Опис проведених теоретичних досліджень …….…….....23
2.1 Скінченні автоматні абстракції …….…….....23
2.2 Мінімізація часових автоматів ……....26
3 Аналіз результатів дослідження …….....29
3.1 Мінімізація часових автоматів по станах ……...29
3.2 Алгоритми автоматів з тайм-аутами і часовими обмеженнями ……..37
3.3 Алгоритми композиції часових автоматів ……40
4 Опис розробленої програмної системи ……56
4.1 Опис проектування кінцевих автоматів …….....56
4.2 Моделювання несправності для синтезу тестів ……62
4.3 Реалізація синтезу тестів для сервісних систем ……...64
5 Опис можливості використання отриманих результатів ……...68
Висновки ……72
Перелік джерел посилання …….....75
Додаток А Перелік джерел посилання за науковими напрямами керівника
та науковців кафедри програмної інженерії ……78
Додаток Б Звіт результатів перевірки на унікальність тексту ……..…….....79
Додаток В Слайди презентації ……81
Додаток Г Листінг модуля 90
Додаток Д Апробація роботи……..……...……..93

Додаток Е Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ ……..95

ВСТУП

Наразі час практично будь-яка сфера діяльності людини залежить від якості роботи тієї або іншої інформаційної системи. Прагнення забезпечити стабільну й надійну роботу програмного й апаратного забезпечення таких систем стимулював розвиток відповідних формальних методів аналізу й синтезу. Повністю автоматичні методи аналізу й синтезу [1] знайшли широке застосування при перевірці функціональних і нефункціональних вимог до дискретних і гібридних керуючих систем, їх оптимізації, синтезу складних систем, як композиції більш простих у деякому значенні.

Модель кінцевого автомату має природню реактивність і дозволяє описувати системи з кінцевим числом станів, вхідних впливів і вихідних реакцій. Однак, системи, дія яких залежить від часу, не можуть бути досить адекватно описані в рамках моделі класичного кінцевого автомата, що привело до розширення звичайно автоматних моделей часовими змінними й введенню поняття темпорального (часового) автомата [2]. Модель автомату з однієї часовою змінною є частним випадком моделі напівавтомата з кінцевою множиною часових змінних, для якої багато завдань алгоритмічно нерозв'язні. У ряді випадків розгляд кінцевого автомату з однією часовою змінною (далі часового автомата) являє собою компроміс між складністю й виразністю моделі. Незважаючи на велику кількість досліджень в області часових автоматів, багато завдань аналізу й синтезу для часових моделей залишаються невирішеними.

Одним із завдань оптимізації в теорії автоматів є побудова еквівалентного автомату з мінімальним числом станів, і це завдання залишається невирішеним для часових автоматів. Методи синтезу автомату, що описує роботу композицій різної структури, досить добре вивчені для класичних автоматів, однак, методи композиції для часових автоматів розроблено тільки для деяких окремих випадків. Для часових автоматів не певні умови замкнутості різних класів

автоматів щодо операцій композиції, що вводяться.

У рамках завдання аналізу для дискретних керуючих систем досліджуються властивості кінцевого автомату, зокрема, питання ідентифікації стану кінцевого автомату й можливість побудови кінцевої множини вхідних послідовностей (тестів з гарантованою повнотою), що дозволяють ідентифікувати заданий автомат з певною точністю у відомій множини інших автоматів, наприклад, у множини автоматів, які описують дію тестованої системи.

Методи синтезу тестів, що перевіряють, для детермінованих і недетермінованих кінцевих автоматів вивчаються досить давно, однак, при спробі «перекласти» ці методи на часові автомати, отримані тести мають занадто більшу довжину навіть для детермінованих часових автоматів і вимагають додаткової оптимізації. У ряді випадків питання про повноту таких тестів для часових автоматів залишається невирішеним. Досить часто автори робіт із синтезу тестів на основі автоматів з однією часовою змінною розглядають тільки відносини конформності, що не дозволяє вважати задачу побудови тестів повністю вирішеною.

В роботі потрібно проаналізувати методи та алгоритми вирішення завдання синтезу, що спрямовані на оптимізацію кінцевих автоматів з однією часовою змінною, у той час як завдання аналізу представлене алгоритмами генерації й оптимізації тестів, що перевіряють, з гарантованою повнотою для таких автоматів.

Метою роботи є розробка методів оптимізації часових автоматів і їх композицій і синтезу тестів з гарантованою повнотою для контролю виконання функціональних і нефункціональних вимог до систем, дію яких описано такою моделлю.

1 АНАЛІЗ МЕТОДІВ ПОБУДОВИ І АНАЛІЗУ СКІНЧЕННИХ АВТОМАТІВ

1.1 Реактивні скінченні автомати

Кінцевий автомат [3] являє собою модель із кінцевим числом станів, яка використовується для опису реактивних або реагентних систем, що виконують переходи між станами під дією вхідного впливу, видаючи при цьому деяку вихідну реакцію, і в наш час достатньо часто називаються трансдьюсерами. При вступі вхідного впливу автомат виконує перехід у наступний стан і видає вихідний сигнал (реакцію) відповідно до поточного стану вхідним сигналом, що й робить, після чого очікує наступний вхідний вплив.

У кваліфікаційній роботі кінцевий автомат розглядається як перетворювач потенційно нескінченної множини вхідних послідовностей у множину вихідних послідовностей такої ж довжини [4]. Кінцевий автомат дозволяє описати роботу дискретної системи, у якій виділені кінцеві непусті множини станів, вхідних впливів і вихідних реакцій. Необхідно відзначити, що у випадку наявності недетермінізму в описуваній системі, в одному стані автомата може існувати кілька переходів по тому самому вхідному впливу.

Під кінцевим автоматом [4] розуміється четвірка

$$S = (\Sigma, I, O, h_s),$$

де I – кінцева непуста множина вхідних символів (вхідний алфавіт),

O – кінцева непуста множина вихідних символів (вихідний алфавіт),

$I \cap O = \varnothing$,

S – кінцева непуста множина станів,

$h_s \subseteq (S \times I \times O \times S)$ – відношення переходів.

Ініціальний автомат є п'ятірка

$$S = (\Sigma, I, O, h_s, s_0),$$

де s_0 – виділений початковий стан.

Якщо будь-який стан автомата може бути обраний в якості початкового, то автомат називається неініціальним.

Для автомата S припустима послідовність пар вхідний символ/вихідний символ i_1/o_1, i_2/o_2, …, i_n/o_n у стані s_1 називається вхідно-вихідною послідовністю в стані s_1, якщо в S існує відповідна послідовність переходів (s_1, i_1, o_1, s_2), (s_2, i_2, o_2, s_3), …, (s_n, i_n, o_n, s_{n1}). При цьому $\alpha = i_1, i_2, \ldots, i_n$ є вхідна послідовність, $\gamma = o_1, o_2, \ldots,$ on – відповідна вихідна послідовність, а вхідно-вихідна послідовність позначається α/γ. Для вхідно-вихідної послідовності α/γ вхідну послідовність α іноді називають вхідною проекцією, а відповідну вихідну послідовність γ – вихідною реакцією. Відношення роботи повністю визначеного автомату звичайним чином поширюється на вхідно-вихідні послідовності: якщо автомат у стані s може видати вихідну реакцію γ на вхідну послідовність α і перейти в стан s', те (s, α, γ, s') $\in h_s$. Під довжиною вхідної (вихідної) послідовності розуміється кількість вхідних (вихідних) символів, що входять у послідовність.

Нехай S – повністю визначений автомат. Префіксом послідовності $\alpha = i_1, i_2, \ldots, i_n$ називається початковий відрізок даної послідовності, тобто послідовність $\alpha' = i_1, i_2, \ldots,$ ik, де $0 < k \leq n$. Послідовність α у стані s автомата S покриває стан s', якщо існує префікс α' послідовності α, такий що (s, α', γ, s') $\in h_s$. Якщо (s, α, γ, s') $\in h_s$ у спостережуваному автоматі S, той стан s' називається α/γ– спадкоємцем стану s по вхідно-вихідній послідовності α/γ. Якщо α/γ є спадкоємцем стану s є порожня множина, то іноді говорять, що α/γ спадкоємець стану s не існує.

Для послідовностей $\alpha = i_1, i_2, \ldots, i_n$ і $\gamma = o_1, o_2, \ldots, o_n$ запис α/γ є скороченням запису для вхідно-вихідної послідовності i_1/o_1, i_2/o_2, …, i_n/o_n. Якщо автомат є недетермінованим, то для вхідної послідовності α може існувати кілька вихідних реакцій $\{\gamma_1, \ldots, \gamma_v\}$, і відповідно, у деяких (усіх) станах існує декілька вхідно-вихідних послідовностей з однією й тією же вхідною проекцією. Крім того, α/γ-спадкоємцем стану s у неспостережуваному автоматі може бути

множина станів, однак для спостережуваного, зокрема, детермінованого автомата, які розглядаються в даній роботі, $\alpha/\ \gamma$ стану або не існує, або є синглтон. Для вхідної послідовності α можна розглядати стан s, яким є множина станів, у які автомат може перейти зі стану s під дією вхідної послідовності α. Відзначимо також, що в спостережуваному автоматі S α/γ підмножини станів S є множина $\alpha/\ \gamma_o\rangle$ станів $s \in S'$.

1.2 Часові автомати

У ряді літературних джерел вхідна послідовність називається адаптивної [6], якщо наступний вхідний символ залежить від реакції автомата на попередні вхідні символи. Адаптивна послідовність зазвичай представляється тестовим прикладом (над вхідним алфавітом *I* і вихідним алфавітом *O*) [7], тобто ініціальним автоматом з ациклічним графом переходів, у кожному стані якого або визначений перехід тільки по одному вхідному символу з усіма вихідними символами, або не визначений жоден перехід. Стан, з якого немає переходів, будемо називати тупиковим або термінальним. Висота адаптивного тестового прикладу дорівнює довжині максимальної вхідної послідовності, яка відповідає переходу з початкового стану в термінальний. Якщо множина *I* містить більше одного вхідного символу, то тестовий приклад над вхідним алфавітом *I* і вихідним алфавітом *O* є частковим автоматом.

Часові автомати, або автомати з однією часовою змінної, є, з одного боку, розширенням кінцевого автомата, а з іншої сторони, обмеженням моделі часового напівавтомата з множиною часових змінних [8]. Багато завдань аналізу й синтезу для часових напівавтоматів є алгоритмічно нерозв'язними, і модель кінцевого автомата з однією часовою змінною в ряді випадків виявляється «гарним компромісом» між складністю й виразністю моделі, описуючи роботу систем з часовими аспектами.

Крім дискретних множини станів, вхідних і вихідних символів, часовий автомат бере до уваги часові аспекти, що впливають на поведінку системи, і відповідно, містить часову змінну, яка може ухвалювати невід'ємні раціональні або цілі значення.

Часова змінна постійно збільшує своє значення подібно секундоміру, що веде відлік часу, і скидається в нуль при виконанні автоматом переходу (механізм скидання часу буде докладніше розглянутий при формальному описі моделі). Для розширення кінцевого автомата часовими аспектами існує кілька підходів. Часовий автомат може містити часові обмеження [8], тайм-аути і вихідні затримки [9]. Таким чином, робота часового автомату визначається не тільки поточним станом і вхідним символом, але й значенням змінної часу. Далі буде визначений тип часового автомата, що містить усі три види розширення і його окремі випадки, також використовувані на практиці.

Під автоматом з тайм-аутами й часовими обмеженнями розуміється п'ятірка $S = (S, I, O, \lambda\Sigma, \Delta S)$, де S – кінцева непуста множина станів, I – вхідний алфавіт, O – вихідний алфавіт, $I \cap O = \varnothing$, $\lambda\Sigma \subseteq (S \times I \times O \times S \times \Pi \times Z)$ – відношення переходів, $\Delta S: S \to S \times (N \cup \{\infty\})$ – функція тайм-ауту, що визначає для кожного стану максимальний час очікування вхідного символу (вхідний тайм-аут). У цих визначеннях N – множина натуральних чисел, Z – множина цілих ненегативних чисел, Π – множина вхідних часових інтервалів із проміжку [0; T) виду α, b, де $a, b \in Z$, а T є величина максимального тайм-ауту в автоматі.

Таким чином, кортеж (s, i, o, s', g, d) описує перехід зі стану *s* у стан *s'* під дією вхідного символу *i*, що вплинув на автомат, часова змінна якого в цей момент мала значення t, $t \in g$, з видачою вихідного символу *o* через d тактів часу після вступу вхідного символу. Іноді час *d* обробки вхідного символу називають вихідним тайм-аутом. Часова змінна вимірює час, проведений автоматом у поточному стані до переходу по вхідному символу або тайм-ауту, і час обробки вхідного впливу між вступом вхідного символу й видачою вихідної реакції. Таким чином, часова змінна скидається в 0 при вступі вхідного символу і при

видачі вихідної реакції. Якщо в деякому стані автомата вхідний сигнал не надходить до досягнення часової змінної вхідного тайм-ауту в поточному стані, то автомат змінює свій стан відповідно до функції тайм-ауту. Перехід по тайм-ауту виконується миттєво й «скидає» значення часової змінної в 0. Наприклад, якщо $\Delta S(s) = (s', T)$ і в стані s протягом T одиниць часу на автомат не було подано жодного вхідного символу, то автомат переходить у стан s'.

Якщо $s = s'$, то після досягнення тайм-ауту в стані s відлік часу починається з 0. Відзначимо, що тайм-аут не виконується між подачею вхідного символу і очікуванням відповідно до реакції. Якщо $\Delta S(s) = (s', \infty)$, то автомат може очікувати вхідний символ у стані s нескінченно довго. Для часового автомату S через BS будемо позначати найбільшу кінцеву межу для вхідних часових інтервалів, яка збігається з максимальною величиною тайм-аутів.

Часовий автомат називається повністю визначеним по вхідних символах, якщо для будь-якої пари $(s, i) \in S \times I$, тобто для будь-якого вхідного символу i, що надходить на вхід автомата в стані s, існує кортеж $(s, i, o, s', g, d) \in \lambda\Sigma$. Часовий автомат називається повністю визначеним, якщо автомат повністю визначений по вхідних символах, і для кожної пари $(s, i) \in S \times I$ об'єднання всіх часових інтервалів g з переходів (s, i, o, s', g, d) у стані s по вхідному символу i рівно [0; T), де $\Delta S(s) = (s', T)$. Аналогічно класичним автоматам, часовий автомат може бути визначений до повністю визначеного різними способами залежно від інтерпретації невизначених переходів, однак питання визначення залишається за межами нашої роботи. Якщо для будь-яких двох кортежів $(s, i, o_1, s_1, g_1, d_1)$, $(s, i, o_2, s_2, g_2, d_2) \in \lambda\sigma$ справедливо, що $g_1 \cap g_2 = \varnothing$, те часовий автомат називається детермінованим, інакше – недетермінованим.

Повністю визначений недетермінований часовий автомат називається спостережуваним, якщо для кожної пари переходів (s, i, o, s_1, g_1, d), $(s, i, o, s_2, g_2, d) \in \lambda\Sigma$, такий що $g_1 \cap g_2 \neq \varnothing$, справедливо, що $s_1 = s_2$, тобто наступний стан визначається єдиним чином.

Для кожного стану s часового автомата введемо функцію time(s, t) = s', яка

визначає, у якому стані s' перебуває автомат через t тактів часу, за умови, що вхідний символ не був поданий протягом цього часу. Часовим вхідним символом називається пара (i, t), де i – символ вхідного алфавіту, t – час вступу вхідного символу, відлічуване від моменту видачі автоматом останнього вихідного символу або початку експерименту для першого вхідного символу. Часовим вихідним символом називається пара (o, d), де o – символ вихідного алфавіту, d – число одиниць часу між подачею вхідного символу й видачою вихідного символу (вихідний тайм-аут).

Для визначення реакції часового автомату на часовий вхідний символ (i, t) у стані s на першому кроці визначається стан s' = time(s, t), у якому перебуває автомат у певний момент часу t. У стан s' автомат переходить по ланцюжковому тайм-ауту $\Delta S(s) = (s_1, T_1)$, $\Delta S(s_1) = (s_2, T_2)$, …, $\Delta S(s_n T_n) = (s', T_n)$, такому, що сума тайм-аутів $\Sigma = T_1 + T_2 + \ldots + T_n$ не перевищує значення t і $t - \Sigma < T_{n+1}$, де $\Delta S(s') = (s'', T_{n+1})$, тобто тайм-аут у стані s' не буде досягнутий. Далі автомат виконує перехід (s', i, o, s", g, d), де $t - \Sigma \in g$, та відповідно, має місце вхідно-вихідна пара (i, t) / (o, d) у стані s і наступному стані s". Вирахування величини Σ із часу вступу вхідного символу відображає той факт, що відлік часу починається з 0 при виконанні автоматом переходу по тайм-ауту. Час вступу t' наступного вхідного символу (i', t'), відлічується від моменту видачі автоматом останнього вихідного символу o. Відзначимо, що часова змінна автомату може скидатися тайм-аутами між видачою вихідного символу й вступом наступного вхідного символу та таким чином, час вступу вхідного символу є глобальним і відлічується в зовнішньому середовищі незалежно від внутрішнього таймеру автомату.

Для часового автомата S і часової вхідної послідовності $\alpha = (i_1, t_1), (i_2, t_2) \ldots, (i_n, t_n)$ відповідна часова вихідна послідовність $\gamma = (o_1, d_1), (o_2, d_2) \ldots, (o_n, d_n)$ у стані s називається (вихідною) реакцією. При цьому послідовність пар «часовий вхідний символ / часовий вихідний символ» називається часовою вхідно-вихідною послідовністю й позначається α/γ. Реакція часового автомату на часову вхідну послідовність у стані s обчислюється, починаючи зі стану s.

У випадку, коли автомат є недетермінованим, для часової вхідної

послідовності α може існувати кілька вихідних реакцій $\gamma_1, \ldots, \gamma_\nu$, які є множиною часових вихідних послідовностей. Поняття спадкоємців по часовим вхідно-вихідним послідовностям вводяться аналогічно таким для класичних кінцевих автоматів.

У ряді робіт та додатків розглядаються автомати тільки з часовими обмеженнями або тільки з тайм-аутами. Під автоматом з часовими обмеженнями розуміється четвірка $\mathbf{S} = (S, I, O, \lambda\Sigma)$, де S – кінцева непуста множина станів, I – вхідний алфавіт, O – вихідний алфавіт, $\lambda\Sigma \subseteq (S \times I \times O \times S \times \Pi \times Z)$ – відношення переходів, Π – множина інтервалів із проміжку $[0; \infty)$ і Z – множина цілих невід'ємних чисел. Відповідно, кортеж (s, i, o, s', g, d) описує перехід зі стану s у стан s' під дією вхідного символу i, що зроблено у момент часу t, $t \in g$, після переходу автомату в поточний стан з видачою вихідного символу *o* через *d* тактів часу після вступу вхідного символу. Повнота (частковість) і детермінізм (недетермінізм) для автомату з часовими обмеженнями визначаються аналогічно часовому автомату загального виду, позначають, що відсутність тайм-аутів є ідентичністю нескінченним тайм-аутам.

Автоматом з тайм-аутами [9] називається п'ятірка

$$\mathbf{S} = (\Sigma, I, O, h_s, \Delta S_i),$$

де S – кінцева непуста множина станів,

I – вхідний алфавіт,

O – вихідний алфавіт,

$h_s \subseteq (S \times I \times O \times S \times Z)$ – відношення переходів,

Z – множина цілих невід'ємних чисел,

ΔS: $S \to S \times (N \cup \{\infty\})$ – функція тайм-ауту, що визначає для кожного стану максимальний час очікування вхідного символу,

N – множина натуральних чисел.

Аналогічно автоматам з тайм-аутами і часовими обмеженнями, якщо в

певному стані автомату вхідний сигнал не надходить протягом певного часу (тайм-ауту), то автомат може змінити свій стан. Якщо в автоматі є перехід (s, i, o, s', d) і вхідний символ буде поданий у стані s менш чим через T одиниць часу, де ΔS(s) = (s', T), то автомат перейде в стан s' та через d одиниць часу видасть вихідний символ.

1.3 Методи конформності автоматів

При аналізі й синтезі систем, роботу яких описано кінцевим автоматом, використовують підходящі відносини відповідності (або конформності). Стани *s* і *p* повністю визначених детермінованих (часових) автоматів S і P називаються еквівалентними (s ≅ p), якщо реакції автоматів у цих станах збігаються на кожну (часову) вхідну послідовність [10]. У такому випадку, говорять, що робота автоматів S і P у станах s і p збігається. Якщо стани s і p не є еквівалентними, то вони називаються помітними а вхідна послідовність, реакції на яку в цих станах не збігаються, називається послідовністю, що розрізняє, автомати у станах *s* і *p*. Аналогічним чином визначаються відношення еквівалентності послідовності, що розрізняє стани автоматів.

Ініціальні (часові) автомати S і P еквівалентні, якщо еквівалентні їхні початкові стани. Неініціальні (часові) автомати S і P еквівалентні, якщо для кожного стану автомата S існує еквівалентний стан в автоматі P, і для кожного стану автомата P існує еквівалентний стан в автоматі S. Повністю визначений автомат називається наведеним, якщо будь-які два різні його стани помітні.

Нехай *s* і *p* – стани повністю визначених (часових), можливо, недетермінованих, автоматів S і P. Стан p є редукція [11] стану s (p ≤ s), якщо множина (часових) вхідних-вихідних послідовностей автомата P у стані p утримується в множини (часових) вхідних-вихідних послідовностей автомату S у стані s. Ініціальний автомат P є редукція ініціального автомату S, якщо

відношення редукції виконується для початкових станів цих автоматів, тобто $p_0 \leq s_0$. Автомат P є напівавтоматом автомату S, якщо $P \subseteq S$, $p_0 = s_0$ і $hp \subseteq hs$.

Кінцеві автомати S і P з однаковими вхідними й вихідними алфавітами ізоморфні [12], якщо між множинами станів і переходів цих автоматів можна встановити взаємно однозначну відповідність, тобто існує взаємно однозначне відображення H: $S \rightarrow P$, таке що $(s_i, i, o, s_j) \in h_s$, якщо й тільки якщо $(H(s_i), i, o, H(s_j)) \in h_p$. Часові автомати S і P з однаковими вхідними й вихідними алфавітами ізоморфні, якщо між множинами станів і переходів цих автоматів можна встановити взаємно однозначну відповідність, тобто існує взаємо-однозначне відображення H: $S \rightarrow P$, таке що $(s_i, i, o, s_j, g, d) \in \lambda\Sigma$, якщо й тільки якщо $(H(si), i, o, H(sj), g, d) \in \lambda\Pi$ і $\Delta S(s_i) = (s_j, T)$, якщо й тільки якщо $\Delta P(H(s_i)) = (H(sj), T)$.

Часовий автомат, ізоморфний заданому часовому автомату, може бути отриманий перейменуванням станів вихідного автомату. Для ініціальних ізоморфних (часових) автоматів *S* та *P* має місце $H(s_0) = p_0$.

Завдання синтезу тестів зазвичай розглядається для зв'язних автоматів, кожний стан яких досяжний до початкового. У цьому випадку, для автомата S з початковим станом s_0, послідовність α називається передатною послідовністю для стану s, якщо s є S ⟩|◊™| ⌊≡⌈∑⌉| стану s_0.

Одним з найпоширеніших методів синтезу тестів для ініціальних детермінованих автоматів є обхід графа переходів автомати-специфікації. Такий тест є повним щодо вихідних помилок і для кожного переходу специфікації містить вхідну послідовність, що проходить по цьому переходу. Для того, щоб побудувати тест, що гарантує перевірку не тільки вихідних реакцій автомату і реалізації, але й перехід тестованої системи в стан, запропонований специфікацією, використовуються більш складні методи, найбільш відомим з яких є W-метод, а також його модифікації: Wp-метод [13], SCR-метод [14] і інші подібні методи. W-метод і більшість його похідних можна розділити на два етапи. На першому етапі встановлюється відповідність між станами реалізації

(автомату, що перевіряється) і специфікації, для чого автомат спочатку переводиться в стан, що перевіряється, а потім досягає стану ідентифікації. Відповідні тестові послідовності формуються конкатенацією передатної (-их), послідовностей, що й розрізняє їх. Якщо реакції реалізації й специфікації на такі послідовності не збігаються, то відповідності між станами реалізації й специфікації не існує, автомат, що й перевіряється, не еквівалентний специфікації.

На другому етапі перевіряється відповідність між переходами специфікації й реалізації. Для цього, після досягнення реалізацією кожного стану на неї подаються всі можливі вхідні символи, і досягнутий в такий спосіб стан ідентифікуються. Відповідні тестові послідовності формуються конкатенацією передатної послідовності, вхідного символу, що й розрізняє їх послідовності. Основною відмінністю різновидів W-подібних методів є підхід до ідентифікації станів. Так наприклад, W-метод використовує множину, що розрізняє, містить для кожної пари різних станів автомату і специфікації вхідну послідовність їх, що розрізняє; в Ds-методі множину, що розрізняє, складається з єдиної діагностичної послідовності, тобто послідовності, реакції на яку не збігаються для будь-якої пари різних станів; Hise-метод використовує ідентифікатори станів, що розрізняють заданий стан з усіма іншими станами.

Для недетермінованих автоматів існують W-подібні методи побудови повних тестів щодо редукції [14].

1.4 Моделі несправності, що перевіряють тести

Зазвичай автоматні методи синтезу тестів, що перевіряють, засновані на моделях несправності [15], що визначають еталонну роботу тестованої системи (специфікацію), множина систем, що перевіряються (звичайно кінцева) і відношення конформності між специфікацією й коректо реалізованою системою,

що перевіряється.

Розглядається завдання синтезу тестів для ініціальних детермінованих і недетермінованих повністю визначених часових автоматів і, відповідно, вводиться дві моделі несправності для цих випадків.

При побудові тестів для детермінованих ініціальних повністю визначених автоматів зазначається, що тестована реалізація є «правильно» функціонуючою, тобто є конформна специфікації, якщо автомат – реалізація й автомат і специфікація еквівалентні та, відповідно, для кожної вхідної послідовності реакції реалізації та специфікації в початкових станах збігаються. Якщо спостережувані реакції системи, що перевіряється, не збігаються з очікуваними, або не видаються у відповідні моменти часу, то реалізація містить несправність, тобто реалізація специфікації неконформна. У такому випадку розглядається модель несправності

$$FMDFSM = <S, \cong, \aleph\mu>,$$

де S – ініціальний повністю визначений детермінований автомат і специфікація,

«≅» – відношення еквівалентності,

$\aleph\mu$ – область несправності.

Область несправності містить кожний ініціальний повністю визначений детермінований автомат з таким же вхідним алфавітом, як і у специфікації, і не більш ніж з *m* станами. Якщо в специфікації присутня опціональність, тобто можливість конформної реалізації різних варіантів функціонування, то ставиться завдання побудови тестів для недетермінованих специфікацій. Ініціальний автомат і специфікація може бути недетермінованим. Для побудови тестів по недетермінованій специфікації розглядається модель несправності

$$FMNDFSM = <S, \leq, \aleph\mu>,$$

де S – ініціальний повністю визначений недетермінований автомат і

специфікація,

≤ – відношення редукції

ℵμ – область несправності, що містить кожний ініціальний повністю визначений детермінований автомат з таким же вхідним алфавітом, як і в специфікації, і не більш ніж з *m* станами.

При цьому діє припущення, що реалізація специфікації є конформною, якщо автомат і реалізація є редукцією специфікації, тобто для кожної вхідної послідовності α реакція реалізації в початковому стані втримується в множини реакцій специфікації на α.

Тест, що перевіряє для кінцевого автомата, являє собою кінцеву множину вхідних послідовностей автомат-специфікації, що дозволяє визначити, конформна чи така, що тестована, реалізація специфікації. Відповідно, для часового автомату тест містить часові вхідні послідовності.

Під довжиною тесту розуміється сума довжин усіх (часових) вхідних послідовностей. Якщо тест містить адаптивні вхідні послідовності, то кожна послідовність тесту представляється тестовим прикладом. У цьому випадку, під довжиною тесту розуміється сума висот усіх тестових прикладів. тест, що перевіряє, називається повним щодо моделі несправності FMNDFSM = <S, p, P_m>, якщо для кожного автомату p ∈ P_m такого що P є редукція S, вихідна реакція на кожну послідовність тесту втримується в множини вихідних реакцій S на цю послідовність, у той час як для кожного автомата <S, p, P_m> такого що P не є редукцією S, у тесті є послідовність, вихідна реакція на яку не належить множини вихідних реакцій Sна цю послідовність.

Аналогічно визначається повнота тесту щодо моделі несправності FMDFSM = <S, ≅, ℵμ>, де кожна не еквівалентна автомату S реалізація з ℵμ виявляється повним тестом, у той час як кожна конформна реалізація «проходить» тест [11].

1.5 Обґрунтування цілей дослідження

Повнота (частковість) і детермінізм (недетермінізм) для автомату з тайм-аутами визначається аналогічно класичному кінцевому автомату. Для окремих випадків часового автомата поняття вхідна і вхідна часова послідовність вводяться аналогічно загальному випадку.

Таким чином, проведений аналіз сучасного стану вирішення проблем і для виконання поставленої мети дослідження, необхідно розв'язати наступні завдання:

– розробити метод мінімізації часового автомату та дослідити, чи існує для часових автоматів єдина наведена форма;

– адаптувати операцію паралельної композиції класичних кінцевих автоматів для часових автоматів;

– адаптувати для часових автоматів класичні звичайно автоматні методи синтезу тестів з гарантованою повнотою.

2 ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

2.1 Скінченні автоматні абстракції

Поведінка часових автоматів у ряді випадків можна адекватно бути описана за допомогою кінцевого автомата, тобто з використанням скінченної автоматної абстракції. Відповідно, ряд завдань аналізу й синтезу для часових автоматів можна вирішувати на основі відомих підходів класичної теорії автоматів.

Для автомата з тайм-аутами й часовими обмеженнями відповідна звичайна автоматна абстракція може бути побудована подібно методу з [15].

Нехай $S = (S, I, O, \lambda\Sigma, \Delta S)$ – повністю визначений, можливо, недетермінований часовий автомат і BS – найбільша кінцева межа для часових вхідних інтервалів, яка не перевищує максимального кінцевого тайм-ауту по визначенню часового автомата, у той час як D визначає найбільшу вихідну затримку. Побудуємо звичайно автоматну абстракцію часового автомату, яка є повністю визначеним кінцевим автоматом $\mathtt{AS} = (SA, IA, OA, \lambda A\Sigma)$, де $SA \subseteq \{(s, 0), (s, (0, 1)), \ldots, (s, (BS - 1, BS)), (s, BS), (s, (BS, \infty)): s \in S\}$, $IA = I \cup \{\mathtt{I}\}$, $OA = \{(o, 0), (o, 1), \ldots, (o, D): o \in O\} \cup \{\mathtt{I}\}$, а $\mathtt{I}$ – спеціальний символ звичайно автоматної абстракції. Для стану (s, t_j), $t_j = 0, \ldots, BS$, автомата $\mathtt{AS}$ і вхідного символу i, множина $\lambda A\Sigma$ містить перехід $((s, t_j), i, (o, d), (s', 0))$, якщо тільки існує перехід $(s, i, o, s', g, d) \in \lambda\Sigma$, такий що $t_j \in g$. Для стану (s, g_j), $g_j = (0, 1) \ldots, (BS - 1, BS), (BS, \infty)$, автомата $\mathtt{AS}$ і вхідного символу i, множина $\lambda A\Sigma$ містить перехід $((s, gj), i, (o, d), (s', 0))$, якщо й тільки якщо існує перехід $(s, i, o, s', g, d) \in \lambda\Sigma$, такий що $g_j \subseteq g$. Переходи по спеціальному символу $\mathtt{I}$ відбивають зміна часової змінної між речовинними значеннями з відкритого інтервалу виду (a, b) і цілим значеннями, або переходи по тайм-аутам між станами. Переходи $((s, n), \mathtt{I}, \mathtt{I}, (s, (n, n + 1)))$, $((s, (n, n + 1)), \mathtt{I}, \mathtt{I}, (s, n + 1)) \in \lambda A\Sigma$, якщо й тільки якщо $n < BS$, де $\Delta S(s) = (s', \infty)$. Переходи $((s, n), \mathtt{I}, \mathtt{I}, (s, (n, n + 1)))$, $((s, (n, n + 1)), \mathtt{I},$

I, (s, n + 1)), ((s, T − 1), **I**, **I**, (s, (T − 1, T))) ∈ λAΣ, якщо й тільки якщо n < T − 1, де ΔS(s) = (s', T). Перехід ((s, (n – 1, n)), **I**, **I**, (s’, 0)) ∈ λAΣ, якщо й тільки якщо n = T < ∞, де ΔS(s) = (s', T). Для кожного стану (s, BS) абстракції перехід ((s, BS), **I**, **I**, (s, (BS, ∞))) ∈ λAΣ, якщо ΔS(s) = (s, ∞). Для кожного стану (s, (BS, ∞)) абстракції перехід ((s, (BS, ∞)), **I**, **I**, (s, (BS, ∞))) ∈ λAΣ. Таким чином, для збереження інформації про часову змінну у звичайно автоматну абстракцію вносяться додаткові абстрактні стани, які являють собою пари (стан часового автомата; значення часовий змінної). Для ініціального часового автомату з початковим станом s_0, скінченна автоматна абстракція також є ініціальним автоматом з початковим станом (s_0, 0).

Часова вхідна послідовність α може бути переведена у відповідну послідовність вхідних символів αΦΣM для скінченної автоматної абстракції. Кожний часовий вхідний символ (i, t) представляється послідовністю вхідних символів абстракції виду **I**, **I**, …, **I**, i, де число символів **I** дорівнює числу переходів часової змінної із цілого числа в інтервал i назад за час t. У той же час вихідна реакція абстракції на послідовність **I**, **I**, …, **I**, i, має вигляд **I**, **I**, …, **I**, (o, d), де число символів **I** у вхідній і вихідній послідовності збігається, (o, d) є реакція вихідного часового автомата на (i, t). Таким чином, вихідна послідовність абстракції γΦΣM може бути переведена у відповідну часову вихідну послідовність γ видаленням усіх символів I.

Для автоматів з тайм-аутами звичайно автоматна абстракція вводиться схожим способом [17], з тією лише різницею, що стану абстракції відповідають тільки цілі значенням часової змінної. Відповідно, для скінченної автоматної абстракції **AS** = (SA, I ∪ {**I**}, O ∪ {**I**}, λAΣ, s_0) автомату з тайм-аутами **S** = (S, I, O, λΣ, ΔS, s_0) множина станів абстракції, породжених станом s часового автомату, має вигляд {(s, 0), (s, 1), …, (s, T – 1))}, де ΔS(s) = (s', T).

Звичайно автоматна абстракція автомату з часовими обмеженнями S = (S, I, O, λΣ, s_0) з найбільшою кінцевою межею для часових вхідних інтервалів BS і найбільшою вихідною затримкою D може бути побудована як повністю

визначений кінцевий автомат AS(BS) = (S, IA, OA, λAΣ, s_0), де IA = {(i, 0), (i, (0, 1)), …, (i, (BS – 1, BS)), (i, BS), (i, (BS, ∞)): i ∈ I}, OA = {(o, 0), (o, 1), …, (o, D): o ∈ O}. Для стану s автомату AS(BS) і вхідного символу (i, t_j), t_j = 0, …, BS, множина λAΣ містить перехід (s, (i, t_j), (o, d), s'), якщо й тільки якщо існує перехід (s, i, o, s', g, d) ∈ λΣ, такий що t_j ∈ g. Для стану s автомату AS(BS) і вхідного символу (i, g_j), g_j = (0, 1) …, (BS – 1, BS), (BS, ∞), множина λAΣ містить перехід (s, (i, g_j), (o, d), s'), якщо й тільки якщо існує перехід (s, i, o, s', g, d) ∈ λΣ, такий що $g_j \subseteq$ g. Таким чином, для побудови абстракції до кожного вхідного символу додаються інтервали часу або цілі значення часової змінної. Між станами автомату з часовими обмеженнями і його скінченною автоматною абстракцією може бути встановлена взаємно однозначна відповідність, що спрощує побудову повних тестів, що перевіряють, для такого класу часових автоматів.

Часова вхідна послідовність α для автомату з часовими обмеженнями може бути переведена у відповідну послідовність вхідних символів αΦΣM для звичайно автоматної абстракції. При цьому кожний часовий вхідний символ (i, t) представляється вхідним символом (i, t), якщо t ціле, або символом (i, (a, b)), де t ∈ (a, b). Аналогічно, вхідна послідовність звичайно автоматної абстракції може бути переведена в часову вхідну послідовність. При цьому, для вхідного символу (i, (a, b)) значенням часовий змінної відповідного вхідного часового символу може бути обране будь-яке значення t ∈ (a, b). У той же час вихідна реакція абстракції на вхідний символ має вигляд (o, d), тобто відповідає реакції вихідного часового автомату, виданої через d тактів часу після вступу вхідного символу. Таким чином, вихідна послідовність абстракції γΦΣM може бути переведена у відповідну часову вихідну послідовність γ.

Скінченна автоматна абстракція «успадковує» такі властивості часового автомату як повнота (частковість) і детермінованість (недетермінованість). Таким чином, якщо часовий автомат S є повністю визначеним (частковим), те і його звичайно автоматна абстракція AS є повністю визначеним (частковим) автоматом. Якщо часовий автомат S є детермінованим (недетермінованим), то і

його звичайно автоматна абстракція AS є детермінованим (недетермінованим) автоматом.

2.2 Мінімізація часових автоматів

Основним завданням кваліфікаційної роботи є розробка алгоритмів мінімізації детермінованих повністю визначених автоматів з тайм-аутами і часовими обмеженнями. Завдання мінімізації є одним із класичних завдань теорії автоматів [4]. Мінімальна форма повністю визначеного детермінованого автомату єдина (з точністю до ізоморфізму) і може розглядатися як канонічна форма класичного кінцевого автомата. Завдання мінімізації розширюється на часові детерміновані автомати, і показується, що дві наведені по станах форми часового детермінованого повністю визначеного автомату не обов'язково ізоморфні. Відповідно, потрібно ввести поняття наведеної за часом форми, як часового автомату з максимально можливими часовими вхідними інтервалами й мінімальними тайм-аутами, і дві наведені по станах і часу форми часового детермінованого повністю визначеного неініціального автомату ізоморфні, тобто таку наведену форму можна розглядати як канонічну форму часового автомату.

Пропонований метод мінімізації часового детермінованого автомата заснований на побудові відповідної звичайно автоматної абстракції, і встановлюється ряд властивостей такої абстракції, необхідних для аналізу можливостей мінімізації часового автомата.

Як часові вхідні й вихідні послідовності для часового автомату можуть бути перетворені у відповідні послідовності для скінченної автоматної абстракції. Для часових вхідних і вихідних послідовностей α і γ часового автомату S відповідні послідовності звичайно автоматної абстракції AS позначаються як αФΣМ і γФΣМ.

Часова вхідно-вихідна послідовність α/γ існує в стані *s* часового автомату *S*, якщо й тільки якщо вхідно-вихідна послідовність αΦΣM/γΦΣM існує в стані (s, 0) скінченної автоматної абстракції AS. Розглянуто стан s часового автомату S і множина станів (s, 0), (s, (0, 1)), …, (s, (T – 1, T)) скінченної автоматної абстракції AS, де ΔS(s) = (s', T). Кожний стан AS відповідає цілому моменту часу або інтервалу виду (a, b), у якому перебуває значення часової змінної автомату S, а переходи по вхідному символу I відображають зміну значення часової змінної автомату S із цілого числа в інтервал виду (a, b) та назад. Таким чином, по побудові множини переходів абстракції вхідно-вихідна пара (i, t) / (o, d) існує в стані s автомату S, якщо й тільки якщо існує вхідно-вихідна послідовність I, I, …, I, i / I, I, …, I, (o, d) у стані (s, g) автомата AS, де число символів I дорівнює числу m переходів часової змінної із цілого числа в інтервал виду (a, b) та назад за час t < T і автомат AS переходить зі стану (s, 0) у стан (s, g) при наявності послідовності з m вхідних символів I. У силу того, що переходу по тайм-ауту ΔS(s) = (s', T) автомату S відповідає перехід ((s, (T – 1, T)), I, I, (s', 0)) автомату AS, аналогічні міркування можна провести для стану s' і множини (s', 0), (s', (0, 1)), …, (s', (T – 1, T)), а також й тільки якщо усіх наступних станів, у які автомат S може перейти по тайм-ауту. У той же час, за правилами побудови переходів скінченної автоматної абстракції, вхідний часовий символ (i, t) переведе автомат S у стан s", якщо й тільки якщо, вхідна послідовність I, I, …, I, переводить автомат AS у стан (s", 0). Далі, для пари станів s" і (s", 0) можна провести аналогічні розрахунки.

Таким чином, еквівалентність станів часового автомату можна перевірити по відповідних до станів скінченної автоматної абстракції.

Стани s_1 і s_2 часового автомату S еквівалентні, якщо й тільки якщо у відповідному кінцевому автоматі AS еквівалентними є стани (s_1, 0) і (s_2, 0). Нехай стани s_1 і s_2 часового автомату S еквівалентні. Припустимо, що стани (s_1, 0) і (s_2, 0) відповідного кінцевого автомату AS не є еквівалентними, тобто для них існує послідовність. Тоді, послідовність α буде такою, що розрізняє для

станів s_1 і s_2, що суперечить твердженню, що стани s_1 і s_2 еквівалентні. Нехай тепер стани $(s_1, 0)$ і $(s_2, 0)$ еквівалентні в побудованому кінцевому автоматі AS, однак стани s_1 і s_2 часового автомату S не є еквівалентними. Тоді, послідовність αФΣМ буде такою, що розрізняє для станів $(s_1, 0)$ і $(s_2, 0)$, що суперечить твердженню, що стани $(s_1, 0)$ і $(s_2, 0)$ еквівалентні.

Таким чином, висновок про еквівалентність або нееквівалентність станів часового автомату (різних часових автоматів) може бути зроблений на основі властивостей відповідних станів звичайно автоматної абстракції (звичайно автоматних абстракцій).

Основні твердження слушні також для автоматів тільки з тайм-аутами і автоматів тільки з часовими обмеженнями.

3 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

3.1 Мінімізація часових автоматів по станах

Завдання мінімізації в теорії автоматів полягає в синтезі наведеного кінцевого автомату, еквівалентного вихідному неініціальному автомату, тобто в побудові автомату із заданою роботою, у якім будь-які два різні стани помітні. В [4] представлений алгоритм мінімізації кінцевих повністю визначених детермінованих автоматів, на який і опирається запропонований нами підхід для мінімізації часових автоматів. Відношення еквівалентності на множини станів повністю визначеного детермінованого автомату індукує розбивка, яка далі називається розбивкою по відношенню еквівалентності й позначається E. Будь-які два стани, що належать одному класу (блоку) розбивки E, є еквівалентними; будь-які два стани, що належать різним класам розбивки E, є помітними. Класичний метод заснований на розбивці множини станів по відношенню еквівалентності, побудова якого містить наступні етапи.

На першому кроці будується розбивка по відношенню 1-еквівалентності. Два стани належать одному класу 1-розбивки, якщо реакції автомата в цих станах на будь-який вхідний символ збігаються. Два стани належать різним класам 1-розбивки, якщо реакції автомату в цих станах не збігаються хоча б на один вхідний символ.

Потім виконується побудова розбивки по відношенню (k+1)-еквівалентності на основі розбивки по відношенню k-еквівалентності. Два стани з одного блоку k-розбивки належать одному блоку (k+1)- розбивки, якщо спадкоємці цих станів по будь-якому вхідному символу перебувають в одному класі k-розбивки. Два стани з одного блоку k- розбивки належать різним блокам (k+1)-розбивки, якщо спадкоємці цих станів хоча б по одному вхідному символу перебувають у різних блоках k- розбивки. Цей крок виконується доти, поки поточне й наступне розбивки не збіжаться, останнє означає, що побудована розбивка являє собою розбивку по відношенню еквівалентності.

Далі, на основі отриманої розбивки будується наведена форма автомату. Класи розбивки по відношенню еквівалентності відповідають станам наведеної форми, а переходи зі стану наведеної форми збігаються з такими для будь-якого стану з відповідного класу.

Для часового автомата S наведена по станах форма являє собою часовий автомат еквівалентний S, у якім будь-які два різні стани помітні. Аналогічно класичному алгоритму мінімізації, ми будуємо наведену по станах форму часового автомата на основі розбивки множини станів по відношенню еквівалентності. Відповідно до результатів попереднього розділу, така розбивка може бути отримана з розбивки множини станів звичайно автоматної абстракції по відношенню еквівалентності.

Алгоритм 3.1 побудови наведеної по станах форми часового неініціального повністю визначеного детермінованого автомату.

Вхід: Часовий повністю певний детермінований автомат S.

Вихід: Наведена по станах форма B автомату S.

Крок 1. По часовому автомату S будується відповідна звичайно автоматна абстракція AS, для якої виконується побудова розбивки EFSM множини станів по відношенню еквівалентності.

Крок 2. Будується розбивка E множини станів автомата S по відношенню еквівалентності: стану s_1 і s_2 автомату S належать одному блоку розбивки E, якщо й тільки якщо стану $(s_1, 0)$ і $(s_2, 0)$ абстракції AS належать одному блоку розбивки EFSM.

Крок 3. Будується наведена по станах форма B автомату S. Вхідний і вихідний алфавіти автомату B збігаються з такими для автомату S, стану b_1, b_2, …, b_l автомату B відповідають блокам B_1, B_2, …, B_l розбивки E. множина переходів $\lambda\beta$ автомату B формується в такий спосіб. З кожного блоку B_i вибирається стан s_i. Для станів b_i і b_j існує кортеж $(b_i, i, o, b_j, g, d) \in \lambda B$, якщо й тільки якщо існує стан $s_j \in B_j$, таке що кортеж $(s_i, i, o, s_j, g, d) \in \lambda\Sigma$. Для функції тайм-ауту $\Delta B(b_i) = (b_j, T)$, якщо й тільки якщо $\Delta S(s_i) = (s_j, T)$, де $s_j \in B_j$.

Кінець алгоритму.

Наступне твердження підтверджує коректність побудови наведеної по станах форми часового автомату по описаному вище алгоритму. Нехай B – автомат, побудований для часового автомату S по алгоритму 3.1. Автомат B є детермінованим повністю визначеним автоматом, і стан bi автомату B еквівалентний стану s автомата S, якщо й тільки якщо $s \in Bi$.

Стани s_1 і s_2 часового автомату S належать одному блоку розбивки E, якщо й тільки якщо стану $(s_1, 0)$ і $(s_2, 0)$ звичайно автоматної абстракції AS належать одному блоку розбивки EFSM. Таким чином, розбивка E, отримана на другому кроці алгоритму, є розбивка множини станів автомату S по відношенню еквівалентності. Автомат B є повністю певним і детермінованим автоматом, оскільки автомат S має ці властивості й для кожного стану b_i автомата B множина переходів по вхідних символах і тайм-ауту «повторює» такі для деякого стану $s \in B_i$.

Показано, що стан $s_i \in B_i$ автомату S і стан b_i автомату B, відповідне до блоку Bi розбивки E, еквівалентні. Усі переходи зі стану b_i дублюють переходи з деякого стану $s'_i \in B_i$, еквівалентного будь-якому стану s_i з B_i у силу того, що E – розбивка по відношенню еквівалентності. У той же час, якщо $\Delta S(s'_i) = (s_k, T)$ і $\Delta B(b_i) = (b_k, T)$, $T < \infty$, то $s_k \in B_k$ є такий стан, що еквівалентний стану s_k, переходи якого «дублюються» у стані b_k. Аналогічні міркування можна провести й для можливих спадкоємців по тайм-аутам станів s_k і b_k. Таким чином, реакції на будь-який часовий вхідний символ у станах s_i і b_i збігаються. Оскільки переходу (si, i, o, si', g, d) вихідного автомата відповідає перехід (bi, i, o, bi', g, d) наведеної форми, де $si' \in Bi'$, те для станів si' і bi', і для всіх їхніх спадкоємців по вхідних часових символах, можна провести аналогічні міркування. Таким чином, реакції автоматів B і S у станах bi і si на будь-яку часову вхідну послідовність збігаються. Покажемо тепер, що стану $sj \in Bj$ і bi, де $i \neq j$, не еквівалентні. Стан $si \in Bi$ еквівалентно стану bi згідно з першою частиною доказу. Стану si і sj не еквівалентні, тому що перебувають у різних блоках розбивки E. Таким чином, стану sj і bi не еквівалентні, і слушна

наступна теорема.

Нехай S – повністю визначений детермінований автомат з тайм-аутами й часовими обмеженнями. Повністю визначений детермінований автомат B, побудований по алгоритму 3.1, є наведений по станах формою автомату S.

Як приклад, на рисунку 3.1 представлені часовий автомат S (вище) і його наведена по станах форма (нижче).

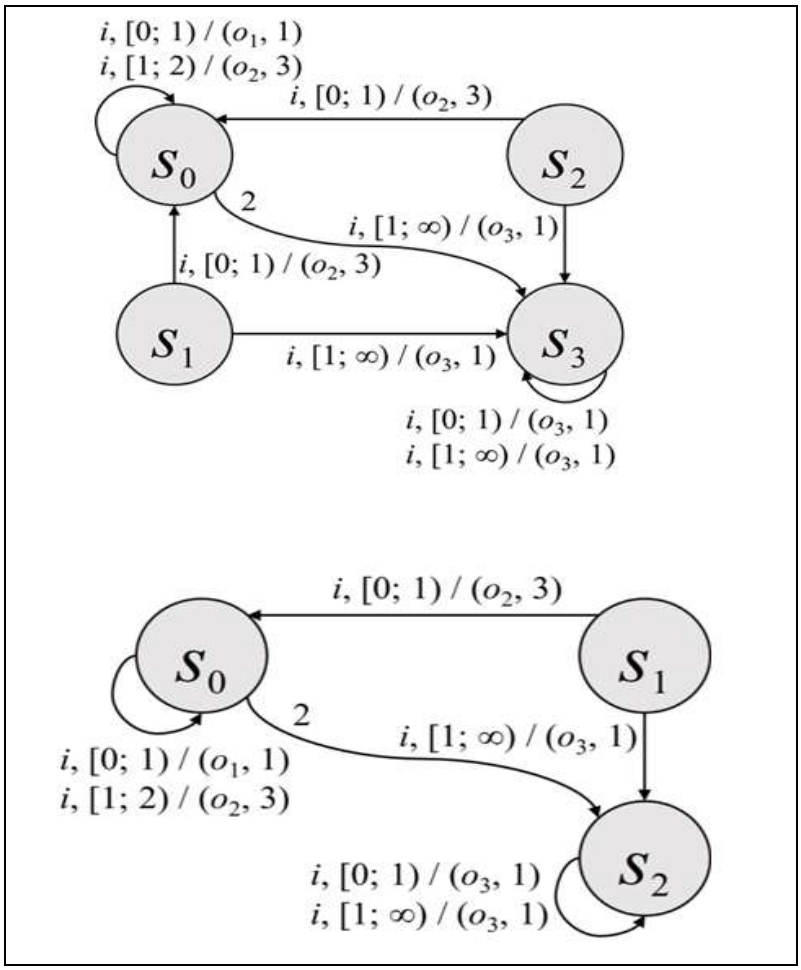

Рисунок 3.1 – Часовий автомат S і його наведена форма

Відомо, що наведена по станах форма кінцевого повністю визначеного детермінованого автомату єдина з точністю до ізоморфізму [4], однак, для часового автомату можуть існувати неізоморфні наведені по станах форми. Представлені часові автомати на рисунку 3.2, де безпосередньою перевіркою можна переконатися, що обоє ці автомата є наведеними по станах формами автомата S на рис. 3.1, але не є ізоморфними.

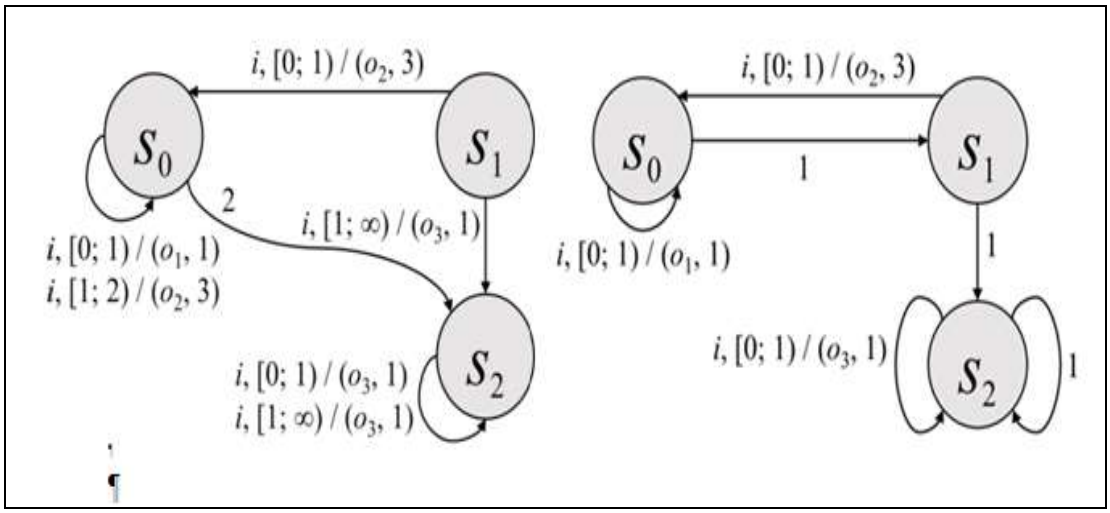

Рисунок 3.2 – Наведені по станах форми автомату S

Причиною описаної ситуації є той факт, що для ізоморфізму двох часових автоматів повинне мати місце необхідна відповідність між часовими аспектами. У даному розділі ми показуємо, як можна привести тайм-аути й часові обмеження автоматів з однієї часовою змінною для побудови єдиної мінімальної форми часового автомату. Автомат з тайм-аутами і часовими обмеженнями S називається наведеним за часом, якщо для будь-яких двох кортежів (s, i, o, s', g_1, d), (s, i, o, s', g_2, d) $\in \lambda\sigma$ справедливо, що g_1 і g_2 не можна об'єднати в один інтервал і для будь-якого стану s, такого що $\Delta S(s) = (s', T)$, не існує стану s" і цілого невід'ємного числа $T' < T$, таких що автомат S', отриманий зміною функції тайм-ауту стану s автомата S на $\Delta S(s) = (s'', T')$, еквівалентний вихідному автомату S.

Якщо для переходів (s, i, o, s', g_1, d), (s, i, o, s', g_2, d) $\in \lambda\sigma$ часового автомата S інтервали g_1 і g_2 можуть бути об'єднані в один інтервал, то ці два переходи можна замінити одним переходом виду (s, i, o, s', $g_1 \cup \gamma_2$, d). Пошук тайм-ауту меншої величини може бути виконаний по звичайно автоматній абстракції AS. Для пошуку мінімального тайм-ауту в стані s автомата S необхідно розглянути стан абстракції виду (s, n), відповідні до роботи часового автомату в стані s у цілі моменти часу. Якщо для деякого стану (s, j) існує еквівалентний стан виду (s', 0), то перехід по спеціальному вхідному символу ((s, (j – 1, j)), I, I, (s, j)) може бути замінений на перехід ((s, (j – 1, j)), I, I, (s', 0)). Відповідно, тайм-аут у стані s автомату S може бути замінений на $\Delta S(s) = (s', j)$. Таким чином, для

кожного стану може бути обраний мінімально можливий тайм-аут, який не змінює роботу часового автомата.

Для побудови наведеної за часом форми припустимі переходи по вхідних символах з часовими обмеженнями поєднуються, а кожний тайм-аут приводиться до мінімального значення. Ми пропонуємо наступний алгоритм побудови наведеної за часом форми часового автомату.

Алгоритм 3.2 побудови наведеної за часом форми автомату з тайм-аутами і часовими обмеженнями S.

Вхід: Повністю визначений детермінований часовий автомат S.

Вихід: Наведена за часом форма автомата S.

Крок 1. Кожна пара переходів (s, i, o, s', g_1, d), (s, i, o, s', g_2, d) $\in \lambda\Sigma$, така, що g_1 і g_2 може бути об'єднана в один інтервал, заміняється на перехід (s, i, o, s', $g_1 \cup \gamma_2$, d).

Крок 2. По вихідному автомату з тайм-аутами й часовими обмеженнями S будується відповідна звичайно автоматна абстракція AS, для якої виконується побудова розбивки по відношенню еквівалентності.

Крок 3. Для кожного стану s автомата S, такого що $\Delta S(s) = (s', T)$, серед станів (s, j) автомата AS, де $j = 1, \ldots, T - 1$, визначається стан з мінімальним j, для якого існує еквівалентний стан виду (s", 0). Якщо такий стан s" існує, то тайм-аут у стані s заміняється на $\Delta S(s) = (s", j)$, інакше тайм-аут у стані s не змінюється.

Крок 4. Для кожного стану s автомата S, такого що $\Delta S(s) = (s', \infty)$, визначається стан виду (s", 0) в автоматі AS, еквівалентне стану $(s, (j, \infty))$. Якщо такий стан s" існує, то тайм-аут у стані s заміняється на $\Delta S(s) = (s", j + 1)$, інакше тайм-аут у стані s не змінюється.

Крок 5. Для кожного стану s, де $\Delta S(s) = (s', T)$ і $T < \infty$, віддаляються всі переходи (s, i, o, s', g, d), такі що $g \cap [0, T) = \varnothing$. Якщо для переходу (s, i, o, s', $\langle \alpha, b \rangle$, d) справедливо, що $\langle \alpha, b \rangle \cap [0, T) \neq \varnothing$ і невірно, що $\langle \alpha, b \rangle \subseteq [0, T)$, те такий перехід (s, i, o, s', $\langle \alpha, b \rangle$, d) заміняється на перехід (s, i, o, s', $\langle \alpha, T)$, d).

Кінець алгоритму.

Доказ коректності алгоритму. Нехай P – часовий автомат, побудований для повністю визначеного детермінованого автомата з тайм-аутами й часовими обмеженнями S по алгоритму 3.2. Автомат P є наведеним за часом і еквівалентний автомату S. Автомати S і P еквівалентні. Об'єднання переходів (s, i, o, s', g_1, d), (s, i, o, s', g_2, d) у перехід (s, i, o, s', $g_1 \cup \gamma_2$, d) (Крок 1) не змінюють роботу автомату.

Процедура заміни тайм-ауту на третьому кроці алгоритму. Нехай для стану p часового автомата P знайдений стан p'' автомата P, такий, що існує стан (p, j) автомата AP, еквівалентний стану (p'', 0) автомата AP. Тут ΔP(p) = (p', T) і j < T. Тобто, реакція автомата P у стані p'' на будь-яку часову вхідну послідовність збігається такою для автомату в стані *p* у момент часу *j*. Таким чином, при заміні функції тайм-ауту ΔP(p) = (p', T) у стані p на ΔP(p) = (p'', j) робота автомату P у стані *p* не міняється. Аналогічно, заміна тайм-ауту на четвертому кроці алгоритму не змінює роботу автомату. На кроці 5 алгоритму віддаляються переходи (p, i, o, p', g, d), часові інтервали яких не перетинаються з інтервалом [0, T), де ΔP(p) = (p', T), відповідно, такі переходи не можуть бути виконані автоматом. Переходи, часові інтервали яких перетинаються з [0, T), але виходять за межу тайм-ауту, обмежуються зверху величиною тайм-ауту, що також не змінює роботу автомату.

Автомат P наведений за часом. Допустимо, що для деякого стану p автомата P таймаут ΔP(p) = (p_1, T_1) може бути замінений на тайм-аут ΔP(p) = (p_2, T_2), де $T_1 > T_2$, таким чином, що робота автомату в стані p не зміниться. Останнє означає, що у звичайно автоматній абстракції AP існує або пари еквівалентних станів (p, T_2) і (p_2, 0), або (p, ($T_2 - 1$, ∞)) і (p_2, 0). Існування еквівалентних станів (p, T_2) і (p_2, 0) неможливо по третьому кроку алгоритму. Існування еквівалентних станів (p, ($T_2 - 1$, ∞)) і (p_2, 0) відображає той факт, що реакція на будь-яку вхідну часову послідовність автомату P у стані p у момент часу $T > (T_2 - 1)$ збігається з такою для стану p_2 у момент часу 0. У такому випадку, стану (p, T_2) не існує по побудові абстракції, однак, по кроку 4

алгоритму, тайм-аут був би замінений на $\Delta P(p) = (p_2, T_2)$. Таким чином, існування еквівалентних станів $(p, (T_2 - 1, \infty))$ і $(p_2, 0)$ неможливо.

У той же час, будь-які два переходи виду (s, i, o, s', g_1, d), (s, i, o, s', g_2, d), які можливо об'єднати в один перехід $(s, i, o, s', g_1 \cup \gamma_2, d)$, поєднуються на кроці 1 алгоритму 3.2. На основі отриманих результатів, можна встановити необхідну й достатню умову еквівалентності часових автоматів.

Детерміновані повністю визначені наведені по станах і часу неініціальні часові автомати S і P еквівалентні, якщо і тільки якщо ці автомати ізоморфні. Нехай S і P – детерміновані повністю визначені наведені по станах і часу часові автомати, які є еквівалентними. Автомати S і P ізоморфні – розглядається відображення H: $S \rightarrow P$, при якім H(s) є стан автомата P, еквівалентний стану s. Відображення H є взаємно однозначним, оскільки автомати S і P є наведеними по станах. Для будь-якої пари станів s і p = H(s), такий що $\Delta S(s) = (s', Ts)$, $\Delta P(p) = (p', Tp)$, справедливо, що Ts = Tp і p' = H(s'). Допустимо, що Tp < Ts; тоді в силу еквівалентності S і P, існує стан s'', еквівалентний стану p'. А оскільки стани s і p також є еквівалентними, то й тайм-аут у стані s може бути замінений на $\Delta S(s) = (s'', Ts')$, де Ts' = Tp < Ts. Останнє неможливо, оскільки S є наведеним за часом автоматом. Нехай тепер Tp > Ts; тоді в силу еквівалентності S і P, існує стан p'', еквівалентний стану s'. А оскільки стани *s* і *p* також є еквівалентними, то й тайм-аут у стані p може бути замінений на $\Delta P(p) = (p'', Tp')$, де Ts = Tp' < Tp. Останнє неможливо, оскільки P – наведений за часом автомат. У силу рівності тайм-аутів Tp = Ts і еквівалентності станів s і p, справедливо, що стани s' = time(s, Ts) і p' = time(p, Tp) еквівалентні, і, відповідно, p' = H(s'). Покажемо далі, що для будь-якого переходу $(s_1, i, o, s_2, g, d) \in \lambda\sigma$ існує перехід $(p_1, i, o, p_2, g, d) \in \lambda\Pi$, де $p_1 = H(s_1)$ і $p_2 = H(s_2)$. Припущення зворотне, нехай в автоматі S існує перехід (s_1, i, o, s_2, g, d), але переходу (p_1, i, o, p_2, g, d) не існує в автоматі P. Оскільки стани *s* і *p* еквіваленти, а також мають тайм-аути однакової величини, робота автомату P у стані p в інтервалі часу g збігається з таким для автомату S у стані *s*. Відповідно, в автоматі P існує перехід (p_1, i, o, p_2, g', d), де $g \subset g'$, через то й же

що автомат P є наведеним за часом і станам. Однак, останнє неможливо, оскільки робота автомату S у стані s, а відповідно й автомату P у стані p в інтервалі часу g відрізняється від роботи в сусідніх інтервалах по тому ж вхідному символу через те, що автомати S і P є наведеними за часом.

Оскільки ізоморфні автомати збігаються з точністю до перейменування станів, то ізоморфні автомати еквівалентні. Відповідно, для часових автоматів уводиться визначення мінімальної форми, що враховує не тільки мінімізацію числа станів, але й відомість часових аспектів до наведеної форми. Мінімальною формою повністю визначеного детермінованого неініціального автомату з тайм-аутами й часовими обмеженнями S називається наведений по станах і часу автомат, еквівалентний S.

Мінімальна форма неініціального автомату з тайм-аутами й часовими обмеженнями єдина з точністю до ізоморфізму. Для наведеної по станах форми автомату S (рис. 3.2) наведена за часом форма побудована на рисунку 3.2 (праворуч). Як видно з рисунку 3.2, тайм-аут $\Delta S(s_0) = (s_2, 2)$ був замінений на тайм-аут $\Delta S(s_0) = (s_1, 1)$, а перехід $(s_0, i, o_2, s_0, [1, 2), 3)$ вилучений. У той же час, два переходи зі стану s_2 минулого об'єднані, а тайм-аути $\Delta S(s_1) = (s_1, \infty)$ і $\Delta S(s_2) = (s_2, \infty)$ замінені на $\Delta S(s_1) = (s_2, 1)$ і $\Delta S(s_2) = (s_2, 1)$ відповідно.

3.2 Алгоритми автоматів з тайм-аутами і часовими обмеженнями

Алгоритм 3.1 дозволяє побудувати наведену по станах форму для автоматів з тайм-аутами і автоматів з часовими обмеженнями, як частину випадків часового автомату. Визначення наведеної за часом форми часових автоматів, містить дві умови, які забезпечують одиничність мінімальної форми. Визначення наведеної за часом єдиної форми для автоматів з часовими обмеженнями й автоматів з тайм-аутами є окремими випадками відповідного визначення для розглянутої вище моделі часового автомата.

Перша умова пов'язана з неможливістю об'єднання переходів по вхідних символах і повністю збігається з визначенням наведеної за часом єдиної форми для автоматів з часовими обмеженнями, наведеної в [14]. Наведена по станах форма автомату з часовими обмеженнями не єдина, оскільки часові інтервали і відповідно переходи можуть бути об'єднані різними способами. Наприклад, це продемонстроване в еквівалентних наведених по станах автоматах на рис. 3.2. Таким чином, автомат з часовими обмеженнями S називається наведеним за часом, якщо для будь-яких двох кортежів (s, i, o, s', g_1, d), (s, i, o, s', g_2, d) $\in \lambda\sigma$ інтервали g_1 і g_2 не можуть бути об'єднані в один інтервал. Детерміновані повністю визначені наведені за часом і станам автомати з часовими обмеженнями S і P еквівалентні, якщо й тільки якщо ці автомати ізоморфні.

Розглянемо далі два еквівалентні наведені по станах автомати з тайм-аутами (рис. 3.3).

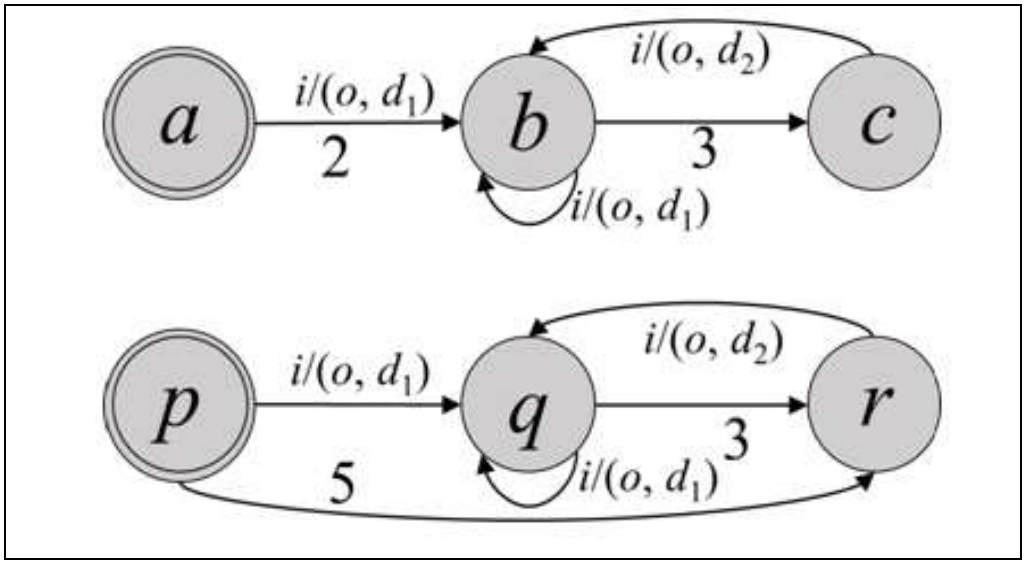

Рисунок 3.3 – Два еквівалентних наведених, але неізоморфних автоматів з тайм-аутами

Цей приклад, ілюструє, що наведені по станах форми автомату з тайм-аутами не завжди є ізоморфними, оскільки деякі ланцюжки переходів по тайм-аутам можуть бути об'єднані в один перехід, або, навпаки, один тайм-аут може бути розділений на частини. Зокрема, у стані *p* відсутній перехід по тайм-ауту 2, який є в стані a іншого автомату, еквівалентному p.

Потрібно ввести клас мінімальних форм для автоматів з тайм-аутами, які

єдині з точністю до ізоморфізму. Друга умова одиничності для часових автоматів накладається на тайм-аути й вимагає, щоб тайм-аут у кожному стані був мінімальний. Ця умова відрізняється від визначення наведених за часом автоматів з тайм-аутами, уведеного в [17], що й припускав пошук максимального тайм-ауту для кожного стану, однак також може бути використаний для одержання єдиної форми даної моделі. Таким чином, автомат з тайм-аутами S називається наведеним за часом, якщо для будь-якого стану s, такого що ΔS(s) = (s', T), не існує стану s" і цілого невід'ємного числа T' < T, таких що автомат S', отриманий зміною функції тайм-ауту стану s автомата S на ΔS(s) = (s", T'), еквівалентний вихідному автомату S твердження, що й справедливо випливає.

Тобто, детерміновані повністю певні наведені за часом і станам автомати з тайм-аутами S і P еквівалентні, якщо й тільки якщо ці автомати ізоморфні.

Аналогічно автоматам з тайм-аутами й часовими обмеженнями, введено визначення мінімальної форми для інших моделей з часовими аспектами. Мінімальна форма повністю визначеного детермінованого автомату тільки з тайм-аутами (тільки з часовими обмеженнями) S є наведений по станах і часу автомат, еквівалентний S і, як результат – мінімальна форма автомата тільки з тайм-аутами (тільки з часовими обмеженнями) єдина з точністю до ізоморфізму.

Для класичних кінцевих автоматів складність алгоритму мінімізації визначається як кількість ітерацій для побудови розбивки множини станів по відношенню еквівалентності й у найгіршому разі рівна n – 1 [4], де n – число станів мінімізованого автомату. Оскільки, пропонований метод мінімізації часових автоматів заснований на побудові такої розбивки для відповідних звичайно автоматних абстракцій, то складність алгоритму мінімізації часових автоматів може бути оцінена в такий спосіб. По визначенню, число станів звичайної автоматної абстракції не перевищує величини 2Bsm, де m – число станів часового автомату, а BS – найбільша кінцева межа для часових інтервалів. Відповідно, для побудови розбивки по відношенню еквівалентності множини станів такої абстракції буде потрібно не більш ніж 2Bsm–1 кроків. Таким чином, складність алгоритму мінімізації залишається поліноміальної щодо числа станів

вихідного часового автомату.

3.3 Алгоритми композиції часових автоматів

Має бути введений алгоритм виконання операції паралельної композиції для ініціальних детермінованих повністю визначених автоматів з часовими обмеженнями, результатом якого є автомат, що описує роботу системи автоматів, що взаємодіють у режимі діалогу. Така паралельна композиція має використовуватися, наприклад, при описі спільної роботи різних клієнтських і серверних додатків. операція, що вводиться, є розширенням паралельної композиції для класичних автоматів і зводиться до неї, якщо автомати не мають часових обмежень.

Для класичних автоматів операція паралельної композиції є замкненою в множини детермінованих автоматів за умови «повільного» зовнішнього середовища, тобто за умови, що наступний вхідний символ надходить на композицію після того, як отриманий вихідний сигнал на попередній вхідний символ.

На відміну від класичних ініціальних автоматів, «повільне» зовнішнє середовище не є достатньою умовою для замкнутості множини часових детермінованих автоматів щодо операції паралельної композиції. Відповідно, потрібно визначати окремі випадки, для яких паралельна композиція часових ініціальних детермінованих автоматів є детермінованим автоматом.

Модель взаємодіючих у режимі діалогу класичних кінцевих автоматів досить добре вивчена й коротко описана на початку даного розділу. Множина детермінованих кінцевих автоматів є замкнена щодо цієї операції при наявності «повільного» зовнішнього середовища, і існують різні алгоритми побудови такої бінарної композиції [30].

Система взаємодіючих автоматів (компонент) [9] являє собою кінцеві

множини автоматів, які можуть обмінюватися повідомленнями (вхідними й вихідними впливами) як із зовнішнім середовищем, так і між собою. Така структура дозволяє описувати роботу складних систем як композицію більш простих у деякому сенсі автоматів-компонентів.

Паралельна композиція – система, у якій автомати (компоненти) обробляють вхідні впливи в режимі діалогу. В кожний інтервал часу в такій системі обробляється не більш ніж одне повідомлення. Для класичних автоматів операція паралельної композиції в ряді випадків є асоціативною. Схематично бінарна композиція (часових) автоматів зображено на рисунку 3.4, де I_1 і I_2 – зовнішні вхідні алфавіти, O_1 і O_2 – зовнішні вихідні алфавіти, U і V – внутрішні алфавіти, тобто вхідний і вихідний алфавіти компонента C_2, а також, відповідно, вихідний і вхідний алфавіти компонента C_1.

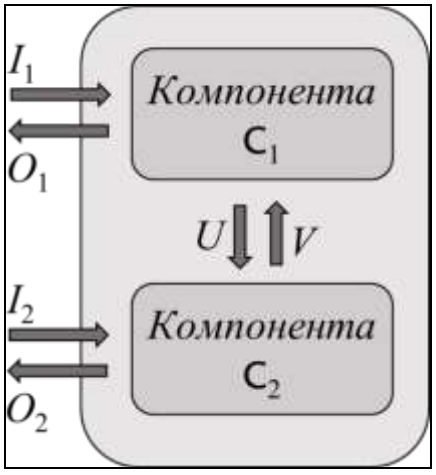

Рисунок 3.4 – Паралельна композиція двох (часових) автоматів

При надходженні на композицію зовнішнього вхідного впливу компоненти по черзі обробляють вхідні символи доти, поки не буде виданий зовнішній вихідний символ, якщо діалог між компонентами закінчується за кінцеве число кроків. Якщо діалог між компонентами при деякому вхідному символі є нескінченним, то говорять, що в композиції існує осциляція, і дія на даному вхідному символі зазвичай залишають невизначеною.

Результатом операції композиції є автомат, що описує роботу системи взаємодіючих компонентів, тобто автомат, дія якого збігається з дією композиції на будь-якій послідовності визначених зовнішніх вхідних символів. Підходи до

побудови паралельної композиції для класичних автоматів досить добре вивчені й звичайно засновані на побудові дерева спадкоємців [30] або відомості операції композиції до операцій над відповідними напівавтоматами [9]. Композиція детермінованих кінцевих автоматів може бути побудована на основі дерева спадкоємців, вершини якого позначені парами станів виду (s, p), де s і p – стани першого й другого компонента відповідно, і є станами автомату і композиції. Вхідний і вихідний алфавіти композиції являють собою об'єднання множини зовнішніх вхідних і вихідних алфавітів компоненту відповідно. Перехід ((s, p), i, o, (s', p')) існує у вершині дерева (стані композиції), якщо композиція, компоненти якої перебувають у станах *s* і *p*, видає зовнішній вихідний символ o у відповідь на зовнішній вхідний символ і таким чином, що при обробці вхідного впливу компоненти переходять у стани *s*' і *p*'. До видачі зовнішнього вихідного сигналу компоненти композиції можуть «проходити» по проміжних станах, у яких компоненти приймають і видають внутрішні сигнали.

Якщо зовнішнє середовище є «повільним», то композиція детермінованих кінцевих автоматів є детермінованим автоматом, і, відповідно, множина класичних детермінованих автоматів замкнута щодо операції паралельної композиції. «Повільнє» зовнішнє середовище припускає, що наступний вхідний символ подається на композицію тільки після одержання реакції композиції на попередній зовнішній вхідний символ. При наявності осциляції або часткових автоматів-компонентів, композиція є частковим автоматом.

Паралельна композиція часових автоматів описує дію системи взаємодіючих у режимі діалогу часових автоматів. Аналогічно композиції класичних автоматів, кожен з компонентів може мати зовнішні й внутрішні вхідні й вихідні символи. У той же час, у випадку часових автоматів, передбачається, що компоненти обробляють вхідні впливи не миттєво, а з деякою затримкою, і відповідно компонент, що бездіє, «накопичує» значення часової змінної й може виконати перехід по тайм-ауту або перетнути межу між часовими інтервалами, не чекаючи впливу від іншого компонента.

Нехай компоненти C_1 і C_2 на рисунку 3.4 є автоматами з тайм-аутами та/або

часовими обмеженнями. Описано принцип роботи такої композиції на прикладі обробки часового вхідного символу (i, t), що подіяв на компонент C_1, де час t відлічується від моменту, коли значення часової змінної було рівне 0 у кожному з компонентів. У цьому випадку часові змінні компонентів мають значення t у момент вступу вхідного символу i, а компонента C_1 може відреагувати внутрішнім або зовнішнім вихідним символом. Якщо був зроблений зовнішній вихідний символ, то композиція готова прийняти наступний зовнішній вхідний вплив. У той же час, до значення часового змінного компонента C_2 додається час, протягом якого компонента C_1 взаємодіяла із зовнішнім середовищем. Якщо компонента C_1 видає внутрішню вихідну реакцію, то компонента C_2 обробляє цей вплив відповідно до запропонованій їй дії і робить внутрішній або зовнішній вихідний вплив. Величина затримки в зовнішніх вихідних часових символах являє собою суму затримок усіх переходів, виконаних між подачею зовнішнього вхідного символу й видачою зовнішнього вихідного символу.

Як ілюстрація, наведено композицію автоматів з часовими обмеженнями на рисунку 3.5 і часові вхідні послідовності (i_1, 0.2), (i_1, 1.1) і (i_1, 0.2), (i_1, 1.9). Нехай часові змінні обох компонентів дорівнюють нулю. Якщо подати на композицію часовий вхідний символ (i_1, 0.2), то компонент S зробить зовнішній вихідний символ o_{12} через один такт після подачі вхідного символу, часова змінна автомату S буде скинута в 0, а часова змінна компоненту P досягне значення 1.2, яке являє собою суму часу подачі вхідного символу й затримки на обробку вхідного символу автоматом S. Після цього, при надходженні часового вхідного символу (i_1, 1.1), компонент S надішле другому компоненту вхідний символ u через один такт, на автомат P, коли часова змінна останнього досягне значення 3.3. Далі компонент P повертає символ v із затримкою в один такт компоненту S, який, у свою чергу, видає зовнішній вихідний символ o_{11}. Таким чином, у відповідь на зовнішній часовий вхідний символ (i_1, 1.1) композиція повернула часовий вихідний символ (o_{11}, 3), де 3 – сума затримок на всіх виконаних переходах. Якщо ж другий вхідний символ i_1 буде поданий не в момент часу 1.1, а в момент 1.9, то композиція надасть вихідний символ o_{12} через

шість тактів.

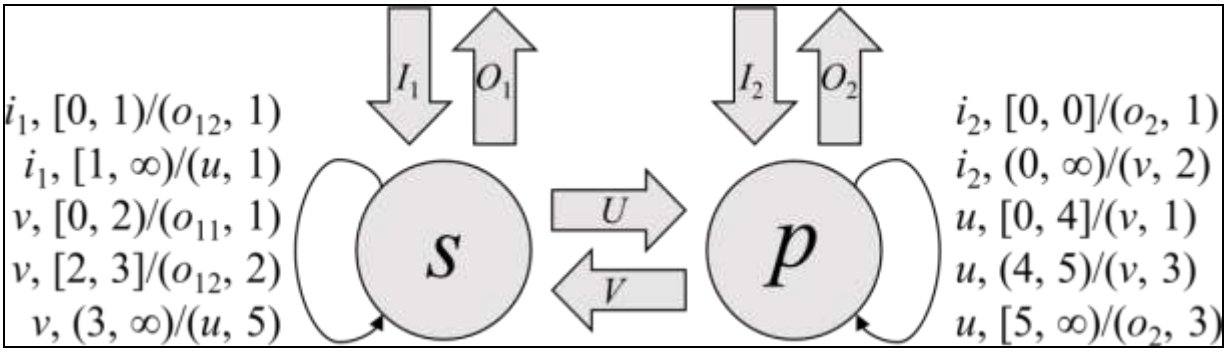

Рисунок 3.5 – Автомати з часовими обмеженнями S і P

Таким чином, після обробки першого зовнішнього вхідного символу часова змінна другого компоненту має раціональне значення й може досягти часового інтервалу, такого що змінює дію цілої межі, менш ніж за один такт. Відповідно, для обчислення реакції композиції на другий зовнішній часовий вхідний символ необхідно знати точне значення двох часових змінних, які можуть приймати раціональні значення. Результатом операції композиції над двома часовими автоматами є часовий автомат, тобто автомат з однієї часовою змінною, що описує роботу системи взаємодіючих компонентів. Таким чином, завдання композиції часових автоматів може трактуватися як оптимізація числа часових змінних у системі.

Кількість часових змінних у моделі в загальному випадку не може бути зменшена, тобто система взаємодіючих детермінованих часових автоматів не завжди може бути описана автоматом того ж класу.

Оскільки дія автомату з часовими обмеженнями може бути досить адекватно описана звичайною автоматною абстракцією, пропонується будувати композицію двох часових автоматів на основі звичайно автоматних абстракцій компоненту. Дія композиції записується в глобальний напівавтомат, у положеннях якого зберігаються стани й значення часових змінних компонентів, після чого отриманий напівавтомат перетвориться у звичайно автоматну абстракцію й, далі, у часовий автомат [27]. Під кінцевим напівавтоматом [25] розуміється трійка $S = (\Sigma, A, \eta\Sigma)$, де S – кінцева непуста множина станів, A – кінцева непуста множина дій, $\eta\Sigma \subseteq (S \times A \times S)$ – відношення переходів, де

кортеж (s, a, s') визначає перехід напівавтомату зі стану *s* у стан *s'* по дії *a*. Для композиції автоматів з часовими обмеженнями S і P (рис. 3.6), звичайні автоматні абстракції AS(3) і AP(3) зображено на рис. 3.7.

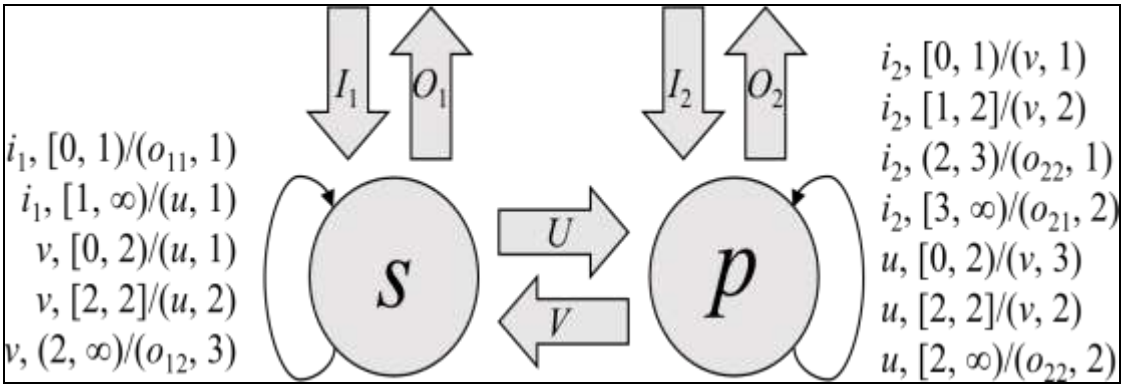

Рисунок 3.6 – Автомати з часовими обмеженнями Si P

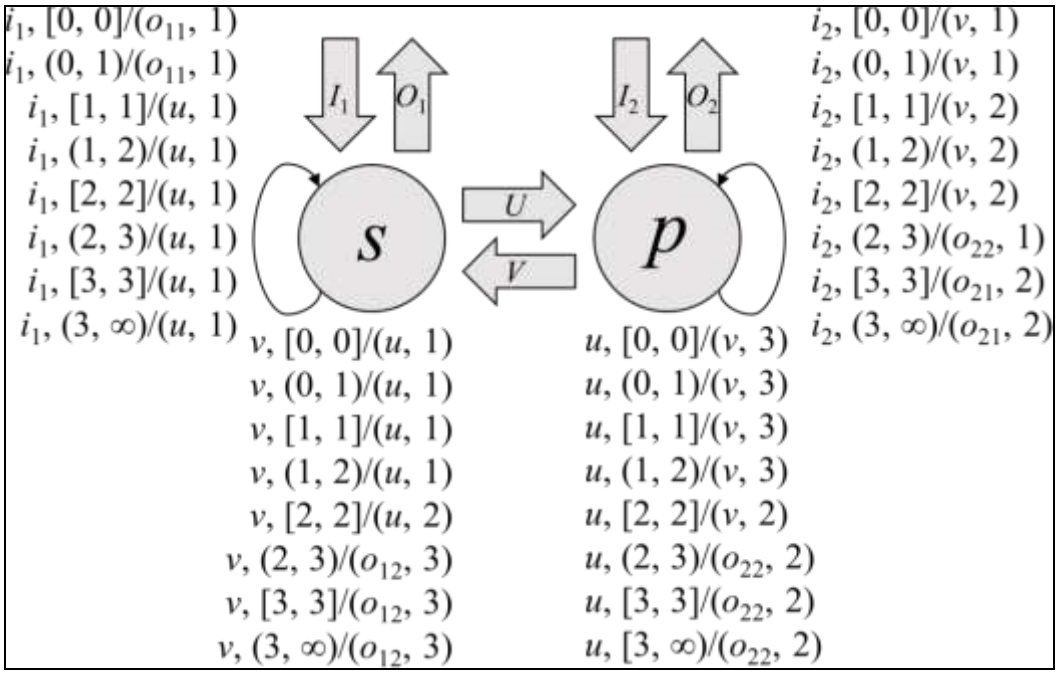

Рисунок 3.7 – Звичайно автоматні абстракції AS(3) і AP(3)

Нехай BS – найбільша кінцева межа часових інтервалів часового автомату S. Введено наступні позначення:

$$G_B = \{[a, a], (a, a + 1), [BS, BS], (BS, \infty): a \in N \text{ і } a < BS\},$$

де N – множина невід'ємних цілих чисел.

Множина N являє собою множину мінімальних непересічних інтервалів, у яких може перебувати часова змінна в інтервалі [0, BS], в об'єднанні з (BS, ∞).

Алгоритм 3.4 побудови паралельної композиції ініціальних повністю визначених детермінованих автоматів з часовими обмеженнями Si P.

Вхід: Ініціальні детерміновані повністю визначені автомати з часовими обмеженнями S = (S, IS = Is ∪ V, OS = Os ∪ U, λΣ, s_0) і P = (P, IP = Ip ∪ U, OP = Op ∪ V, λΠ, p_0), де Is і Os (Ip і Op) – зовнішні вхідний і вихідний алфавіти компоненту S (P), V і U (U і V) – внутрішні вхідний і вихідний алфавіти компоненту S (P); алфавіти Is, Ip, V, U попарно не перетинаються, Is ∪ Ip ≠ ∅, V ∪ U ≠ ∅.

Вихід: Паралельна композиція часових автоматів S і P у вигляді ініціального повністю визначеного детермінованого автомату з часовими обмеженнями або одне з повідомлень: «У системі є осциляція», «Дія системи не може бути описана детермінованим автоматом».

Крок 1. Будуються звичайно автоматні абстракції AS(B) і AP(B), де B – найбільша із меж часових інтервалів BS і BP.

Крок 2. Формується глобальний напівавтомат Q = (Q, {(i, g)/(r, d): i ∈ IS ∪ IP, r ∈ OS ∪ OP, g ∈ G_B, d ∈ Z}, q_0, λΘ), де Z – множина цілих невід'ємних чисел, і множина станів Q містить стабільні й проміжні стани. Під стабільним станом розуміється четвірка (s, gs, p, gp) ∈ S × G_B × P × G_B; під проміжним станом розуміється п'ятірка (s, gs, p, gp, (a, g)), де (s, gs, p, gp) ∈ S × G_B × P × G_B, (a, g) ∈ (U ∪ V) × G_B. Стан є стабільним, якщо й тільки якщо кожний вхідний перехід позначений дією (r, g)/(o, d), де o ∈ Os ∪ Op, тобто (o, d) є зовнішнім часовим вихідним символом часового автомата S або P і відповідним вихідним символом звичайно автоматної абстракції AS(B) або AP(B). У кожному стабільному стані визначений перехід по дії (i, g)/(r, d), де i ∈ Is ∪ Ip, тобто (i, g) є зовнішнім вхідним символом звичайно автоматної абстракції AS(B) або AP(B). Початковий стан q_0 = (s_0, [0, 0], p_0, [0, 0]) є стабільним. У кожному проміжному (нестабільному) стані (s, gs, p, gp, (a, g)), визначений єдиний перехід по дії (a, g)/(r, d), r ∈ OS ∪ OP.

Для кожного стабільного стану (s, [0, 0], p, gp) глобального напівавтомата Q і переходу (s, (i, g), (r, d), s') звичайної автоматної абстракції AS(B), такого що i ∈ Is, g ∈ G_B.

Для кожного стабільного стану (s, gs, p, [0, 0]) глобального напівавтомата Q і переходу (p, (i, g), (r, d), p') звичайно автоматної абстракції AP(B), такого що $i \in Ip$, $g \in G_B$

Для кожного проміжного стану (s, [0, 0], p, gp, (r, [d, d])) глобального напівавтомата Q із вхідним переходом, позначеним (i, g)/(r, [d, d]), визначається перехід (p, (r, gp), (m, k), p') звичайно автоматної абстракції AP(B); Якщо $m \in V$, те перехід ((s, [0, 0], p, gp, (r, [d, d])), (r, [d, d])/(m, k), (s, [k, k], p', [0, 0], (m, [k, k]))) додається в Q;

Інакше перехід ((s, [0, 0], p, gp, (r, [d, d])), (r, [d, d])/(m, k), (s, [k, k], p', [0, 0])) додається в Q;

Для кожного проміжного стану (s, gs, p, [0, 0], (r, [d, d])) глобального напівавтомата Q із вхідним переходом по дії (i, g)/(r, [d, d]) визначається перехід (s, (r, gs), (m, k), s') звичайно автоматної абстракції AS(B); Якщо $m \in U$, те перехід ((s, gs, p, [0, 0], (r, [d, d])), (r, [d, d])/(m, k), (s', [0, 0], p, [k, k], (m, [k, k]))) додається в Q;

Інакше перехід ((s, gs, p, [0, 0], (r, [d, d])), (r, [d, d])/(m, k), (s', [0, 0], p, [k, k])) додається в Q;

Крок 3. Якщо в глобальному напівавтоматі Q існує цикл, позначений тільки внутрішніми вхідними символами, то вертається повідомлення «У системі є осциляція», інакше будується звичайно автоматна абстракція AT:

– стану AT відповідають стабільним станом глобального напівавтомата Q;

– для кожної послідовності $\gamma = (i_1, g_1)/(o_1, d_1), \ldots, (i_n, g_n)/(o_n, d_n)$, яка переводить напівавтомат Q зі стабільного стану q_1 у стабільний стан q_2 і містить єдиний зовнішній вихідний символ (o_j, d_j), в AT додається перехід $(q_1, (i_1, g_1), (o_n, d), q_2)$, де $d = d_1 + \ldots + d_n$ є сумою вихідних затримок, якими позначені дії послідовності γ;

Якщо автомат AT недетермінований, то вертається повідомлення «Дія системи не може бути описана детермінованим автоматом», інакше звичайно автоматна абстракція AT переводиться в часовий автомат T.

Кінець алгоритму.

Таким чином, кожна послідовність переходів глобального напівавтомата між двома стабільними станами відповідає послідовності переходів, виконаних парою взаємодіючих часових автоматів під час діалогу між подачею зовнішнього вхідного символу й видачою зовнішньої реакції. Зауважимо також, що алгоритм 3.1 закінчує роботу за кінцеве число кроків, оскільки множини $S \times G_B \times P \times G_B$ і $(U \cup V) \times G_B$ є кінцевими. Крім того, запропонований алгоритм є розширенням відповідного алгоритму для класичних кінцевих автоматів, тобто справедливо наступне твердження.

Результатом роботи алгоритму 3.1 для повністю визначених детермінованих кінцевих автоматів S і P, є паралельна композиція цих автоматів при наявності «повільного» зовнішнього середовища й відсутності осциляцій. Дія кінцевого автомату може бути описана відповідним часовим автоматом, для якого П = {[0, ∞)} і всі вихідні затримки дорівнюють нулю. Нехай S і P – кінцеві автомати, а St і Pt – відповідні часові автомати, отримані додаванням інтервалів [0, ∞) і нульових затримок у кожний перехід автоматів S і P. Звичайно автоматні абстракції таких автоматів мають переходи тільки по вхідних символах (i, [0, 0]) і (i, (0, ∞)), реакції на які збігаються для кожної пари «стан, вхідний символ i». У такому випадку, для кожного стану q глобального напівавтомата Q, побудованого для автоматів St і Pt на другому кроці алгоритму, і вхідного символу i часового автомату S або P, в Q існують лише дії (i, [0, 0])/(r, 0) і (i, (0, ∞))/(r, 0). Відповідно, побудований з напівавтомата Q кінцевий автомат AT має переходи тільки по вхідних символах (i, [0, 0]) і (i, (0, ∞)), реакції на які збігаються для кожної пари «стан, вхідний символ i», і може бути перетворений у часовий автомат T, для якого П {[0, ∞)} і всі вихідні затримки дорівнюють нулю. Таким чином, автомат T, побудований по алгоритму 3.1 для автоматів S і P є кінцевим автоматом. У той же час, процедура побудови глобального напівавтомата ідентична для класичного випадку [30] і відповідно, побудований у такий спосіб кінцевий автомат є паралельною композицією кінцевих автоматів S і P. На

рисунку 3.8 представлений фрагмент глобального напівавтомата, побудований на Кроці 2 алгоритму 3.1 для пари часових автоматів S і P.

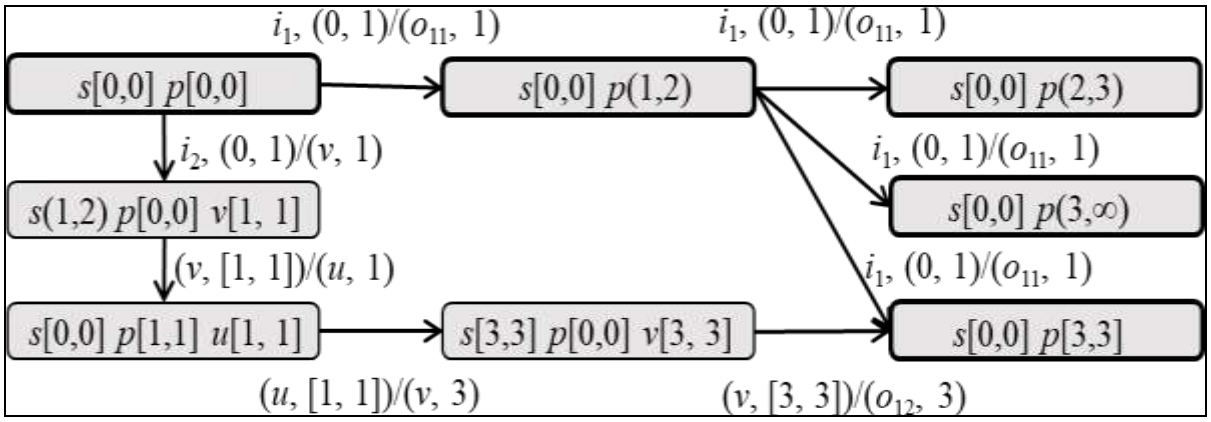

Рисунок 3.8 – Автомати з часовими обмеженнями Si P

Фрагмент кінцевого автомата композиції, відповідний до фрагмента глобального напівавтомату, зображено на рисунку 3.9.

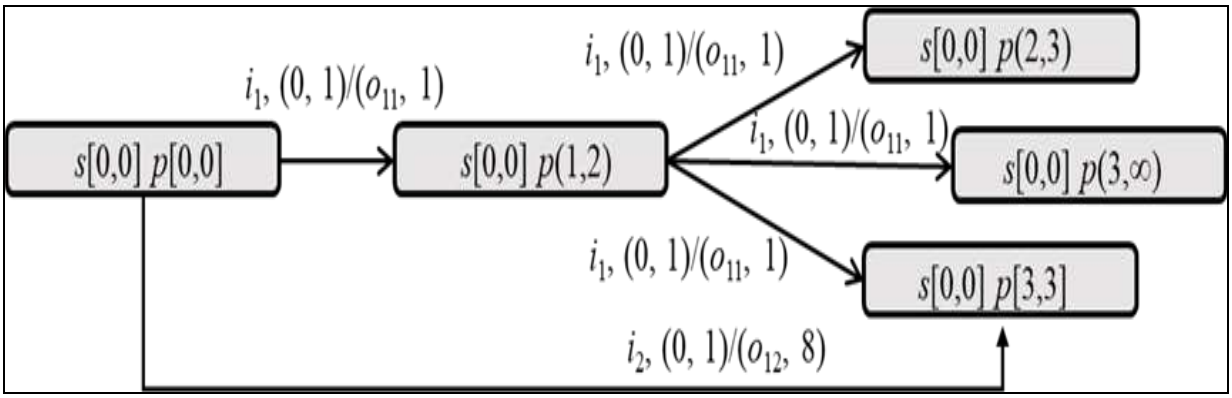

Рисунок 3.9 – Фрагмент глобального напівавтомату

Qкомпозиції автоматів Si P

Глобальний напівавтомат, фрагмент якого представлено на рис. 3.8 кінцевий автомат, що й відповідає, на рис. 3.9 мають недетерміновані переходи, і, як наслідок, композиція не може бути описана детермінованим часовим автоматом.

Таким чином, множина детермінованих автоматів не є замкненою щодо операції паралельної композиції, і в наступному розділі ми досліджуємо причини виникнення недетермінізму при побудові композиції й визначає класи паралельних композицій детермінованих часових автоматів, які можуть бути описані автоматом того ж класу.

Нехай паралельна композиція перебуває в стані (s, [0, 0], p, (1, 2)), і на

компонент S надходить вхідний символ i_1 в інтервалі (0, 1). Якщо компонента S видасть зовнішній вихідний символ, то стан композиції зміниться на одне з наступних станів: (s, [0, 0], p, (2, 3)), (s, [0,0], p, [3, 3]), (s, [0, 0], p, (3, ∞)). Таким чином, недетермінізм проявляється в композиції, коли значення часової змінної однієї з компонентів відповідає інтервалу ненульової довжини. У такому випадку, не завжди можливо однозначно визначити реакцію відповідного компонента на вхідний символ, який також подається в ненульовому інтервалі.

У свою чергу, ненульові інтервали з'являються, коли один з компонентів може видати зовнішню вихідну реакцію у відповідь на зовнішній вхідний символ, а саме, обробити зовнішній вплив на цілий момент часу без діалогу із другим компонентом. У цьому випадку часова змінна, що не приймає в обробці зовнішнього впливу компоненти накопичує час із ненульового інтервалу, яке не завжди може бути однозначно визначене, і в стані глобального напівавтомата з'являються інтервали довжини більше одиниці, які в алгоритмі 3.1 представляються декількома станами з інтервалами часу з G_B. Однак, якщо будь-який зовнішній вплив приводить до діалогу між компонентами, то така ситуація не виникає.

Паралельна композиція автоматів з часовими обмеженнями S і P є детермінованим часовим автоматом, якщо для кожного переходу (sp, i, a, sp', g, d) $\in \lambda\Sigma \cup \lambda\Pi$, такого що i – зовнішній вхідний символ, а є внутрішнім вихідним символом. Якщо визначені ініціальні детерміновані часові автомати S і P, у яких для кожного зовнішнього вхідного символу в кожному стані відповідний вихідний символ є внутрішнім. У цьому випадку, кожна з компонентів обов'язково виконає хоча б один перехід між вступом зовнішнього вхідного символу й видачою зовнішнього вихідного символу й скине значення часової змінної в 0. Через останній, у момент видачі зовнішнього вихідного символу, значення часового змінного компонента, що видав зовнішній вихідний символ, «скидається» в 0, у той час як значення часового змінного іншого компонента стає рівним затримці останнього виконаного композицією переходу, тобто має ціле значення. Оскільки в момент видачі зовнішнього вихідного символу

значення часової змінної кожної з компонентів ціле, та множина стабільних станів глобального напівавтомата втримується в $S \times G_B^{int} \times P \times G_B^{int}$, де $G_B^{int} = \{[0, 0], [1, 1], \ldots, [B, B], (B, \infty)\}$.

Останнє виключає можливість недетермінізму в стабільних станах глобального напівавтомата, у той час як переходи із проміжних станів є детермінованими по визначенню. Таким чином, автомат з часовими обмеженнями, відповідний до глобального напівавтомата, є детермінованим.

Тобто, якщо в композиції часових автоматів S і P кожний зовнішній вхідний символ приводить до діалогу між компонентами, то паралельна композиція автоматів S і P є часовим детермінованим автоматом.

До даного класу, наприклад, ставляться послідовні композиції. У послідовній композиції лише одна з компонентів (головний компонент) приймає зовнішні вхідні символи, у той час як другий компонент (хвостовий компонент) приймає вхідні впливи тільки від першого компонента, при цьому зовнішні вихідні реакції може видавати лише хвостовий компонент (рисунок 3.10).

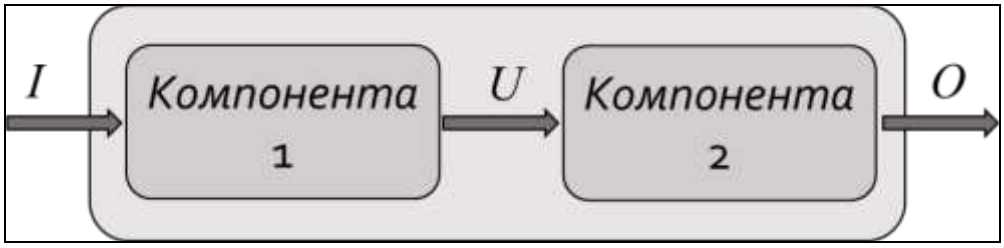

Рисунок 3.10 – Послідовна композиція

Оскільки головний компонент може видавати лише внутрішні вихідні реакції, умови виконуються. Таким чином, множина ініціальних часових детермінованих повністю визначених автоматів замкнута щодо операції послідовної композиції.

Аналогічно можна виділити інший клас детермінованих часових автоматів, для яких операція паралельної композиції є замкненою.

Паралельна композиція автоматів з часовими обмеженнями S і P є детермінованим автоматом з часовими обмеженнями, якщо зовнішні вхідні символи подаються на композицію тільки в цілі моменти часу. Через те, що момент вступу вхідного символу й вихідні затримки є цілими числами, множина

стабільних станів глобального напівавтомата втримується в $S \times G_B^{int} \times P \times G_B^{int}$, де $G_B^{int} = \{[0, 0], [1, 1], \ldots, [B, B], (B, \infty)\}$ і, більше того, для проміжних станів виду (s, gs, p, gp, (a, g)), справедливо, що $(s, gs, p, gp) \in S \times G_B^{int} \times P \times G_B^{int}$ і $(a, g) \in (U \cup V) \times G_B^{int}$. Таким чином, глобальний напівавтомат і відповідний автомат з часовими обмеженнями є детермінованими.

Можна розглянути ще одну умову, при якім паралельна композиція часових детермінованих автоматів є детермінованим автоматом.

Паралельна композиція автоматів з часовими обмеженнями S і P є детермінованим автоматом з часовими обмеженнями, якщо для кожного переходу $(sp, i, a, sp', g, d) \in \lambda\Sigma \cup \lambda\Pi$, справедливо що $d > B$, де B – максимальна з кінцевих границь часових інтервалів BS і BP.

Дійсно, якщо затримки переходів в автоматах і компонентах перевищують верхню межу часових інтервалів, то множина стабільних станів глобального напівавтомата втримується в $S \times \{[0, 0], (B, \infty)\} \times P \times \{[0, 0], (B, \infty)\}$ і реакція композиції на зовнішній вхідний вплив може бути визначена однозначно.

Ще одним класом композицій, дія яких описується детермінованим часовим автоматом, є композиція контексту й вбудованого компонента (Рисунок 3.11).

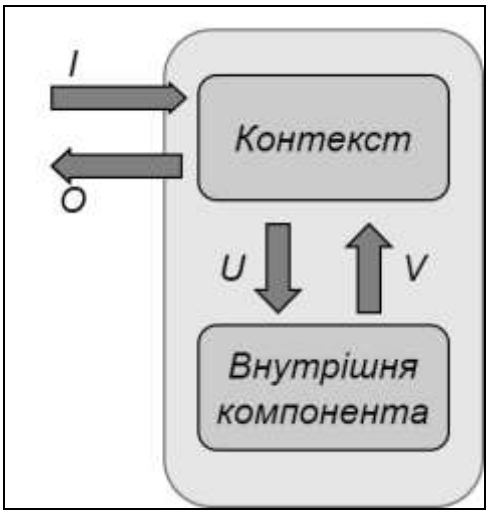

Рисунок 3.11 – Композиція з контекстом

У цьому випадку лише один з компонентів (контекст) взаємодіє із зовнішнім середовищем. Якщо дія вбудованого компонента не залежить від часу, то значення часового змінного зовнішнього компонента (контексту) завжди

«скидається» в 0 при видачі композицією зовнішнього вихідного символу.

Композиція контексту з вбудованим компонентом досить часто використовується в різних додатках, наприклад, при композиції серверних і клієнтських додатків. Композиція із двох автоматів з часовими обмеженнями S і P (рис. 3.12), отримана з компонентів композиції заміною переходів по вхідним-вихідним парам із зовнішніх вхідних і вихідних символів на відповідні переходи із внутрішніми вихідними символами.

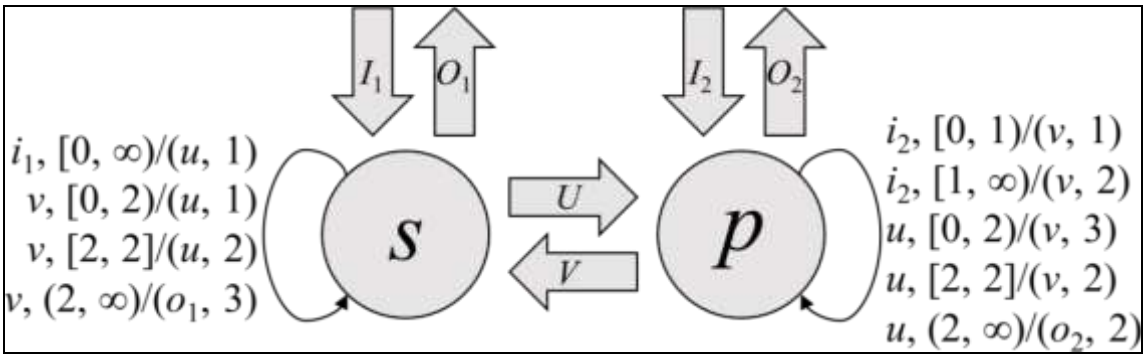

Рисунок 3.12 – Автомати з часовими обмеженнями Si P

Звичайні автоматні абстракції автоматів S і P представлені на рисунку 3.13. Оскільки найбільша із меж часових інтервалів у композиції автоматів S і P рівна 3, то переходи абстракцій будуються до вхідного символу (i, (3, ∞)).

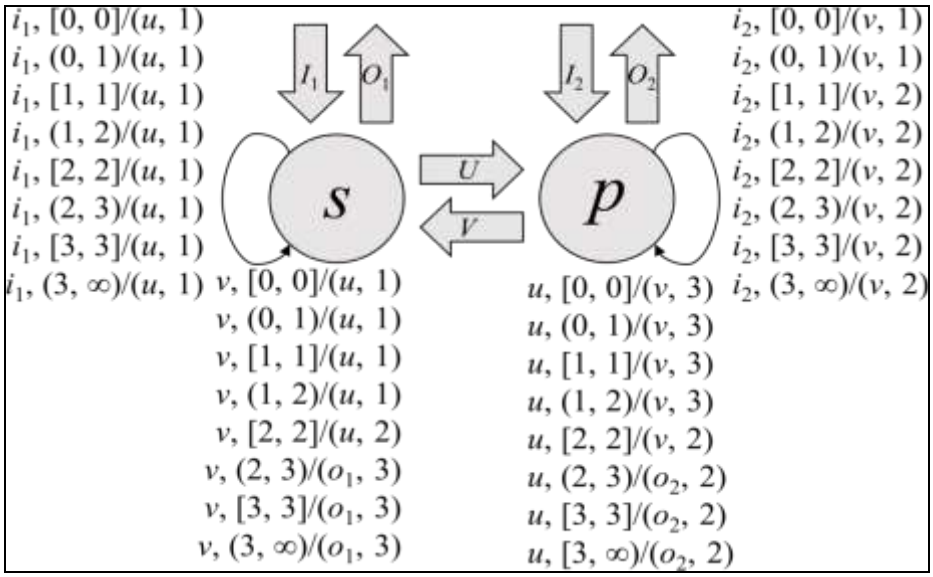

Рисунок 3.13 – Звичайні автоматні абстракції з часовими обмеженнями

Побудовано фрагмент глобального напівавтомата Q (рис. 3.14), що описує дію композиції автоматів з часовими обмеженнями S і P.

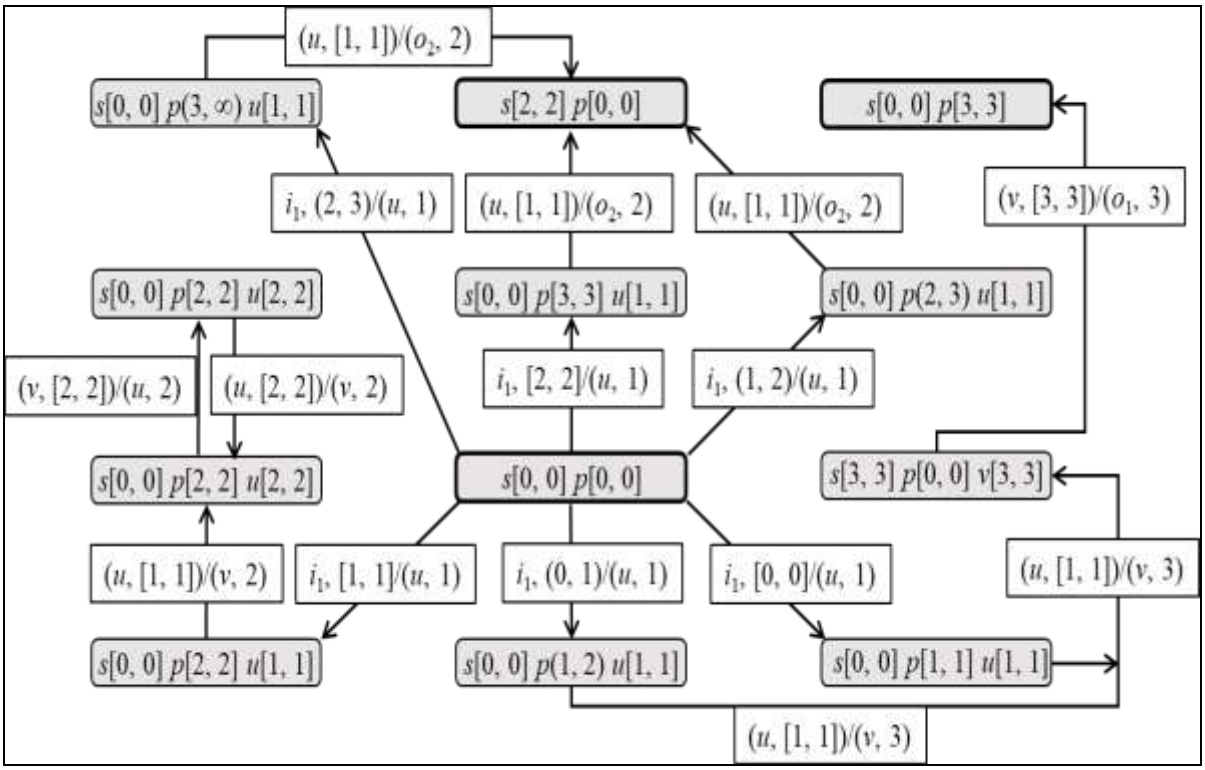

Рисунок 3.14 – Фрагмент глобального напівавтомата

Відзначимо, що глобальний напівавтомат має цикл, що проходить тільки через проміжні стани й відповідний до наявності осциляції в композиції часових автоматів, і в такому випадку, побудувати повністю визначений автомат та композицію неможливо. Однак, якщо замінити перехід (p, u, v, p, [2, 2], 2) автомата P на перехід (p, u, v, p, [2, 2], 3), то цикл зникає, і відповідний до глобального напівавтомату повністю визначений часовий автомат та композиція може бути побудовані так, як це показано на кроці 3 алгоритму 3.3 (рис. 3.15).

Аналогічно операції композиції над класичними кінцевими автоматами, можна довести наступне твердження.

Паралельна композиція повністю визначених автоматів з часовими обмеженнями S і P є повністю визначеним автоматом з часовими обмеженнями, якщо відповідний глобальний напівавтомат не має циклів, що проходять тільки через проміжні стани.

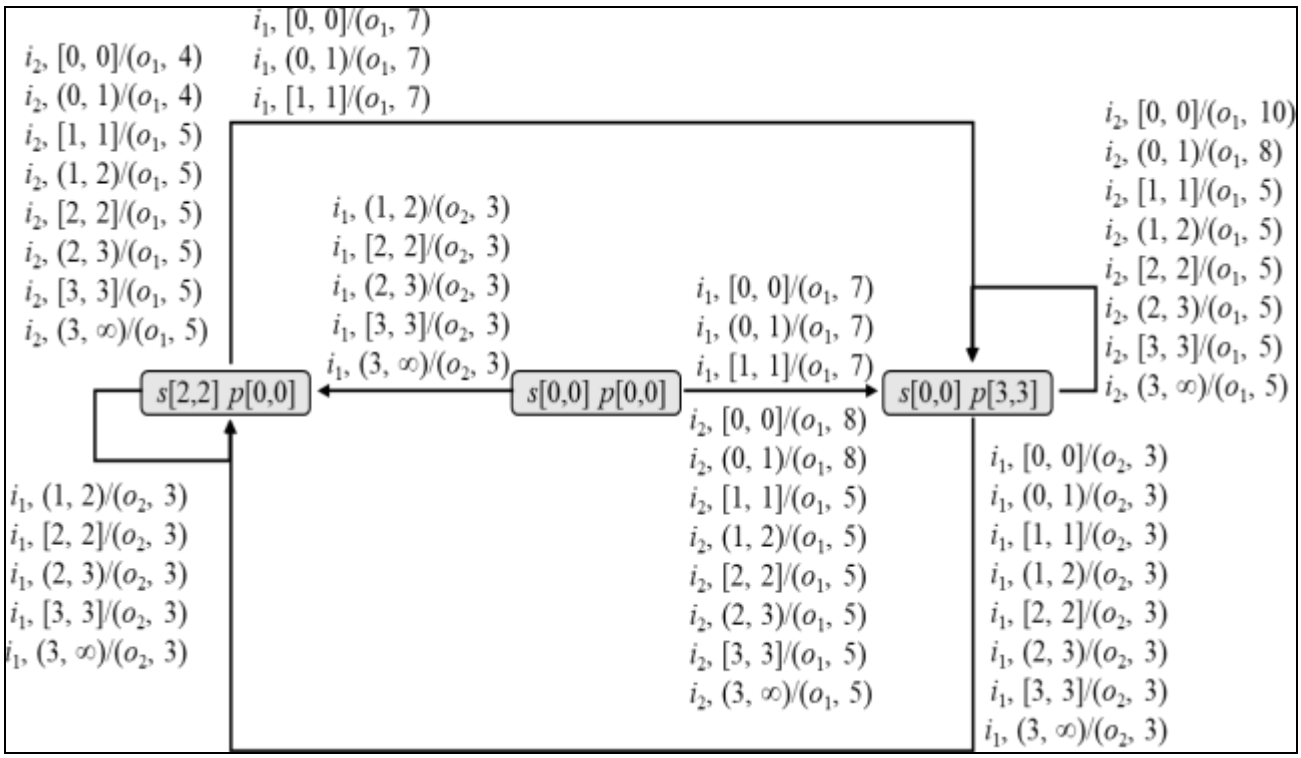

Рисунок 3.15 – Композиція автоматів S і P

Таким чином, операція паралельної композиції визначена для ініціальних детермінованих автоматів з часовими обмеженнями; результатом операції є часовий автомат, що описує дію системи взаємодіючих часових автоматів. Введена операція є розширенням операції паралельної композиції над класичними кінцевими автоматами й зводиться до неї, якщо компонентами композиції є кінцеві автомати без часових аспектів.

На відміну від класичних ініціальних автоматів, «повільне» зовнішнє середовище не є достатньою умовою для опису паралельної композиції двох часових детермінованих автоматів автоматом цього ж класу, тобто множина ініціальних детермінованих часових автоматів не є замкненою щодо паралельної композиції.

4 ОПИС РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ

4.1 Опис проектування кінцевих автоматів

Багато сучасних інтернет сервісів використовують двокрокову аутентифікацію для одержання доступу до облікового запису (також названу акаунтом), вимагаючи не тільки ввести пароль, але й підтвердити особистість користувача, увівши спеціальний код, відправлений по електронній пошті. В якості практичної реалізації розроблено фрагмент серверної реалізації такої системи, автоматний опис якої (див. рис. 4.1) містить два стани, два вхідні та два вихідні символи.

Рисунок 4.1 – Кінцевий автомат для системи аутентифікації

У стані «Очікування запиту» система чекає запит від користувача на дозвіл входу в обліковий запис; коли надходить вхідний символ «Запит» автомат відправляє користувачеві лист із кодом доступу й переходить у стан «Очікування коду». Далі система чекає введення правильного коду користувачем (або можливо повторення запиту), у відповідь на який надає або не надає доступ до облікового запису. Для простоти на рисунку ми не розглядаємо випадок, коли введений код неправильний; у цьому випадку система не дозволяє доступ до облікового запису.

Кортеж (s, i, o, s') описує перехід зі стану s у стан s' під дією вхідного символу i з видачею вихідного символу *o*. Для наочності, кінцевий автомат часто

представляється орієнтованим графом (діаграмою переходів), вершини якого відповідають станам автомата, а дуги позначені парою (вхідний символ/вихідний символ). Таким чином, якщо в автоматі існує перехід (s, i, o, s'), то у відповідній графі існує дуга з вершини s у вершину s', позначена парою i/o.

Якщо в автоматі S для будь-якої пари (s, i) ∈ S × I існує перехід (s, i, o, s') ∈ h_s, то кінцевий автомат називається повністю визначеним (по вхідним символам), інакше – частковим.

Частковість автомата суттєво підвищує складність рішення багатьох завдань теорії автоматів [5]. У ряді випадків повністю визначений автомат може бути досить просто отриманий із часткового довизначення невизначених переходів, механізм якого суттєво залежить від інтерпретації таких переходів [2]. Різні види довизначення широко використовуються при аналізі систем, зокрема, при аналізі реалізацій телекомунікаційних протоколів, і в даній роботі ми розглядаємо тільки повністю визначені автомати. Автомат на рис. 4.1 є частковим, але може бути довизначеним до повністю визначеного переходом по вхідному символу «Код» у стані «Очікування запиту» з вихідним символом NULL. При такому довизначенню дозвіл на вхід в обліковий запис буде як і раніше даний тільки після відповідного запиту й правильно введеного коду (рис. 4.2).

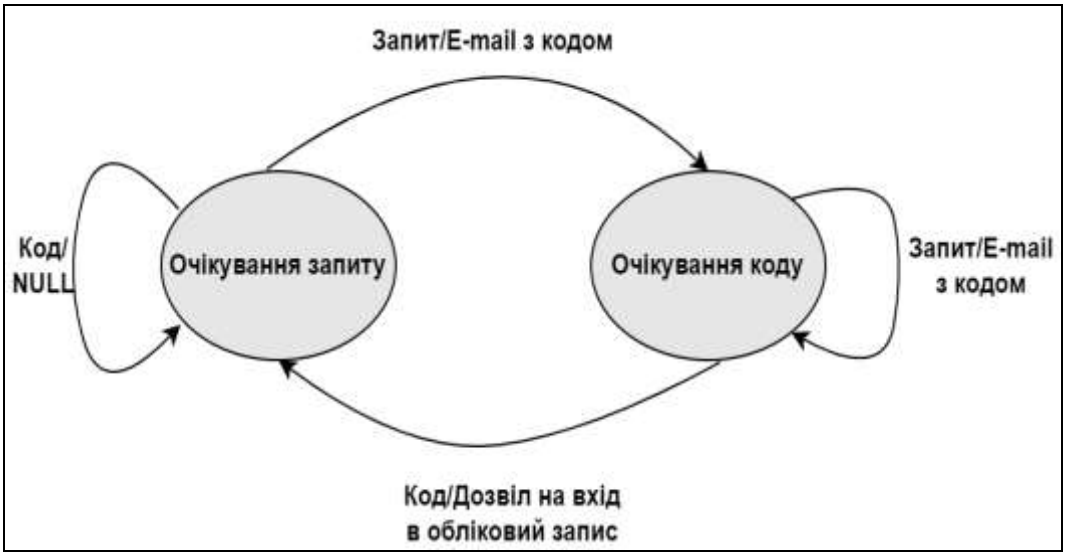

Рисунок 4.2 – Довизначення кінцевого автомату для системи аутентифікації

Якщо в автоматі S для будь-якої пари (s, i) ∈ S × I існує не більш одного

переходу (s, i, o, s') ∈ h_s, те кінцевий автомат називається детермінованим, інакше – недетермінованим. Наприклад, у специфікацію системи на рис. 4.2 можна внести недетермінізм, додавши можливість відправляти коди користувачеві не тільки по електронній пошті, але й за допомогою SMS-повідомлень.

Відповідний автомат зображено на рис. 4.3 і являє приклад недетермінізму як наслідку опціональності в специфікації. Повністю визначений недетермінований автомат називається спостережуваним, якщо для кожної трійки (s, i, o) ∈ S × I × O існує єдиний перехід (s, i, o, s') ∈ h_s, тобто наступний стан визначається єдиним чином. Автомат на рис. 4.3 є спостережуваним.

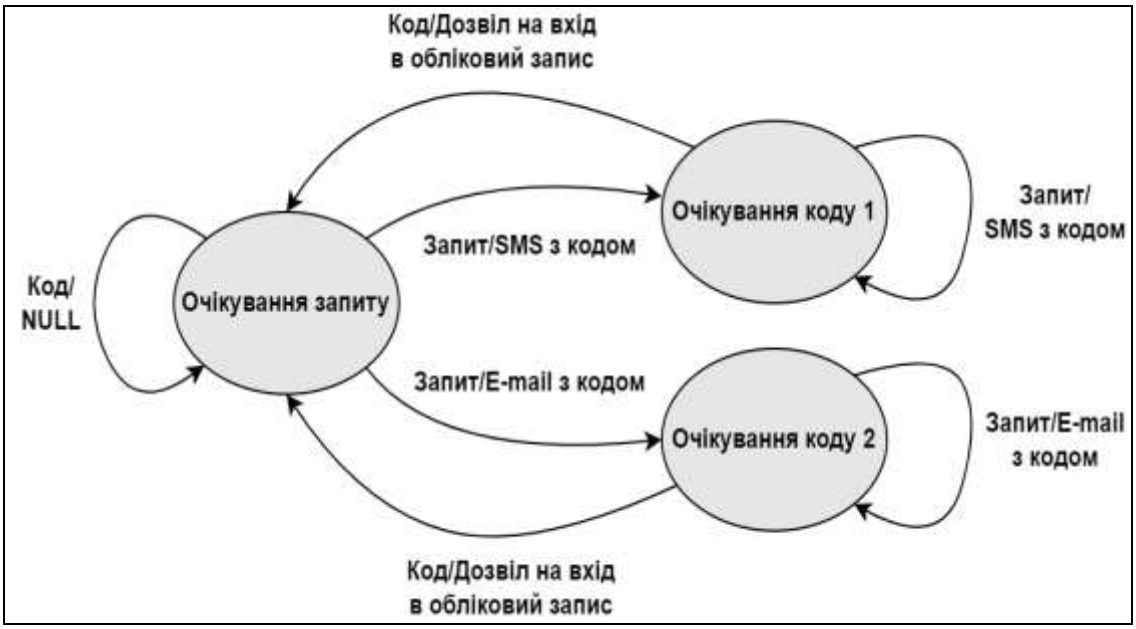

Рисунок 4.3 – Повністю визначений недетермінований кінцевий автомат для системи аутентифікації

Якщо в автоматі зафіксований єдиний початковий стан, то автомат називається ініціальним. Ініціальні автомати звичайно використовуються для опису роботи систем, що мають надійний сигнал скидання з будь-якого стану в деякий (початковий) стан.

Ініціальний автомат є п'ятірка S = (Σ, I, O, hs, s_0), де s_0 – виділений початковий стан. Якщо будь-який стан автомата може бути обраний в якості початкового, то автомат називається неініціальним.

Для автомата S припустима послідовність пар вхідний символ/вихідний символ i_1/o_1, i_2/o_2, …, i_n/o_n у стані s_1 називається вхідно-вихідною послідовністю в

стані s_1, якщо в S існує відповідна послідовність переходів (s_1, i_1, o_1, s_2), (s_2, i_2, o_2, s_3), …, (s_n, i_n, o_n, s_{n1}). При цьому $\alpha = i_1, i_2, \ldots, i_n$ є вхідна послідовність, $\gamma = o_1, o_2, \ldots,$ on – відповідна вихідна послідовність, а вхідно-вихідна послідовність позначається α/γ. Для вхідно-вихідної послідовності α/γ вхідну послідовність α іноді називають вхідною проекцією, а відповідну вихідну послідовність γ – вихідною реакцією. Відношення роботи повністю визначеного автомату звичайним чином поширюється на вхідно-вихідні послідовності: якщо автомат у стані s може видати вихідну реакцію γ на вхідну послідовність α і перейти в стан s', те $(s, \alpha, \gamma, s') \in h_s$. Під довжиною вхідної (вихідної) послідовності розуміється кількість вхідних (вихідних) символів, що входять у послідовність.

Нехай S – повністю визначений автомат. Префіксом послідовності $\alpha = i_1, i_2, \ldots, i_n$ називається початковий відрізок даної послідовності, тобто послідовність $\alpha' = i_1, i_2, \ldots, ik$, де $0 < k \leq n$. Послідовність α у стані s автомата S покриває стан s', якщо існує префікс α' послідовності α, такий що $(s, \alpha', \gamma, s') \in h_s$. Якщо $(s, \alpha, \gamma, s') \in h_s$ у спостережуваному автоматі S, той стан s' називається α/γ– спадкоємцем стану s по вхідно-вихідній послідовності α/γ. Якщо α/γ є спадкоємцем стану s є порожня множина, то іноді говорять, що α/γ спадкоємець стану s не існує.

Для послідовностей $\alpha = i_1, i_2, \ldots, i_n$ і $\gamma = o_1, o_2, \ldots, o_n$ запис α/γ є скороченням запису для вхідно-вихідної послідовності $i_1/o_1, i_2/o_2, \ldots, i_n/o_n$. Якщо автомат є недетермінованим, то для вхідної послідовності α може існувати кілька вихідних реакцій $\{\gamma_1, \ldots, \gamma_v\}$, і відповідно, у деяких (усіх) станах існує декілька вхідно-вихідних послідовностей з однією й тією же вхідною проекцією. Крім того, α/γ-спадкоємцем стану s у неспостережуваному автоматі може бути множина станів, однак для спостережуваного, зокрема, детермінованого автомата, які розглядаються в даній роботі, α/γ стану або не існує, або є синглтон. Для вхідної послідовності α можна розглядати стан s, яким є множина станів, у які автомат може перейти зі стану s під дією вхідної послідовності α. Відзначимо також, що в спостережуваному автоматі S α/γ підмножини станів S є множина $\alpha/\gamma_o\rangle$ станів $s \in S'$.

Під множиною входів вихідних послідовностей ініціального автомата розуміється множина таких послідовностей у початковому стані автомата, і в ряді випадків ця множина називається мовою автомата.

Для автомата на рис. 4.3 у стані «Очікування запиту» вхідної послідовності α = “Запит”, “Запит” відповідають вихідні послідовності “SMS з кодом”, “SMS з кодом” і “E-mail з кодом”, “E-mail з кодом”, а множина містить стан “Очікування коду 1” і “Очікування коду 2”. Однак для будь-якої вихідної послідовності $\gamma \in$ {“SMS з кодом”, “SMS з кодом”; “E-mail з кодом”, “E-mail з кодом”} початковий стан має тільки один α/γ.

На рис. 4.4 представлений тестовий приклад для системи аутентифікації.

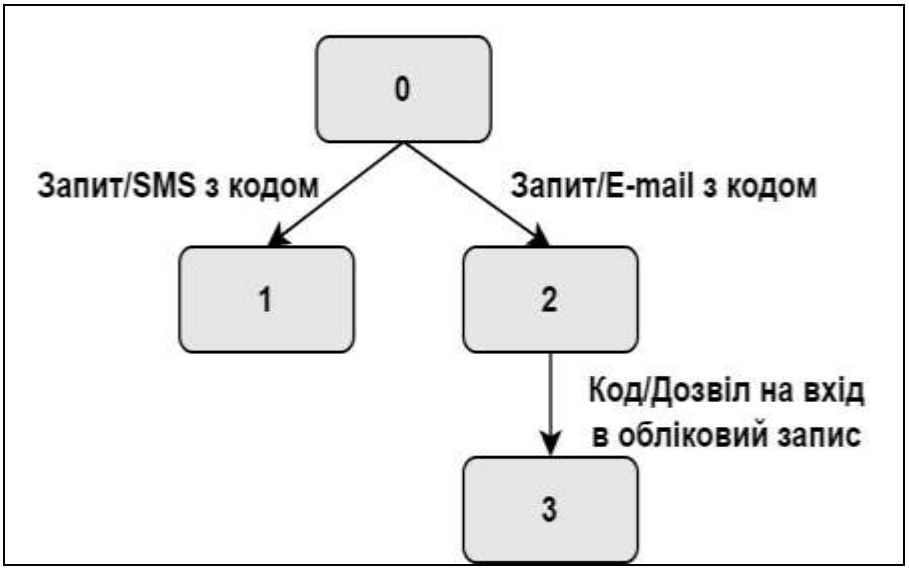

Рисунок 4.4 – Тестовий приклад для системи аутентифікації

Відповідно до якого спочатку буде поданий вхідний символ “Запит”. Далі, у випадку одержання реакції “E-mail з кодом” буде також поданий наступний вплив “Код”, у той час як при одержанні реакції “SMS з кодом” тестовий приклад перейде в тупиковий стан і подача адаптивної вхідної послідовності завершиться.

На рисунку 4.5 зображена система аутентифікації, розширена тайм-аутами і часовими обмеженнями (див. рис. 4.6).

У цьому випадку, система очікує код у термін 5 хвилин, після чого скидає аутентифікацію та повертається в стан “Очікування запиту”.

Рисунок 4.5 – Система аутентифікації з тайм-аутами

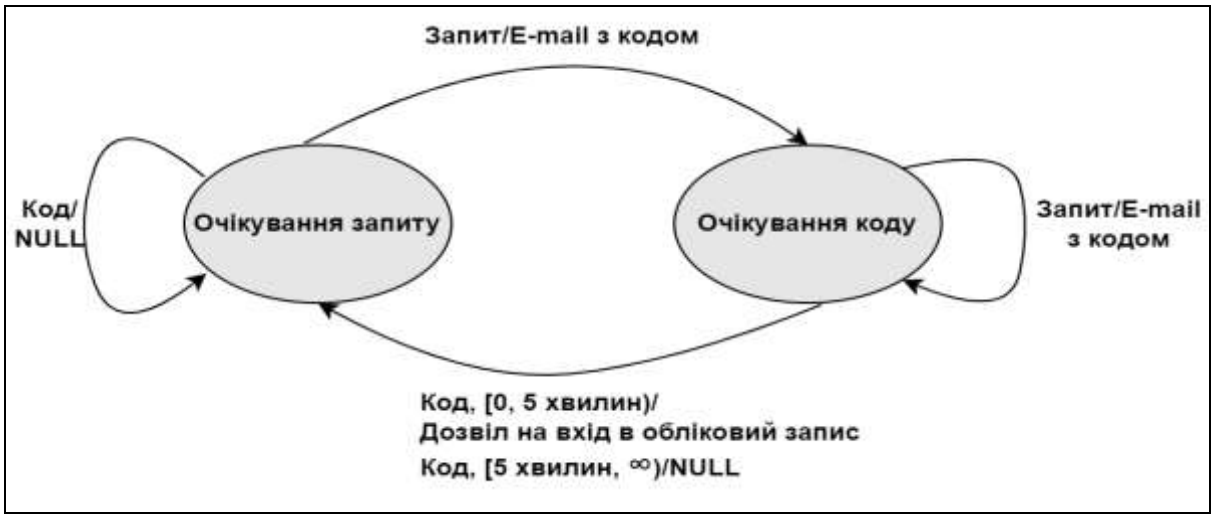

Рисунок 4.6 – Скінченний автомат системи аутентифікації з часовими обмеженнями

Повнота (частковість) і детермінізм (недетермінізм) для автомату з тайм-аутами визначена аналогічно класичному кінцевому автомату.

Для окремих випадків часового автомата поняття вхідна і вхідна часова послідовність вводяться аналогічно загальному випадку.

Таким чином, проведений аналіз сучасного стану вирішення проблем і для виконання поставленої мети дослідження, необхідно розв'язати наступні завдання:

– розробити метод мінімізації часового автомату та дослідити, чи існує для

часових автоматів єдина наведена форма;

−адаптувати операцію паралельної композиції класичних кінцевих автоматів для часових автоматів;

−адаптувати для часових автоматів класичні звичайно автоматні методи синтезу тестів з гарантованою повнотою.

4.2 Моделювання несправності для синтезу тестів

Наведено результати комп'ютерних експериментів, проведених із програмною реалізацією запропонованого вище методу синтезу тесту. Приведено отримані оцінки довжини тесту для випадково згенерованих часових автоматів і деяких реальних сервісних систем. Дослідження довжини тестів, що перевіряють, для часових автоматів. Для оцінки довжини тесту, що перевіряє, щодо часового автомату-специфікації, був реалізований мовою програмування C++. Комп'ютерні експерименти проводилися з випадково згенерованими автоматами з тайм-аутами й часовими обмеженнями, а також класичними кінцевими автоматами. Експериментальна вибірка генерувалася з параметрами |I| = |O| = 10, у той час як тест будувався W і H-методами [33] для класичних автоматів і звичайних автоматних абстракцій часових автоматів. Залежності середньої довжини побудованих тестів від числа станів досліджуваних автоматів представлено на рис. 4.7, де W(FSM) і H(FSM) – тести для класичних кінцевих автоматів, W(TFSM) і H(TFSM) – тести для часових автоматів.

Згідно із проведеними експериментами, різниця між довжиною H(FSM) і H(TFSM) тестів не занадто велика, у той час як H(TFSM) тест виявляє всі некоректні реалізації, для яких мінімальна форма звичайно автоматної абстракції має не більше станів, ніж для автомату-специфікації.

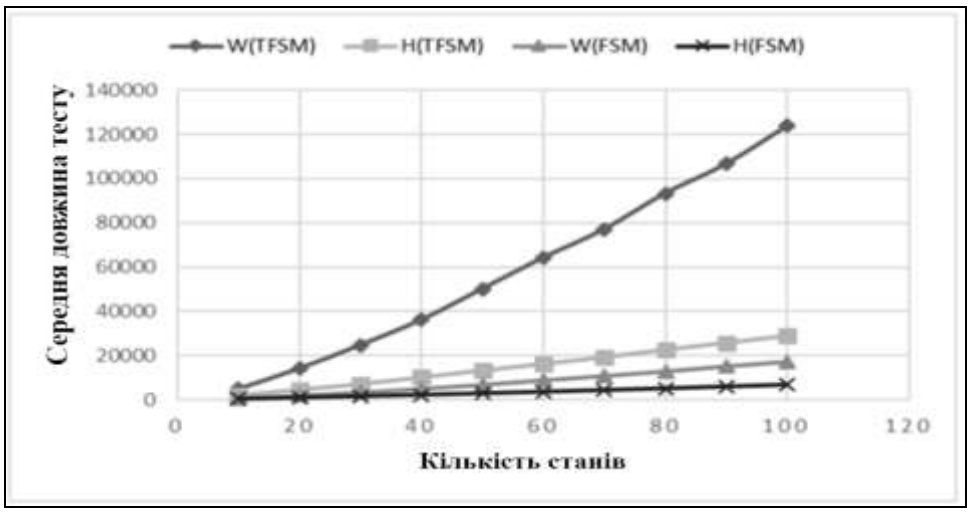

Рисунок 4.7 – Залежність середньої довжини тесту, що перевіряє, від числа станів часового автомату і специфікації

Ми також провели експерименти по оцінці впливу величини тайм-аутів на довжину тесту. Розглядалися тайм-аути величини 5, 10, і 15 для автоматів з 10 і 20 станами; результати представлено на рисунку 4.8.

Довжина H(TFSM) тесту збільшується вдвічі, коли тайм-аут збільшується у два рази.

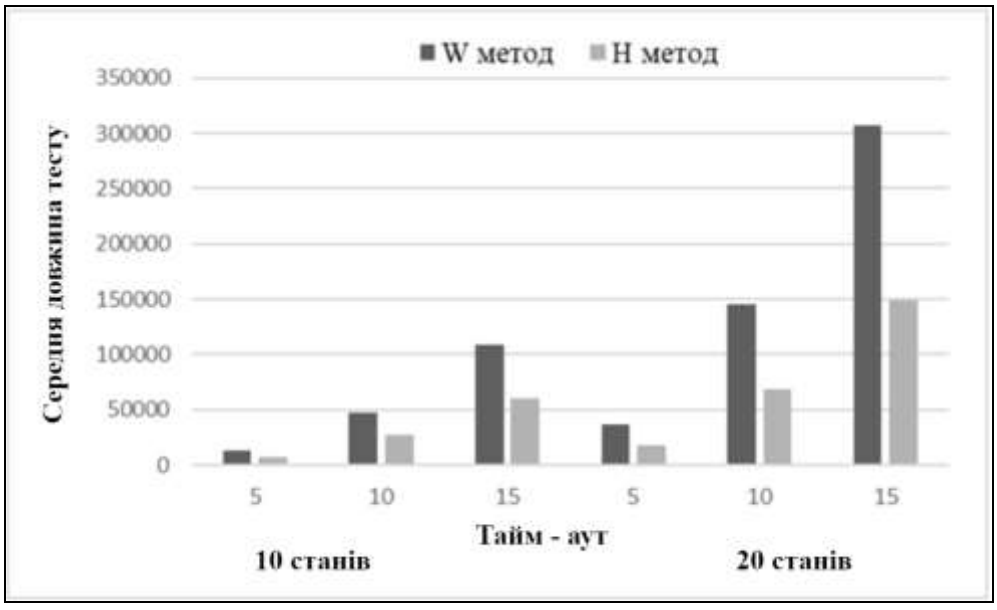

Рисунок 4.8 – Результати експериментів для автоматів-специфікацій з різними тайм-аутами

4.3 Реалізація синтезу тестів для сервісних систем

Програмна реалізація виконує оцінку довжини повних тестів, що перевіряють, для сервісних систем, поведінка яких залежить від часу. Розглянуто, як приклад, сервісну систему італійської національної залізничної компанії, яка дозволяє пасажирам одержувати компенсації у випадку затримки поїздів (рис 4.9).

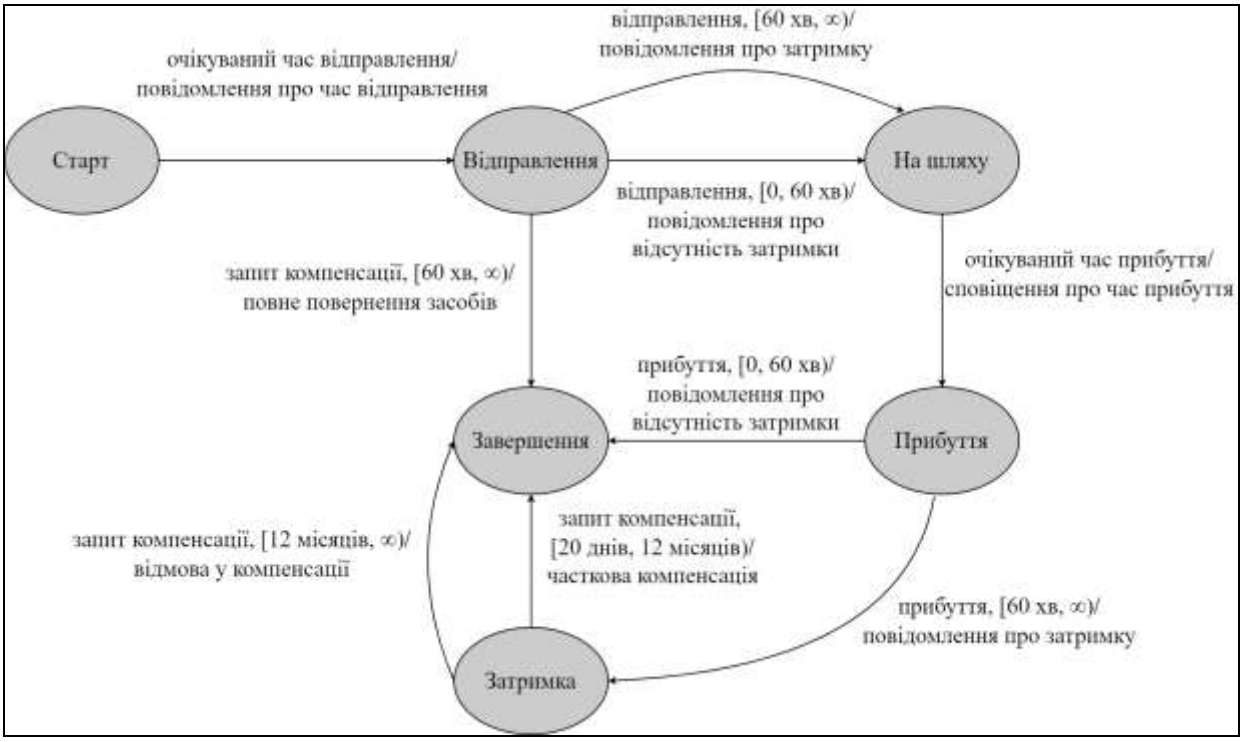

Рисунок 4.9 – Часовий автомат, що описує сервісну систему італійської національної залізничної компанії

Пасажири інформуються про затримки (перехід “очікуваний час відправлення/повідомлення про час відправлення”) і можуть запросити компенсацію до відправлення поїзда при запізненні на 60 хвилин і більше поки автомат перебуває в стані “Відправлення”. У цьому випадку полагається виплата повної суми компенсації. Після відправлення поїзду система переходить у стан “У дорозі” і далі повідомляє пасажирів про час прибуття. Якщо поїзд спізнився на 60 хвилин і більш, то автомат переходить у стан “Затримка”, і пасажири можуть запросити компенсацію, а якщо ні, то, система завершує обслуговування,

виконавши перехід “прибуття, [0, 60 хв)/повідомлення про відсутність затримки”. Після поїздки компенсація може бути отримана в період від 20 днів до 12 місяців. Однак, у цей період можна одержати лише часткову компенсацію, у той час як через 12 місяців після поїздки одержання компенсації неможливо. Побудова тестових послідовностей для часових автоматів з використанням описаних вище методів вимагає побудови звичайно автоматних абстракцій, число станів яких суттєво залежить від значень тайм-аутів і часових обмежень на переходах. Відповідно, для того щоб кінцевий автомат мав «розумні» розміри, скорочено значення часових змінних у стані “Затримка” з 20 днів і 12 місяців до двох днів і одного місяця відповідно.

Довжина побудованого тесту з використанням HSI-методу, який є однієї з модифікацій W-методу, представлено в таблиці 4.1 (стовпець *m*). Побудований тест виявляє кожну нееквівалентну реалізацію часового автомату, мінімальна форма звичайно автоматної абстракції якої має не більше станів, ніж мінімальна форма відповідної абстракції часового автомата-специфікації. Область несправності може бути розширена реалізаціями (часовими автоматами), звичайно автоматні абстракції яких містять більше станів, ніж звичайно автоматна абстракція специфікації. Відповідно, також програмно оцінено побудовані тести, що дозволяють виявити кожну несправну реалізацію (часовий автомат), мінімальна звичайно автоматна абстракція якої містить не більш ніж (m+1) і (m+2) станів, де *m* – число станів мінімальної форми звичайно автоматної абстракції часового автомату-специфікації. Довжини тестів, що перевіряють, для звичайно автоматних абстракцій і відповідних часових автоматів, побудованих методом HSI [33], наведено в таблиці 4.1.

У таблиці 4.1 також представлені відповідні результати для системи керування ліфтом, опис якої презентовано на рисунку 4.10.

У стані “Двері закриті” система чекає виклику ліфта, або запиту на відкриття дверей. При вступі запиту на відкриття дверей автомат переходить у стан “Двері відкриті” з вихідним символом ←| .□| У стані “Двері відкриті”, реакція на вхідні символи залежить від часу подачі вхідного впливу. Якщо

вхідний символ “нагору” або “униз” поданий у проміжку [0, 7) система перейде в стан “Закр. тайм-аут”, вижде п'ять тактів часу й почне закривати двері перейшовши в стан “Закр. двері”. У той же час, у проміжку [7, 10), двері вже почали закриватися й автомат відразу перейде в стан “Закр. двері”, з якого перейде в стан “Рух” після повідомлення про закриття дверей. Якщо в стані “Двері відкриті” вхідний вплив не був подано в продовж 10 тактів, то двері закриються й автомат перейде у відповідний стан по тайм-ауту. Зі стану “Рух” система переходить у стан “Двері відкриті” по команді відкриття двері.

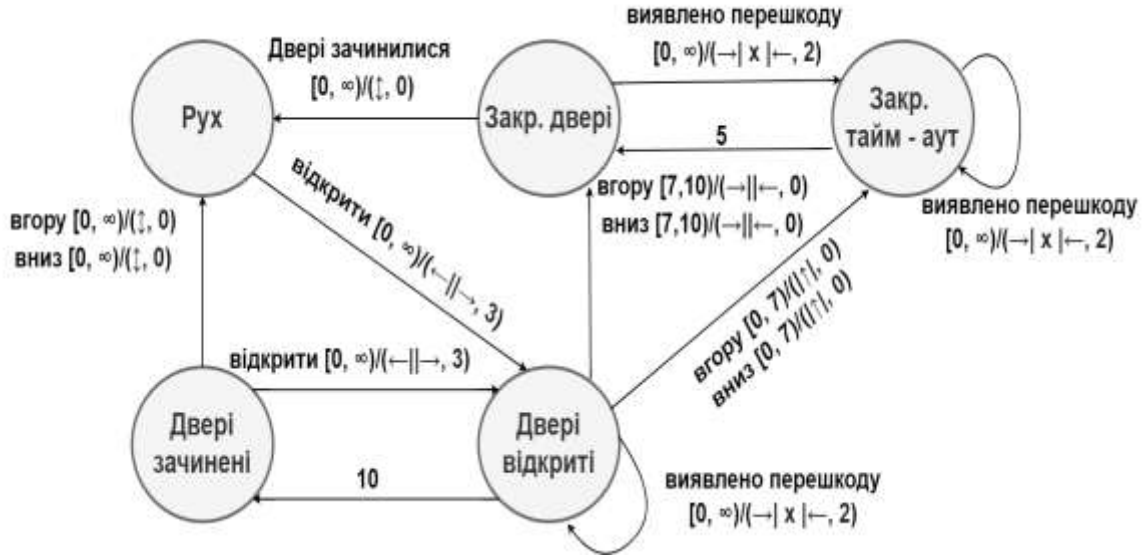

Рисунок 4.7 – Система керування ліфтом

Таблиця 4.1 – Довжини тестів для сервісних систем, побудованих HSI методом

	Число станів реалізації	m	m+1	m+2
Залізнична система повернення засобів	Довжина тесту для часового автомату	20107	137681	932644
	Довжина тесту для відповідної абстракції	112514	692006	4278194
Система керування ліфтом	Довжина тесту для часового автомату	6181	46420	335504
	Довжина тесту для відповідної до абстракції	29388	184697	1167346

При близьких значеннях числа станів і вхідних символів у розглянутих

прикладах, тести для першої з розглянутих систем значно довші, що є наслідком більш високих значень тайм-аутів та/або часових обмежень. Проте, тести мають досить розумну довжину при дуже високій якості, тобто виявленні великої кількості несправностей.

Оскільки мінімальна форма ініціальних детермінованих часових автоматів не є єдиною з точністю до ізоморфізму, завдання синтезу, що перевіряє тест з гарантованою повнотою для таких автоматів не вдається звести до перевірки відповідності між переходами специфікації, що й перевіряється реалізації, що є основою синтезу таких тестів для класичних автоматів.

Визначена модель несправності для ініціальних детермінованих повністю визначених часових автоматів і запропонований відповідний метод синтезу тестів щодо еквівалентності з гарантованою повнотою покриття несправності. Область несправності й метод синтезу тестів засновані на звичайно автоматній абстракції часового автомата й дозволяють звести перевірку еквівалентності між часовими автоматами до перевірки ізоморфізму між їхніми абстракціями.

5 ОПИС МОЖЛИВОСТІ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Приведено результати комп'ютерних експериментів по оцінці довжини адаптивних послідовностей, що розрізняють.

Проведені експерименти (рис. 5.1) показали, що для випадково згенерованих автоматів тестовий приклад, що розрізняє, існує вкрай рідко, якщо nd > 3, і відповідно, наступні експерименти проводилися з автоматами, для яких nd ≤ 3.

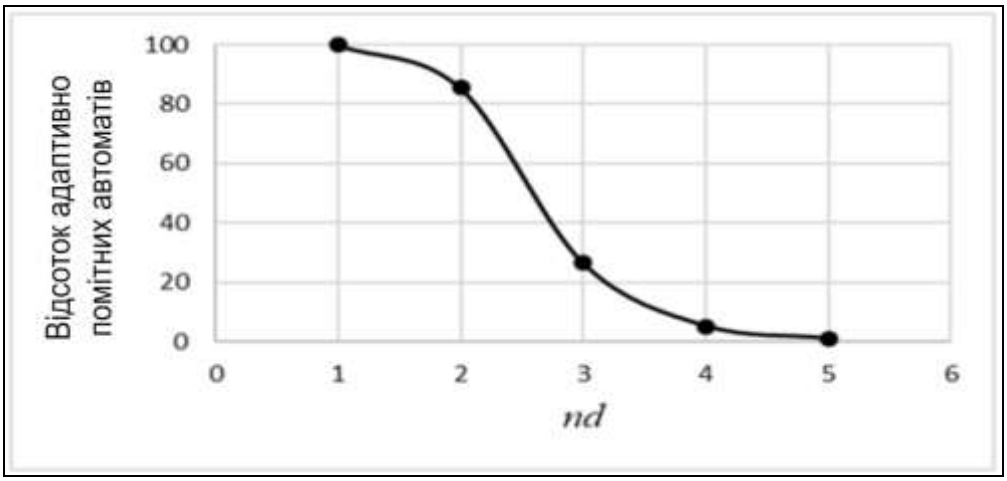

Рисунок 5.1 – Відсоток адаптивно помітних автоматів

Перша серія експериментів проводилася з випадково згенерованими кінцевими автоматами для оцінки ймовірності існування адаптивної, що розрізняє послідовності залежно від числа недетермінованих переходів автомата. Кількість nd переходів автомата для кожної пари ‘стан, вхідний символ’ задається заздалегідь.

На рисунку 5.2 представлена середня висота тестового прикладу, що розрізняє, залежно від числа станів автомата. Експерименти проводилися з автоматами, для яких |I| = |O| = 10, при різнім числі недетермінованих переходів. Нижня крива побудована для детермінованих автоматів, середня й верхня криві для значень nd, що дорівнює 2 і 3 відповідно.

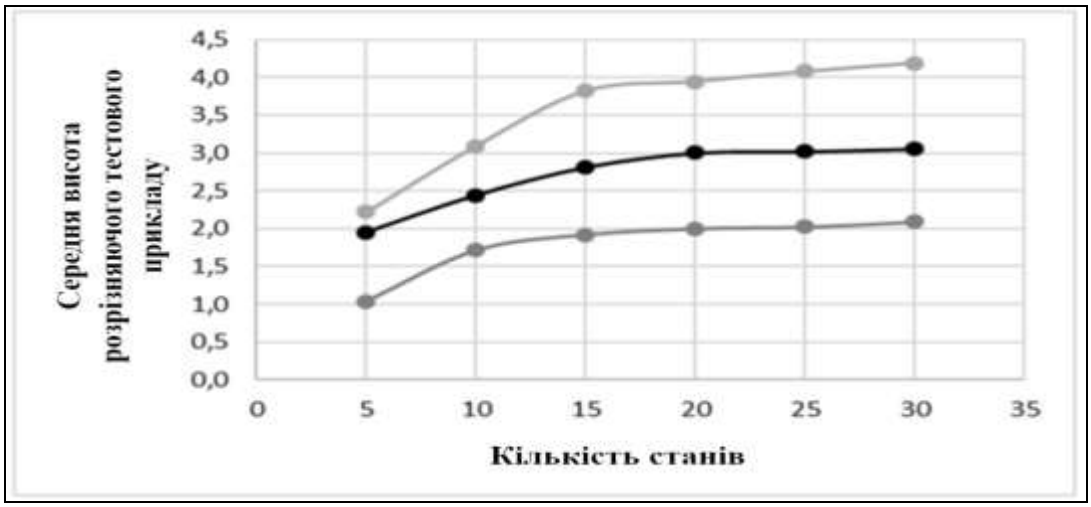

Рисунок 5.2 – Середня висота тестового прикладу, що розрізняє, для детермінованих і недетермінованих автоматів

Із проведених експериментів випливає, що для випадково згенерованих автоматів тестовий приклад, що розрізняє, або має невелику висоту, або не існує. Однак, незважаючи на невелику висоту, що розрізняють тестових прикладів, необхідні для їхньої побудови, що розрізняють автомати в деяких випадках містять більшу надмірність, і крім того, описаний вище підхід не гарантує побудови найкоротшого тестового прикладу, що розрізняє (якщо такий приклад існує).

Проведено комп'ютерні експерименти по оцінці часу, затрачуваного на виконання завдання мінімізації, залежно від числа станів часового автомату. Експеримент проводився для випадково згенерованих часових автоматів з десятьома вхідними й вихідними символами й двома часовими інтервалами виду [a, a+2); число станів змінювалося від 10 до 100. Для кожного значення числа станів генерувалася 1000 автоматів з тайм - аутами й часовими обмеженнями, для яких визначався середній час побудови мінімальному по станах і часу форми автомату. Відповідний графік показано на рис. 5.3.

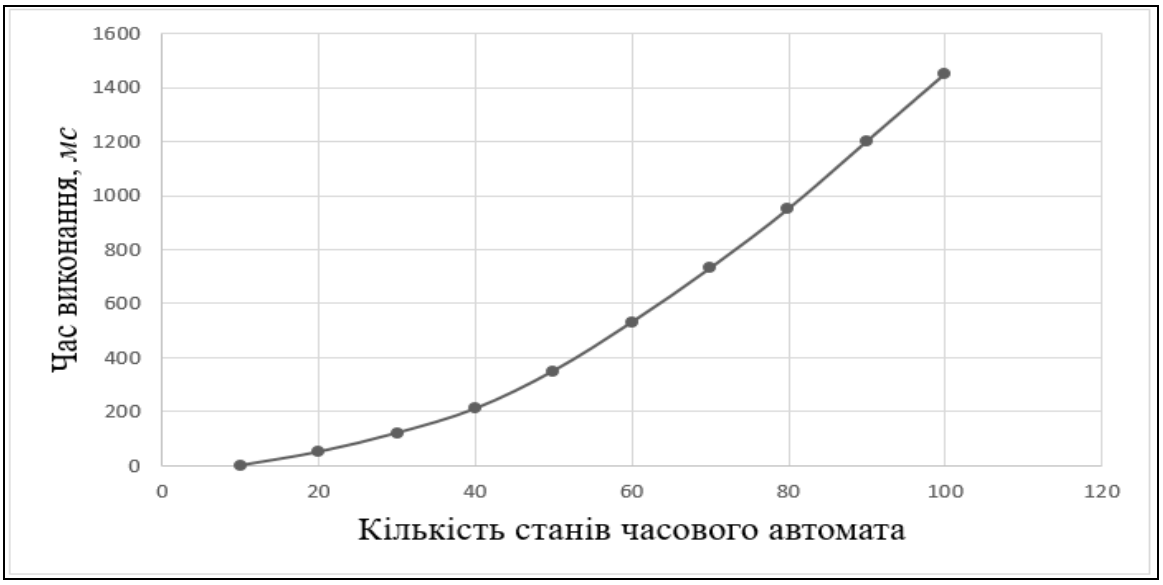

Рисунок 5.3 – Залежність часу виконання алгоритму мінімізації від числа станів часового автомату

Як видно з наведених результатів, час, протягом якого виконується алгоритм мінімізації для автоматів з тайм-аутами й часовими обмеженнями, зростає практично лінійний щодо числа станів автомату, коли число останнього більш 40 і границя B_S порівнянна із числом станів, незважаючи на те, що теоретична оцінка пропорційна добутку числа станів на величину B_S.

Для повністю визначеного спостережуваного, можливо, недетермінованого автомату S існує тестовий приклад, що розрізняє, якщо й тільки якщо у відповідному автоматі, що розрізняє, Sdist не існує повністю визначеного підавтомату. Довжина тестового прикладу, що розрізняє, може досягати значення $n^2 - n - 1$ для повністю визначеного недетермінованого спостережуваного автомата S з *n* станами. У цьому випадку, усі можливі підмножини станів автомата S, що не є одноелементними, можуть бути станами автомату, що розрізняє, Sdist, і відповідно, процедура побудови автомату, що розрізняє, вимагає експонентного числа кроків. З іншого боку, побудова тестового прикладу по автомату, що розрізняє, за допомогою ітеративного видалення станів з невизначеним вхідним символом не гарантує побудови тестового прикладу, що розрізняє, найменшої висоти. Для того щоб побудувати найкоротшу адаптивну послідовність, що розрізняє, якщо така існує, за оптимальної число кроків, в

результаті програмних експериментів пропонується обмежувати побудову автомата, що розрізняє, припустимою довжиною адаптивної послідовності.

Скорочений автомат, що розрізняє, Sldist містить лише стани, досяжні з початкового стану по вхідно-вихідним послідовностям довжини не більше деякого $L \geq 0$, і дозволяє знайти тестові приклади, що розрізняють, висоти не більше L. Запропонований підхід узгодиться з результатами, що довжина адаптивної послідовності, що розрізняє, не перевищує 15 для випадково згенерованих кінцевих автоматів із більшим (більш ніж 100) числом станів. Через те, що висота тестового прикладу, що розрізняє, для більшості автоматів не занадто велика, має сенс обмежити довжину, на якій здійснюється пошук тестового прикладу.

Таким чином, під автоматом Sldist розуміється автомат, що розрізняє, містить лише стани, досяжні з початкового стану по вхідно-вихідним послідовностям довжини не більше L, де L – ціле невід'ємне число, і стан F. Всі переходи S зі станів, що досяжні тільки по вхідній послідовності довжини L ведуть у стан F. Наприклад, для автомату на рисунку 5.3 автомат, що розрізняє, Sdist містить тільки початковий стан {0, 1, 2, 3} і стан F, у той час як S1dist містить ще чотири стани, що досяжні з початкового стану по вхідно-вихідної послідовності довжини 1.

Для повністю певного спостережуваного, можливо, недетермінованого кінцевого автомату S, існує тестовий приклад, що розрізняє, висоти не більше L, якщо й тільки якщо автомат, що розрізняє, Sldist не має повністю визначенного підавтомату.

ВИСНОВКИ

Робота присвячена рішенню завдань аналізу й синтезу для кінцевих автоматів з однієї часовою змінною. Розглянуті завдання синтезу спрямовані на оптимізацію опису системи у вигляді часового автомату, у той час як завдання аналізу представлене проблемами генерації й оптимізації тестів, що перевіряють, з гарантованою повнотою.

В результаті проведеного дослідження показано, що для виконання поставленої мети, необхідно розробити метод мінімізації часового автомату та дослідити, чи існує для часових автоматів єдина (з точністю до ізоморфізму) наведена форма.

Адаптувати операцію паралельної композиції класичних кінцевих автоматів для часових автоматів та дослідити замкнутість класу детермінованих часових автоматів щодо такої операції.

Адаптувати для часових автоматів автоматні методи синтезу тестів з гарантованою повнотою. Запропонувати метод оптимізації довжини тестів, що перевіряють, з гарантованою повнотою для повністю визначених недетермінованих, можливо, часових автоматів.

Показано, що паралельна композиція взаємодіючих часових детермінованих автоматів не завжди є детермінованим часовим автоматом. Виділені класи детермінованих часових автоматів, обмежені щодо операції паралельної композиції.

Для вирішення завдань аналізу й синтезу автоматів з однією часовою змінною використовується апарат дискретної математики, зокрема, апарат класичної теорії автоматів і математичної логіки.

Для аналізу ефективності запропонованих алгоритмів використовуються комп'ютерні експерименти.

Запропонована й досліджена операція паралельної композиції для системи взаємодіючих часових детермінованих автоматів, яка є розширенням такої

операції для класичних кінцевих автоматів. Показане, що на відміну від класичних кінцевих автоматів, клас детермінованих часових автоматів не є замкненим щодо цієї операції при наявності «повільної» зовнішнього середовища. Виділені класи детермінованих часових автоматів, замкнені щодо операції паралельної композиції.

Запропонований алгоритм синтезу тестів, що перевіряють, з гарантованою повнотою для недетермінованих часових автоматів щодо редукції; запропоновані алгоритми мінімізації адаптивних тестів, що перевіряють, можливо, зі зменшенням повноти тесту.

Мінімальна за часом і станам форма неініціальних часових автоматів може розглядатися як канонічна форма представлення системи, однак, якщо в автоматі виділений початковий стан, то його мінімальна форма не є єдиною. Останнє, зокрема, суттєво утрудняє визначення моделей несправності для часових автоматів, для яких існують методи синтезу тестів, що перевіряють, з гарантованою повнотою без перехід до звичайно автоматної абстракції, число станів якої може суттєво збільшитися в порівнянні з вихідним автоматом-специфікацією.

Більші тайм-аути в сервісних системах можуть викликати істотний ріст станів у відповідній абстракції. Таким чином, інтерес представляє визначення й побудова канонічної мінімальної форми для ініціальних часових автоматів.

У роботі розглянута лише одна операція композиції часових автоматів, а саме, бінарна паралельна композиція детермінованих повністю визначених автоматів з часовими обмеженнями. Становить інтерес дослідження цієї операції для автоматів, які мають і тайм-аути, і часові обмеження, як найбільш загальної моделі автомата з однієї часовою змінною.

Інші операції композиції, зокрема, операція синхронної композиції, яка в основному використовується при композиції апаратних засобів, залишилися за рамками даної роботи й вимагають додаткових досліджень. Також, становить інтерес оптимізація багатомодульних композицій та/або їх компонентів на основі рішення рівнянь для часових автоматів.

Запропоновані методи синтезу тестів для детермінованих і недетермінованих часових автоматів досить часто «повертають» занадто довгі тести, що перевіряють, і, відповідно, питання розробки нових моделей несправності й методів синтезу тестів з гарантованої повнотою щодо цих моделей для систем реального часу, залишаються актуальними. Цікавим завданням також є адаптація запропонованих моделей несправності й підходів до синтезу тестів для інших некласичних автоматних моделей, наприклад, розширених автоматів або вхідно-вихідних напівавтоматів з кінцевим числом часових змін.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Агибалов Г. П. Лекции по теории конечных автоматов / Г. П. Агибалов, А. М. Оранов. – Издательство ТГУ, 1984. – 185 с.

2. Бурдонов И. Б. Теория соответствия для систем с блокировками и разрушениями / И. Б. Бурдонов, А. С. Косачев, В. В. Кулямин. – М. : ФИЗМАТЛИТ. – 2008. – 412 с.

3. Василевский М. П. О распознавании неисправности автоматов / М. П. Василевский // Кибернетика. – 1973. – № 4. – С. 93–108.

4. Гилл А. Введение в теорию конечных автоматов / А. Гилл. – М. : Издательство Наука, 1966. – 272 с.

5. Евтушенко Н. В. Метод построения эксперимента для произвольного детерминированного автомата / Н. В. Евтушенко, А. Ф. Петренко // Автоматика и вычислительная техника. – 1990. – № 5. – С. 73–76.

6. Твардовский А. К минимизации автоматов с таймаутами / А. С. Твардовский // Труды Института системного программирования РАН. – Т. 26, вып. 6. – 2014. – С. 77–84.

7. Software testing / G. V. Bochmann, A. Petrenko // Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis. – USA, 2014. – P. 109–124.

8. Bresolin D. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power / D. Bresolin, K. El-Fakih, T. Villa, N. Yevtushenko // Proceedings of the Fifth International Symposium on Games, Automata, Logics and Formal Verification (GANDALF 2014). EPTCS. – 2018. – Vol. 161. – P. 203–216.

9. Cavalli A. R. Testing Methods for SDL Systems / A. R. Cavalli, B.-M. Chin, K. Chon // Computer Networks and ISDN Systems. – 1996. – Vol. 28, is. 12. – P. 1669–1683.

10. Cavalli A. Fault detection within a component of a system communicating FSMs / A. Cavalli, S. Prokopenko, N. Yevtushenko // Proceedings of the 14th

International conference TestCom. – 2012. – P. 317–332.

11. Кнут Д. Искусство программирования. Т. 1 / Д. Кнут. – М. : Мир, 1976. – 734 с.

12. Кондратьева О.В. Параллельная композиция конечных автоматов с таймаутами / О. В. Кондратьева, Н. В. Евтушенко, А. Р. Кавалли // Управление, вычислительная техника и информатика. – 2014. – № 2. – С. 73–81.

13. Alam M. G. R., Tun Y. K., Hong C. S. Multi-agent and reinforcement learning based code offloading in mobile fog // 2016 International Conference on Information Networking (ICOIN), 2016. IEEE, 2016. P. 285–290.

14. Bonomi F. et al Fog Computing: A Platform for Internet of Things and Analytics 2014. 169–186 p.

15. Bonomi F. et al Fog computing and its role in the internet of things // Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC ’12, 2012, New York, New York, USA. ACM Press, 2012. P. 13.

16. Fielding R. T., Taylor R. N. Principled design of the modern Web architecture // ACM Transactions on Internet Technology (TOIT). 2002. Vol. 2, No 2. P. 115–150.

17. Hu P. et al Survey on fog computing: architecture, key technologies, applications and open issues // Journal of Network and Computer Applications. 2017. Vol. 98. P. 27–42.

18. Ionescu V. M. The analysis of the performance of Rabbitmq and Activemq // 2015 14th Roedunet International Conference - Networking in Education and Research (Roedunet NER), 2015. IEEE, 2015. P. 132–137.

19. Iorga M. et al Fog computing conceptual model. Gaithersburg, MD, 2018.

20. Karakaya Z., Yazici A., Alayyoub M. A Comparison of Stream Processing Frameworks // 2017 International Conference on Computer and Applications (ICCA), 2017. IEEE, 2017. P. 1–12.

21. Mebrek A., Merghem-Boulahia L., Esseghir M. Efficient green solution for a balanced energy consumption and delay in the Iot-fog-cloud computing // 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), 2017.

IEEE, 2017. P. 1–4.

22. Openfog Consortium Architecture Working Group Openfog Reference Architecture for Fog Computing // Openfog. 2017. P. 162.

23. Rabay'a A., Schleicher E., Graffi K. Fog Computing with P2P: Enhancing Fog Computing Bandwidth for Iot Scenarios // 2019 International Conference on Internet of Things (ithings) and IEEE Green Computing and Communications (Greencom) and IEEE Cyber, Physical and Social Computing (Cpscom) and IEEE Smart Data (Smartdata), 2019. IEEE, 2019. P. 82–89.

24. Shi W. et al Edge Computing: Vision and Challenges // IEEE Internet of Things Journal. 2016. Vol. 3, No 5. P. 637–646.

25. Empowering App Development for Developers | Docker. URL: https://www.docker.com/.

26. Production-Grade Container Orchestration - Kubernetes. URL: https://kubernetes.io/

27. Protocol Buffers | Google Developers. URL: https://developers.google.com/protocol-buffers.

28. Swarm mode overview | Docker Documentation. URL: https://docs.docker.com/engine/swarm/.

29. The most popular database for modern apps | Mongodb. URL: https://www.mongodb.com/ .

30. Welcome to Python.org. URL: https://www.python.org/

31. Дудар З.В., Склярук Д.О. Дослідження алгоритмів підвищення надійності відеозв'язку / Информационные системы и технологии (ИСТ-2018): материалы 7-й Международ. Науч.-техн. конф., Харьков-Коблево, 10-15 сентября 2018, – тезисы докладов. – Х.: ХНУРЕ, 2018

32. Shubin, I., Snisar, S., Zhyrnov, V., Slavhorodskyi, V.// Practical Application of Formal Representation of Information for Intelligent Radar Systems 2018 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2018 - Proceedings, 2019, pp. 433-436, 8632103