

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИНТЕЗА СПЕЦИАЛЬНЫХ
ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ КОНЕЧНОГО
ДЕТЕРМИНИРОВАННОГО АВТОМАТА
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Ишкенова Егора Сергеевича

Научный руководитель
доцент, к. ф.-м. н.

О. В. Мещерякова

Заведующий кафедрой
доцент, к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2020

ВВЕДЕНИЕ

Формализация различных ситуаций функционирования технических и экономических объектов различной природы (отрасль промышленности, интегрированная структура, предприятие, вычислительная система) особенно необходима при исследовании процессов реструктуризации этих объектов. Для этого требуется формально определить понятия объекта, состояния объекта, близости состояний и некоторые другие. Состояние объекта в определенный момент времени характеризуется совокупностью показателей, индикаторов и характеристик различных сторон его деятельности. В зависимости от целей моделирования и анализа состояние объекта можно формально описать вектором показателей различной размерности от двух – трех до нескольких десятков. Функционирование объекта — это его переход из одного состояния в другое. При нормальной работе объекта переход в другое состояние плавен (эволюционен), но в ряде случаев происходит реструктуризация объекта – резкое изменение состояния объекта с изменением его внутренней структуры. Если условиться понимать состояния объекта как элементы некоего множества состояний (множества вершин ориентированного графа состояний), а переход из состояния в состояние – условиться считать ребрами этого ориентированного графа, то получаем простейшую модель функционирования объекта в виде прокладывания определенного пути по ориентированному графу состояний. Характеристиками ребра в таком случае могут быть затраты ресурсов на переход из одного состояния в другое или другие аддитивные показатели, характеризующие этот переход (например, выпуск продукции и прибыль, энергетические затраты). Ориентированный граф состояний конечен, в нем есть недопустимые состояния. Очевидно также, что невозможно попасть из любого состояния в любое, то есть граф незамкнут. Анализируя графическую модель перехода объекта из состояния в состояние, легко увидеть, что она почти полностью адекватна другой дискретной математической модели конечного автомата, использующей аналогичные понятия состояния, входа, выхода и функции перехода [1, 2].

Целью данной работы является программная реализация синтеза входных последовательностей специального вида для конечного детерминированного автомата.

КРАТКОЕ СОДЕРЖАНИЕ

Первый параграф дипломной работы посвящен рассмотрению основных понятий теории автоматов. Параграф содержит 6 подразделов, в первом из которых введено понятие автомата, различия автомата Мили и Мура.

Модель Мили представляет собой следующие законы:

$$s(t + 1) = \delta(s(t), x(t))$$

$$y(t) = (s(t), x(t))$$

- t – текущий момент времени;
- $t+1$ - следующий момент времени;
- $s(t+1)$ – состояние автомата в следующий момент времени;
- $s(t), x(t), y(t)$ – элементы описания автомата в текущий момент времени.

Модель Мура определяется следующими законами:

$$s(t + 1) = \delta(s(t), x(t))$$

$$y(t) = (s(t))$$

В модели Мура выходной сигнал явно зависит только от состояния, а косвенно – и от входного сигнала. Любой автомат можно спроектировать по той или иной модели. В работе в качестве основной модели выбран автомат Мили. Во втором подразделе приведены виды автоматов, такие как: конечные и бесконечные, детерминированные и недетерминированные. В работе в качестве основной модели выбран конечный детерминированный автомат. В третьем подразделе приведены способы задания автоматов, такие как, табличный и графический.

		δ_v				λ_{v+1}			
		x_1	x_2	...	x_n	y_1	y_2	...	y_m
s_v	x_v								
s_1		В клетках таблицы помещаются значения из множества $\{s_1, s_2, s_3, \dots, s_k\}$				В клетках таблицы помещаются значения из множества $\{y_1, y_2, \dots, y_m\}$			
s_2									
s_3									
.									
s_k									

Рисунок 1 – Пример таблица переходов-выходов

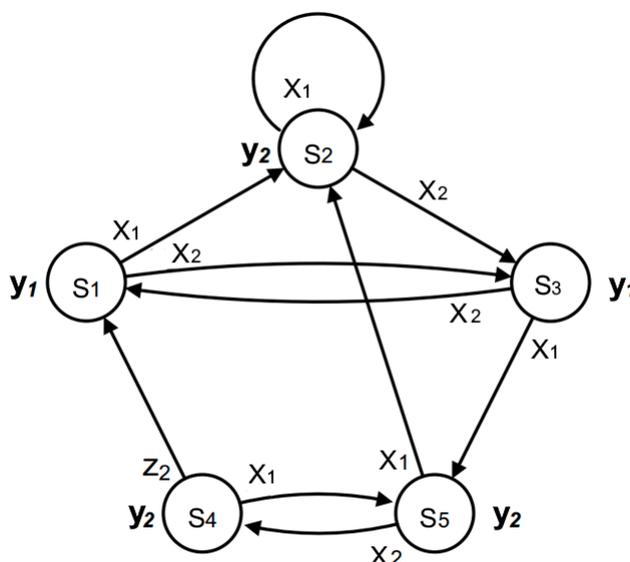


Рисунок 2 – Пример графа автомата

Четвертый подраздел содержит описание и процедуру построения структуры, называемой деревом приемников для автомата. Построение дерева приемников потребовало введения определений некоторых понятий.

Определение

σ -множество автомата – это любая конечная совокупность состояний автомата. σ -множество является *простым*, если состоит из одного элемента. σ -множество называется *кратным*, если содержит два или более одинаковых элементов. σ -множество называется *однородным*, если все его элементы совпадают друг с другом.

Определение

A-группа автомата – это множество, состоящее из σ -множеств, общее число элементов которой равно $l=|S_0|$. *A-группа* называется *простой* (однородной), если все σ -множества в ней просты (однородны).

Дерево состоит из ветвей, расположенных в последовательных уровнях, нумерация которых начинается с нуля. Нулевой уровень дерева содержит начальную ветвь, которая связана с *A-группой* S_0 , где S_0 - множество начальных состояний автомата. Если входной алфавит $X \{ x_1, x_2, \dots, x_n \}$ автомата *A* содержит n символов, то каждая ветвь в k -м уровне расщепляется на n ветвей, представляющих символы x_1, x_2, \dots, x_n соответственно и являющихся ветвями $(k+1)$ -го уровня. В пятом подразделе приведены основные понятия теории экспериментов с конечными автоматами. Поскольку целью работы стала программная реализация классических алгоритмов синтеза последовательностей специального вида, то возникла необходимость оценки алгоритмической сложности. Понятие алгоритмической сложности приведено в шестом подразделе. Второй параграф дипломной работы посвящен задачам построения последовательностей специального вида, таких как: установочная, диагностическая и синхронизирующая. Параграф содержит 3 подраздела. В первом подразделе введены понятия установочной последовательности и установочного эксперимента.

Последовательность $p=x_{i_1}, x_{i_2}, \dots, x_{i_n}$ называется *установочной* для автомата *A* и с множеством начальных состояний S_0 , если

$$\forall s_1, s_2 \in S_0, (s_1, p) = (s_2, p) \rightarrow (s_1, p) = (s_2, p).$$

Смысл определения в том, что по наблюдаемой реакции на установочную последовательность автомата *A*, находящегося в неизвестном состоянии для наблюдателя начальном состоянии, всегда однозначно определяется его конечное состояние. Эксперимент, связанный с подачей установочной последовательности называется установочным.

Для реализации эксперимента построено установочное дерево, как дерево преемников, описаны правила обрыва ветвей. Ветвь становится конечной если:

1. она является однородной;
2. она является простой;

3. на уровнях, предшествующих k-му, имеется вершина S^* такая, что $S = S^*$ но значение флажка вершины S больше значения флажка вершины S^* ;
4. на уровнях, предшествующих k-му, имеется однородная или простая вершина S^* , значение флажка которой меньше значения флажка вершины S^* .

И для произвольного автомата в приложении А приведена программная реализация алгоритма построения установочной последовательности на языке Python. В втором подразделе введены понятия диагностической последовательности и диагностического эксперимента.

Последовательность $p = x_{i_1}, x_{i_2}, \dots, x_{i_n}$ называется *диагностической* для автомата А и с множеством начальных состояний S_0 , если

$$\forall s_1, s_2 \in S_0, \lambda(s_1, p) = \lambda(s_2, p) \rightarrow s_1 = s_2.$$

Смысл определения в том, что по наблюдаемой реакции на диагностическую последовательность всегда однозначно идентифицируется его неизвестное начальное состояние. Эксперимент, связанный с подачей диагностической последовательности называется *диагностическим*.

Эксперимент начинается с построения диагностического дерева, как дерева преемников, и описаны правила обрыва ветвей. Ветвь становится оконечной если:

1. она является простой;
2. она является кратной;
3. на уровнях, предшествующих k-му, имеется вершина S^* такая, что $S = S^*$ но значение флажка вершины S больше значения флажка вершины S^* ;
4. на уровнях, предшествующих k-му, имеется простая вершина S^* , значение флажка которой меньше значения флажка вершины S.

На рисунке показан пример диагностического дерева.

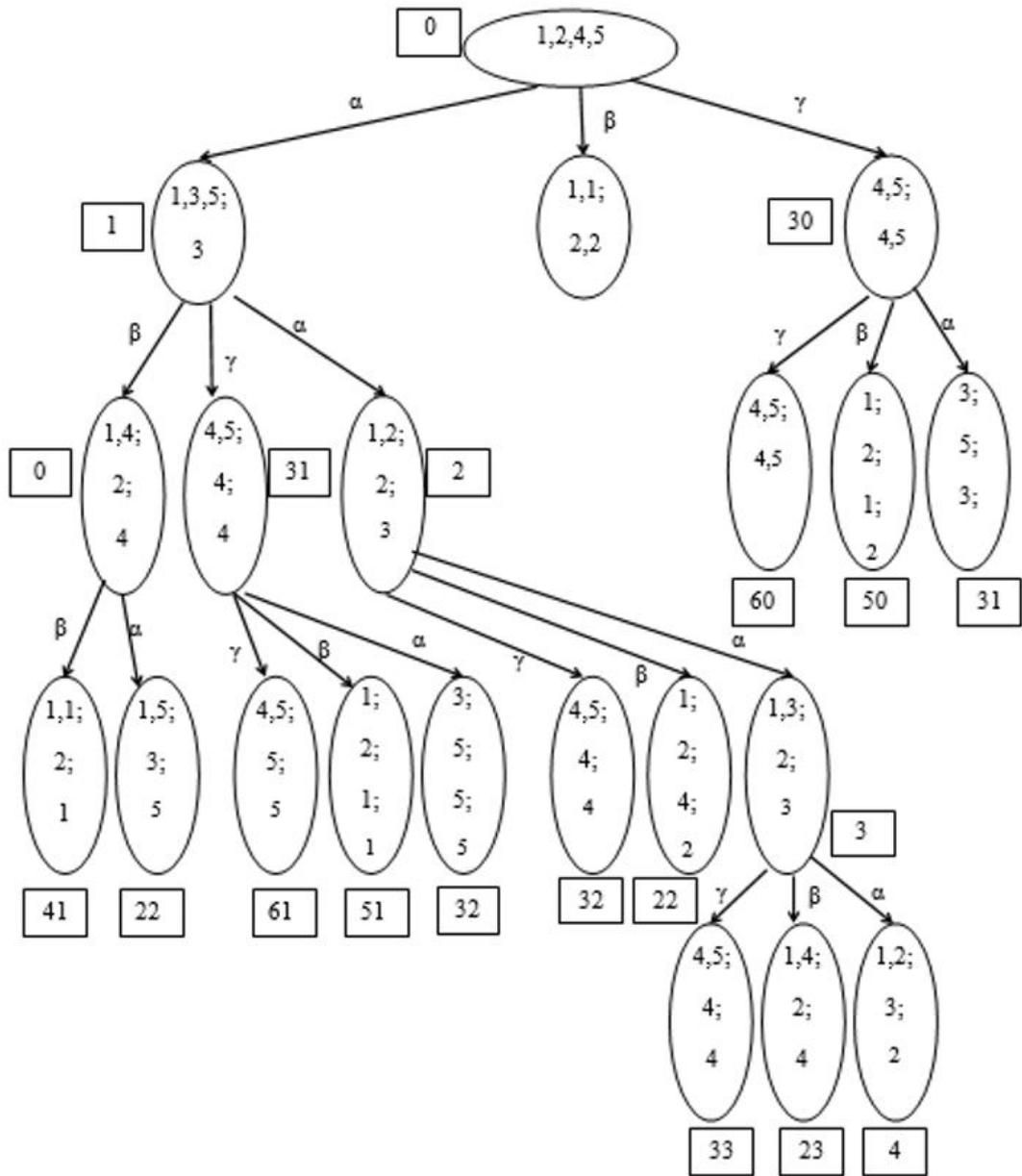


Рисунок 3 – Диагностическое дерево автомата А

И для произвольного автомата в приложении А приведена программная реализация алгоритма построения диагностической последовательности на языке Python. В третьем подразделе введены понятия синхронизирующей последовательности и синхронизирующего эксперимента.

Последовательность $p = x_{i_1}, x_{i_2}, \dots, x_{i_n}$ называется *синхронизирующей* для автомата А и с множеством начальных состояний S_0 , если

$$\forall s_1, s_2 \in S_0, \delta(s_1, p) = \delta(s_2, p).$$

Смысл определения в том, что независимо от начального состояния ав-

томата после подачи на его вход синхронизирующей последовательности он оказывается в одном и том же конечном состоянии. Эксперимент, связанный с подачей синхронизирующей последовательности называется синхронизирующим. Отметим, что синхронизирующая последовательность является частным случаем установочной, когда конечное состояние автомата идентифицируется без наблюдения его выходной реакции. Синхронизирующее дерево, как дерево преемников, строится по аналогии с установочным и диагностическим деревьями. Для синхронизирующего дерева введены правила обрыва ветвей. Ветвь становится оконечной если:

1. $|S| = 1$;
2. на уровнях, предшествующих k -му, имеется вершина S^* такая, что $S=S^*$.
3. на уровнях, предшествующих k -му, имеется такая вершина S^* , что $|S^*| = 1$, а значение флажка вершины S больше значения флажка вершины S^* .

В приложении Б приведен код программной реализации построения синхронизирующей последовательности на языке C++. В каждом подразделе второго параграфа описаны особенности и условия алгоритмов и оценена алгоритмическая сложность.

ЗАКЛЮЧЕНИЕ

К числу классических задач теории автоматов относятся задачи изучения их свойств, задачи анализа и синтеза, задачи минимизации и т.п. Особый класс в теории автоматов составляют задачи распознавания, расшифровки, идентификации, установки, контроля и диагностики, которые требуют для своего решения проведения экспериментов с автоматами. Чрезвычайно широко применение результатов и методов теории экспериментов в области технической диагностики дискретных устройств. Одной из важных проблем классической теории экспериментов с автоматами является разработка методов построения различных типов входных последовательностей, с помощью которых осуществляется эксперимент. В работе использовалась математическая модель, которая называется автомат Мили, как модель, с помощью которой описывается функционирование любого преобразователя информации. Для автомата Мили в классической теории разработаны методы и алгоритмы построения специальных последовательностей, у которых каждый символ имеет вес, равный единице, и оптимальными входными последовательностями являются те, которые имеют минимальную длину. В работе исследован автомат, для которого каждый входной символ имеет свой вес, и тогда классические методы построения оптимальных входных последовательностей оказываются непригодными. Поэтому возникает необходимость разработки новых методов построения оптимальных экспериментов. В предложенном исследовании приведены способы построения дерева преемников и его модификации, сформулированы правила выделения конечной структуры из бесконечного дерева преемников, что позволило по-другому интерпретировать понятие оптимального эксперимента. Реализация методов построения установочной, диагностической и синхронизирующей последовательностей связана с специфическими особенностями частных классов преобразователей информации, и поэтому методы, применяемые к общей модели (автомат Мили) могут оказаться достаточно эффективными. В работе предложена программная реализация построения синтеза входных последовательностей специального вида для конечного детерминированного автомата со взвешенным входным алфавитом, оценена алгоритмическая сложность.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Яблонский С. В. Основные понятия кибернетики – В кн. Проблемы Кибернетики – М. ; Физматгиз. 1959, вып. 2, с 23.
- 2 Гилл А. Введение в теорию конечных автоматов. М.:Наука, 1964
- 3 Глушков В. М. Абстрактная теория автоматов // Успехи матем. наук. Т. 16, вып. 5 (101). 1961. с.3-62.
- 4 Глушков В. М. Синтез цифровых автоматов – М.: Физматгиз. 1962
- 5 Лупанов О. Б. О синтезе некоторых классов управляющих систем. – в кн. Проблемы кибернетики. – М.: физматгиз. 1963, выпуск 10.
- 6 Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. - М.: Наука, 1985. - 320 с.
- 7 Оре О. Теория графов. М. Наука. 1968.
- 8 Зыков А. А. Теория конечных графов. 1. – Новосибирск: Наука. 1969.
- 9 Мур Э. Умозрительные эксперименты с последовательностными машинами // Автоматы: Сб. науч. тр. М.; Л., 1956. с. 179—210. Агибалов Г. П., Юфат Я. Г. О простых экспериментах для линейных инициальных автоматов // Автоматика и вычислительная техника. № 2. 1972. с. 17—19.
- 10 Богомолов А. М., Барашко А. С., Грунский И. С. Эксперименты с автоматами. — Киев: Наукова думка, 1973. — 144 с.
- 11 Твердохлебов В. В. Логические эксперименты с автоматами. – Саратов: издательство Саратовского ун-та, 1988.
- 12 Барашко А. С., Скобцов Ю. А., Сперанский Д. В. Моделирование и тестирование дискретных устройств. — Киев: Наукова думка, 1992. -288 с.
- 13 Спивак М. А. Введение в абстрактную теорию автоматов. – Саратов: Изд-во Саратов. ун-та 1970.
- 14 Спивак М. А. Обобщенные задачи диагноза и установки для конечных автоматов // Известия АН СССР, Технич. Киб-ка. №3. 1969. С. 82-87.
- 15 Богомолов А. М., Грунский И. С., Сперанский Д. В. Контроль и преобразования дискретных автоматов. — Киев: Наукова думка, 1975. — 176 с.

- 16 Грунский И. С. Анализ поведения конечных автоматов. – Луганск : Изд-во Луганск. гос. пед. Ун-та, 2003
- 17 Breuer. M., Friedman A. Diagnosis and reliable design of digital systems – Computer Science Press. – 300 p.
- 18 Yannakakis M., Lee D. Principles and methods of testing finite-state machines/ - A survey / Proc. IEEE. V. 84. №8. 1996. P. 1090-1123.
- 19 Петренко Л.С. Эксперименты над протокольными объектами // Автоматика и вычислительная техника. №1 1987
- 20 Агафонов В.Н. Спецификация программ: понятийные средства и их организация