

УДК 519.713

ОБЗОР МЕТОДОВ АВТОМАТНОГО ПРОГРАММИРОВАНИЯ

А.И. Мартышкин¹, Д.А. Трокоз²

кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Alexey314@yandex.ru

кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Dmitriy.trokoz@gmail.com

Аннотация. В статье приведены теоретические сведения о теории автоматного программирования. Дается определение конечного автомата, детерминированного конечного автомата и недетерминированного конечного автомата. Проведен сравнительный анализ детерминированного конечного автомата и недетерминированного конечного автомата, на основании чего сделан вывод о параллелизме недетерминированного конечного автомата. Описываются основные способы задания и моделирования недетерминированного конечного автомата.

Ключевые слова: теория автоматов, конечный автомат, детерминированный автомат, недетерминированный автомат, параллелизм, способы задания автоматов.

OVERVIEW OF AUTOMATIC PROGRAMMING METHODS

A.I. Martyshkin¹, D.A. Trokoz²

¹Ph.D., Associate Professor, Associate Professor of the Department of Computers and Systems, FGBOU VO 'Penza State Technological University', Penza, Russia, e-mail: Alexey314@yandex.ru

²Ph.D., Associate Professor, Associate Professor of the Department of Computers and Systems, FGBOU VO 'Penza State Technological University', Penza, Russia, e-mail: Dmitriy.trokoz@gmail.com

Abstract. The article provides theoretical information about the theory of automatic programming. We define a finite automaton, a deterministic finite automaton, and a nondeterministic finite automaton. Comparative analysis of deterministic finite state machine and nondeterministic finite automata, based on what conclusion is made about concurrency automaton. The main methods of defining and modeling a nondeterministic finite state machine are described.

Keywords: automata theory, finite automata, deterministic automata, nondeterministic automata, parallelism, methods for specifying automata.

Введение. Термин «автомат» имеет греческое происхождение. Автомат – абстрактное понятие, описывающееся как некое вычислительное

устройство, или машина, автоматически выполняющая заданную последовательность операций. Автомат с конечным числом состояний называется конечным автоматом. В зарубежной литературе встречаются два термина для обозначения конечного автомата: Finite Automaton и Finite State Machine [1].

Цель работы. В статье приведены сведения о теории автоматного программирования. Цель работы состоит в проведении сравнительного анализа детерминированного конечного автомата (ДКА) и недетерминированного конечного автомата (НКА).

Материал и результаты исследований. Формальное определение конечного автомата. Для формального описания автомата пользуются так называемой пятеркой $(Q, \Sigma, \delta, q_0, F)$ компонентов, где первые три компонента – Q, Σ, F – множества [2]. Q содержит множество состояний автомата. Σ содержит множество входных символов, или алфавит автомата. F представляет собой множество принимающих состояний автомата. Если каждое из этих множеств конечно, такой автомат – конечный [3]. Следующие два компонента – δ и q_0 . Символом δ обозначают функцию переходов, а q_0 – начальное состояние автомата, причем $q_0 \in Q$ [4].

Недетерминированный конечный автомат. В зарубежных источниках упоминается Non-deterministic Finite Machine (NFM) или Non-deterministic Finite Automaton (NFA) [5]. Во многом определение НКА схоже с формулировкой ДКА. Для сравнения сопоставим ДКА и НКА в таблице 1.

Таблица 1 – Сопоставление ДКА и НКА

	ДКА	НКА
Множество состояний	Q	Q
Алфавит (множество символов)	Σ	Σ
Начальное состояние	$q_0, (q_0 \in Q)$	$q_0, (q_0 \in Q)$
Множество принимающих состояний	$F, (F \subseteq Q)$	$F, (F \subseteq Q)$
Функция переходов	$\delta: Q \times \Sigma \rightarrow Q$	$\delta: Q \times \Sigma \rightarrow 2Q$

Формальное определение ДКА [6]:

Q – конечное множество всех состояний автомата.

Σ – конечное множество символов (алфавит).

δ – функция перехода, где $\delta: Q \times \Sigma \rightarrow Q$.

q_0 – начальное состояние автомата ($q_0 \in Q$).

F – множество конечных состояний, причем $F \subseteq Q$.

Формальное определение НКА:

Q – конечное множество всех состояний автомата.

Σ – конечное множество символов (алфавит).

δ – функция перехода, где $\delta: Q \times \Sigma \rightarrow 2Q$.

q_0 – начальное состояние автомата ($q_0 \in Q$).

F – множество конечных состояний, причем $F \subseteq Q$.

Из таблицы понятно, что оба автомата имеют аналогичное описание, за исключением функции переходов [7]. Согласно определению ДКА, функция переходов ДКА ставит в соответствие паре $\{qi, sj\}$ только одно состояние, тогда как в НКА это условие нарушается и паре $\{qi, sj\}$ может соответствовать множество состояний либо оно будет пустым.

Отличительной особенностью НКА является то, что переход из одного состояния в другое определен не однозначно. Это значит, что при возникновении ситуации альтернативных переходов, автомат совершит переход одновременно в несколько состояний при заданных условиях. В этом и заключается способность автомата к параллельной обработке процессов [8].

Таким образом, НКА – обобщение ДКА, т.к. может находиться сразу в нескольких состояниях одновременно [9]. Анализируя вышесказанное, приведем итоговую сравнительную таблицу 2 [10].

Таблица 2 – Сравнение ДКА и НКА

ДКА	НКА
Если из одного состояния по каждому входному символу возможен только один переход в следующее состояние, такой автомат называется детерминированным	Если из одного состояния возможно сделать несколько переходов, то такой автомат называется недетерминированным
ДКА не допускает пустых переходов	НКА допускает пустые переходы
Есть поиск с возвратом	Нет поиска с возвратом
Занимает больше памяти	Занимает меньше памяти
Если ДКА после чтения входной цепочки оказался в допускающем состоянии, то он принимает ее, иначе – нет	Если НКА после чтения входной цепочки из всего множества состояний оказался хотя бы в одном допускающем, то он принимает ее

Моделирование НКА. НКА – простейшая модель дискретно-событийной модели. Чтобы представить НКА в виде модели необходимо задать его начальное и конечное состояние, определить основные события и условия для перехода из одного состояния в другое. Моделирование НКА, как правило, происходит двумя способами: с помощью таблицы переходов; с помощью диаграммы/графа переходов [11].

Для примера построим модели НКА, основываясь на этих методах.

Представим некоторую последовательность символов входной цепочки, которую должен прочитать НКА. В качестве такой последовательности возьмем цепочку символов {a, a, c, b, c, b, c}, которая, согласно определению НКА, является алфавитом Σ . На рисунке 1 изображена последовательность символов в виде входной цепочки [12].

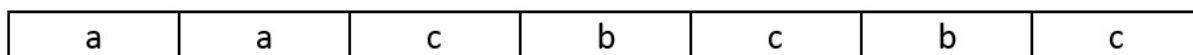


Рисунок 1 – входная цепочка

Начальное состояние НКА $q_0 = a$, конечное состояние $F = c$. Опираясь на эти данные, получаем изображение НКА в виде графа (рисунок 2).

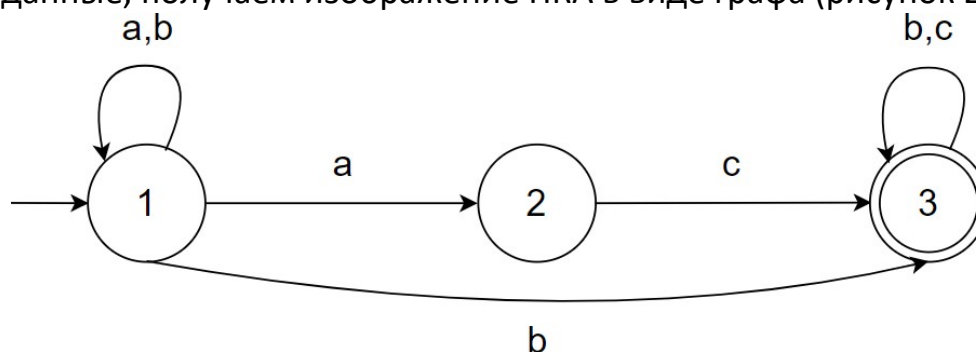


Рисунок 2 – модель НКА в виде графа

В соответствии с правилами моделирования графов кружками отмечены состояния автомата, принимающее состояние отмечено двойным кружком. Каждое состояние пронумеровано внутри кружка. Данная модель НКА имеет три состояния. Дуги, изображенные в виде стрелок, отображают переходы между состояниями, а также циклические переходы, т.е. переход в то же самое состояние. Циклических переходов на рисунке два. Входящая слева стрелка указывает начальное состояние автомата. Каждая дуга-переход сверху имеет обозначение символа, по которому совершается переход из одного состояния в другое.

Таблица переходов – метод, основанный на функции переходов $\delta = \{qi, sj\}$, где qi – состояние, sj – входной символ. Следовательно, каждое состояние данного автомата НКА можно описать следующим образом:

$\delta(1, a) = \{1,2\}$	$\delta(1, b) = \{1,3\}$	$\delta(1, c) = \emptyset$
$\delta(2, a) = \emptyset$	$\delta(2, b) = \emptyset$	$\delta(2, c) = \{3\}$
$\delta(3, a) = \emptyset$	$\delta(3, b) = \{3\}$	$\delta(3, c) = \{3\}$

Преобразуем в таблицу переходов (таблица 3).

Таблица 3 – Таблица переходов НКА

sj	a	b	c
------	---	---	---

qi			
1	1,2	1,3	-
2	-	-	3
3	-	3	3

Выводы. В статье рассмотрены основные понятия теории конечных автоматов. Определение недетерминированного конечного автомата выводится через детерминированный конечный автомат с уточнением их различия в трактовке функции переходов. Также приводятся описания двух способов моделирования конечных автоматов и их модели.

Статья публикуется при поддержке гранта РФФИ «Конкурс на лучшие проекты фундаментальных научных исследований» (Грант № 19-07-00516 А).

ЛИТЕРАТУРА

1. Byoung Kyu Choi, DongHun Kang John, Modeling and Simulation of Discrete Event Systems // Wiley & Sons, 2013. – 432 с.
2. Бикташев Р.А., Вашкевич Н.П. Модели событийных недетерминированных автоматов для формального представления основных свойств систем управления параллельными процессами и ресурсами // Инфокоммуникационные технологии. – 2013. – Т. 11. – № 3. – С. 95-98.
3. Вашкевич Н.П., Бикташев Р.А. Недетерминированные автоматы и их использование для реализации систем параллельной обработки информации: Монография – Пенза: Ид-Во Пгу, 2016, – 394 С.
4. Вашкевич Н.П., Бикташев Р.А. Достоинство формального языка, основанного на концепции недетерминизма, при структурной реализации параллельных Систем логического управления процессами и ресурсами // Известия высших учебных заведений. Поволжский регион. Технические науки. –2011. – № 1. – С. 3-11.
5. Вашкевич Н.П., Бикташев Р.А., Пашенко Д.В., Кутузов В.В., Сауанова К.Т. Использование моделей событийных недетерминированных автоматов для формального описания параллельных алгоритмов логического управления // Вестник Нан Рк, 2015, № 4, С. 48-63.
6. Волчихин В.И., Вашкевич Н.П., Бикташев Р.А. Модели событийных недетерминированных автоматов представления алгоритмов управления взаимодействующими процессами в многопроцессорных вычислительных системах на основе использования механизма монитора // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2013. –№ 2 (26). – С. 5-14
7. Дубинин В.Н., Будаговский Д.А., Дроздов Д.Н., Артамонов Д.В. Проектирование и реализация систем управления дискретными событийными системами на основе иерархических модульных недетерминированных автоматов (Ч. 2. Методы и средства) Известия высших учебных заведений. Поволжский регион. Технические науки. – 2016. – № 2 (38). – С. 18-32.

8. Дубинин В.Н., Дроздов Д.Н. Проектирование и реализация систем управления дискретными событийными системами на основе иерархических модульных недетерминированных автоматов (Ч. 1. Формальная модель). Известия высших учебных заведений. Поволжский регион. Технические науки. – 2016. – № 1 (37). – С. 28-39.

9. Моделирование распределенных многокомпонентных программных систем и их тестирование на основе автоматных вероятностных моделей /С.М. Старолетов. – Барнаул: Изд-во АлтГТУ, 2011. – 107 с.

10. Мартин Фаулер Предметно-ориентированные языки программирования: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 576 с.

11. Н.П.Вашкевич Недетерминированные автоматы в проектировании систем параллельной обработки [Текст]/Н.П.Вашкевич: учебное пособие. – Пенза: Изд-во Пенз.гос.ун-та, 2004. – 280 с.

12. Мозговой М.В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход. – СПб.: Наука и Техника, 2006. – 320 с.

УДК 004.451

АНАЛИЗ НАИБОЛЕЕ ПЕРСПЕКТИВНЫХ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

А.И. Мартышкин

кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Alexey314@yandex.ru

Аннотация. В статье рассмотрены основные принципы работы алгоритмов планирования операционных систем реального времени. Выявлено, что задачи в операционных системах реального времени делятся на периодические и аperiodические и для их планирования используются статические и динамические алгоритмы планирования.

Ключевые слова: операционная система, алгоритмы планирования, планировщик, приоритет, эффективность, производительность, система реального времени.

ANALYSIS OF THE MOST PROMISING ALGORITHMS FOR PLANNING REAL-TIME OPERATING SYSTEMS

A.I. Martyshkin

Ph.D., Associate Professor, Associate Professor of the Department of Computers and Systems, FGBOU VO 'Penza State Technological University', Penza, Russia, e-mail: Alexey314@yandex.ru

Abstract. The article describes the basic principles of real-time operating system planning algorithms. It is revealed that tasks in real-time operating systems are divided into periodic and aperiodic, and static and dynamic scheduling algorithms are used for their planning.