



Ссылка на статью:

// Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн. 2018, № 2, с. 24-27.

Поступила: 01.11.2018

Окончательный вариант: 18.12.2018

© УлГУ

УДК 519.713:004.42

## Детерминированные и недетерминированные конечные автоматы в современном программировании

Жаркова Г.А.<sup>1,\*</sup>, Лаптев К.Н.<sup>2</sup>

\* [zharkovaga@inbox.ru](mailto:zharkovaga@inbox.ru)

<sup>1</sup>УлГУ, Ульяновск, Россия

---

В статье описываются детерминированные, недетерминированные и вероятностные конечные автоматы. Разбиение исполнения программы на шаги автомата и передача информации от шага к шагу через состояние необходимо при построении событийно-ориентированных приложений. Применение вероятностных автоматов затрагивает такие области, как создание систем искусственного интеллекта, работа с нечеткими множествами, нечеткая логика, распознавание сложных образов и др.

**Ключевые слова:** детерминированные и недетерминированные автоматы, вероятностные автоматы, распознавание образов.

---

На протяжении нескольких десятков лет вычислительные системы успешно и активно развиваются. Теория больших систем, изучение функционирования биологических структур, исследование надежности систем, строящихся из большого числа элементов, поиски методов управления объектами, привели к развитию раздела теоретической кибернетики — теории вероятностных автоматов.

Многие части электронно-вычислительной машины, такие как аппаратное и программное обеспечение строятся на основе модели конечного автомата. Конечный автомат в программировании можно представить в виде «чёрного ящика», на входе которого имеется сигнал входного алфавита, на выходе – сигнал выходного алфавита, и сам он находится в одном из состояний, число которых конечно. Результат работы автомата определяется по его финальному состоянию.

Конечные автоматы применяются для моделирования поведения системы, которая реагирует на внутренние или внешние воздействия. Такая модель показывает состояние системы и события, которые служат причиной перехода системы в следующее состояние, не описывая поток данных внутри системы. Вот некоторые из них:

Создание программных средств отладки электронных схем компьютера.

Программирование разбиения исходного текста на различные лексические единицы в компиляторе.

Нахождение в больших объёмах текстовой информации заданных запросов или последовательности различных символов.

Программирование проверки различных систем, которые могут находиться в конечном числе различных состояний.

Детерминированный конечный автомат (ДКА) – абстрактная машина, распознающая слова, в которых для каждой последовательности входных символов существует лишь одно состояние, в которое автомат может перейти из текущего. Детерминированный конечный автомат – это автомат, который при получении данных может находиться в одном состоянии. Если для представления функции переходов ДКА используется таблица, то каждая запись в ней представляет собой единственное состояние.

Недетерминированный конечный автомат (НКА) – абстрактная машина, которая читает символы из вводимого слова и решает, допустить или отвергнуть это слово. Она может изменить состояние, перейдя из одного состояния в другое. Внутреннюю структуру такого автомата можно представить графом переходов НКА, который тоже распознаёт слова. Слово считается допустимым, если после его обработки множество состояний, в котором оказался автомат, содержит хотя бы одно допустимое. Таким образом, НКА также задаёт некоторый язык.

Любой недетерминированный конечный автомат может быть преобразован в детерминированный так, чтобы их языки совпадали, и такие автоматы называются эквивалентными. Но так как количество состояний в эквивалентном ДКА в худшем случае растёт экспоненциально с ростом количества состояний исходного НКА, на практике подобная детерминизация не всегда возможна. Недетерминированный конечный автомат обладает свойством находиться в нескольких состояниях одновременно. Эту особенность часто представляют как свойство автомата пытаться «догадаться», каковы его входные данные. Для многих языков построить соответствующий НКА гораздо легче, чем ДКА. Кроме того, такой ДКА имеет примерно столько же состояний, сколько и НКА, правда, может содержать больше переходов. В худшем случае наименьший ДКА может содержать  $2^n$  состояний, в то время как НКА для того же самого языка имеет всего  $n$  состояний.

Мышление в терминах конечных автоматов (то есть разбиение исполнения программы на шаги автомата и передача информации от шага к шагу через состояние) необходимо при построении событийно-ориентированных приложений. В этом случае программирование в стиле конечных автоматов оказывается единственной альтернативой порождению множества процессов или потоков управления [2]. Часто понятие состояний

и машин состояний используется для спецификации программ. Так, при проектировании программного обеспечения с помощью UML для описания поведения объектов используются диаграммы состояний. Кроме того, явное выделение состояний используется в описании сетевых протоколов [1].

Мышление в терминах вероятностных автоматов еще сложнее, чем в терминах детерминированных. Зато и применение вероятностных автоматов затрагивает области пока недоступные «обычному» программированию: создание искусственного интеллекта, работа с нечеткими множествами, нечеткая логика, распознавание сложных образов, наконец, моделирование человеческого мышления [3]. Вероятностный автомат можно попытаться синтезировать как композицию генератора случайных импульсов и обычного детерминированного конечного автомата (метод Меррея). В этом вся логика работы автомата детерминирована, а случайность реализуется как случайный поток входных импульсов, подаваемых на вход этого автомата.

Пусть необходимо синтезировать автомат, матрица переходных вероятностей которого имеет вид:

$$\begin{pmatrix} 1/3 & 0 & 2/3 \\ 0 & 1/2 & 1/2 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}$$

Граф переходов соответствующего конечного автомата показан на рис. 1.

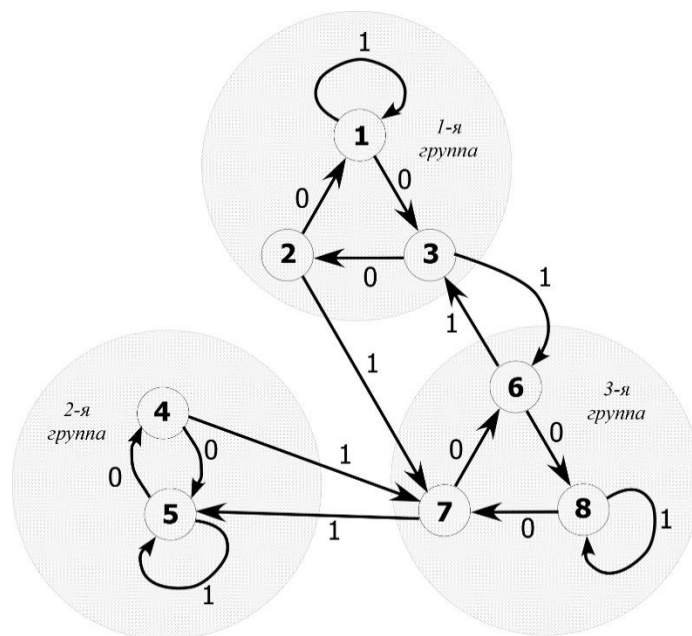


Рис.1. Граф переходов вероятностного автомата

По этому графу нами была смоделирована система с датчиком случайных сигналов и изучалось наличие устойчивых состояний, которые можно трактовать как некие распознанные образы.

### **Список литературы**

1. Жаркова, Г.А., Жарков А.В. *Программная реализация конечных автоматов: учебно-методическое пособие*. Ульяновск: УлГУ, 2011.
2. Поликарпова, Н. И., Шалыто А.А. *Автоматное программирование*. СПб.: СПбГПУ, 2008.
3. Поспелов, Д.А. *Вероятностные автоматы*. М.: Энергия, 1970.