

## ПРИМЕНЕНИЕ JUPYTER NOTEBOOK ДЛЯ ОЗНАКОМЛЕНИЯ С ОСНОВНЫМИ ПОНЯТИЯМИ ТЕОРИИ ДЕТЕРМИНИРОВАННЫХ КОНЕЧНЫХ АВТОМАТОВ

### Аннотация

В данной статье приводится описание использования Jupyter Notebook при начальном изучении конечных автоматов в учебной дисциплине «Теория автоматов» у студентов направления «Компьютерные науки» УрФУ.

### 1. Введение

Статья посвящена проблеме получения студентами практических навыков в начальном изучении основных понятий теории детерминированных конечных автоматов. В монографии И. А. Липского и М. С. Чвановой [1] взаимосвязь теоретической и практической деятельности названа основным признаком того, что полученное знание является методологическим.

Данная работа является продолжением статьи [2], в которой приводится описание использования Jupyter Notebook для изучения методов и алгоритмов решения задач на графах. Аналогично формулировкам статьи [2], обычным способом тренировки студентов в работе с конечными автоматами, теоретические сведения о которых преподаются в форме лекции, является решение задач «вручную». Но для применения в хозяйственной деятельности нужны программные продукты, в которых выполняется комбинирование различных алгоритмов. Следовательно, получение студентами навыков реализации алгоритмов в виде компьютерной программы является важной методологической задачей для преподавателя.

При использовании Jupyter Notebook преподаватель получает возможность тренировать студентов в создании компьютерных программ на языке высокого уровня Python. При этом у преподавателя есть возможность давать подсказки с целью уменьшения трудностей, связанных с техническими особенностями процесса программирования.

### 2. Краткое описание Jupyter Notebook

В соответствии с описанием в статье [2], система Jupyter Notebook является средой для написания и выполнения программ на языке высокого уровня Python.

Основным объектом в системе является файл с расширением `.ipynb`, называемый «блокнот». Блокнот может включать в себя обычный текст, рисунки, ссылки на внешние ресурсы, и тексты программ на языке Python. Такой блокнот, как и любой другой файл, можно копировать, пересылать по почте, выкладывать в «облако» и т.д. Идея объединения в одном документе текста, рисунков, формул и выполняемых программ, вычисляющих значения формул, также используется в файлах системы Mathcad, получившей широкое применение в инженерных дисциплинах.

Для выполнения программ, записанных в блокноте, используется веб-приложение и сервер Jupyter. Также имеется возможность установить на свой компьютер приложение для локального запуска Jupyter Notebook. Подробное руководство для работы с Jupyter Notebook содержится в [3].

### 3. Основные понятия в теории детерминированных конечных автоматов

Определение. Детерминированным конечным автоматом (сокращенно ДКА) называется математический объект, представляющий собой набор

$(Q, \Sigma, \varphi, q_0, Q_F)$ , где

$Q$  – конечное множество (внутренних) состояний автомата;

$\Sigma$  – конечное множество (входных) символов, «алфавит»;

$q_0 \in Q$  – начальное состояние;

$Q_F \subseteq Q$  – множество заключительных состояний;

$\varphi$  – функция переходов (всюду определенная):

$\varphi: Q \times \Sigma \rightarrow Q$ .

Применяются два способа задания ДКА.

Первый способ называется расширенной таблицей переходов. Схематичное составление такой таблице представлено в рисунке 1.

		символы алфавита $\Sigma$				заключ.
		...	$a$	...		
состояния из $Q$	$q_0$					
	...					
	$q$			$\varphi(q, a)$		0 или 1

Рисунок 1.

Строки таблицы подписаны элементами множества  $Q$ , причем первая строка подписана начальным состоянием  $q_0$ . Столбцы таблицы, кроме последнего, подписаны элементами множества  $\Sigma$ . На пересечении каждой строки (например  $q$ ) и каждого столбца (например  $a$ ) в таблице записано состояние  $q'$ , равное значению функции переходов  $\varphi(q, a)$ . В последнем столбце записана информация о заключительных состояниях. Если состояние в названии строки принадлежит  $Q_F$ , то в последнем столбце записывают 1, в противном случае записывают 0.

Второй способ задания ДКА называется диаграммой переходов.

Диаграммой переходов ДКА является ориентированный граф, в котором вершины соответствуют состояниям из множества  $Q$ , каждое ориентированное ребро (дуга) имеет пометку элементом множества  $\Sigma$  (подписано буквой из алфавита  $\Sigma$ ). Схематичное составление такого графа представлено в рисунке 2.

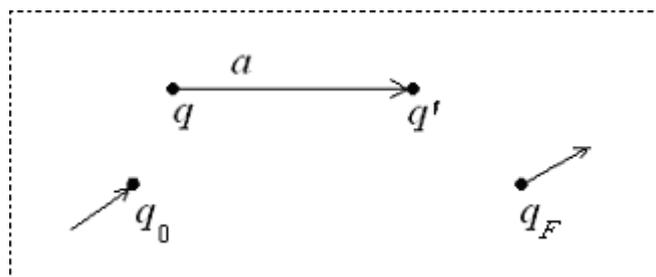


Рисунок 2.

Для каждой пары  $(q, a)$  в диаграмме переходов рисуется дуга из  $q$  в  $q'$ , подписанная буквой  $a$ , где состояние  $q'$  равно значению функции переходов  $\varphi(q, a)$ . Начальное состояние  $q_0$  отмечается входящей стрелкой. Каждое заключительное состояние отмечается выходящей стрелкой.

Изложенные определения соответствуют методическому пособию [4].

#### 4. Полный текст блокнота с заданием на тему «ДКА – часть 1»

##### ДКА - часть 1

Этот набор задач предназначен для самостоятельного решения.

В задачах используется параметр  $N$ , значение которого равно номеру студента в списке группы.

##### Задача 1. Создание таблицы переходов ДКА.

Допишите функцию `make_dfa_table`, которая создаёт двухмерный массив символов, являющийся расширенной таблицей переходов ДКА. Состояниями автомата являются цифры: 0, 1, 2, 3. Алфавит состоит из двух английских букв  $a, b$ . Столбец переходов по букве  $a$  равен (1230) для студентов группы КН-201. Столбец переходов по букве  $a$  равен (2301) для студентов группы КН-202. Столбец переходов по букве  $a$  равен (3012) для студентов группы КН-203. Столбец переходов по букве  $b$  вычисляется по формулам: для 0 переход в  $(N \bmod 2)$ , для 1 переход в  $(N \bmod 3)$ , для 2 переход в  $(N \bmod 4)$ , для 3 переход в 3. Начальным состоянием является 0. Конечным состоянием является 2. Вывести на печать получившуюся таблицу.

```
def make_dfa_table(N):  
    # здесь мог бы быть ваш код  
    dfa =  
    return dfa
```

Допишите программу вывода на печать и проверьте работу функции `make_dfa_table`:

```
import  
# ввести вручную значение  
N =  
make_dfa_table(N)  
# здесь мог бы быть ваш код  
# вывод на печать
```

##### Задача 2. Создание рисунка диаграммы переходов ДКА.

Допишите функцию `make_dfa_picture`, которая по таблице переходов, созданной в задаче 1, создаёт рисунок диаграммы переходов ДКА. Стрелки переходов по букве  $a$

сделать красным цветом, а по букве b - зелёным. Вершины диаграммы подписать номерами состояний, стрелки можно не подписывать.

```
def make_dfa_picture(dfa):  
    # здесь мог бы быть ваш код
```

Проверьте работу функции `make_dfa_picture`:

```
import  
# ввести вручную значение  
N =  
make_dfa_table(N)  
make_dfa_picture(dfa)
```

## 5. Обзор результатов применения блокнота в учебном процессе

В осеннем семестре 2024/2025 учебного года задание на тему «ДКА – часть 1» было выдано студентам трех групп второго курса направления «Компьютерные науки» УрФУ. Общее количество в этих группах составляет 82 студента. Из этого количества не приступали к выполнению задания 24 студента, что составляет примерно 29%.

Для отчета по выполнению задания студентам было предложено прислать на электронную почту преподавателя заполненный блокнот и скриншоты выводимых программой на экран таблицы и рисунка диаграммы переходов ДКА. В отчетах студентов, выполнивших задание, встречались разного вида ошибки. Наиболее частыми ошибками были таблицы ДКА без последнего столбца заключительных состояний. Также встречались ошибки в диаграмме переходов ДКА, когда не были отмечены начальное и заключительное состояния. Ещё одной ошибкой при рисунке диаграммы переходов ДКА было неправильное использование в языке Python библиотеки `graphviz` визуализации графов. Выполнили задание без ошибок 14 студентов, что составляет примерно 17%.

Как пример отчета на рисунках 3 и 4 приведены скриншоты результатов задания, выполненного студентом К.

состояние	a	b	F
0	3	1	0
1	0	0	0
2	1	3	1
3	2	3	0

Рисунок 3.

Использование специальных символов вместо разделительных линий таблицы сильно упрощает выполнение задания и не считается ошибкой.

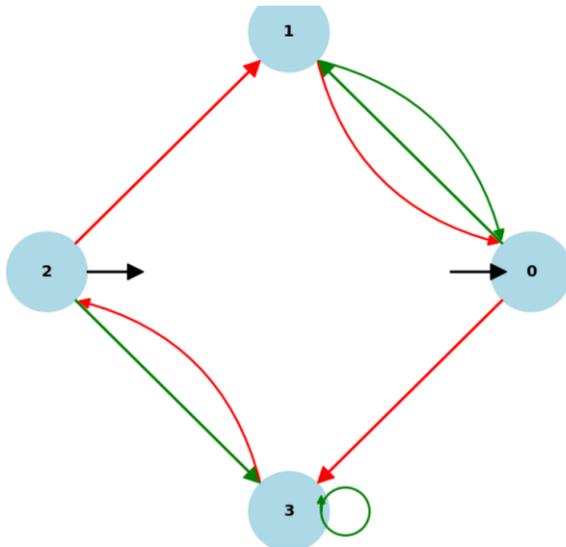


Рисунок 4.

В соответствии с заданием вместо подписей дуг нужно было выполнить рисунок дуг красным и зеленым цветом.

## 6. Заключение

Файлы в формате блокнот являются удобным инструментом формулировки и выполнения заданий. Этот инструмент комфортен как для преподавателя, так и для студентов. Использование в учебном процессе заданий с применением Jupyter Notebook улучшает у студентов взаимосвязь теоретических сведений и практической деятельности.

## Список литературы

1. Чванова М. С., Липский И. А. Информатизация системы непрерывной подготовки специалистов: методология, теория, практика. Монография. - Тамбов-Москва, Изд. ТГУ, 2000. - 518 с.
2. Щербакова В.А. Применение Jupyter Notebook для изучения алгоритмов обхода в ширину и обхода в глубину на неориентированных графах. Электронный ресурс размещения научных статей preprint.ru, URL: <https://doi.org/10.24108/preprints-3112979>, Дата публикации 05.02.2024.- 4 с.
3. Руководство для начинающих. URL: <https://pythonru.com/baza-znanij/jupyter-notebook-dlja-nachinajushhih>.
4. Замятин А.П. Автоматы, сети, алгоритмы. Учебное пособие. - Екатеринбург: Изд. УрГУ, 2001.- 89 с.