

Разработка программы, использующей математическую модель конечного автомата

Черкашин Александр Михайлович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описан процесс написания программы по использованию алгоритма конечного автомата. В процессе работы использовался язык программирования Python для написания программы, и алгоритм конечного автомата. В результате работы разработана программа который применяет алгоритм математической модели конечного автомата.

Ключевые слова: Python, конечный автомат, КА.

Development of a program using a mathematical model of a finite automaton

Cherkashin Alexander Mihailovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of writing a program using the finite state machine algorithm. In the process of work, the Python programming language was used to write the program, and the finite automaton algorithm was used. As a result of the work, a program was developed that uses the algorithm of a mathematical model of a finite automaton.

Keywords: Python, finite-state machine, FSM.

1 Введение

1.1 Актуальность исследования

Данная статья описывает возможность написание программы для решения задачи математической модели алгоритма конечного автомата. Математическая модель конечного автомата применяют в синтаксических и лексических анализаторах, тестировании программы.

1.2 Цель исследования

Целью работы является написание программы для решения задачи математической модели конечного автомата.

1.3 Обзор исследований

В работе Й. Ц. Монтеиро, А. Л. Оливейра представляют использование конечных автоматов для синхронизации часов в последовательной логической схеме [1].

Т. Биманторо, Х. Харянто предлагает программу которая использует конечный автомат для игры с образовательными задачами по химии [2].

Л. Юан, Г. Яу описывает конечный автомат для сокрытия избыточной информации [3].

2. Решение

Алгоритм конечный автомат представляет собой дискретное устройство, имеющий один вход, один выход, и в каждый момент времени находящийся в одном состоянии из множества возможных, число возможных внутренних состояний которого конечно [4].

В данной статье применялась математическая модель алгоритма конечного автомата для написания программы на языке программирования Python. Исходный код класса StateMachine взят из источника [5].

Листинг 2.1. Класс конечный автомат

```
1 class StateMachine:
2     def __init__(self):
3         self.handlers = {}
4         self.startState = None
5         self.endStates = []
6     def add_state(self, name, handler, end_state=0):
7         name = name.upper()
8         self.handlers[name] = handler
9         if end_state:
10            self.endStates.append(name)
11
12    def set_start(self, name):
13        self.startState = name.upper()
14
15    def run(self, cargo):
16        try:
17            handler = self.handlers[self.startState]
18        except:
19            raise InitializationError("must call .set_start() before
20.run()")
21        if not self.endStates:
22            raise InitializationError("at least one state must be an
23.end_state")
24
25        while True:
26            (newState, cargo) = handler(cargo)
27            if newState.upper() in self.endStates:
28                print("reached ", newState)
29                break
```

30	else:
31	handler = self.handlers[newState.upper()]

В листинг 2.1. Переменная `handlers` — список функций, ключ — имя функция, значение — тело функции. `StartState` — Входное состояние, `endStates` — выходное состояние.

Если `handlers` в функция не входит `StartState` и `endStates` то внутреннее состояние.

Функция `add_state` добавляет состояние.

Функция `set_start` устанавливает начальное состояние (точки входа).

Функция `run` запускает автомат, аргумент `cargo` используется для передачи функции состояния. Запускает начальное состояние получая аргумент `cargo`.

Листинг 2.2. Использование алгоритма конечного автомата

1	<code>def s0(text):</code>
2	<code>print("s0: " + str(text))</code>
3	<code>return ("s2", text)</code>
4	<code>def s1(text):</code>
5	<code>print("s1: " + str(text))</code>
6	<code>return ("s0", text)</code>
7	<code>def s2(text):</code>
8	<code>print("s2: " + str(text))</code>
9	<code>return ("stop", text)</code>
10	
11	<code>m = StateMachine()</code>
12	<code>m.add_state("s0", s0, 0)</code>
13	<code>m.add_state("s1", s1, 0)</code>
14	<code>m.add_state("s2", s2, 0)</code>
15	<code>m.add_state("stop", None, 1)</code>
16	<code>m.set_start("s1")</code>
17	<code>m.run("arg 0")</code>
18	<code>m.run("arg 1")</code>
19	<code>m.run("arg 2")</code>

Листинг 2.2. Строки 1 — 9. Состояние автомата.

Строка 11. Создание экземпляра класса `StateMachine`.

Строки 12 — 15. Добавляем список состояния.

Строка 16. задаем входное состояние.

Строки 17 — 19. Запускаем автомат.

```
user@Desktop:~  
user@Desktop:~$ ./my_1.py  
s1: arg 0  
s0: arg 0  
s2: arg 0  
reached stop  
s1: arg 1  
s0: arg 1  
s2: arg 1  
reached stop  
s1: arg 2  
s0: arg 2  
s2: arg 2  
reached stop  
user@Desktop:~$
```

Рисунок. 2.1. Результат выполнения программы

3 Выводы

В результате работы была написана программа на языке Python с использованием математической модели алгоритма конечного автомата.

Библиографический список

1. Monteiro J. C., Oliveira A. L. Finite state machine decomposition for low power //Proceedings of the 35th annual Design Automation Conference. 1998. С. 758-763.
2. Bimantoro T., Haryanto H. Pemodelan perilaku musuh menggunakan Finite State Machine (FSM) pada game pengenalan unsur kimia //Journal of Applied Intelligent System. 2016. Т. 1. №. 3. С. 210-219.
3. Yuan L., Qu G. Information hiding in finite state machine //International Workshop on Information Hiding. Springer, Berlin, Heidelberg, 2004. С. 340-354.
4. Конечный автомат — Википедия // Википедия URL: https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82 (дата обращения: 2023-01-26).
5. Finite State Machine in Python | Applications | python-course.eu // python-course URL: <https://python-course.eu/applications-python/finite-state-machine.php> (дата обращения: 2022-09-05).