

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное агентство по образованию

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики-процессов управления

Трефилов Андрей Романович

Курсовая работа
Проверочная работа
по МП-автоматам

2 курс, 232 группа

Руководитель курсовой работы
А.В. Матросов
2016 г.

Санкт-Петербург, 2016 г.

1 Постановка задачи

Задать порождающую грамматику языка $\{(0^n 1^{2n}) 0^{2m} 1^m | n, m \geq 0\}$ и построить по ней распознающий МП НКА. Разработать программу, распознающую цепочки этого языка на основе программной реализации МП НКА.

2 Задание грамматики

$$G = \left(\{S, R, T\}, \{0, 1, (,)\}, \left\{ \begin{array}{l} S \rightarrow (T)R \\ T \rightarrow 0T11|\varepsilon \\ R \rightarrow 00R1|\varepsilon \end{array} \right\}, S \right)$$

3 Задание МП НКА

$$P = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0, 1, (,)\}, \{0, 1, (,)\}, \delta, q_0, Z_0, \{q_3\})$$

Функция δ определяется следующими правилами:

- 1) $\delta(q_0, (, Z_0) = \{(q_0, (Z_0)\}$
- 2) $\delta(q_0, 0, () = \{(q_0, 00())\}$
- 3) $\delta(q_0,), () = \{(q_0, \varepsilon)\}$
- 4) $\delta(q_0, 0, 0) = \{(q_0, 000)\}$
- 5) $\delta(q_0, 1, 0) = \{(q_1, \varepsilon)\}$
- 6) $\delta(q_1, 1, 0) = \{(q_1, \varepsilon)\}$
- 7) $\delta(q_1,), () = \{(q_4, \varepsilon)\}$
- 8) $\delta(q_4, \varepsilon, Z_0) = \{(q_3, Z_0)\}$
- 9) $\delta(q_4, 0, Z_0) = \{(q_4, 0Z_0)\}$
- 10) $\delta(q_4, 0, 0) = \{(q_4, 00)\}$
- 11) $\delta(q_4, 1, 0) = \{(q_5, \varepsilon\varepsilon)\}$
- 12) $\delta(q_5, 1, 0) = \{(q_5, \varepsilon\varepsilon)\}$
- 13) $\delta(q_5, \varepsilon, Z_0) = \{(q_3, \varepsilon)\}$

Все остальные сочетания состояний, терминальных символов и терминальных символов на стеке приводят к переходу автомата в состояние q_2 , которое является «дьявольским» состоянием. Состояние q_3 является финальным.

На рисунке цифры у стрелок-переходов означают пронумерованные выше переходы. «oth» — «other», все остальные переходы.

4 Описание МП НКА

Состояние q_0 является состоянием чтения, в котором автомат читает сперва символ $($, а затем символы 0 и заносит их в стек. Встретив первый символ 1 автомат переходит в состояние q_1 в котором происходит чтение символов 1 и снятие со стека эквивалентного числа символов 0 . Встретив символ $)$, автомат может перейти в 2 состояния: финальное состояние q_3 (то есть $m = 0$) или в состояние q_4 , которое является аналогом состояния q_1 , за тем лишь исключением, что вначале не читается скобка. Аналогично, встретив первую единицу, автомат переходит в состояние q_5 , которое является аналогом состояния чтения

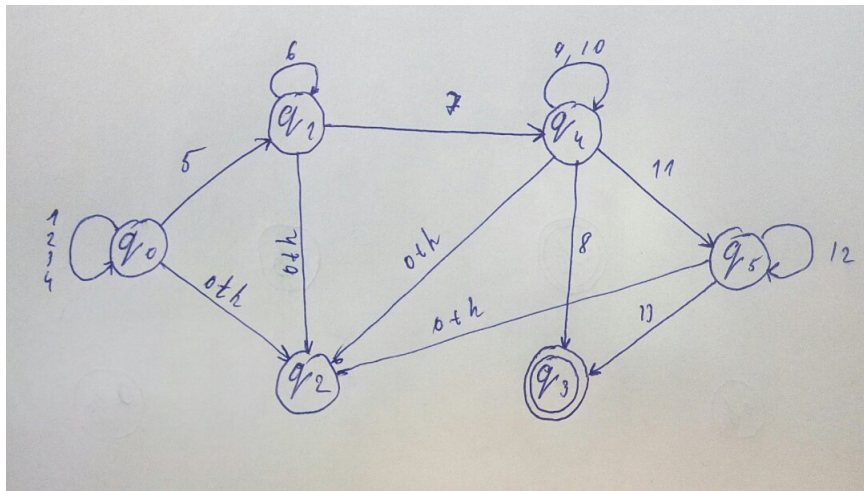


Рис. 3.1:

q_1 . Необходимость состояний q_4, q_5 обуславливается различным количеством символов 0, 1 подающихся на вход при первом чтении (в скобках) и при втором.

5 Код программы

Язык программирования — python версии 3.5

```
stack = []
step = 0 #состояние
iterator = 0
string = input('Введите строку: ')
string += '*'; #маркер конца строки
for symbol in string:
    iterator += 1 #для обозначения места первого несоответствия
    if step == 0 and symbol == '(' and len(stack) == 0 :
        step = 0
        stack.append(symbol)
    elif step == 0 and symbol == '0' and len(stack) != 0 and stack[len(stack) - 1] ==
        step = 0
        stack.append(symbol)
        stack.append(symbol)

    elif step == 0 and symbol == ')' and len(stack) != 0 and stack[len(stack) - 1] ==
        step = 4
        stack.pop()
    elif step == 0 and symbol == '0' and len(stack) != 0 and stack[len(stack) - 1] ==
        step = 0
        stack.append(symbol)
        stack.append(symbol)
    elif step == 0 and symbol == '1' and len(stack) != 0 and stack[len(stack) - 1] ==
        step = 1
        stack.pop()
    elif step == 1 and symbol == '1' and len(stack) != 0 and stack[len(stack) - 1] ==
```

```

    step = 1
    stack.pop()

elif step == 1 and symbol == ')' and len(stack) != 0 and stack[len(stack) - 1] == '(':
    step = 4
    stack.pop()
elif step == 4 and symbol == '*' and len(stack) == 0 :
    step = 3
    print('Входная цепочка принадлежит данному языку.')
elif step == 4 and symbol == '0' and len(stack) == 0 :
    step = 4
    stack.append(symbol)
elif step == 4 and symbol == '0' and len(stack) != 0 and stack[len(stack) - 1] == '0':
    step = 4
    stack.append(symbol)
elif step == 4 and symbol == '1' and len(stack) > 1 and stack[len(stack) - 1] == '1':
    step = 5
    stack.pop()
    stack.pop()
elif step == 5 and symbol == '1' and len(stack) > 1 and stack[len(stack) - 1] == '1':
    step = 5
    stack.pop()
    stack.pop()
elif step == 5 and symbol == '*' and len(stack) == 0 :
    step = 3
    print('Входная цепочка принадлежит данному языку.')
else :
    step = 2
    print('Входная цепочка не принадлежит данному языку.')
    print('Первое несоответствие цепочки языку имеет позицию ', iterator)
    print('Это символ', symbol)
    break;

```

6 Описание работы программы

Программа представляет собой наивную реализацию переходов конечного автомата в различные состояния. Переменная *step* является хранителем состояния, переменная *stack* — реализацию стека при помощи методов стандартной библиотеки. При обработке каждого нового символа из введенной строки программа последовательно проверяет соответствие текущего положения и возможности перехода в другое состояние по текущему символу и символу, находящемуся на вершине стека. Если это возможно, происходит переход, в противном случае состояние сохраняется. При запуске программы отображается приглашение ко вводу, после которого необходимо ввести строку, которая будет проверена на соответствие заданной КС грамматике. Если соответствие найдено, то будет выведено извещение об этом. Если же какой-то символ строки не соответствует данной КС грамматике, то будет выведено предупреждение об этом, позиция данного символа и сам символ.

7 Тестирование

Тестовая строка	Вывод программы
	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 1 Это символ *
()	Входная цепочка принадлежит данному языку.
(0)	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 3 Это символ)
(011)	Входная цепочка принадлежит данному языку.
011	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 1 Это символ 0
(00111)	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 7 Это символ)
(001111)	Входная цепочка принадлежит данному языку.
()001	Входная цепочка принадлежит данному языку.
()1	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 3 Это символ 1
(011)001	Входная цепочка принадлежит данному языку.
()000011	Входная цепочка принадлежит данному языку.
()01	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 4 Это символ 1
(001111)001	Входная цепочка принадлежит данному языку.
def	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 1 Это символ d
(())	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 2 Это символ (
()()	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 3 Это символ (
(011	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 5 Это символ *
()001()	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 6 Это символ (
(011)000000111	Входная цепочка принадлежит данному языку.
(0)(0)	Входная цепочка не принадлежит данному языку. Первое несоответствие цепочки языку имеет позицию 3 Это символ)