

Дисциплина: АНАЛИЗ АЛГОРИТМОВ

РУБЕЖНЫЙ КОНТРОЛЬ 2

Нахождение подстроки в строке с помощью регулярных  
выражений и конечного автомата.

Студент группы ИУ7-55,  
Аминов Тимур Саидович

# 1 Аналитическая часть

В рамках раздела будет дано аналитическое описание регулярных выражений и конечного автомата.

## 1.1 Описание алгоритмов

### 1.1.1 Регулярные выражения

**Регулярные выражения** – формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов. Для поиска используется строка-образец, состоящая из символов и метасимволов и задающая правило поиска. Для манипуляций с текстом дополнительно задаётся строка замены, которая также может содержать в себе специальные символы.

### 1.1.2 Конечный автомат

**Конечный автомат** можно охарактеризовать множеством состояний (вершин) и переходов (дуг, соединяющих вершины). Среди состояний есть два специальных - состояние начала и конца. Если строка читается данным автоматом, то после прохода по строке, автомат должен оказаться в одном из заключительных состояний. На этом основывается алгоритм поиска подстроки в строке с помощью конечного автомата.

## 2 Конструкторская часть

В данной части будут рассмотрены схемы разработанных автоматов.

### 2.1 Разработка алгоритмов

На рисунках 1-4 приведены схемы автоматов поиска даты в русском, английском и текстовом форматах.

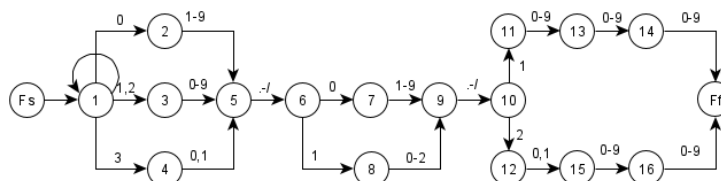


Рис. 1: Автомат для поиска даты в русском формате

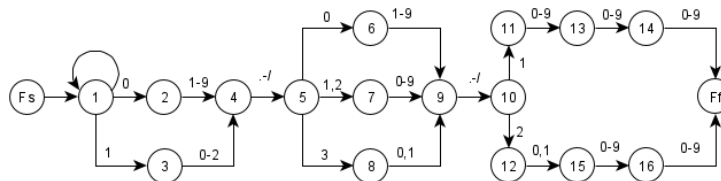


Рис. 2: Автомат для поиска даты в английском формате

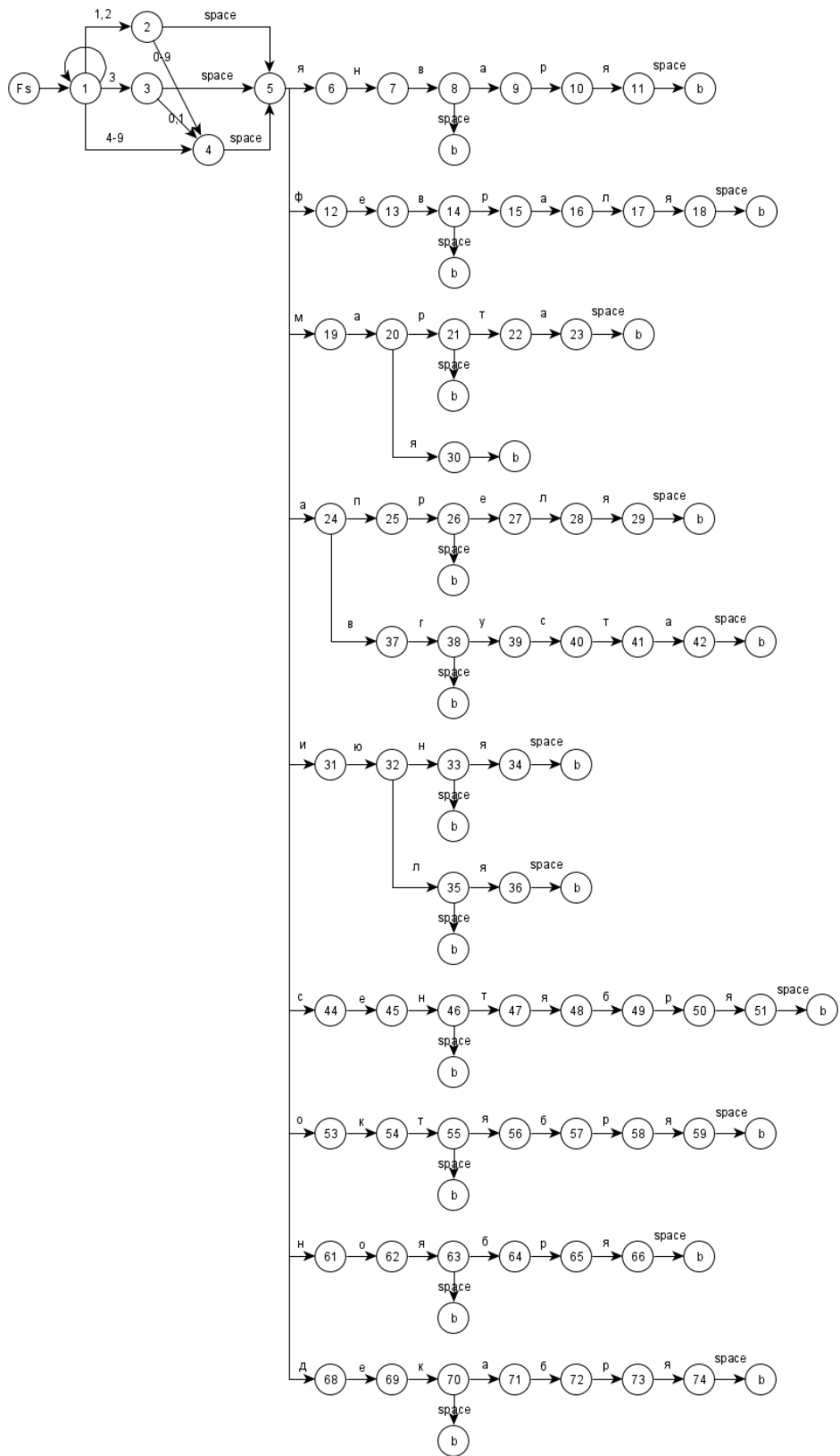


Рис. 3: Автомат для поиска даты в текстовом формате

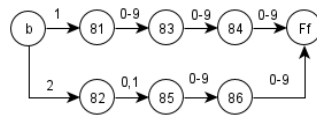


Рис. 4: Автомат для поиска даты в текстовом формате (продолжение)

## 3 Технологическая часть

В рамках этого раздела будут описаны инструментарии разработки, выбор среды, требования к ПО. Также будут предоставлены листинги конкретных реализаций алгоритмов.

### 3.1 Средства реализации

Программа была написана на языке программирования python 3.8 в среде разработки PyCharm. Я выбрал данный язык, так как уже имею определенный опыт работы с ним.

### 3.2 Требования к программному обеспечению

На вход программа должна получать текст, на выход отображать все найденные даты в формате dd.mm.yyyy, mm.dd.yyyy, а также даты, в которых название месяца записано целым или сокращенным словом.

### 3.3 Листинг кода

В листингах 1-5 представлена реализация поиска даты с помощью регулярных выражений и конечных автоматов. На листинге 1 представлен код регулярных выражений для даты в русском, английском и текстовом формате соответственно. На листинге 2 представлен класс и функции для работы с автоматом. На листинге 3 представлены инструкции автомата для поиска даты в русском формате. На листинге 4 представлены инструкции автомата для поиска даты в английском формате. На листинге 5 представлены инструкции автомата для поиска даты в текстовом формате.

Листинг 1: Листинг разработанных регулярных выражений

```
1 months = [ 'января', 'января',
2             'февраля', 'фев',
3             'марта', 'мар',
4             'апреля', 'апр',
5             'мая',
6             'июня', 'июнь',
7             'июля', 'июль',
8             'августа', 'авг',
9             'сентября', 'сен',
10            'октября', 'окт',
11            'ноября', 'ноя',
12            'декабря', 'дек' ]
13
14 months_str = get_months_str(months)
15
16 res_re_ru_date = re.findall(r'(0[1-9]|[12][0-9]|3[01])[ -/.](0[1-9]|1[0-2])[ -/.](1\d\d\d|
17 \d|2[01]\d\d)', text)
18 res_re_us_date = re.findall(r'(0[1-9]|1[0-2])[ -/.](0[1-9]|[12][0-9]|3[01])[ -/.](1\d\d\d|
19 \d|2[01]\d\d)', text)
20 res_re_text_date = re.findall(r'([1-9]|[12][0-9]|3[01]) (' + months_str + r') (1\d\d\d|
21 \d|2[01]\d\d)', text)
```

Листинг 2: Класс и функции для работы с автоматом

```
1 # Классинструкцииавтомата
2 class Transition:
3     def __init__(self, state_1, state_2, symbols, sym_ins, direction):
4         self.state_1 = state_1
5         self.state_2 = state_2
6         self.symbols = symbols
7         self.sym_ins = sym_ins
8         self.direction = direction
9
10 # Возвращаетмассивпариндексовподходящихусловиюподстрока
11 def run_automat(instructions, text, final_state):
12     pos = 0
13     end = len(text)
```

```

14 state = 1
15 start_pos = 0
16 found = []
17
18 while pos != end:
19     # print(state, text[pos])
20     tran = find_transition(state, text[pos], instructions)
21
22     if tran.state_1 == 1 and tran.state_2 != 1:
23         start_pos = pos
24
25     if tran.state_2 == final_state:
26         found.append((start_pos, pos + 1))
27
28     if tran.direction == 'r':
29         pos += 1
30     elif tran.direction == 'l':
31         pos -= 1
32
33     state = tran.state_2
34
35 return found
36
37 # Возвращает нужную инструкцию из массива инструкций
38 def find_transition(state, sym, instructions):
39     for i in instructions:
40         if i.state_1 == state:
41             if i.sym_ins == 'in':
42                 if sym in i.symbols:
43                     return i
44             elif i.sym_ins == 'not':
45                 if sym not in i.symbols:
46                     return i
47
48 return None

```

Листинг 3: Автомат для поиска даты в формате dd.mm.yyyy

```

1  # Автомат для поиска даты в формате dd.mm.yyyy
2  def date_ru_automat(text):
3      instructions = [Transition(1, 2, ['0'], 'in', 'r'), # Первая цифра дня
4                      Transition(1, 3, ['1', '2'], 'in', 'r'),
5                      Transition(1, 4, ['3'], 'in', 'r'),
6                      Transition(1, 1, ['0', '1', '2', '3'], 'not', 'r'),
7
8                      Transition(2, 5, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in',
9                      'r'), # Вторая цифра дня
10                     Transition(3, 5, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
11                     'in', 'r'),
12                     Transition(4, 5, ['0', '1'], 'in', 'r'),
13                     Transition(2, 1, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not',
14                     'r'),
15                     Transition(3, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
16                     'not', 'r'),
17                     Transition(4, 1, ['0', '1'], 'not', 'r'),
18
19                     Transition(5, 6, ['.', '-', '/'], 'in', 'r'), # Разделитель
20                     Transition(5, 1, ['.', '-', '/'], 'not', 'r'),
21
22                     Transition(6, 7, ['0'], 'in', 'r'), # Первая цифра месяца
23                     Transition(6, 8, ['1'], 'in', 'r'),
24                     Transition(6, 1, ['0', '1'], 'not', 'r'),
25
26                     Transition(7, 9, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in',
27                     'r'), # Вторая цифра месяца
28                     Transition(8, 9, ['0', '1', '2'], 'in', 'r'),
29                     Transition(7, 1, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not',
30                     'r'),
31                     Transition(8, 1, ['0', '1', '2'], 'not', 'r'),
32
33                     Transition(9, 10, ['.', '-', '/'], 'in', 'r'), # Разделитель
34                     Transition(9, 1, ['.', '-', '/'], 'not', 'r'),
35
36                     Transition(10, 11, ['1'], 'in', 'r'), # Первая цифра года
37                     Transition(10, 12, ['2'], 'in', 'r'),
38                     Transition(10, 1, ['1', '2'], 'not', 'r'),
39
40                     Transition(11, 13, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
41                     'in', 'r'), # 1000 - 1999
42                     Transition(13, 14, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
43                     'in', 'r'),
44                     Transition(14, 100, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
45                     'in', 'r'),
46                     Transition(11, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
47                     'not', 'r'),
48                     Transition(13, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
49                     'not', 'r'),
50                     Transition(14, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
51                     'not', 'r'),
52
53                     Transition(12, 15, ['0', '1'], 'in', 'r'), # 2000 - 2199
54                     Transition(15, 16, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
55                     'in', 'r'),
56                     Transition(16, 100, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
57                     'in', 'r'),
58                     Transition(12, 1, ['0', '1'], 'not', 'r'),
59                     Transition(15, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
60                     'not', 'r'),
61                     Transition(16, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
62                     'not', 'r'),
63
64                     Transition(100, 1, [], 'not', 'r'), # Продолжаем поиск
65
66 ]
67
68 final_state = 100

```

```

52 found = run_automat(instructions, text, final_state)
53 res = []
54
55 for couple in found:
56     res.append(text[couple[0]:couple[1]])
57
58 return res

```

Листинг 4: Автомат для поиска даты в формате mm.dd.yyyy

```

1  # Автомат для поиска даты в формате mm.dd.yyyy
2  def date_us_automat(text):
3      instructions = [
4          Transition(1, 2, ['0'], 'in', 'r'), # Первая цифра месяца
5          Transition(1, 3, ['1'], 'in', 'r'),
6          Transition(1, 1, ['0', '1'], 'not', 'r'),
7
8          Transition(2, 4, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in',
9              'r'), # Вторая цифра месяца
10         Transition(3, 4, ['0', '1', '2'], 'in', 'r'),
11         Transition(2, 1, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not',
12             'r'),
13         Transition(3, 1, ['0', '1', '2'], 'not', 'r'),
14
15         Transition(4, 5, ['.', '-', '/'], 'in', 'r'), # Разделитель
16         Transition(4, 1, ['.', '-', '/'], 'not', 'r'),
17
18         Transition(5, 6, ['0'], 'in', 'r'), # Первая цифра дня
19         Transition(5, 7, ['1', '2'], 'in', 'r'),
20         Transition(5, 8, ['3'], 'in', 'r'),
21         Transition(5, 1, ['0', '1', '2', '3'], 'not', 'r'),
22
23         Transition(6, 9, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in',
24             'r'), # Вторая цифра дня
25         Transition(7, 9, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
26             'in', 'r'),
27         Transition(8, 9, ['0', '1'], 'in', 'r'),
28         Transition(6, 1, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not',
29             'r'),
30         Transition(7, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
31             'not', 'r'),
32         Transition(8, 1, ['0', '1'], 'not', 'r'),
33
34         Transition(9, 10, ['.', '-', '/'], 'in', 'r'), # Разделитель
35         Transition(9, 1, ['.', '-', '/'], 'not', 'r'),
36
37         Transition(10, 11, ['1'], 'in', 'r'), # Первая цифра года
38         Transition(10, 12, ['2'], 'in', 'r'),
39         Transition(10, 1, ['1', '2'], 'not', 'r'),
40
41         Transition(11, 13, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
42             'in', 'r'), # 1000 - 1999
43         Transition(13, 14, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
44             'in', 'r'),
45         Transition(14, 100, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
46             'in', 'r'),
47         Transition(11, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
48             'not', 'r'),
49         Transition(13, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
50             'not', 'r'),
51         Transition(14, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
52             'not', 'r'),
53
54         Transition(12, 15, ['0', '1'], 'in', 'r'), # 2000 - 2199
55         Transition(15, 16, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
56             'in', 'r'),
57         Transition(16, 100, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
58             'in', 'r'),
59         Transition(12, 1, ['0', '1'], 'not', 'r'),

```



```

46         Transition(15, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
47                     'not', 'r'),
48         Transition(16, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
49                     'not', 'r'),
50
51         Transition(100, 1, [], 'not', 'r') # Продолжаем поиск
52     ]
53
54     final_state = 100
55     found = run_automat(instructions, text, final_state)
56     res = []
57
58     for couple in found:
59         res.append(text[couple[0]:couple[1]])
60
61     return res

```

Листинг 5: Автомат для поиска даты в текстовом формате

```

1  # Автомат для поиска даты в текстовом формате
2  def date_text_automat(text):
3      a = 5
4      b = 200
5      c = 300
6
7      instructions = [
8          Transition(1, 2, ['1', '2'], 'in', 'r'),
9          Transition(1, 3, ['3'], 'in', 'r'),
10         Transition(1, 4, ['4', '5', '6', '7', '8', '9'], 'in', 'r'),
11         Transition(2, 4, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in', 'r'),
12         Transition(2, a, [''], 'in', 'r'),
13         Transition(3, 4, ['0', '1'], 'in', 'r'),
14         Transition(3, a, [''], 'in', 'r'),
15         Transition(4, a, [''], 'in', 'r'),
16
17         Transition(1, 1, ['1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not', 'r'),
18         Transition(2, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', ''], 'not',
19                     'r'),
20         Transition(3, 1, ['0', '1', ''], 'not', 'r'),
21         Transition(4, 1, [''], 'not', 'r'),
22
23         # января, янв
24         Transition(a, 6, ['я'], 'in', 'r'),
25         Transition(6, 7, ['н'], 'in', 'r'),
26         Transition(7, 8, ['в'], 'in', 'r'),
27         Transition(8, 9, ['а'], 'in', 'r'),
28         Transition(9, 10, ['п'], 'in', 'r'),
29         Transition(10, 11, ['я'], 'in', 'r'),
30         Transition(11, b, [''], 'in', 'r'),
31         Transition(8, b, [''], 'in', 'r'),
32
33         Transition(a, 1, ['я', 'ф', 'м', 'а', 'и', 'с', 'о', 'н', 'д'], 'not', 'r'),
34         Transition(6, 1, ['н'], 'not', 'r'),
35         Transition(7, 1, ['в'], 'not', 'r'),
36         Transition(8, 1, ['а', ''], 'not', 'r'),
37         Transition(9, 1, ['п'], 'not', 'r'),
38         Transition(10, 1, ['я'], 'not', 'r'),
39         Transition(11, 1, [''], 'not', 'r'),
40
41         # февраля, фев
42         Transition(a, 12, ['ф'], 'in', 'r'),
43         Transition(12, 13, ['е'], 'in', 'r'),
44         Transition(13, 14, ['в'], 'in', 'r'),
45         Transition(14, 15, ['п'], 'in', 'r'),
46         Transition(15, 16, ['а'], 'in', 'r'),
47         Transition(16, 17, ['л'], 'in', 'r'),
48         Transition(17, 18, ['я'], 'in', 'r'),
49         Transition(18, b, [''], 'in', 'r'),
50         Transition(14, b, [''], 'in', 'r'),

```

```

50
51 Transition(12, 1, ['e'], 'not', 'r'),
52 Transition(13, 1, ['в'], 'not', 'r'),
53 Transition(14, 1, ['р', ' '], 'not', 'r'),
54 Transition(15, 1, ['а'], 'not', 'r'),
55 Transition(16, 1, ['л'], 'not', 'r'),
56 Transition(17, 1, ['я'], 'not', 'r'),
57 Transition(18, 1, [' '], 'not', 'r'),
58
59 # марта, мар
60 Transition(a, 19, ['м'], 'in', 'r'),
61 Transition(19, 20, ['а'], 'in', 'r'),
62 Transition(20, 21, ['р'], 'in', 'r'),
63 Transition(21, 22, ['т'], 'in', 'r'),
64 Transition(22, 23, ['а'], 'in', 'r'),
65 Transition(23, b, [' '], 'in', 'r'),
66 Transition(21, b, [' '], 'in', 'r'),
67
68 Transition(19, 1, ['а'], 'not', 'r'),
69 Transition(20, 1, ['р', 'я'], 'not', 'r'),
70 Transition(21, 1, ['т', ' '], 'not', 'r'),
71 Transition(22, 1, ['а'], 'not', 'r'),
72 Transition(23, 1, [' '], 'not', 'r'),
73
74 # апреля, апр
75 Transition(a, 24, ['а'], 'in', 'r'),
76 Transition(24, 25, ['п'], 'in', 'r'),
77 Transition(25, 26, ['р'], 'in', 'r'),
78 Transition(26, 27, ['е'], 'in', 'r'),
79 Transition(27, 28, ['л'], 'in', 'r'),
80 Transition(28, 29, ['я'], 'in', 'r'),
81 Transition(29, b, [' '], 'in', 'r'),
82 Transition(26, b, [' '], 'in', 'r'),
83
84 Transition(24, 1, ['п', 'в'], 'not', 'r'),
85 Transition(25, 1, ['р'], 'not', 'r'),
86 Transition(26, 1, ['е', ' '], 'not', 'r'),
87 Transition(27, 1, ['л'], 'not', 'r'),
88 Transition(28, 1, ['я'], 'not', 'r'),
89 Transition(29, 1, [' '], 'not', 'r'),
90
91 # мая
92 # Transition(, , м[''], 'in', 'r'),
93 # Transition(, , а[''], 'in', 'r'),
94 Transition(20, 30, ['я'], 'in', 'r'),
95 Transition(30, b, [' '], 'in', 'r'),
96
97 Transition(30, 1, [' '], 'not', 'r'),
98
99 # июня, июн
100 Transition(a, 31, ['и'], 'in', 'r'),
101 Transition(31, 32, ['ю'], 'in', 'r'),
102 Transition(32, 33, ['н'], 'in', 'r'),
103 Transition(33, 34, ['я'], 'in', 'r'),
104 Transition(34, b, [' '], 'in', 'r'),
105 Transition(33, b, [' '], 'in', 'r'),
106
107 Transition(31, 1, ['ю'], 'not', 'r'),
108 Transition(32, 1, ['н', 'л'], 'not', 'r'),
109 Transition(33, 1, ['я', ' '], 'not', 'r'),
110 Transition(34, 1, [' '], 'not', 'r'),
111
112 # июля, июл
113 # Transition(, , и[''], 'in', 'r'),
114 # Transition(, , ю[''], 'in', 'r'),
115 Transition(32, 35, ['л'], 'in', 'r'),
116 Transition(35, 36, ['я'], 'in', 'r'),
117 Transition(36, b, [' '], 'in', 'r'),
118 Transition(35, b, [' '], 'in', 'r'),

```

```

119
120 Transition(35, 1, ['я', ' '], 'not', 'r'),
121 Transition(36, 1, [' '], 'not', 'r'),
122
123 # августа, авг
124 # Transition(, a[''], 'in', 'r'),
125 Transition(24, 37, ['в'], 'in', 'r'),
126 Transition(37, 38, ['г'], 'in', 'r'),
127 Transition(38, 39, ['й'], 'in', 'r'),
128 Transition(39, 40, ['с'], 'in', 'r'),
129 Transition(40, 41, ['т'], 'in', 'r'),
130 Transition(41, 42, ['а'], 'in', 'r'),
131 Transition(42, b, [' '], 'in', 'r'),
132 Transition(38, b, [' '], 'in', 'r'),
133
134 Transition(37, 1, ['г'], 'not', 'r'),
135 Transition(38, 1, ['й', ' '], 'not', 'r'),
136 Transition(39, 1, ['с'], 'not', 'r'),
137 Transition(40, 1, ['т'], 'not', 'r'),
138 Transition(41, 1, ['а'], 'not', 'r'),
139 Transition(42, 1, [' '], 'in', 'r'),
140
141 # сентября, сен // 43 — отсутствует
142 Transition(a, 44, ['с'], 'in', 'r'),
143 Transition(44, 45, ['е'], 'in', 'r'),
144 Transition(45, 46, ['н'], 'in', 'r'),
145 Transition(46, 47, ['т'], 'in', 'r'),
146 Transition(47, 48, ['я'], 'in', 'r'),
147 Transition(48, 49, ['б'], 'in', 'r'),
148 Transition(49, 50, ['р'], 'in', 'r'),
149 Transition(50, 51, ['я'], 'in', 'r'),
150 Transition(51, b, [' '], 'in', 'r'),
151 Transition(46, b, [' '], 'in', 'r'),
152
153 Transition(44, 1, ['е'], 'not', 'r'),
154 Transition(45, 1, ['н'], 'not', 'r'),
155 Transition(46, 1, ['т', ' '], 'not', 'r'),
156 Transition(47, 1, ['я'], 'not', 'r'),
157 Transition(48, 1, ['б'], 'not', 'r'),
158 Transition(49, 1, ['р'], 'not', 'r'),
159 Transition(50, 1, ['я'], 'not', 'r'),
160 Transition(51, 1, [' '], 'not', 'r'),
161
162 # октября, окт // 52 — отсутствует
163 Transition(a, 53, ['о'], 'in', 'r'),
164 Transition(53, 54, ['к'], 'in', 'r'),
165 Transition(54, 55, ['т'], 'in', 'r'),
166 Transition(55, 56, ['я'], 'in', 'r'),
167 Transition(56, 57, ['б'], 'in', 'r'),
168 Transition(57, 58, ['р'], 'in', 'r'),
169 Transition(58, 59, ['я'], 'in', 'r'),
170 Transition(59, b, [' '], 'in', 'r'),
171 Transition(55, b, [' '], 'in', 'r'),
172
173 Transition(53, 1, ['к'], 'not', 'r'),
174 Transition(54, 1, ['т'], 'not', 'r'),
175 Transition(55, 1, ['я', ' '], 'not', 'r'),
176 Transition(56, 1, ['б'], 'not', 'r'),
177 Transition(57, 1, ['р'], 'not', 'r'),
178 Transition(58, 1, ['я'], 'not', 'r'),
179 Transition(59, 1, [' '], 'not', 'r'),
180
181 # ноября, ноя // 60 — отсутствует
182 Transition(a, 61, ['н'], 'in', 'r'),
183 Transition(61, 62, ['о'], 'in', 'r'),
184 Transition(62, 63, ['я'], 'in', 'r'),
185 Transition(63, 64, ['б'], 'in', 'r'),
186 Transition(64, 65, ['р'], 'in', 'r'),
187 Transition(65, 66, ['я'], 'in', 'r'),

```

```

188 Transition(66, b, [' '], 'in', 'r'),
189 Transition(63, b, [' '], 'in', 'r'),
190
191 Transition(61, 1, ['o'], 'not', 'r'),
192 Transition(62, 1, ['я'], 'not', 'r'),
193 Transition(63, 1, ['6', ' '], 'not', 'r'),
194 Transition(64, 1, ['p'], 'not', 'r'),
195 Transition(65, 1, ['я'], 'not', 'r'),
196 Transition(66, 1, [' '], 'not', 'r'),
197
198 # декабря, дек // 67 — отсутствует
199 Transition(a, 68, ['д'], 'in', 'r'),
200 Transition(68, 69, ['е'], 'in', 'r'),
201 Transition(69, 70, ['к'], 'in', 'r'),
202 Transition(70, 71, ['а'], 'in', 'r'),
203 Transition(71, 72, ['6'], 'in', 'r'),
204 Transition(72, 73, ['p'], 'in', 'r'),
205 Transition(73, 74, ['я'], 'in', 'r'),
206 Transition(74, b, [' '], 'in', 'r'),
207 Transition(70, b, [' '], 'in', 'r'),
208
209 Transition(68, 1, ['е'], 'not', 'r'),
210 Transition(69, 1, ['к'], 'not', 'r'),
211 Transition(70, 1, ['а', ' '], 'not', 'r'),
212 Transition(71, 1, ['6'], 'not', 'r'),
213 Transition(72, 1, ['p'], 'not', 'r'),
214 Transition(73, 1, ['я'], 'not', 'r'),
215 Transition(74, 1, [' '], 'not', 'r'),
216
217 # Год
218 Transition(b, 81, ['1'], 'in', 'r'), # Первая цифра года
219 Transition(b, 82, ['2'], 'in', 'r'),
220 Transition(b, 1, ['1', '2'], 'not', 'r'),
221
222 Transition(81, 83, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in', 'r')
223 , # 1000 — 1999
224 Transition(83, 84, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in', 'r')
225 ,
226 Transition(84, c, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in', 'r'),
227 Transition(81, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not', 'r')
228 ,
229 Transition(83, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not', 'r')
230 ,
231 Transition(84, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not', 'r')
232 ,
233 Transition(85, 86, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in', 'r')
234 ,
235 Transition(86, c, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'in', 'r'),
236 Transition(82, 1, ['0', '1'], 'not', 'r'),
237 Transition(85, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not', 'r')
238 ,
239 Transition(86, 1, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'], 'not', 'r')
240 ,
241 Transition(c, 1, [], 'not', 'r')
242 ]
243
244 final_state = c
245 found = run_automat(instructions, text, final_state)
246 res = []
247
248 for couple in found:
249     res.append(text[couple[0]:couple[1]])
250
251 return res

```

# Заключение

Выполнен рубежный контроль по нахождению подстроки в строке при помощи регулярных выражений и конечного автомата.