Регулярные выражения

Регулярные языки Клини

Пусть Σ – алфавит, тогда

- Пустой язык $\varnothing$ является регулярным
- Язык $\{\varepsilon\}$, состоящий из пустого слова ε, является регулярным
- Для всех $a \in \Sigma$ язык, состоящий из одной буквы $\{a\}$, регулярный
- Если L, L_1 и L_2 - регулярные языки, то регулярными будут
  - объединение $L_1 \cup L_2$

$\Sigma = \{a, b\}$
$L_1 = \{\varepsilon, bb, ba\}$
$L_2 = \{a, aab\}$
$L_1 \cup L_2 = \{\varepsilon, bb, ba, a, aab\}$

Регулярные языки Клини

Пусть Σ – алфавит, тогда

- Пустой язык $\varnothing$ является регулярным
- Язык $\{\varepsilon\}$, состоящий из пустого слова ε, является регулярным
- Для всех $a \in \Sigma$ язык, состоящий из одной буквы $\{a\}$, регулярный
- Если L, L_1 и L_2 - регулярные языки, то регулярными будут
  - объединение $L_1 \cup L_2$
  - конкатенация $L_1 \bullet L_2$

$\Sigma = \{a, b\}$
$L_1 = \{\varepsilon, bb, ba\}$
$L_2 = \{a, aab\}$
$L_1 \bullet L_2 = \{a, aab, bba, bbaab, baa, baaab\}$

Регулярные языки Клини

Пусть Σ – алфавит, тогда

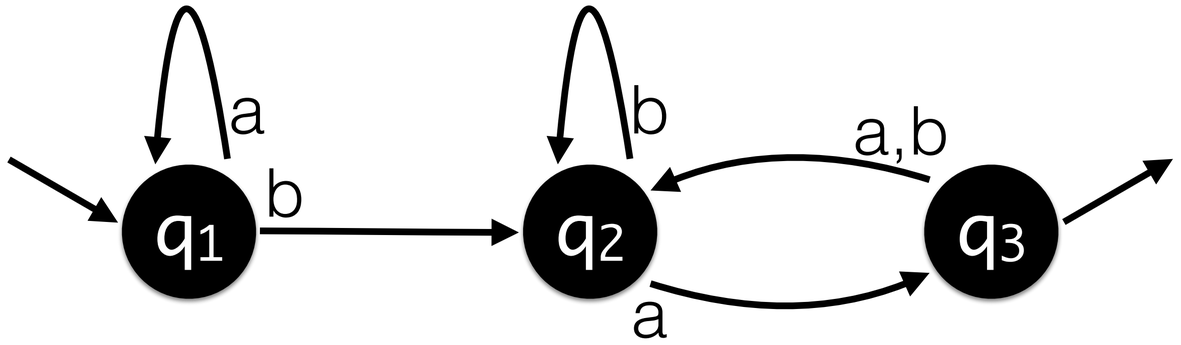
- Пустой язык $\varnothing$ является регулярным
- Язык $\{\varepsilon\}$, состоящий из пустого слова ε, является регулярным
- Для всех $a \in \Sigma$ язык, состоящий из одной буквы $\{a\}$, регулярный
- Если L, L_1 и L_2 – регулярные языки, то регулярными будут
  - объединение $L_1 \cup L_2$
  - конкатенация $L_1 \bullet L_2$
  - результат применения звезды Клини L^*

$\Sigma = \{a, b\}$
$L = \{ab, b\}$
$L^* = \{\varepsilon, ab, b, abab, bab, abb, bb, (ab)^3, \ldots\}$

Регулярные языки Клини

Регулярный язык ⟷ Детерминированный конечный автомат

a*bb*a((a|b)b*a)* ⟷



Регулярные выражения в Python

Проблемы записи регулярных выражений

```
len('\\') == 1
len(r'\\') == 2
```

Raw strings

Задачи, решаемые регулярными выражениями

- Проверка соответствия строки шаблону
- Извлечение данных из строки по шаблону
- Изменение данных в строке, подходящих под шаблон

Модуль **re**

Специальные символы в записи регулярных выражений

Специальные символы в записи регулярных выражений

- **.** . ~ a, b, c, … \n (re.DOTALL)
- ***** a^* ~ ε, a, aa, aaa, …; ab^* ~ a, ab, abb, …; $(ab)^*$ ~ ε, ab, abab
- **+** $a+ \equiv aa^*$
- **?** ab?c ~ ac, abc

Про жадность

- ***** → ***?**
- **+** → **+?**
- **?** → **??**

Специальные символы в записи регулярных выражений

- **{m}** a{3} ~ aaa
- **{m, n}** a{2, 4} ~ aa, aaa, aaaa
- **{m,}** a{2,} ≡ aa+
- **{,n}** a{,3} ~ ε, a, aa, aaa
- **[…]** [abc] ~ a, b, c; [0-9] ~ 0, 1, …, 9; [_0-9a-zA-Z]; [^abc]
- **|** abc | de | f ~ abc, de, f

Специальные символы в записи регулярных выражений

- **()**
- **^**
- **$**

Специальные символы в записи регулярных выражений

- **\…**

\d ≡ [0-9], \D ≡ [^0-9]

\s ≡ [\t\n\r\f\v], \S ≡ [^\s]

\w ≡ [a-zA-Z0-9_], \W ≡ [^\w]

\1, \2, \3, …

\\, \., *, …

Специальные символы в записи регулярных выражений

- **(?:…)** non-grouping
- **(?=…)** positive lookahead
- **(?!…)** negative lookahead
- **(?<=…)** positive lookbehind
- **(?<!…)** negative lookbehind

Вопросы производительности

```
for string in very_big_data:
    re.search(r'something interesting', string)
```

```
regex = re.compile(r'something interesting’)
for string in very_big_data:
    regex.search(string)
```

Извлечение данных из строки по шаблону

Изменение данных в строке, подходящих под шаблон