文本数据管理与分析

复旦大學
FUDAN UNIVERSITY

正则表达式
--语言的形式化描述

邱锡鹏
复旦大学
http://nlp.fudan.edu.cn/xpqiu

需求

- 文本处理中的常见需求
  - 匹配
    - *天气*
  - 抽取
    - 我要买明天从北京到上海的机票
  - 数据验证
    - Email的合法性
    - 密码
  - 替换
    - 替换所有数字

如何描述规则！

语言

- 语言是在一个特定的字符集上，通过一定的组合规则产生的字符序列的集合。
  - 有限字母表（词表）
  - 英文
    - 英文字母
  - 中文
    - 汉字
  - 不是任何字符串都符合语言规则

让一只猴子在打字机上随机地按键，当按键时间达到无穷时，几乎必然能够打出任何给定的文字，比如莎士比亚的全套著作。

--无限猴子定理

形式语言

- 文法分成四种类型
  - 无限制文法
  - 上下文相关文法
  - 上下文无关文法
  - 正规文法

Avram Noam Chomsky
1928年12月7日 —

字符串的属性

- 字符串(String) 是零个或多个字符组成的有限序列。
  - E.g. tech是长度为4的字符串
- 属性
  - 长度
  - 空字符串ϵ
  - 前缀
  - 后缀
  - 子串
  - 子序列

字符串操作-拼接

- 如果x和y是字符串，那么x和y的拼接（concatenation）xy是将y 附加在x 的后面。
  - x 为ba，y 为na，则xy 为bana，xyy 为banana。
- 对于一个字符串和它自身的连接，我们可以用指数运算来表示。
  - $x^2 = xx$, $x^3 = xxx$, etc. $x^0 = \varepsilon$.
  - $xy^2 = banana$.

字符串集合操作

- 字符串集合（语言）操作
  - 并
  - 拼接
  - 闭包（Closure）

字符串集合操作

- 并
  - L U M { s | s is in L or in M}
- 拼接
  - LM {st | s is in L and t is in M}
- Kleen闭包
  - $L^* = \{\epsilon\} \cup L \cup LL \cup LLL \cup LLLL \cup \cdots$.
- 正则闭包
  - $L^+ = L \cup LL \cup LLL \cup LLLL \cup \cdots$.

例子

- L = {A,B,⋯.,Z,a,b,⋯,z}
- D = {0,1,2,⋯.,9}

  - L∪D
  - LD
  - L4 = LLLL
  - L^*
  - L(L ∪ D)*
  - D

正则表达式（Regular Expression）

- 正则表达式由字母表和算子组成，可以表示字符串的集合和在这些集合上的运算。
- 字母表
  - {0,1}
  - ASCII
  - 英文字母
  - 汉字
- 运算
  - 并、连接、闭包

https://zh.wikipedia.org/wiki/正则表达式

正则表达式的递归定义

- ε 是正则表达式
  - 代表{ε}
- 如果a字母表Σ中的符号，则a是正则表达式
  - 代表{a}
- 如果r和s是正则表达式，则
  - r|s是正则表达式，代表L(r) U L(s)
  - rs是正则表达式，代表L(r)L(s)
  - r*是正则表达式，代表(L(r))*

优先级

- * > 拼接 > |
- 所有操作是左结合的

- 例子
  - a(b|c)*d|e

简写

- 如果r是正则表达式，则
  - r+ 表示r r*
  - r?表示r | Є
  - .表示any chars
  - a-z表示a 到z中的所有字母

例子：电话号码

- (+86)-21-65642222

- d=0-9
- nation= $+d^2$
- area = d^2
- phone = d^8
- phone_number = '(' nation')-' area'-' phone

例子：Email

- anyone@fudan.edu
- anyone@fudan.edu.cn

- l=A-Z | a-z
- str = l^+
- address = str '@' str '. str ('.' str)?

例子：无符号数

- d=0-9

- digits = d^+
- opt_frac = '.' digits | Є
- opt_exp → (E('+' | '-' | Є) digits) | Є
- num → digits opt_frac opt_exp

题目

- 构造四个正则表达式用于验证“密码字符串”，依次满足如下要求:
  - 1. 只能由大小写字母、数字和横线（-）组成;
  - 2. 满足条件1，并且开头和结尾不允许是横线;
  - 3. 满足条件2，并且不允许有连续（超过一个）的横线。
  - 4. 满足条件3，并且不允许全部是数字;

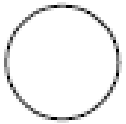
如何实现？

- 有限状态自动机 Finite Automata

- $\{Q, \sum, \delta, q_0, F\}$
  - Q状态集合
  - $\sum$输入符号集合
  - f 状态转移函数 $Q \times \sum \rightarrow Q$
  - q_0, δ, 起始状态和终止状态

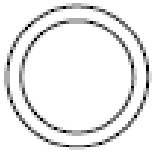
有限状态自动机

- A state

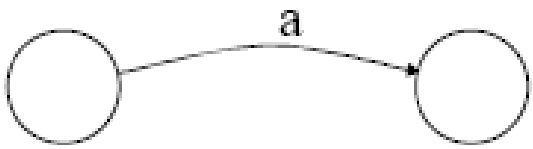


- The start state

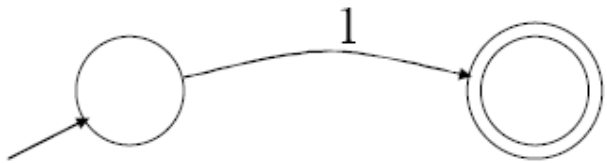
- An accepting state

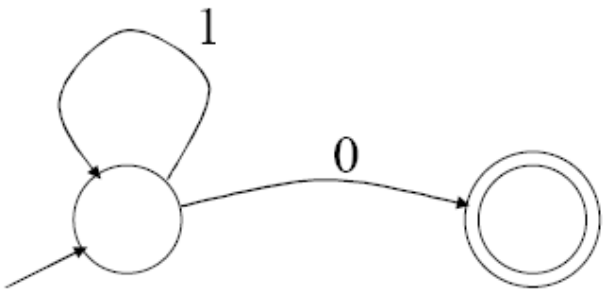


- A transition

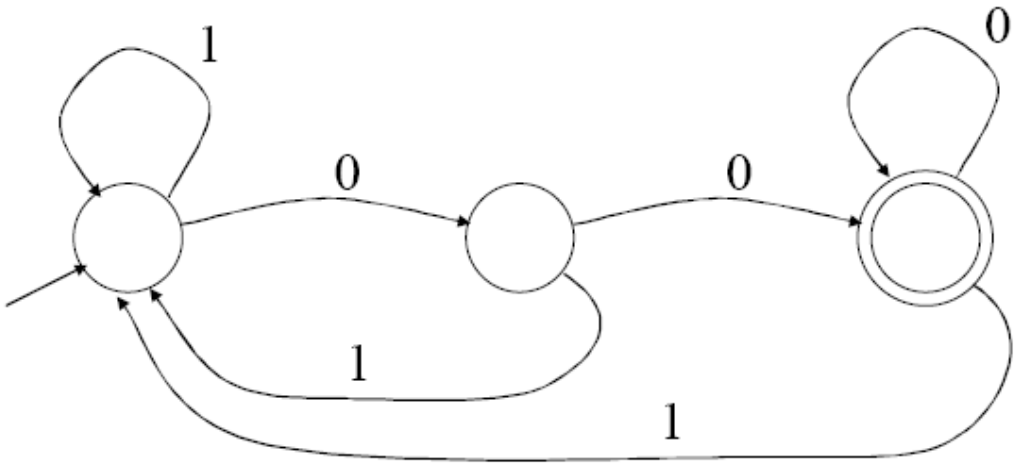


例子





例子

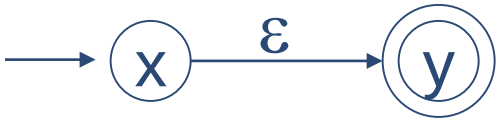


相互转换

- 正则表达式到自动机
- 自动机到正则表达式

正则表达式到自动机

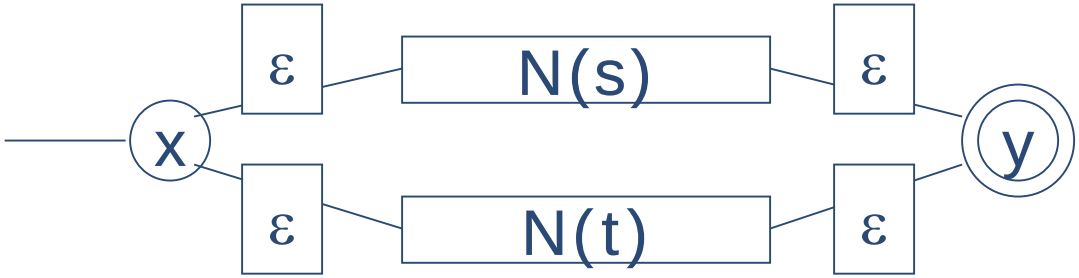
- For ε,



- For a,

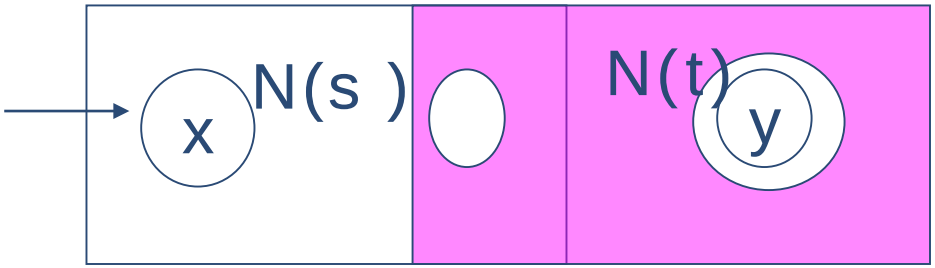
正则表达式到自动机

- For s | t,



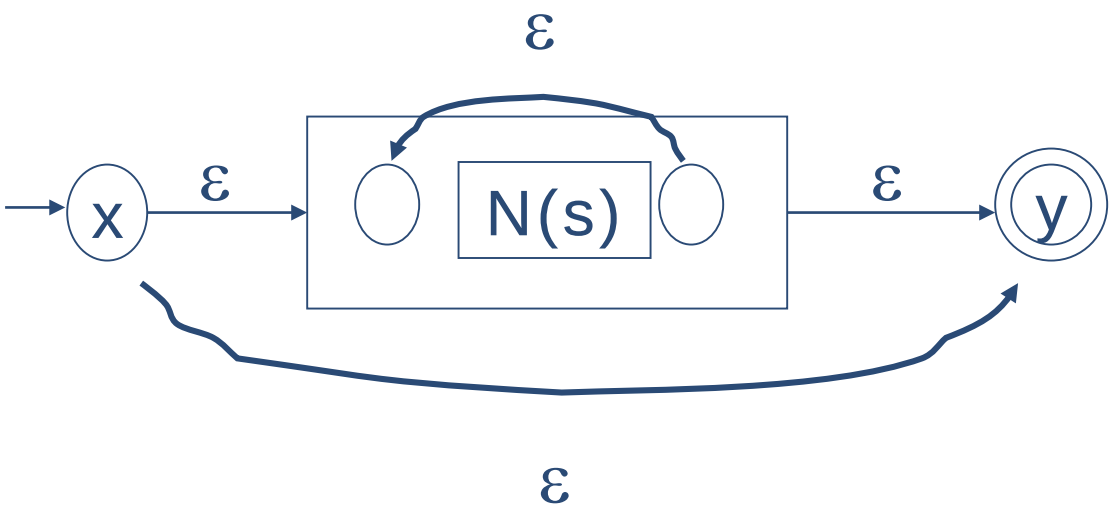
正则表达式到自动机

- For st,



正则表达式到自动机

- For s*,



例子

- (a|b)*abb

正则表达式引擎

- 正则表达式引擎分为两类
  - NFA (Nondeterministic Finite Automata，非确定型有穷状态自动机)
  - DFA (Deterministic Finite Automaton，确定型有穷状态自动机)

- 在Java中，使用NFA和DFA结合方法。

正则表达式的特点

- 灵活性、逻辑性和功能性非常强；
- 可以迅速地用极简单的方式达到字符串的复杂控制。

正则表达式的不足

- 可读性不高。当描述一个复杂的语言集合时，正则表达式的可读性就变得很差。
- 描述能力有限。
  - 比如对与语言集合anban（n为变量）。正则表达式并不能描述这个集合。
- 正则表达式只适合匹配文本字面，不适合匹配文本意义)

正则表达式的不足

- 容易引起性能问题
  - 像.*这种贪婪匹配符号很容易造成大量的回溯，性能有时候会有上百万倍的下降，编写好的正则表达式要对正则引擎执行方式有很清楚的理解才可以。
- 正则的替换功能较差

自然语言的文法

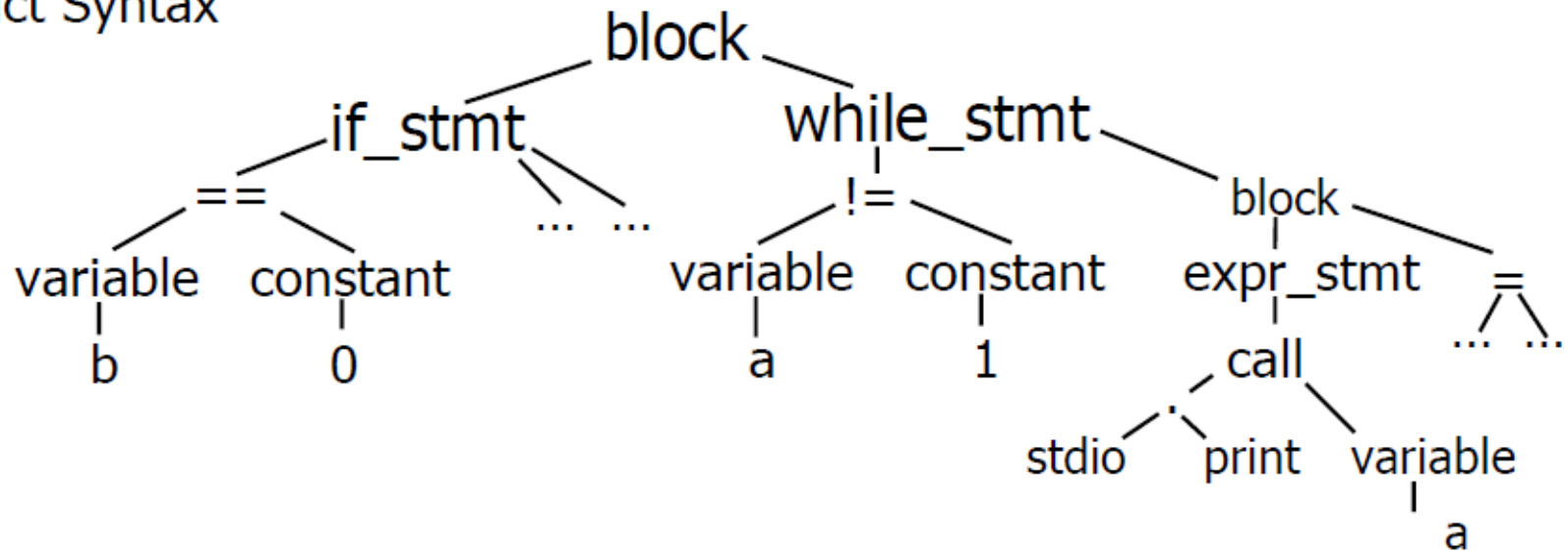

"I gave him the book"
sentence
subject: I
verb:gave
indirect object:
him
object
noun phrase
article: the
noun: book

程序语言的文法

Source code
(token stream)

```
{
    if (b == (0)) a = b;
    while (a != 1) {
        stdio.print(a);
        a = a - 1;
    }
}
```

Abstract Syntax Tree



上下文无关文法

- 上下文无关文法
  - 终止符 terminals T
  - 非终止符non-terminals N
  - 开始符号S (非终止符)
  - 产生式 productions r

- $X \in N$
  - $X \rightarrow \varepsilon$, or
  - $X \rightarrow Y1\ Y2 \ldots Yn$ where $Yi \subseteq N \cup T$

如何生成语言？

- 替换规则
  - $X \rightarrow Y1 \ldots Yn$
    - 表示X可以被Y1 ... Yn替换
  - $X \rightarrow \varepsilon$
    - 表示X可以被删除
- 生成语言
  1. 由开始符号“S”开始
  2. 替换其中的非终止符X
  3. 重复(2)，直到字符串中全部为终止符

文法G对应的语言

- 文法G的语言为：

$$\{ a_1 \cdots a_n \mid S \Rightarrow a_1 \cdots a_n \}$$

  - 其中，a_i 为终止符

例子：算术表达式

$$
\begin{aligned}
E &\rightarrow E * E \\
&\mid E + E \\
&\mid (E) \\
&\mid id
\end{aligned}
$$

例子：匹配的括号

- $\{ (^i)^i \mid i \geq 0\}$

- 文法
  - $S \rightarrow (S)$
  - $S \rightarrow \varepsilon$

总结

- 正则表达式是文本处理中很重要的技术
  - 几乎用在所有的文本处理系统
- 不足
  - 需要一定的专家知识，维护成本很高
  - 不能处理自然语言的歧义现象
  - 只能描述部分语言现象，只能在单一、封闭的任务中使用
- 对于更难的任务，需要使用更加复杂的方法
  - 上下文无关文法
  - 机器学习（正则表达式也可以作为一种特征使用）

谢 谢