Regular Expressions

Thomas Schwarz, SJ

Regular Expressions

- Two related concepts:
  - Used for searching:
    - Module re in Java and Python, regex in C++
  - In theoretical computer science:
    - A way to describe a large set of languages
    - Used e.g. in compiler theory
  - Nota bene: Regular expressions as implemented in most regular expression packets are more powerful than regular expressions as described here

Languages

- Theory of Computation can be based on the notion of a language
    - Imagine a language to be with which we can calculate
    - Fortunately, this is quite intuitive if you do not mind "abstract nonsense".

Languages

- Consists of finite length strings over a finite length alphabet Σ .
- Think about a C-program:
  - Written with characters that can be produced by an English keyboard
- Set of finite length strings over Σ is called Σ^*
- Regular expressions describe subset of the set of finite length strings

Languages

- Given languages (subsets of Σ^*) we can define operations
  - Concatenation: (with . denoting concatenation of strings)
    $L_1, L_2 \subset \Sigma^* : L_1 \cdot L_2 := \{x \,.\, y \,|\, x \in L_1, y \in L_2\}$
  - Powers: defined inductively
    - $L \subset \Sigma^*$: $\quad L^0 = \{\epsilon\}$ (set with empty string)
    - $L^1 = L$
    - $L^{n+1} = L^n \cdot L$

Languages

- The Kleene Closure is for $L \subset \Sigma^*$:
  - $L^* = \cup_{i \in \mathbb{N}} L^i$
- This is the set of all strings that can be formed by concatenating elements in L
  - It includes the empty string ϵ

Regular Expressions

- Regular expressions describe "regular" languages over a finite alphabet Σ
- They are defined inductively
  - The empty set $\emptyset$ is a regular expression
  - The empty string ϵ is a regular expression
    - And these things are different

Regular Expressions

- Singletons are regular expressions:
  - Let $a \in \Sigma^*$. We define a regular expression $\mathbf{a}$ that stands for the set $\{a\}$.
  - It is customary to denote the regular expression in bold-face to distinguish it from the element, but not everyone does that

Regular Expressions

- If r and s are regular expressions denoting the sets R and S respectively, then the following are also regular expressions:
  - $r + s$ for $R \cup S$ (union)
  - rs for $R \cdot S$ (concatenation)
  - $r*$ for $R*$ (clean closure)

Regular Expression Examples

- Let $\Sigma = \{0,1\}$
- $\mathbf{01}$ is $\{01\}$, the set consisting of the string 01.
- $\mathbf{0} + \mathbf{1}$ is $\{0,1\}$, the set consisting of the two strings 0 and 1
- $(\mathbf{0} + \mathbf{1}) \cdot \mathbf{0} \cdot \mathbf{1} \cdot (\mathbf{0} + \mathbf{1})$ is the set $\{0010,0011,1010,1011\}$
- $\mathbf{1}* = \{\epsilon,1,11,111,1111,\ldots\}$ the set of strings that can be written with a finite number of ones, including none
- $\mathbf{1}* \cdot \mathbf{0} \cdot \mathbf{1}*$, the set of strings with any number of ones, but exactly one zero.

Regular Expressions Examples

- As an abbreviation, we use $L^+ = \cup_{i=1\ldots\infty} L^i$
  - This just excludes the empty string ϵ
- $\mathbf{1}^+ \cdot \mathbf{00} \cdot \mathbf{1}*$: All strings that start out with at least one one, followed by a double zero, followed by none or several ones
- $\mathbf{01}^+ = \{01,011,0111,01111,\ldots\}$: all strings that consists of a single 0 followed by at least one one
- $(\mathbf{01})^+ \qquad = \{01,0101,010101,01010101,\ldots\}$: all strings that start out with a zero, followed by a 1, followed possibly by another zero followed by a one, etc.