Automata and Formal Languages

Regular Expressions

Expressions in arithmetic: $(5 + 8) \cdot 3$

We can use the **regular operations** to build expressions describing **languages**.

The value of a regular expressions is a **language**.

Example:

$(0 \cup 1) \cdot 0*, \qquad \Sigma = \{0,1\}$

First part: $\{0\} \cup \{1\} = \{0,1\}$

Second part: $\{0\}*$

Concatenate first with second part:
$\{0,1\} \cdot \{\epsilon, 0, 00, \ldots\} = \{w \mid w$ such that the string is made up by a one or zero followed by any amount of 0s$\}$

In applications involving text, you might want to search for strings satisfying certain **patterns**.

$(0 \cup 1)*$

$\{0\} \cup \{1\} = \{0,1\}$

$\{0,1\} = \Sigma^1, \qquad \Sigma = \{0,1\}$

Shorthand: *Given* $\Sigma = \{0,1\}$*, we use* Σ *as a shorthand for the regular expression* $0 \cup 1$

More examples:

$(\Sigma*)1 \implies$ All the possible strings of 0s and 1s ending with a 1

$(0\Sigma*) \cup (\Sigma * 1) \implies$ All strings that start in 0 or end in 1

$0\Sigma * 1 = 0\Sigma * \Sigma * 1$

$\Sigma 101 \Sigma \implies \{01010, 11010, 01011, 11011\}$

$\Sigma * 101 \Sigma * \implies$ all strings containing 101

$\Sigma(1 \cup \epsilon) \implies (1 \cup 0)(1 \cup \epsilon) = \{0, 1, 01, 11\}$

$1 * \cup 0 * \implies$ all strings containing only 1s or only 0s

$L = \varnothing$ is the complement of $\Sigma *$, $L \neq \{\epsilon\}$

Formal Definition of a RegEx

R is a regular expressions over an alphabet Σ if R is:

1. a, for some a in the alphabet Σ
2. ϵ
3. $\varnothing$
4. $R1 \cup R2$
5. $R1R2$
6. $R1 *$, where R1 and R2 are regular expressions

Example: Numerical Constant

$DD *, \qquad D = \{0,1,2,3,4,5,6,7,8,9\}$

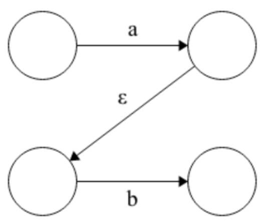
Equivalence with Finite Automata

A language is regular if and only if a RegEx describes it.

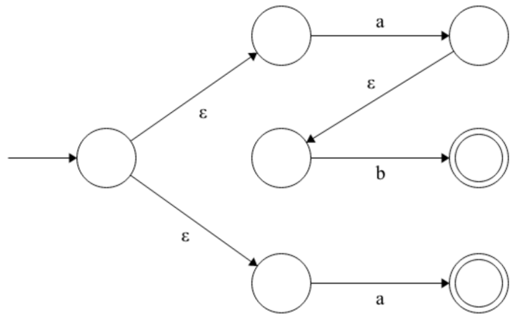
Construction of NFA given a RegEx

$(ab \cup a) * =$

$\{a\} \cdot \{b\}$



$\{a, b\} \cup \{a\}$



$(ab \cup a)*$