MCS-265: The Theory of Computation Handout #R3
San Skulrattanakulchai
Gustavus Adolphus College February 27, 2013

Regular Expressions

Sipser Ch 1: p63–66

Regular Expressions (R.E.'s)

Given an alphabet Σ, a *regular expression* over Σ is recursively defined as follows.

1. Each $a \in \Sigma$ is a regular expression.
2. $\emptyset$ is a regular expression.
3. ε is a regular expression.
4. If R_1 and R_2 are regular expressions, then $(R_1 \cup R_2)$ is a regular expression.
5. If R_1 and R_2 are regular expressions, then $(R_1 \circ R_2)$ is a regular expression.
6. If R is a regular expression, then (R^*) is a regular expression.

Something is a regular expression if and only if it follows from one of the above rules.
To make regular expressions easy to write and also unambiguous, we

- use juxtaposition instead of $\circ$,
- declare that * has higher precedence than $\circ$, and that $\circ$ has higher precedence than $\cup$, and omit enclosing parentheses when possible, and
- retain pairs of enclosing parentheses only when needed to override the default precedence rules.

Therefore, 01^* means $(0 \circ (1^*))$, which is different from $((0 \circ 1)^*)$. Similarly, $10 \cup 01$ means $((1 \circ 0) \cup (0 \circ 1))$, which is different from $((1 \circ (0 \cup 0)) \circ 1)$ or $(1 \circ ((0 \cup 0) \circ 1))$.
We associate each R.E. R with its language $L(R)$ as follows.

1. Each $a \in \Sigma$ is associated with $\{a\}$.
2. $\emptyset$ is associated with $\emptyset$

3. ε is associated with $\{\varepsilon\}$.

4. If $L(R_1)$ is the language of R_1 and $L(R_2)$ is the language of R_2, then $L(R_1) \cup L(R_2)$ is the language of $(R_1 \cup R_2)$.

5. If $L(R_1)$ is the language of R_1 and $L(R_2)$ is the language of R_2, then $L(R_1) \circ L(R_2)$ is the language of $(R_1 \circ R_2)$.

6. If $L(R)$ is the language of R, then $L(R)^*$ is the language of (R^*).

Exercise.

1. Show that the rule "ε is an regular expression" is superfluous.

Notes.

1. For an R.E. R and nonnegative integer n, R^+ is short for $(R \circ (R^*))$, and R^n is short for n copies of R's concatenated (in any order!).

2. The three operations $\cup$, $\circ$ and * on languages are termed *regular operations*. A language representable by an R.E. is called a *regular language*.