4 Regular Expressions

4.1 Recall

- Discussion of Closure Properties
  1. Complement
  2. Reverse
  3. Union
  4. Concatenation
  5. Star
- Brief discussion of Regular Expressions

4.2 Plan

- Discuss Exercise 2
- Regular Expressions
- All language described by regular expressions are regular
- All regular languages are described by regular expressions

4.3 Exercise 2 discussion: Symbols, Strings, and Languages

Unary numbers. How do we add unary numbers (concatenation). How do we represent 0 in unary? This is why we have $\epsilon$!

A symbol is taken from a finite, non-empty set called an alphabet.

A string is a finite sequence of symbols. The string of length 0 is written $\epsilon$.

A language is a set of strings over a common alphabet.

A machine can be seen to accept or reject a string. In doing so we say it recognizes a language. The language recognized by the machine is the set of strings accepted.
4.4 Regular Expressions

Definition. (Example of inductive)

The atomic regular languages over $\Sigma$ are:

1. For some $a \in \Sigma$: The language containing exactly the singleton string $a$.
2. $\epsilon$: The language containing exactly the empty string.
3. $\emptyset$: The language containing no strings.

If $R_1$ and $R_2$ are regular expressions, the following composite forms are regular expressions:

- $R_1 \cup R_2$
- $R_1 \circ R_2$
- $R_1^*$

Examples.

Conventions. $R^+ = R \circ R^*$. Sometimes use $+$ instead of $\cup$. Sometimes use juxtaposition for concatenation. That is, $RS$ is read formally as $R \circ S$.

Algebra. Explore the analogy:

<table>
<thead>
<tr>
<th>0</th>
<th>$\emptyset$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>+</td>
<td>$\cup$</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\circ$</td>
</tr>
</tbody>
</table>

Look at rules for identity, anihilation, distributive laws, associativity, commutativity. What works? What doesn’t?

Thm: Every language described by a REGEXP is regular

What about the other way?