Lecture #7: Regular Expressions and Regular Operations
Lecture Presentation

The following definitions were given in the lecture notes. They will be useful when solving problems that concern regular expressions.

Definition: A string of symbols ω is a ***regular expression*** for the alphabet Σ if and only if it can be formed using a finite number of applications of the following rules.

1. $\omega = \sigma$, for some symbol $\sigma \in \Sigma$.
2. $\omega = \lambda$.
3. $\omega = \emptyset$.
4. $\omega = \Sigma$.
5. If R_1 and R_2 are both regular expressions over the alphabet Σ, and ω is the string $(R_1 \cup R_2)$, then ω is regular expression over the alphabet Σ as well.
6. If R_1 and R_2 are both regular expressions over the alphabet Σ, and ω is the string $(R_1 \circ R_2)$, then ω is regular expression over the alphabet Σ as well.
7. If R_1 is a regular expression over the alphabet Σ, and ω is the string $(R_1)^\star$, then ω is a regular expression over the alphabet Σ as well.

Definition: If ω is a regular expression for the alphabet Σ then the ***language*** $L(\omega)$ of ω is as follows.

1. If $\omega = \sigma$, for $\sigma \in \Sigma$, then $L(\omega) = L(\sigma) = \{\sigma\}$.
2. If $\omega = \lambda$ then $L(\omega) = L(\lambda) = \{\lambda\}$.
3. If $\omega = \emptyset$ then $L(\omega) = L(\emptyset) = \emptyset$.
4. If $\omega = \Sigma$ then $L(\omega) = L(\Sigma) = \Sigma$ (the set of all strings in $\Sigma^\star$ with length one).
5. If ω is the string $(R_1 \cup R_2)$ where R_1 and R_2 are regular expressions over Σ then the language $L(\omega)$ of ω is the set $L(R_1) \cup L(R_2)$.
6. If ω is the string $(R_1 \circ R_2)$ where R_1 and R_2 are regular expressions over Σ then the language $L(\omega)$ of ω is the set $L(R_1) \circ L(R_2)$.
7. If ω is the string $(R_1)^\star$ where R_1 is a regular expression over Σ then the language $L(\omega)$ of ω is the set $(L(R_1))^\star$.

Interpretation of Regular Expressions

Consider the following regular expression, ω, over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$\omega = ((((\Sigma)^{\star} \circ \mathtt{a}) \circ (\Sigma)^{\star}) \circ (\mathtt{a} \circ (\Sigma)^{\star}))$$

Note that $\omega = (\omega_1 \circ \omega_2)$ where

$$\omega_1 = (((\Sigma)^{\star} \circ \mathtt{a}) \circ (\Sigma)^{\star}) \qquad \text{and} \qquad \omega_2 = (\mathtt{a} \circ (\Sigma)^{\star}).$$

If ω_1 and ω_2 are regular expressions over Σ, then it follows by part 6, of the definition of a regular expression over Σ, that ω is are regular expression over Σ as well.

In order to ***confirm that ω is a a regular expression over Σ***, we might proceed as follows:

Now recall that

(a) Σ is, by definition, a regular expression over Σ, and the ***language*** $L(\Sigma)$ of this regular expression is the *language* (that is, *set*) $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

(b) $(\Sigma)^\star$ is a regular expression over Σ, since Σ is, and the ***language*** of $(\Sigma)^\star$ is the Kleene star of the language of Σ. This is the Kleene star of the set $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, that is, the set (which we already call $\Sigma^\star$) of all strings over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

We might continue, as follows, in order to ***identify the language of the regular expression ω:***

Development of Regular Expressions

Once again, let $\Sigma = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$ and consider the language $L \subseteq \Sigma^\star$ the includes all strings over Σ with an even number of copies of "a" — that is,

$$L = \{\mu \in \Sigma^\star \mid \text{the number of copies of "a" in } \mu \text{ is divisible by } 2\}.$$

Suppose that we want to ***discover a regular expression ω over Σ such that $L(\omega) = L$.***

- It can be helpful to ***examine the description of the language*** (rephrasing it, if needed, as long you do not change its meaning) to try to identify ***simpler languages that can be used to produce the desired one, using one of the regular operations***.

- Notice that another way to write that "the number of copies of 'a' in ω is divisible by 2" is to write that "for some non-negative integer k,

  $$\mu = \mu_0 \circ (\mu_1 \cdot \mu_2 \cdot \cdots \cdot \mu_k)$$

  where μ is some string in $\Sigma^\star$ that does not include any copies of 'a', and $\mu_1, \mu_2, \dots, \mu_k$ are all strings in $\Sigma^\star$ that include exactly two copies of 'a' ".

  In other words, $L = \widetilde{L} \circ (\widehat{L})^\star$, where $\widetilde{L}$ is the set of strings in $\Sigma^\star$ that do not include any copies of "a", and $\widehat{L}$ is the set of strings in $\Sigma^\star$ that include exactly two copies of "a".

  It follows that if $\widetilde{\omega}$ and $\widehat{\omega}$ are regular expressions over Σ such that $\widetilde{\omega} = \widetilde{L}$ and $L(\widehat{\omega}) = \widehat{L}$, and we set ω to be $(\widetilde{\omega} \circ (\widehat{\omega})^\star)$, then ω is a regular expression over Σ such that

  $$L(\omega) = L(\widetilde{\omega}) \circ (L(\widehat{\omega}))^\star = \widetilde{L} \circ (\widehat{L})^\star = L,$$

  as desired. We have now identified two (one hopes, simpler) goals: Discover a regular expression $\widetilde{\omega}$ over Σ whose language is the language

  $$\widetilde{L} = \{\mu \in \Sigma^\star \mid \text{there are no copies of "a" in } \mu\},$$

  and discover a regular expression $\widehat{\omega}$ over Σ whose language is the language

  $$\widehat{L} = \{\mu \in \Sigma^\star \mid \text{there are exactly two copies of "a" in } \mu\}.$$

This process could proceed as follows: