Section 11.1 Regular Languages

Problem: Suppose the input strings to a program must be strings over the alphabet \( \{a, b\} \) that contain exactly one substring \( bb \). In other words, the strings must be of the form \( xby \), where \( x \) and \( y \) are strings over \( \{a, b\} \) that do not contain \( bb \), \( x \) does not end in \( b \), and \( y \) does not begin with \( b \). In a few minutes, we’ll see how to describe the strings formally.

A regular language over alphabet \( A \) is a language constructed by the following rules:

- \( \emptyset \) and \( \{\Lambda\} \) are regular languages.
- \( \{a\} \) is a regular language for all \( a \in A \).
- If \( M \) and \( N \) are regular language, then so are \( M \cup N \), \( MN \), and \( M^* \).

Example. Let \( A = \{a, b\} \). Then the following languages are a sampling of the regular languages over \( A \):

\[ \emptyset, \{\Lambda\}, \{a\}, \{b\}, \{a, b\}, \{ab\}, \{a\}^* = \{\Lambda, a, aa, aaa, \ldots, a^n, \ldots\} \].

A regular expression over alphabet \( A \) is an expression constructed by the following rules:

- \( \emptyset \) and \( \Lambda \) are regular expressions.
- \( a \) is a regular expression for all \( a \in A \).
- If \( R \) and \( S \) are regular expressions, then so are \( (R), \ R+S, \ R\cdot S, \) and \( R^* \).

The hierarchy in the absence of parentheses is, * (do it first), \( \cdot \), + (do it last). Juxtaposition will be used in place of \( \cdot \).

Example. Let \( A = \{a, b\} \). Then the following expressions are a sampling of the regular expressions over \( A \):

\[ \emptyset, \Lambda, a, b, ab, a + ab, (a + b)^* \].
Regular expressions represent regular languages
Regular expressons represent regular languages by the following correspondence, where
$L(R)$ denotes the regular language of the regular expression $R$.

$L(\emptyset) = \emptyset,$
$L(\Lambda) = \{\Lambda\},$
$L(a) = \{a\}$ for all $a \in A,$
$L(R + S) = L(R) \cup L(S),$ 
$L(RS) = L(R)L(S),$ 
$L(R^*) = L(R)^*.$

Example. The regular expression $ab + a^*$ represents the following regular language:

$L(ab + a^*) = L(ab) \cup L(a^*)$
$= L(a)L(b) \cup L(a)^*$
$= \{a\}\{b\} \cup \{a\}^*$
$= \{ab\} \cup \{\Lambda, a, aa, aaa, \ldots, a^n, \ldots \}$
$= \{ab, \Lambda, a, aa, aaa, \ldots, a^n, \ldots \}.$

Example. The regular expression $(a + b)^*$ represents the following regular language:

$L((a + b)^*) = (L(a + b))^* = \{a, b\}^*, $ the set of all possible strings over $\{a, b\}.$

Back to the Problem: Suppose the input strings to a program must be strings over the alphabet $\{a, b\}$ that contain exactly one substring $bb.$ In other words, the strings must be of the form $xbby$, where $x$ and $y$ are strings over $\{a, b\}$ that do not contain $bb,$ $x$ does not end in $b,$ and $y$ does not begin with $b.$ How can we describe the set of inputs formally?

Solution: let $x = (a + ba)^*$ and $y = (a + ab)^*.$
Quiz. Find a regular expression for \( \{ab^n \mid n \in \mathbb{N}\} \cup \{ba^n \mid n \in \mathbb{N}\} \).
Answer. \( ab^* + ba^* \).

Quiz. Use a sentence to describe the language of \((b + ab)^* (\Lambda + a)\).
Answer. All strings over \( \{a, b\} \) whose substrings of \( a \)'s have length 1.

The Algebra of Regular Expressions

Equality: Regular expressions \( R \) and \( S \) are equal, written \( R = S \), when \( L(R) = L(S) \).
Examples. \( a + b = b + a, a + a = a, aa^* = a^*a, ab \neq ba \).

Properties of +, \( \cdot \), and closure

+ is commutative, associative, \( \emptyset \) is identity for +, and \( R + R = R \).
\( \cdot \) is associative, \( \Lambda \) is identity for \( \cdot \), and \( \emptyset \) is zero for \( \cdot \).
\( \cdot \) distributes over +
(closure properties)

\( \emptyset^* = \Lambda^* = \Lambda \).
\( R^* = R^*R^* = (R^*)^* = R + R^* \).
\( R^* = \Lambda + R^* = \Lambda + RR^* = (\Lambda + R)^* = (\Lambda + R)R^* \).
\( R^* = (R + R^2 + \ldots + R^k)^* = R + R^2 + \ldots + R^{k-1} + R^kR^* \) for any \( k \geq 1 \).
\( R^*R = RR^* \).
\( (R + S)^* = (R^* + S^*)^* = (R^*S^*)^* = (R^*S)^*R^* = R^*(SR^*)^* \).
\( R(SR)^* = (RS)^*R \).
\( (R^*S)^* = \Lambda + (R + S)^*S \) and \( (RS^*)^* = \Lambda + R(R + S)^* \).

Proof: All properties can be verified by showing that the underlying regular languages are equal as sets. QED.
**Quiz.** Explain each inequality.

(1) \((a + b)^* \neq a^* + b^*\).  
(2) \((a + b)^* \neq a^*b^*\).

**Answers.**  
(1) \(ab \in L((a + b)^*) - L(a^* + b^*)\).  
(2) \(ba \in L((a + b)^*) - L(a^*b^*)\).

**Quiz.** Simplify the regular expression \(aa(b^* + a) + a(ab^* + aa)\).

**Answer.**  
\[ aa(b^* + a) + a(ab^* + aa) = aa(b^* + a) + aa(b^* + a) \]  
\[ = aa(b^* + a) \quad \text{· distributes over +} \]  
\[ = aa(b^* + a) \quad R = R + R. \]

**Example/Quiz.** Show that \((a + aa)(a + b)^* = a(a + b)^*\).

**Proof:**  
\[(a + aa)(a + b)^* = (a + aa)a^*(ba^*)^* \]  
\[= a(\Lambda + a)a^*(ba^*)^* \]  
\[= a\Lambda a^*(ba^*)^* \]  
\[= a^*(ba^*)^* \]  
\[= a(a + b)^* \]  
\[= (R + S)^* = R^*(SR^*)^* \]  
\[R = R\Lambda \text{ and · distributes over +} \]
\[ (\Lambda + R)R^* = R^* \]  
\[R^* = (R + S)^* = R^*(SR^*)^* \quad \text{QED.} \]

**Example/Quiz.** Show that \(a^*(b + ab^*) = b + aa^*b^*\).

**Proof:**  
\[a^*(b + ab^*) = (\Lambda + aa^*)(b + ab^*) \]  
\[= b + ab^* + aa^*b + aa^*ab^* \]  
\[= b + (ab^* + aa^*ab^*) + aa^*b \]  
\[= b + (\Lambda + aa^*)ab^* + aa^*b \]  
\[= b + a^*ab^* + aa^*b \]  
\[= b + aa^*b^* + aa^*b \]  
\[= b + aa^*(b^* + b) \]  
\[= b + aa^*b^*. \]  
\[\text{· distributes over +} \]
\[R^* = R^* + R \quad \text{QED.} \]
Example. Show that \(a^* + abb*a = a^* + ab*a\).

**Proof.** Starting on the RHS, we have

\[
\begin{align*}
a^* + ab*a &= a^* + a(\Lambda + bb^*)a \\
&= a^* + aa + abb*a \\
&= (\Lambda + aa^*) + aa + abb*a \\
&= \Lambda + (aa^* + aa) + abb*a \\
&= \Lambda + a(a^* + a) + abb*a \\
&= \Lambda + aa^* + abb*a \\
&= a^* + abb*a
\end{align*}
\]

\[R^* = \Lambda + RR^* \quad \cdot \text{distributes over +}
\]

\[R^* = \Lambda + RR^* \quad + \text{is associative}
\]

\[= a^* + ab^*a \quad \text{induction assumption}
\]

\[R^* = R + R^* \quad \text{induction assumption}
\]

\[= \Lambda + \left(\Lambda + R R^*\right) \quad \text{QED.}
\]

Example. Show that \( (a + aa + \ldots + a^n)(a + b)^* = a(a + b)^* \) for all \(n \geq 1\).

**Proof (by induction):** If \(n = 1\), the statement becomes \(a(a + b)^* = a(a + b)^*\), which is true. If \(n = 2\), the statement becomes \((a + aa)(a + b)^* = a(a + b)^*\), which is true by a previous example. Let \(n > 2\) and assume the statement is true for \(1 \leq k < n\). We need to prove the statement is true for \(n\). The LHS of the statement for \(n\) is

\[
\begin{align*}
(a + aa + \ldots + a^n)(a + b)^* &= a(a + b)^* + (aa + \ldots + a^n)(a + b)^* \\
&= a(a + b)^* + a(a + aa + \ldots + a^{n-1})(a + b)^* \\
&= a(a + b)^* + aa(a + b)^* \\
&= (a + aa)(a + b)^* \\
&= a(a + b)^*
\end{align*}
\]

\[\cdot \text{distributes over +}
\]

\[\cdot \text{distributes over +}
\]

\[\text{induction assumption}
\]

\[\text{induction assumption}
\]

The last line is the RHS of the statement for \(n\). So the statement is true for \(n\). Therefore, the statement is true for all \(n \geq 1\). QED.
Example/Quiz: Use regular algebra to show that $R + (R + S)^* = (R + S)^*$.  

Proof:

\[
R + (R + S)^* = R + [\Lambda + (R + S) + (R + S)(R + S)^*] = \Lambda + (R + R) + S + (R + S)(R + S)^* = \Lambda + (R + S) + (R + S)(R + S)^* = (R + S)^* \]

\[
R^* = \Lambda + R + R^2 + \ldots + R^{k-1} + R^k R^* \quad \text{is associative}
\]

\[
R + R = R \quad R^* = \Lambda + R + R^2 + \ldots + R^{k-1} + R^k R^* \quad \text{is associative}
\]

\[
QED.
\]

Take-home quiz: Use regular algebra to show that $(a + ab)^*a = a(a + ba)^*$.  
