Chapter 9: Regular Languages *

Peter Cappello
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106
cappello@cs.ucsb.edu

• The corresponding textbook chapter should be read before attending this lecture.

• These notes are not intended to be complete. They are supplemented with figures, and other material that arises during the lecture period in response to questions.

*Based on Theory of Computing, 2nd Ed., D. Cohen, John Wiley & Sons, Inc.
CLOSURE PROPERTIES

**Definition:** The language denoted by a regular expression is a **regular language.**
**Theorem:** If $L_1$ and $L_2$ are regular languages, then $L_1 \cup L_2$, $L_1 L_2$, and $L_1^*$ are regular languages.

**Proof (by regular expression):**

1. Since $L_1$ and $L_2$ are regular languages, each is denoted by some regular expression, say $r_1$ and $r_2$, respectively.
2. Given regular expressions $r_1$ and $r_2$, $r_1 + r_2$, $r_1 r_2$, and $r_1^*$ are regular expressions, by the inductive rules for forming regular expressions.
3. The languages denoted by these regular expressions are $L_1 \cup L_2$, $L_1 L_2$, and $L_1^*$, respectively.
4. Thus, these languages are regular.
Proof (by machine):

1. Since $L_1$ and $L_2$ are regular languages, there exist TGs that accept them, say $TG_1$ and $TG_2$, respectively.

2. Assume, without loss of generality, that each has a single initial state and a single final state.

3. Given these TGs, it is easy to construct TGs that accept $L_1 \cup L_2$, $L_1L_2$, and $L_1^*$. Produce on blackboard.

4. Thus, these languages are regular.
Example

Let $\Sigma = \{a, b\}$.

- Let $L_1 = a(a + b)^*a + b(a + b)^*b = \{\text{the set of all strings of length } \geq 2 \text{ that begin and end with the same letter.}\}$
- Let $L_2 = (a + b)^*aba(a + b)^* = \{\text{the set of all strings that contain "aba" as a substring.}\}$

Then:

- $L_1 \cup L_2 = (a(a + b)^*a + b(a + b)^*b) + ((a + b)^*aba(a + b)^*)$.
- $L_1L_2 = (a(a + b)^*a + b(a + b)^*b)((a + b)^*aba(a + b)^*)$.
- $L_1^* = (a(a + b)^*a + b(a + b)^*b)^*$.

Produce machine compositions on the blackboard.
Complements and Intersections

**Theorem:** If $L$ is a regular language, $\overline{L}$ is regular.

**Proof:**

1. Since $L$ is regular, there is an FA, $A$, that accepts it.
2. Create a new FA, $\overline{A}$, which is the same as $A$, except $F_{\overline{A}} = Q_A - F_A$.
3. Word $w$ is accepted by $A$ if and only if it is rejected by $\overline{A}$.
4. Since $\overline{A}$ is an FA, $L(\overline{A})$ is regular.

Apply the construction on even − odd, the set of strings with an even number of $a$’s and an odd number of $b$’s.
**Theorem:** If $L_1$ and $L_2$ are regular languages, $L_1 \cap L_2$ is regular.

**Proof:** By DeMorgan’s law, $L_1 \cap L_2 = \overline{L_1} \cup \overline{L_2}$, a regular language.

Illustrate DeMorgan’s law with a Venn Diagram.
**Proof:** (machine-based)

Replicate the FA construction for the union of 2 regular languages, but final states are those where *both* component states are final in the given machines.

Thus, a word is accepted by the constructed FA if and only if it is accepted by both given finite automata.

Illustrate on the set of words that begin with *a* and end with *b*.