Definitions related to regular expressions
CS 161, Dr. Ostheimer

Definition 1 *Let A be an alphabet. A* regular expression *over A is defined recursively as follows:*

1. *$\emptyset$ is a regular expression over A.*

2. *Λ is a regular expression over A.*

3. *$\forall a \in A$, a is a regular expression over A.*

4. *If r is a regular expression over A, then (r) is a regular expression over A.*

5. *If r is a regular expression over A, then r^* is a regular expression over A.*

6. *If r_1 and r_2 are regular expressions over A, then $r_1 + r_2$ is a regular expression over A.*

7. *If r_1 and r_2 are regular expressions over A, then $r_1 r_2$ is a regular expression over A.*

Definition 2 *For a regular expression r over alphabet A, we define the* language defined by *r, $lang(r)$, recursively as follows.*

1. *$lang(\emptyset) = \{\}$.*

2. *$lang(\Lambda) = \{\Lambda\}$.*

3. *If $a \in A$, then $lang(a) = \{a\}$.*

4. *If r is a regular expression over A, then $lang((r)) = lang(r)$.*

5. *If r is regular expression over A, then $lang(r^*) = (lang(r))^*$.*

6. *If r_1 and r_2 are regular expressions over A, then*
$$lang(r_1 + r_2) = lang(r_1) \cup lang(r_2).$$

7. *If r_1 and r_2 are regular expressions over A, then*
$$lang(r_1 r_2) = lang(r_1) lang(r_2).$$

Definition 3 *Let L be a language over alphabet A. L is* regular *if there exists a regular expression r such that $lang(r) = L$.*

Note. There is an order of operations defined for regular expressions, much as there are for arithmetic expressions and logical operations. Rather than teaching this order to you, we will fully parenthesize our expressions so there is no ambiguity about the order of operations.