MA/CSSE 474 Day 11 Summary

Announcements (more may be added on-line after this is printed):

1. **Regular expressions.** A way of describing languages in a "by example" way. Each regular expression over alphabet Σ defines language over Σ. We will prove (tomorrow?) that every language so-defined is regular.
   a. The alphabet of "regular expressions over Σ" is $\Sigma \cup \{\emptyset, \varepsilon,), (, \cup, *, ^{+} \}$.
   b. If r is a reg. exp. (RE), then L(r), ***the language denoted by r***, is defined recursively:

RE r	**Language L(r) defined by r**
$\emptyset$	$\emptyset$
ε	$\{ \varepsilon \}$
a (a symbol from Σ)	$\{ a \}$
$\alpha\beta$ (α and β are REs)	$L(\alpha)L(\beta)$ (concatenation)
$\alpha\cup\beta$	$L(\alpha) \cup L(\beta)$
α^*	$L(\alpha)^*$
α^+	$L(\alpha)^+$
(α)	$L(\alpha)$

   c. $^{+}$ and ε are very convenient REs, but can be defined in terms of the other ones (syntactic sugar).

2. **Precedence of operators:** (1) * and $^{+}$, (2) concatenation, (3) union. Use parentheses as needed to override.
3. **Examples:**
   a. $L = \{w \in \{a, b\}^*: |w| \text{ is even}\}$
   b. $L = \{w \in \{0, 1\}^*$: w is a binary representation of a positive multiple of 4$\}$
   c. $L = \{w \in \{a, b\}^*$: w contains an odd number of a's$\}$
   d. $L = \{w \in \{a, b\}^*$: there is no more than one b in $w\}$
   e. $L = \{w \in \{a, b\}^*$: no two consecutive letters in w are the same$\}$
   f. $a^* \cup b^* \neq (a \cup b)^*$
   g. $(ab)^* \neq a^*b^*$
   h. $L(\ (aa^*) \cup \varepsilon\ ) =$
   i. $L(\ (a \cup \varepsilon)^*\ ) =$
   j. $L_1 = \{w \in \{a, b\}^*$: every a is immediately followed a b$\}$
   k. $L_2 = \{w \in \{a, b\}^*$: every a has a matching b somewhere$\}$

4. **Kleene's Theorem:** Finite state machines and regular expressions define the same class of languages.
   a. How we will show it:
      i. If L = L(r) for some RE r, then L=L(M) for some NDFSM M. [fairly easy]
      ii. If L=L(M) for some DFSM M, then L = L(r) for some RE r. [a bit more complicated]

5. **For Every Regular Expression there is a Corresponding NDFSM.** Show it by construction.

   a. ∅:

   b. ε

   c. A single element c of Σ:

   d. Union

   e. Concatenation

   f. Kleene Star

6. **For Every DFSM, there is an equivalent regular expression** (different algorithm than the textbook's)**:**

   a. Number the states $q_1, \ldots, q_n$.

   b. Define $\mathbf{R_{ijk}}$ to be the set of all strings $x \in \Sigma^*$ such that

      $(q_i,x) \mid\!-_M^* (q_j, \varepsilon)$, and

      if $(q_i,y) \mid\!-_M^* (q_\ell, \varepsilon)$, for any prefix y of x (except y= ε and y=x), then $\ell \leq k$

   c. That is, R_{ijk} is the set of all strings that take us from q_i to q_j without passing through any intermediate states numbered higher than k.

      i. In this case, "passing through" means both entering and leaving.

      ii. Note that either i or j (or both) may be greater than k.