MA/CSSE 474 Day 07 Main ideas from today (and tomorrow):

1. Example: From the NDFSM on the slides, produce a DFSM based on "state sets"

2. **Theme for today:** Given a DFSM. Among all DFSMs M' that are equivalent to M (in the sense that L(M) = L(M'), there is a minimal number of states that M' can have. Can we find that minimal number and an equivalent machine that has that many states? If so, is it unique (except for renaming of states)?
3. First step: **remove unreachable states**. Easier to find the reachable states and remove the others. Algorithm:

4. **Remove redundant states**. This is trickier. The rest of this document addresses it.
5. A bridge to finding equivalent states: equivalent strings.
6. Given a language L, two strings w and x in Σ_L^* are *indistinguishable* with respect to L, written $w \approx_L x$, iff (English statement):

   (first-order logic statement):

   a. $\approx_L$ s an equivalence relation.
   b. The equivalence classes of $\approx_L$ partition Σ^*.
7. If $L = \{ w \in \{a, b\}^*$: every a is immediately followed by b }, what are the equivalence classes of $\approx_L$?

8. If $L = \{w \in \Sigma^* : |w| \text{ is even}\}$, what are the equivalence classes of $\approx_L$?

9. If $L = \{aa\}\{b\}^*\{a\}$, what are the equivalence classes of $\approx_L$? (Do this one for practice later)
10. $L=\{w \in \{a, b\}^*$: no two adjacent chars in w are the same$\}$: multiple equivalence classes may be subsets of L.

11. If $L = A^nB^n = \{ a^nb^n: n \geq 0 \}$, what are the equivalence classes of $\approx_L$?

12. If L is regular, the number of equivalence classes of is a lower bound on the number of states in any DFSM M such that L = L(M).
13. ***Theorem*:** Let L be a regular language over some alphabet Σ. Then there is a DFSM M that accepts L and that has precisely n states where n is the number of equivalence classes of $\approx_L$. Any other FSM that accepts L must either have more states than M or it must be equivalent to M except for state names.
14. **Construction:** $M = (K, \Sigma, \delta, s, A)$, where:
    - K contains n states, one for each equivalence class of $\approx_L$.
    - $s = [\varepsilon]$, the equivalence class of ε under $\approx_L$.
    - $A = \{[x] : x \in L\}$.
    - $\delta([x], a) = [xa]$. In other words, if M is in the state that contains some string x, then, after reading the next symbol, a, it will be in the state that contains xa.
15. Three things to show:
    a. K is finite.

    b. **δ is a function.** i.e., δ is defined for all *(state, input)* pairs and produces, for each pair, a unique value.

    c. ***L* = *L*(*M*).** To prove this, we must first show that $\forall s,t \in \Sigma^*(([\varepsilon], st) \vdash_M^* ([s], t))$.
    We do this by induction on $|s|$. The base case is trivial.
16. Induction step. Assume that the claim is true for strings of length k. What can we say when $|s| = k+1$?
    Since $|s| \geq 1$, we know that s = yc, where $y \in \Sigma^*$ and $c \in \Sigma$.