Outline for January 8, 2008

1. Access control matrix and entities
   a. Subject, objects (includes subjects)
   b. State is \((S, O, A)\) where \(A\) is access control matrix

2. Transitions modify access control matrix entries; primitive operations
   a. enter \(r\) into \(A[s, o]\)
   b. delete \(r\) from \(A[s, o]\)
   c. create subject \(s\) (note that for all \(x\), \(A[s', x] = A[x, s'] = \emptyset\))
   d. create object \(o\) (note that for all \(x\), \(A[x, o'] = \emptyset\))
   e. destroy subject \(s\)
   f. destroy object \(o\)

3. Commands and examples
   a. Regular command: \textit{create-file}
   b. Mono-operational command: \textit{make-owner}
   c. Conditional command: \textit{grant-rights}
   d. Biconditional command: \textit{grant-read-if-r-and-c}
   e. Doing “or” of 2 conditions: \textit{grant-read-if-r-or-c}
   f. General form

4. Miscellaneous points
   a. Copy flag and right
   b. \textit{Own} as a special right
   c. Principle of attenuation of privilege

5. What is the safety question?
   a. An unauthorized state is one in which a generic right \(r\) could be leaked into an entry in the ACM that did not previously contain \(r\). An initial state is safe for \(r\) if it cannot lead to a state in which \(r\) could be leaked.
   b. Question: in a given arbitrary protection system, is safety decidable?
   c. Theorem: there is an algorithm that decides whether a given mono-operational system and initial state is safe for a given generic right.

6. General case: It is undecidable whether a given state of a given protection system is safe for a given generic right.
   a. Approach: represent Turing machine tape as access control matrix, transitions as commands
   b. Reduce halting problem to it