A deterministic finite automaton (DFA) is a 5-tuple \( M = (Q, \Sigma, \delta, s, F) \) where:

- \( Q \) is a finite set of states
- \( \Sigma \) is a finite alphabet
- \( s \), a member of \( Q \), is the start state
- \( F \), a subset of \( Q \), is the set of accepting states
- \( \delta \), a function from \( Q \times \Sigma \) to \( Q \).

The function \( \delta \) is defined as:

\[
\delta : Q \times \Sigma \rightarrow Q
\]

Example value:

\[
\delta(p, c) = q
\]

This defines the nondeterministic finite automaton (NFA). It is a DFA when the set of tuples has the property that for all \( p \in Q \) and \( c \in \Sigma \), there is exactly one \( q \in Q \) such that \((p, c, q) \in \delta\).

The rest of the lecture was a demo of the "Turing Kit" software for a DFA (and TM).