1

Finite Automaton
DFA, NFA, NFA-epsilon, GNFA
Regular Languages
Regular Expressions
(Regular Grammars)

2

All are equivalent in terms of the languages they recognize/generate.

I HAVE ONE OF THESE, CAN I CONSTRUCT ANY ONE OF THE OTHERS?

Sure!
Look at the class notes, Chapter 1, and the exercises, problems, and the homeworks.

3

I WANT TO PROVE THAT A LANGUAGE L IS REGULAR.

Either show me a DFA (or NFA, NFA-epsilon, or GNFA) that recognizes L (and only L),
or, write down a regular expression (set) for it,
(or, write a regular grammar that generates L.)

4

WHY ARE FINITE AUTOMATON SO NEAT?

Because they recognize infinite languages with a finite number of states.

WOW! HOW DO THEY DO THIS?

They have loops!

THIS IS TOO GOOD TO BE TRUE, IS THERE A CATCH?

Unfortunately, yes. We cannot control these loops. We cannot count how many times they execute. Could be anything between 0 and infinity.

HMM.... TOO BAD.

5

Don't go. You can use this to prove that some languages are not regular.

INTERESTING... HOW?

Use the pumping lemma! It tells you that there should be a loops somewhere if your language is infinite. If your language cannot tolerate (uncounted) loops, voila... It is not regular!

6

TELL ME HOW I CAN PROVE THAT A LANGUAGE L IS NOT REGULAR.

Show that no FA can recognize it, or use the pumping lemma to show that no decomposition works.