CMSC 330: Organization of Programming Languages

Discussion Section, 2/11
Examples of REs & Finite Automata
Whenever I learn a new skill I concoct elaborate fantasy scenarios where it lets me save the day.

Oh no! The killer must have followed her on vacation!

But to find them we'd have to search through 200 mb of emails looking for something formatted like an address! It's hopeless!

Everybody stand back.

I know regular expressions.
What do we know about REs?

• The Formal Vocab
  – Symbol
  – Alphabet
  – String
  – Language

• The Special Symbol: $$\epsilon$$

• The Formal Operations
  – Concatenation, e.g. /ab/
  – Union, e.g. /a|b/
  – Kleene Closure, e.g. a*
More About REs

• Operator Precedence
  – Kleene closure * > concatenation > union |
  – Like arithmetic, use parentheses to override

• How’d we get from here to DFA again?

Anything but DFA!!!!
Describing Regular Expressions

- $0(0|1)^*0$
  - All strings beginning and ending in 0
- $(((\varepsilon|0)1^*)*)$
  - All strings
- $(0|1)^*0(0|1)(0|1)$
  - All strings with 0 as third digit from right
Creating Regular Expressions

For all strings of 0’s and 1’s that...

a) Begin in 1
   - 1(0|1)*

a) End in 1
   - (0|1)*1

a) Contain 00
   - (0|1)*00(0|1)*

a) Do not contain 00
   - (01|1)*(<empty>|0)
Finite Automata

• We can further formalize regular expressions as finite automata.
  – Why? To reason about them.

• What are the choices?
  – Deterministic
  – Non-deterministic

• Why have two choices?

• What are the differences?
Creating DFA

For all strings of 0’s and 1’s that…

a) Begin in 1

b) End in 1

c) Contains 00

d) Do not contain 00

Swap final / non-final states!
Creating NFA

For all strings of 0’s and 1’s that...

a) Begin in 1

b) End in 1

c) Contain 00

d) Do not contain 00

Based on regular expression