A Hands-on Approach to FLA with JFLAP

JFLAP and Regular Languages

Susan Rodger, Duke University
Thomas Finley, Cornell University
Peter Linz, University of California, Davis

SIGCSE 2006
March 4, 2006
Finite Automata Editing and Simulation

• The most basic feature of JFLAP has always been the creation of automata, and simulation of input on automata.
• Here we demonstrate the creation and simulation on a simple NFA.
FA Edit & Simulation
Start up JFLAP

• When we start up JFLAP we have a choice of structures.
• The first of these is the Finite Automata!
FA Edit & Simulation
Start Editing!

• We start with an empty automaton editor window.
FA Edit & Simulation
Create States

• We create some states ...
We create some transitions ...
FA Edit & Simulation
Initial and Final State

- We set an initial and final state.
- Now we can simulate input on this automaton!
When we say we want to simulate input on this automaton, a dialog asks us for the input.
FA Edit & Simulation
Start Simulation!

• When simulation starts, we have a configuration on the initial state with all input remaining to be processed.
FA Edit & Simulation

After One Step

• This is a nondeterministic FA, and on this input we have multiple configurations after we “Step.”
The previous configurations on $q_1$ and $q_2$ are rejected, and are shown in red.

The remaining uncolored configurations paths are not rejected, and are still open.
FA Edit & Simulation
After Three Steps

• Yet another step.
FA Edit & Simulation
After Four Steps

• One of the final configurations has been accepted!
FA Edit & Simulation
Traceback

• One can then see a traceback to see the succession of configurations that led to the accepting configuration.
FA Multiple Run

• Select Multiple Run
• One can then enter many strings and receive acceptance info.
Exercise

• Build the FA we just did in JFLAP
• Test it with input strings
  – Both step with closure and multiple run
• Build a second FA for one of these:
  – Strings over \{a,b\} such that there are an odd number of b’s and the number of a’s is divisible by 3
  – Strings over \{a,b\} such that the number of a’s is divisible by 2 or 3
JFLAP – Regular Languages

• Create
  – DFA and NFA
  – regular grammar
  – regular expression

• Conversions
  – NFA to DFA to minimal DFA
  – NFA $\leftrightarrow$ regular expression
  – NFA $\leftrightarrow$ regular grammar
NFA to DFA

1. Start with NFA

2. Construct new DFA
   On q0 with an a, go to q0, q1 and q2

3. Final DFA
DFA to Min DFA

- Start with DFA
DFA to Min DFA(2)

- Start tree of distinguishable states
- Complete tree!
DFA to Min DFA (3)

- Determine states in min DFA

- Add arcs to complete it
Regular Expression (RE) to FA

1. Start with RE

2. Generalized Transition Graph (GTG)

3. De-oring

\[ a^*+bc \]
RE to FA (cont)

4. De-concatenate

5. De-staring
FA to RE

1. Start with FA
2. FA only one final state
3. Convert to GTG
4. Select node to remove, q1
5. Compute new RE transitions

6. Remove state q1

7. Resulting RE

\[ a^*(a+ab)(c+bb)^* \]
Exercises

1. Load file ex2-nfa2dfa-c, NFA to DFA
2. Load file ex2-dfa2mindfa-e, DFA to min DFA
3. Load file ex3.rgfa-b, RG to FA
4. Load file ex3.fa2rg-c, FA to RG
5. Convert (a+b)*cd to an FA
6. Load file ex4-nfa2re-d, FA to RE