Increasing Interaction and Support in the Formal Languages and Automata Theory Course

Susan H. Rodger
Duke University

ITiCSE 2007
June 25, 2007

Supported by NSF Grant DUE 0442513.
Outline

• Overview of JFLAP

• Increasing Interaction in the course with JFLAP

• New Items in JFLAP
  – Moore and Mealy Machines
  – Pumping Lemma
  – Batch Testing Mode

• JFLAP Study and Future
Formal Languages and Automata Theory

• Traditionally taught
  – Pencil and paper exercises
  – No immediate feedback

• Different
  – More mathematical than most CS courses
  – Less hands-on than most CS courses
## Why Develop Tools for Automata?

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Textual</strong></td>
<td>[ ({q_0, q_1, q_2}, {a, b}, \delta, q_0, {q_2} ) ]</td>
</tr>
<tr>
<td></td>
<td>[ \delta = {(q_0, b, q_0), (q_0, a, q_1), (q_1, a, q_0), (q_1, b, q_2), (q_2, a, q_1)} ]</td>
</tr>
<tr>
<td><strong>Tabular</strong></td>
<td><img src="image" alt="Tabular Automaton" /></td>
</tr>
<tr>
<td><strong>Visual</strong></td>
<td><img src="image" alt="Visual Automaton" /></td>
</tr>
<tr>
<td><strong>Interactive</strong></td>
<td><img src="image" alt="Interactive Automaton" /></td>
</tr>
</tbody>
</table>
Why Develop Tools for Automata?
Examined 10 Automata textbooks

• One had software with book (not all topics)
• Only 6 had pictures of PDA, 2 or 3 states
• Only 6 had pictures of Turing machines, three of those switched representation
• Only 2 had picture of CFG to NPDA
• None had picture of parse tree for unrestricted grammar
Impact on Courses

• Not very visual
• Simple examples
• Exercises are limited
• Don’t provide feedback
Overview of JFLAP

- **Java Formal Languages and Automata Package**
- Instructional tool to learn concepts of Formal Languages and Automata Theory
- Topics:
  - Regular Languages
  - Context-Free Languages
  - Recursively Enumerable Languages
  - Lsystems

- With JFLAP your creations come to life!
Thanks to Students - Worked on JFLAP and Automata Theory Tools

• NPDA - 1990, C++, Dan Caugherty
• FLAP - 1991, C++, Mark LoSacco, Greg Badros
• JFLAP - 1996-1999, Java version
  Eric Gramond, Ted Hung, Magda and Octavian Procopiuc
• Pâté, JeLLRap, Lsys
  Anna Bilska, Jason Salemme, Lenore Ramm, Alex Karweit, Robyn Geer
• JFLAP 4.0 – 2003, Thomas Finley, Ryan Cavalcante
• JFLAP 6.0 – 2005-2007 Stephen Reading, Bart Bressler, Jinghui Lim, Chris Morgan, Jason Lee

Over 15 years!
JFLAP – Regular Languages

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

• Conversions
  – NFA to DFA to minimal DFA
  – NFA \leftrightarrow regular expression
  – NFA \leftrightarrow regular grammar
JFLAP – Regular languages (more)

- Simulate DFA and NFA
  - Step with Closure or Step by State
  - Fast Run
  - Multiple Run
- Combine two DFA
- Compare Equivalence
- Brute Force Parser
- Pumping Lemma
JFLAP – Context-free Languages

• Create
  – Nondeterministic PDA
  – Context-free grammar
  – Pumping Lemma

• Transform
  – PDA $\rightarrow$ CFG
  – CFG $\rightarrow$ PDA (LL & SLR parser)
  – CFG $\rightarrow$ CNF
  – CFG $\rightarrow$ Parse table (LL and SLR)
  – CFG $\rightarrow$ Brute Force Parser
JFLAP – Recursively Enumerable Languages

• Create
  – Turing Machine (1-Tape)
  – Turing Machine (multi-tape)
  – Building Blocks
  – Unrestricted grammar

• Parsing
  – Unrestricted grammar with brute force parser
JFLAP - L-Systems

• This L-System renders as a tree that grows larger with each successive derivation step.
Students love L-Systems
Increasing Interaction in the Course with JFLAP
Using JFLAP during Lecture

• Use JFLAP to build examples of automata or grammars
• Use JFLAP to demo proofs
• Load a JFLAP example and students work in pairs to determine what it does, or fix it if it is not correct.
Example 1: JFLAP during Lecture

• Ask students to write on paper an NPDA for palindromes of even length
• Build one of their solutions using JFLAP
  – Shows students how to use JFLAP
Example 1: JFLAP during Lecture (cont)

- Run input strings on the NPDA
  - Shows the nondeterminism
Example 2: JFLAP during Lecture

• Brute Force Parser
  – Give a grammar with a lambda-production and unit production
  – Run it in JFLAP, see how long it takes (LONG)
    • Is aabbab in L?
  – Transform the grammar to remove the lambda and unit-productions
  – Run new grammar in JFLAP, runs much faster!
Example 2 (cont)

Parse Tree Results

• First Grammar – 1863 nodes generated
• Second Grammar – 40 nodes generated
• Parse tree is the same.
With JFLAP, Exploring Concepts too tedious for paper

• Load a Universal Turing Machine and run it
• See the exponential growth in an NFA or NPDA
• Convert an NPDA to a CFG
  – Large grammar with useless rules
  – Run both on the same input and compare
  – Transform grammar (remove useless rules)
JFLAP’s use Outside of Class

• Homework problems
  – Turn in JFLAP files
  – OR turn in on paper, check answers in JFLAP
• Recreate examples from class
• Work additional problems
  – Receive immediate feedback
Ordering of Problems in Homework

• Order questions so they are incremental in the usage of JFLAP

1. Load a DFA. What is the language?
   Students only enter input strings.

2. Load a DFA that is not correct. What is wrong? Fix it.
   Students only modifying a small part.

3. Build a DFA for a specific language.
   Last, students build from scratch.
There is another way to get interaction in this course…
Students Work in Groups to Solve Problems

- Lecture some, then stop
- Students work on problem with JFLAP
- Bring students back together
Interaction in Class – Props

Edible Turing Machine

• TM for $f(x) = 2x$ where $x$ is unary

• TM is not correct, can you fix it? Then eat it!

• States are blueberry muffins
Students building DFA with cookies and icing
The Smart Waitress vs Customer

- Four cups on a revolving tray (each up or down)
- Waitress blindfolded and wears boxing gloves
- Goal is to turn all cups up
- Game – Repeat:
  - W turns 1-4 cups
    - If all up wins
  - Customer rotates tray 0, 90, 180 or 270 degrees
- Is there a winning strategy?
  - This is a DFA problem

From an old EATCS bulletin
New Items in JFLAP
New in JFLAP: Mealy Machines

• Similar to finite automata
  – No final states
  – Produce an output on their transitions
  – deterministic
Example – Vending Machine

• Dispenses candy once enough money has been inserted
  – Money – n(nickel), d(dime) q(quarter)
  – Candy bars – 20 cents
  – Returns the appropriate amount of change – the number of nickels
  – C4 means “candy and 4 nickels”

• From Carroll and Long’s *Theory of Finite Automata* book
Mealy Vending Machine Example

<table>
<thead>
<tr>
<th>Input</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>ndq</td>
<td>c4</td>
</tr>
<tr>
<td>qqqq</td>
<td>c1c1c1</td>
</tr>
<tr>
<td>nq</td>
<td>c2</td>
</tr>
<tr>
<td>dq</td>
<td>c3</td>
</tr>
<tr>
<td>dd</td>
<td>c0</td>
</tr>
<tr>
<td>ddn</td>
<td>c0</td>
</tr>
</tbody>
</table>
New in JFLAP: Moore Machine

• Similar to Mealy Machine
  – No final state
  – Output is produce by states, not transitions
Example – Halve a Binary Number

<table>
<thead>
<tr>
<th>Input</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>0010</td>
</tr>
<tr>
<td>11111</td>
<td>001111</td>
</tr>
<tr>
<td>01001</td>
<td>000100</td>
</tr>
<tr>
<td>100</td>
<td>0010</td>
</tr>
<tr>
<td>110</td>
<td>0011</td>
</tr>
<tr>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>0</td>
<td>00</td>
</tr>
</tbody>
</table>
New in JFLAP:
Regular Pumping Lemma Game

Pumping Lemma: Let $L$ be an infinite regular language. $\exists$ a constant $m > 0$ such that any $w \in L$ with $|w| \geq m$ can be decomposed into three parts as $w = xyz$ with

$$|xy| \leq m$$
$$|y| \geq 1$$
$$xy^iz \in L \text{ for all } i \geq 0$$
Pick an Example

$L = \{a^n b^n : n \geq 0\}$

$L = \{w \in \{a, b\}^*: n_a(w) < n_b(w)\}$

$L = \{ww^R : w \in \{a, b\}^*\}$

$L = \{a^n : n \geq 0\}$

$L = \{(ab)^n a^k : n > k, k \geq 0\}$

$L = \{a^n b^k c^{n+k} : n \geq 0, k \geq 0\}$

$L = \{a^n b^l a^k : n > 5, l > 3, k \leq l\}$

$L = \{a^n : n \geq 2, n \text{ is a prime number}\}$

$L = \{a^n : n \text{ is even}\}$
JFLAP Pump lemma Game

User enters in steps 1 and 3

$L = \{a^n b^n : n \geq 0\}$ Regular Pumping Lemma

Messages

I WIN. Do you want to play again or concede that the language is not regular?
I have selected $i$ to give a contradiction. It is displayed in Box 4.
Click "Step" in Box 5 to step the animation.

1. Select integer $m$

   6

2. Given integer $m$, here’s string $w$ such that $|w| \geq m$

   aaaaaabbbbb

3. Select decomposition of $w$ into $xyz$

   $x$: aaa
   $|x|$: 3
   $y$: aa
   $|y|$: 2
   $z$: abbbbbb
   $|z|$: 7

4. A choice of $i$ to give contradiction

   $i$: 0

   pumped string: aaaaabbbbb

5. Animation

   $w = \text{aaa aa abbbbbb}$

   $xy^0z = a^4b^6 = \text{aaabbbbbb}$ is NOT in the language: CONTRADICTION.
New in JFLAP: Context-Free Pumping Lemma Game

Pumping Lemma for CFL’s Let $L$ be any infinite CFL. Then there is a constant $m$ depending only on $L$, such that for every string $w$ in $L$, with $|w| \geq m$, we may partition $w = uvxyz$ such that:

$|vxy| \leq m$, (limit on size of substring)
$|vy| \geq 1$, (v and y not both empty)
For all $i \geq 0$, $uv^i x y^i z \in L$
Similar CFL pump lemma game

$L = \{a^n b^n c^n : n \geq 0\}$ Context-Free Pumping Lemma

<table>
<thead>
<tr>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>I WIN. Do you want to play again or concede that the language is not...</td>
</tr>
<tr>
<td>I have selected ( i ) to give a contradiction. It is displayed in Box 4.</td>
</tr>
<tr>
<td>Click &quot;Step&quot; in Box 5 to step the animation or &quot;Add&quot; in the right panel...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1. Select integer ( m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 5 )</td>
</tr>
</tbody>
</table>

| 2. Given integer \( m \), here's string \( w \) such that \(|w| \geq m\) |
|--------------------------|
| \( aaaaaabbabbccccc \) |

<table>
<thead>
<tr>
<th>3. Select decomposition of ( w ) into ( uvxyz )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u: \text{aa} ) (</td>
</tr>
<tr>
<td>( v: \text{a} ) (</td>
</tr>
<tr>
<td>( x: \text{a} ) (</td>
</tr>
<tr>
<td>( y: \text{abb} ) (</td>
</tr>
<tr>
<td>( z: \text{bbbcccc} ) (</td>
</tr>
<tr>
<td>( a \ a \ a \ a \ a \ b \ b \ b \ b \ b \ b \ b \ c \ c \ c \ c \ c \ c )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. A choice of ( i ) to give contradiction</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i: 2 ) \text{ pumped string: } aaaaaabbbabbabbccccc</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cases:</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
</tr>
</tbody>
</table>

**Cases:**

- **#**
- **Description**
CFL pump lemma (cont)

• Last step shows the contradiction

• In CFL – there are lots of cases to consider
New in JFLAP: Batch Testing Mode

- Select several files for testing
- Then select input file
Batch Testing Mode (cont)

<table>
<thead>
<tr>
<th>File</th>
<th>Input</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>fa1.jff</td>
<td>aabbaabb</td>
<td>Reject(Accept)</td>
</tr>
<tr>
<td>fa1.jff</td>
<td>aaaa</td>
<td>Reject(Accept)</td>
</tr>
<tr>
<td>fa1.jff</td>
<td>aab</td>
<td>Accept</td>
</tr>
<tr>
<td>fa1.jff</td>
<td>aabaaab</td>
<td>Accept(Reject)</td>
</tr>
<tr>
<td>fa2.jff</td>
<td>aabbaabb</td>
<td>Reject(Accept)</td>
</tr>
<tr>
<td>fa2.jff</td>
<td>aaaa</td>
<td>Reject(Accept)</td>
</tr>
<tr>
<td>fa2.jff</td>
<td>aab</td>
<td>Accept</td>
</tr>
<tr>
<td>fa2.jff</td>
<td>aabaaab</td>
<td>Accept(Reject)</td>
</tr>
<tr>
<td>fa5.jff</td>
<td>aabbaabb</td>
<td>Accept</td>
</tr>
<tr>
<td>fa5.jff</td>
<td>aaaa</td>
<td>Accept</td>
</tr>
<tr>
<td>fa5.jff</td>
<td>aab</td>
<td>Accept</td>
</tr>
<tr>
<td>fa5.jff</td>
<td>aabaaab</td>
<td>Accept(Reject)</td>
</tr>
</tbody>
</table>
JFLAP Study

• Study of JFLAP’s effectiveness in learning
  – Pretest/Posttest
  – Interviews

• Supported by National Science Foundation,
  grant NSF DUE 0442513
Fourteen Participants

• Duke
• UNC-Chapel Hill
• Emory
• Winston-Salem State University
• United States Naval Academy
• Rensselaer Polytechnic Institute
• UC Davis
• Virginia State University
• Norfolk State University
• University of Houston
• Fayetteville State University
• University of Richmond
• San Jose State University
• Rochester Institute of Technology
JFLAP’s Use Around the World

• JFLAP web page has over 110,000 hits since 1996
• Google Search
  – JFLAP appears on over 20,000 web pages
  – Note: search only public web pages
• JFLAP been downloaded in over 160 countries
JFLAP Future Work

- Adding a user-control parser
- CYK Parser
- More work on pumping lemmas
Questions?

• JFLAP is free!

• www.jflap.org

• JFLAP book (Jones & Bartlett, 2006)
  – Use as supplement to a textbook