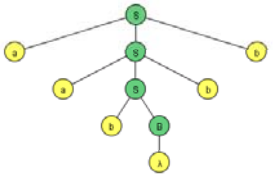
Increasing Engagement in Automata Theory with JFLAP

Susan H. Rodger
Duke University
rodger@cs.duke.edu



SIGCSE 2009
March 7, 2009

Supported by NSF Grant DUE 0442513

Co-Authors

- Evaluators - NCSU
  - Eric Wiebe
  - Kareem Omar
- Undergraduates - Duke
  - Kyung Min Lee
  - Chris Morgan
  - Jonathan Su

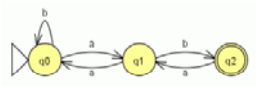
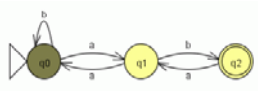
Outline

- Introduction and Overview of JFLAP
- JFLAP Two-year Study and Results
- New Items in JFLAP
  - User Control Parser
  - CYK Parser
  - Expanding Pumping Lemma
  - Turing machine to Unrestricted Grammar
  - Others
- Conclusions and Future Work

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
  - Programming is in most of their CS courses, not here

Why Develop Tools for Automata?

Textual	$(\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$ $\delta = \{(q_0, b, q_0), (q_0, a, q_1), (q_1, a, q_0), (q_1, b, q_2), (q_2, a, q_1)\}$
Tabular	$\begin{array}{c\|c\|c} & a & b \\ \hline q_0 & q_1 & q_0 \\ \hline q_1 & & q_2 \\ \hline q_2 & & \end{array}$
Visual	
Interactive	

Overview of JFLAP

- **J**ava **F**ormal **L**anguages and **A**utomata **P**ackage
- 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!

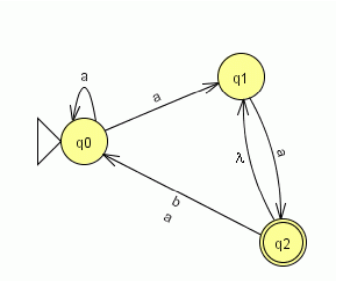
Thanks to Students - Worked on JFLAP and Automata Theory Tools

Over 17 years!

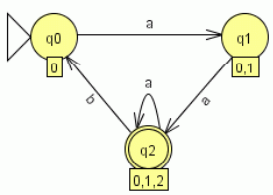
- NPDA - 1990, C++, Dan Caugherty
- FLAP - 1991, C++, Mark LoSacco, Greg Badros
- JFLAP - 1996-1999, Java version
  Eric Gramond, Ted Hung, Magda and Octavian Procopiuc
- Pâté, JeLLRap, Lsys
  Anna Bilska, Jason Salemme, Lenore Ramm, Alex Karweit, Robyn Geer
- JFLAP 4.0 – 2003, Thomas Finley, Ryan Cavalcante
- JFLAP 6.0 – 2005-2008 Stephen Reading, Bart Bressler, Jinghui Lim, Chris Morgan, Jason Lee, Jonathan Su

JFLAP – Regular Languages

- Create
  - DFA and NFA
  - Moore and Mealy
  - regular grammar
  - regular expression

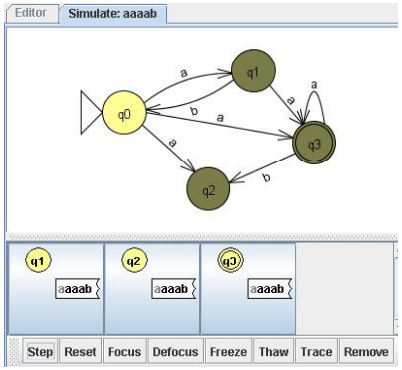


- Conversions
  - NFA to DFA to minimal DFA
  - NFA ←→ regular expression
  - NFA ←→ regular grammar



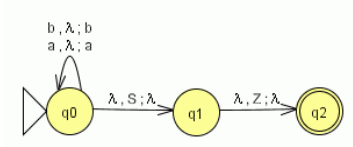
JFLAP – Regular languages (more)

- 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

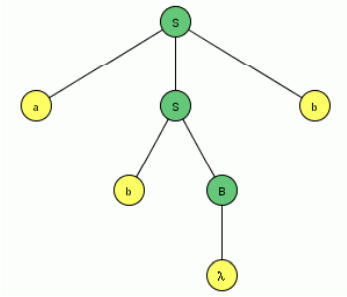


JFLAP – Context-free Languages

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

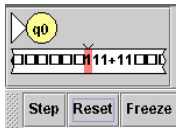


- Transform
  - PDA → CFG
  - CFG → PDA (LL & SLR parser)
  - CFG → CNF
  - CFG → Parse table (LL and SLR)
  - CFG → Brute Force Parser

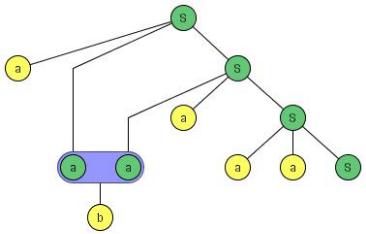


JFLAP – Recursively Enumerable Languages

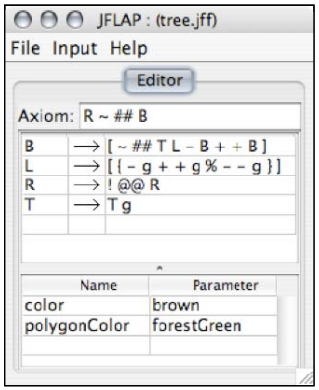
- Create
  - Turing Machine (1-Tape)
  - Turing Machine (multi-tape)
  - Building Blocks
  - Unrestricted grammar



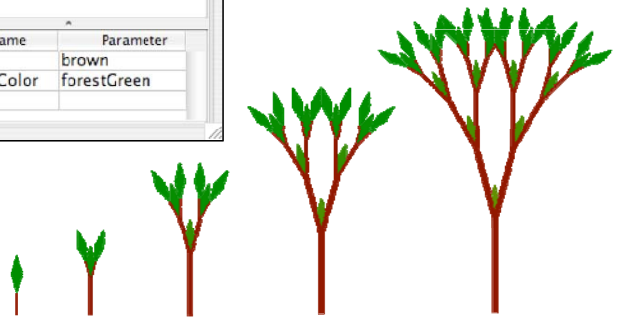
- Parsing
  - Unrestricted grammar with brute force parser



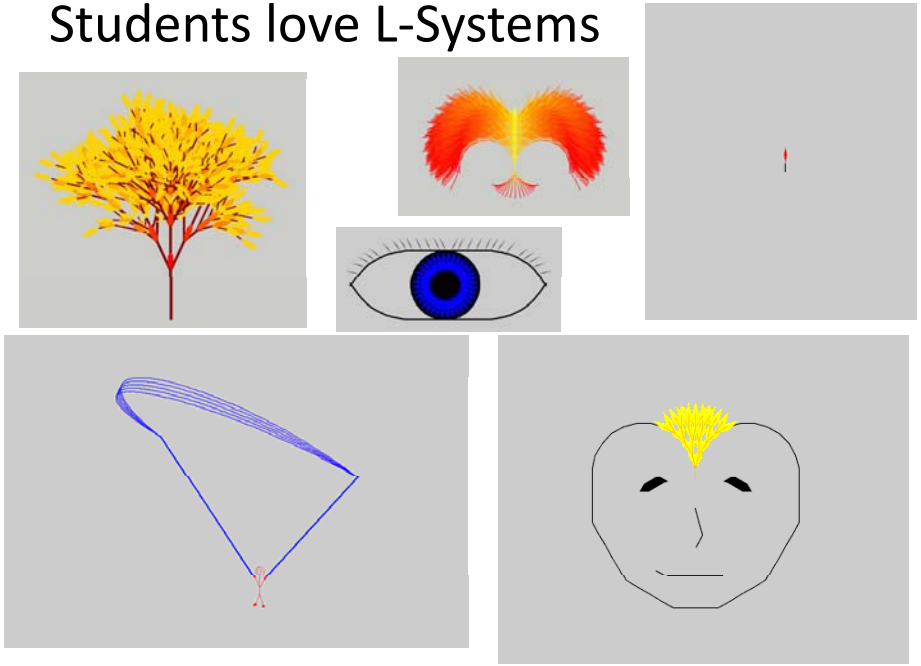
JFLAP - L-Systems



- This L-System renders as a tree that grows larger with each successive derivation step.



Students love L-Systems



Other Tools for Automata

- Turing's World (Barwise and Etchemendy)
- Deus Ex Machina (Taylor and Savoiu)
- Theory of Computing Hypertextbook (Ross)
- Many others
  - L-System tools
  - Compiler tools
  - Finite State machine tools

JFLAP Study

- Study of JFLAP's effectiveness in learning
  - Two year study
  - Fourteen Faculty Adopters
  - Two 2-day faculty Adopter Workshops – June 2005, June 2006
  - Collect data 2005-06 and 2006-07 Academic years
  - Pretest/Posttest
  - Interviews
  - Team of three evaluators
    - Eric Weibe – Education
    - Rocky Ross – Computer Science Theory
    - Joe Bergin – Computer Science Tools

Fourteen Faculty Adopter Participants

-small, large
- public, private
- includes minority institutions

- 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

We hoped to show with this learning approach…

- Students gain a better and deeper understanding of FLA
- Students are happier and more confident in learning FLA
- Students are more interested in using the tools on their own
- Instructors can easily use the tools in class
- Instructors can easily grade electronic submissions

Goals of the JFLAP Study - Formal Languages and Automata (FLA)

- Present FLA in a visual and interactive manner in addition to the more traditional approach
  - Integrated
- Present Applications of FLA
- Provide a tool for allowing students to explore FLA in a computational manner
- Provide Materials for instructors to integrate this approach in their courses

Running a Study is hard!

- Hit by the drop In enrollments in after dot-com burst
- IRBs are different process at every institution
  - One page writeup ok'd (simplest)
  - Full medical IRB (many pages)
- One institution shut down all IRB research projects – we could not use data already collected.
- One University - Control Group – different times means different types of students, different professors.
- Some faculty came to workshop and did not follow through
- There were also some fantastic faculty!

Year One Instructor Interviews

- Used JFLAP in their courses
  - Primary use in class – demonstrations
  - Some used it to generate the graphics for their lecture
  - Extensive use – homeworks – includes electronic submission
  - One used it in office hours

Year One – Software Implementation

Question	YES/NO	No.	%
Did you use JFLAP software to study for inclass exams?	YES NO	20 16	55% 45%
Did you feel you had time to learn how to use the JFLAP software?	YES NO	33 2	94% 6%
Did you feel that using the software took time away from other study activities?	YES NO	3 33	8% 92%
Did the time and effort it took to use JFLAP help you get a better grade in the course?	YES NO	23 13	64% 36%
Was it easier to use JFLAP software or was it easier to draw it out by hand?	software by hand	30 6	83% 17%
Did you feel you would have done as well in the course if you had not used JFLAP?	YES NO NA	18 13 5	50% 36% 14%

Years 1 and 2: Usability Survey

Question		
How easy was it to use the drawing tool of JFLAP? (134 respondents)	Very Easy Easy Neither Difficult Very Difficult	31% 48% 15% 5% 0%
How easy was it to run the automata you designed in JFLAP? (134 respondents)	Very easy Easy Neither Difficult Very Difficult	33% 47% 12% 6% 2%
How easy was it to interpret results from the test run in JFLAP? (134 respondents)	Very Easy Easy Neither Difficult Very Difficulty	23% 45% 19% 10% 3%
What is your overall assessment of the JFLAP software? (133 respondents)	Very Poor Poor Neither Good Very Good	2% 4% 11% 63% 20%

Year 2 – Implementation Survey

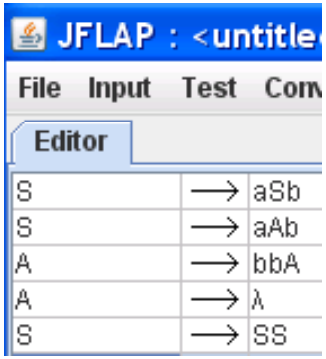
QUESTION	Time	Response
When preparing for exams what percentage of study time involved the use of JFLAP software? (100 responses)	0-20% 21-40% 41-60% 61-80% 81-100%	68% 16% 11% 3% 2%
How often did you use JFLAP to do additional practice problems? (99 responses)	Never Rarely Occasionally Often Very Often	46% 25% 21% 4% 4%

Year 2 – Usability Survey

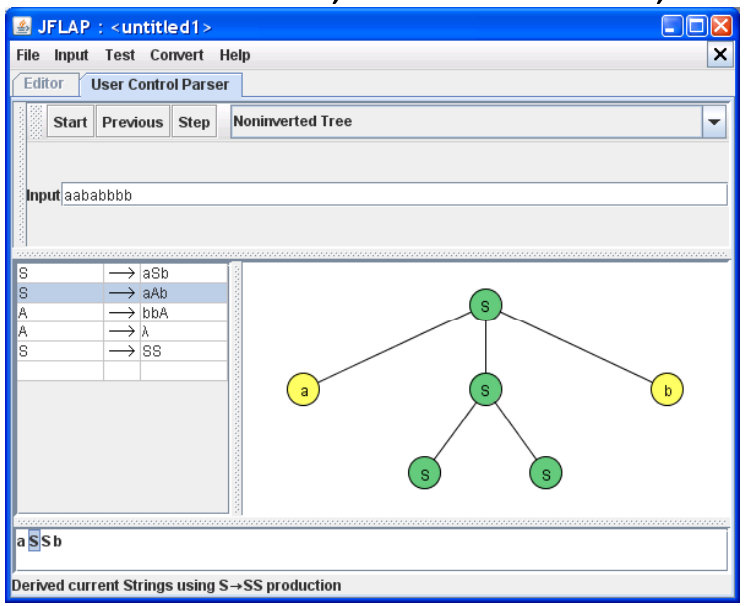
Question		
Using JFLAP made the course more enjoyable for me. (98 responses)	Strongly Agree Agree Neither Disagree Strongly Disagree	12% 51% 25% 6% 4%
Using JFLAP made me feel more engaged in the course. (98 responses)	Strongly Agree Agree Neither Disagree Strongly Disagree	13% 59% 15% 9% 3%
Having access to JFLAP made learning course concepts ... (97 responses)	Much harder Harder Neither Somewhat easier Much easier	1% 5% 26% 54% 14%

New Items in JFLAP

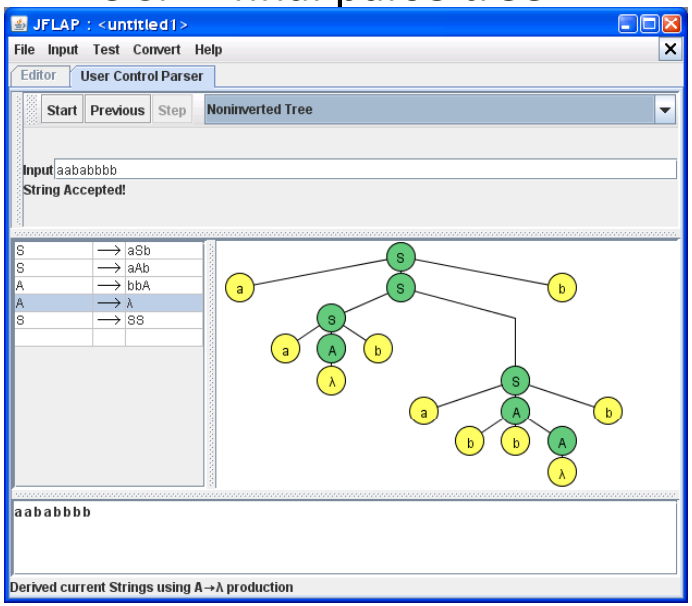
- User-Control Parser (UCP)
  - Allows for more exploration



UCP – select rule, select variable, Step



UCP – final parse tree

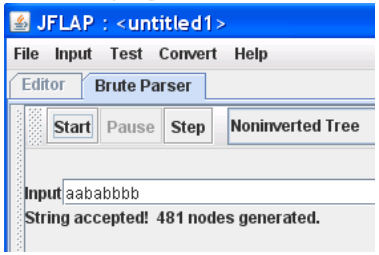


CYK Parser

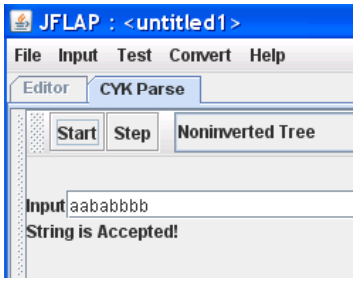
- Cocke-Younger-Kasami (CYK)
- How it works in JFLAP
  - Internally, convert to CNF grammar
  - Then map back to original grammar

CYK – substantially faster than Brute Force Parser for many grammars

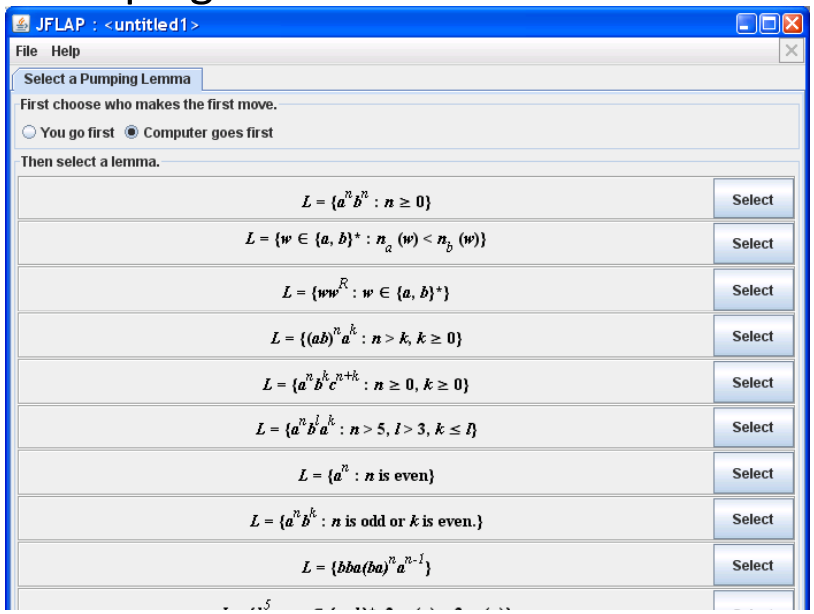
- Brute Force Parser
  8 secs



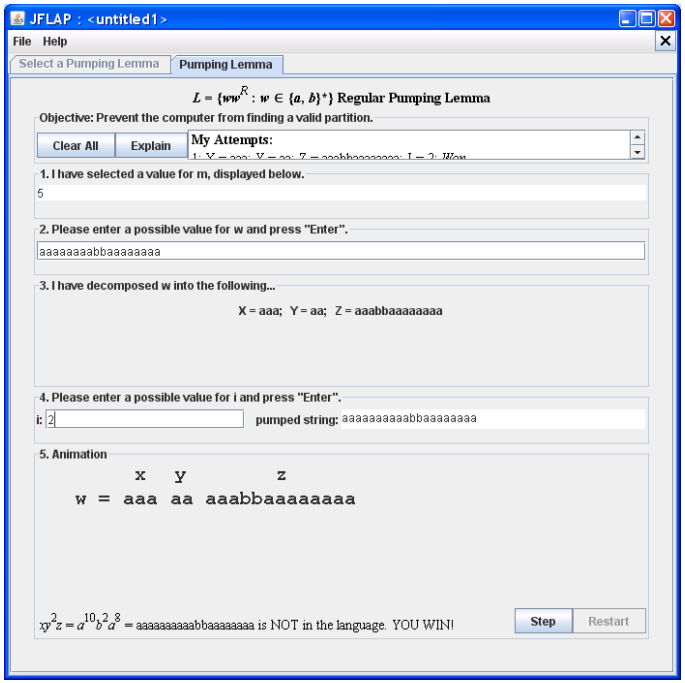
- CYK parser
  1 sec



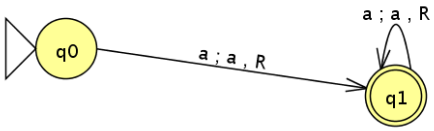
Pumping Lemma Game – Linz Book



ww^R



Turing Machine to Unrestricted Grammar



S	⟶	V(==)S
S	⟶	SV(==)
S	⟶	T
aV(aa)	⟶	aa
aV(=a)	⟶	a=
V(a0a)V(aa)	⟶	V(aa)V(a1a)
V(a1a)	⟶	a
V(aa)a	⟶	aa
V(a0a)V(=a)	⟶	V(aa)V(=1a)
V(aa)=	⟶	a=
V(a1a)V(aa)	⟶	V(aa)V(a1a)
V(a1a)V(=a)	⟶	V(aa)V(=1a)
V(=0a)V(=a)	⟶	V(=a)V(=1a)
V(=1a)V(aa)	⟶	V(=a)V(a1a)
V(=1a)V(=a)	⟶	V(=a)V(=1a)
V(=a)=	⟶	==
V(=1a)	⟶	=
V(=a)a	⟶	=a
V(=0a)V(aa)	⟶	V(=a)V(a1a)
T	⟶	TV(aa)
T	⟶	V(a0a)
=V(aa)	⟶	=a
=V(=a)	⟶	==
=	⟶	λ

Other New Items

- Can automatically add a trap state
- Additional graph layout choices
- Can load an input string from a file
  - This is useful for universal Turing machine
- Online JFLAP Tutorial
- JFLAP wiki and listserv

Getting interaction into the automata theory course……

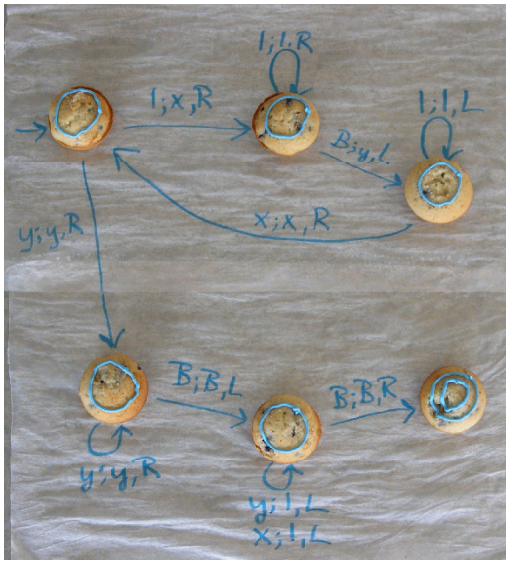
There are lots of ways to get interaction in this course…

Students Work in Groups to Solve Problems – With or W/out JFLAP

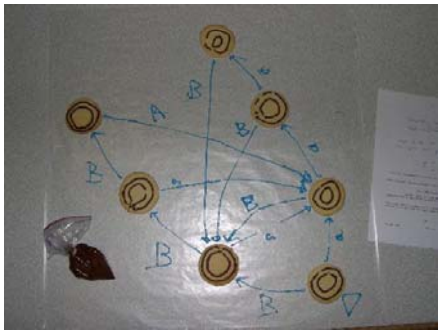
- Lecture some, then stop
- Students work on problem with JFLAP
- Bring students back together

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



JFLAP's Use Around the World

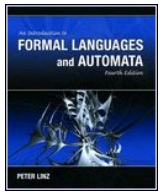
- JFLAP web page has over 200,000 hits since 1996
- Google Search
  - JFLAP appears on over 9830 web pages
  - Note: search only public web pages
- JFLAP been downloaded in over 160 countries

Conclusions From Study

- Results of Study showed
  - All the faculty used JFLAP in their courses, mostly for homework, some in lecture
  - Students had a high opinion of JFLAP
  - Majority of students felt access to JFLAP
    - Made learning course concepts easier
    - Made them feel more engaged
    - Made the course more enjoyable
  - Over half the students used JFLAP to study for exams
  - Over half the student thought time and effort using JFLAP helped them get a better grade.

Resources/Questions?

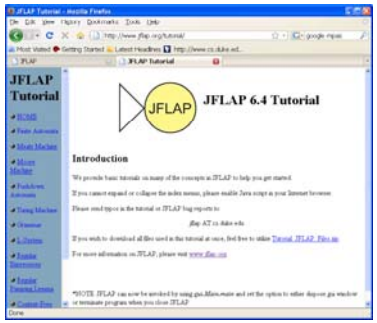
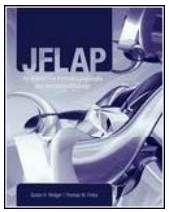
JFLAP works well with Linz book



New CD supplement with JFLAP exercises to go with this book

www.jflap.org

JFLAP tutorial



JFLAP book