Increasing Interaction and Support in the Formal Languages and Automata Theory Course
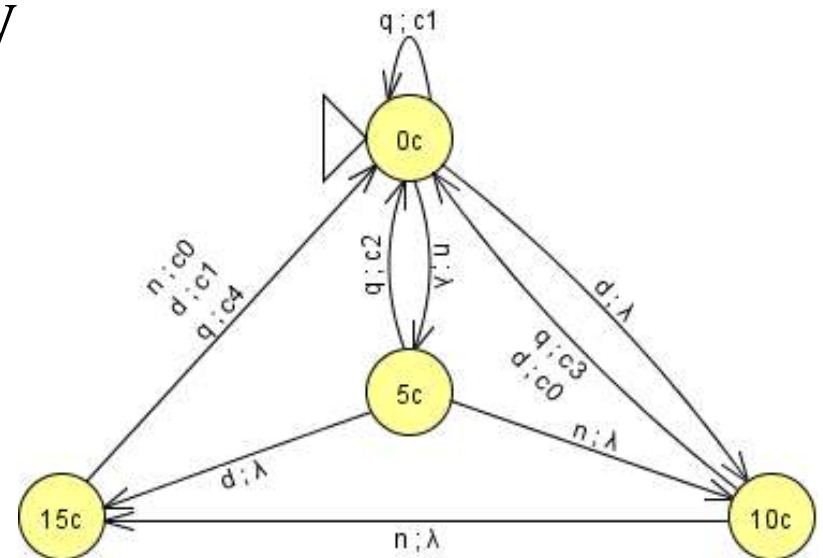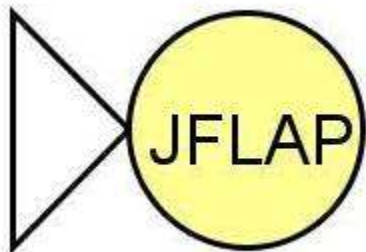
Susan H. Rodger
Duke University
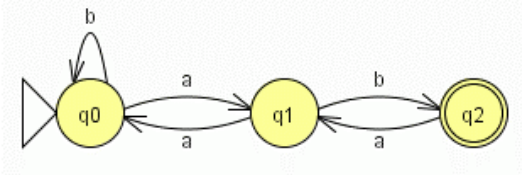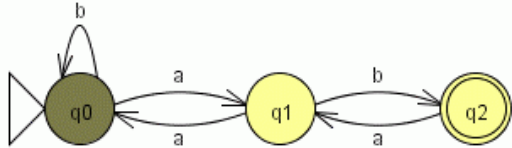
ITiCSE 2007
June 25, 2007

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?

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	
Visual	
Interactive	

Why Develop Tools for Automata? Examined 10 Automata textbooks
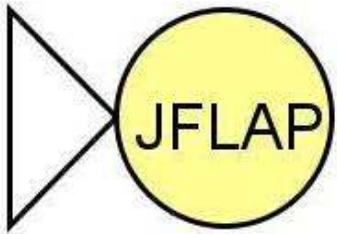
- One had software with book (not all topics)
- Only 6 had pictures of PDA, 2 or 3 states
- Only 6 had pictures of Turing machines, three of those switched representation
- Only 2 had picture of CFG to NPDA
- None had picture of parse tree for unrestricted grammar

Impact on Courses

- Not very visual
- Simple examples
- Exercises are limited
- Don't provide feedback

Overview of JFLAP

- **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 15 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 – 2005-2007 Stephen Reading, Bart Bressler, Jinghui Lim, Chris Morgan, Jason Lee

JFLAP – Regular Languages

- Create
  - DFA and NFA
  - Moore and Mealy
  - regular grammar
  - regular expression
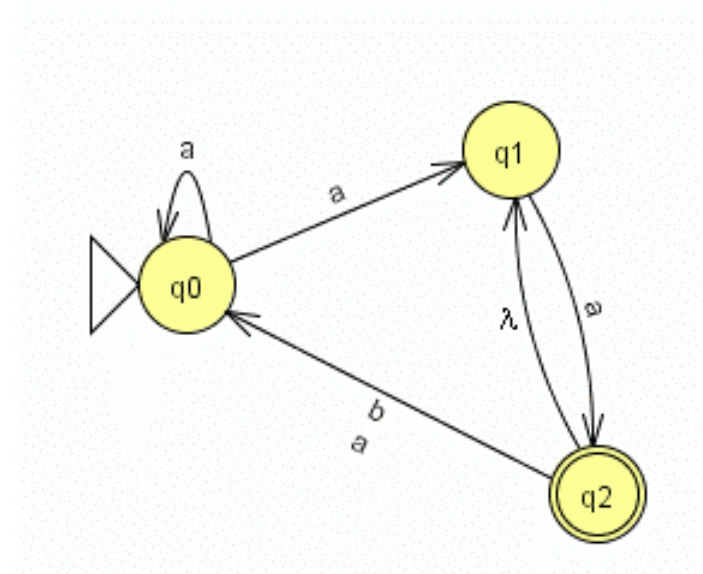
- Conversions
  - NFA to DFA to minimal DFA
  - NFA ←→ regular expression
  - NFA ←→ regular grammar
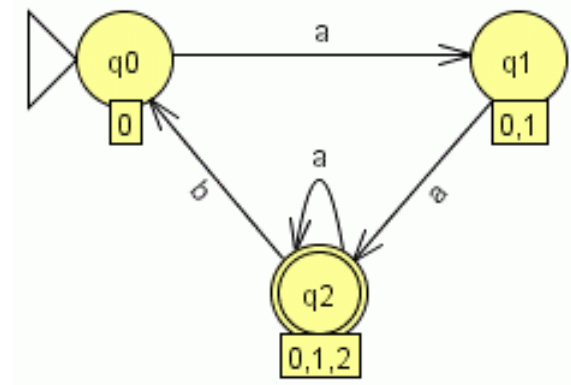
JFLAP – Regular languages (more)

- 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
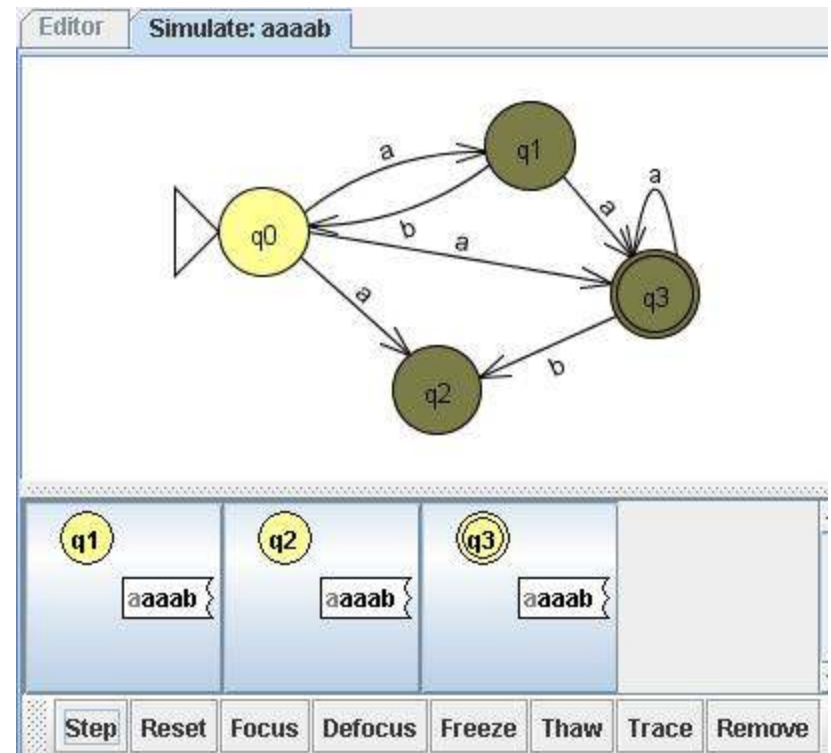
JFLAP – Context-free Languages

- Create
  - Nondeterministic PDA
  - Context-free grammar
  - Pumping Lemma
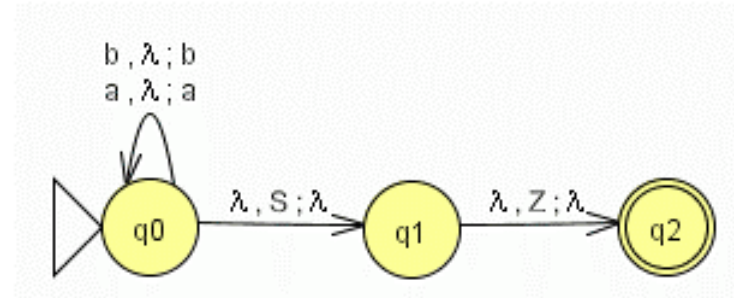
- Transform
  - PDA → CFG
  - CFG → PDA (LL & SLR parser)
  - CFG → CNF
  - CFG → Parse table (LL and SLR)
  - CFG → Brute Force Parser
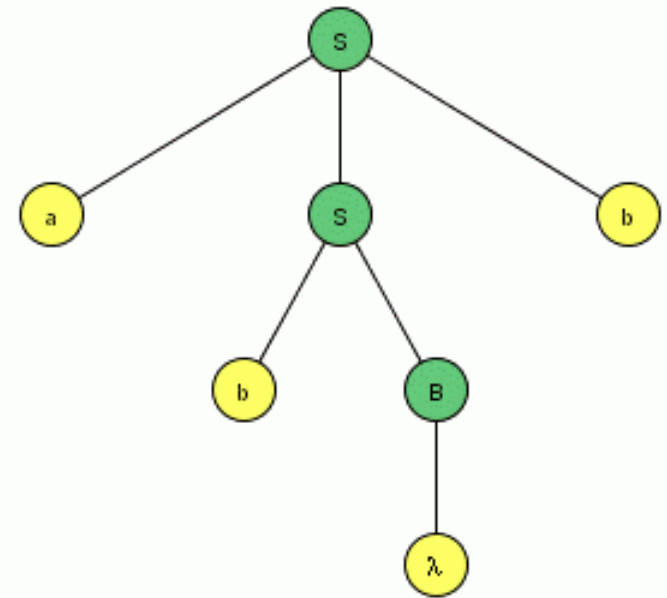
JFLAP – Recursively Enumerable Languages

- Create
  - Turing Machine (1-Tape)
  - Turing Machine (multi-tape)
  - Building Blocks
  - Unrestricted grammar
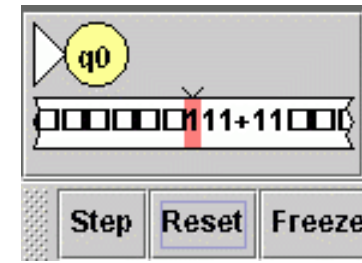
- Parsing
  - Unrestricted grammar with brute force parser
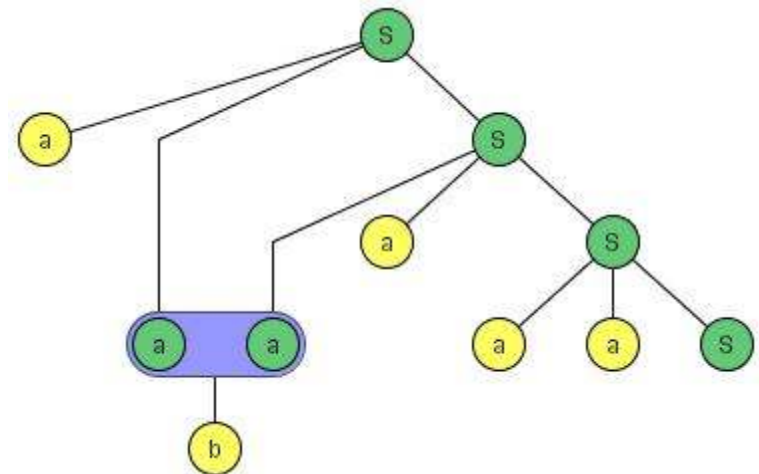
JFLAP - L-Systems
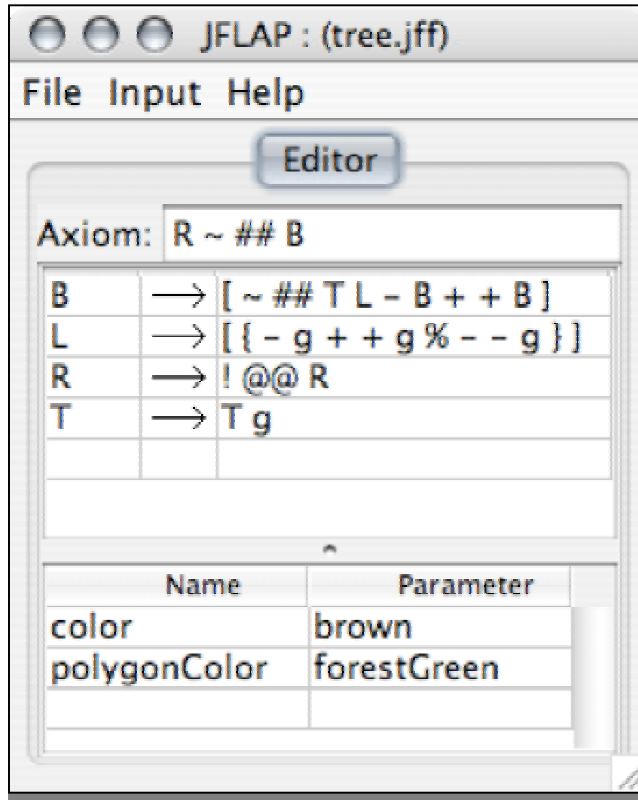
•This L-System renders as a tree that grows larger with each successive derivation step.
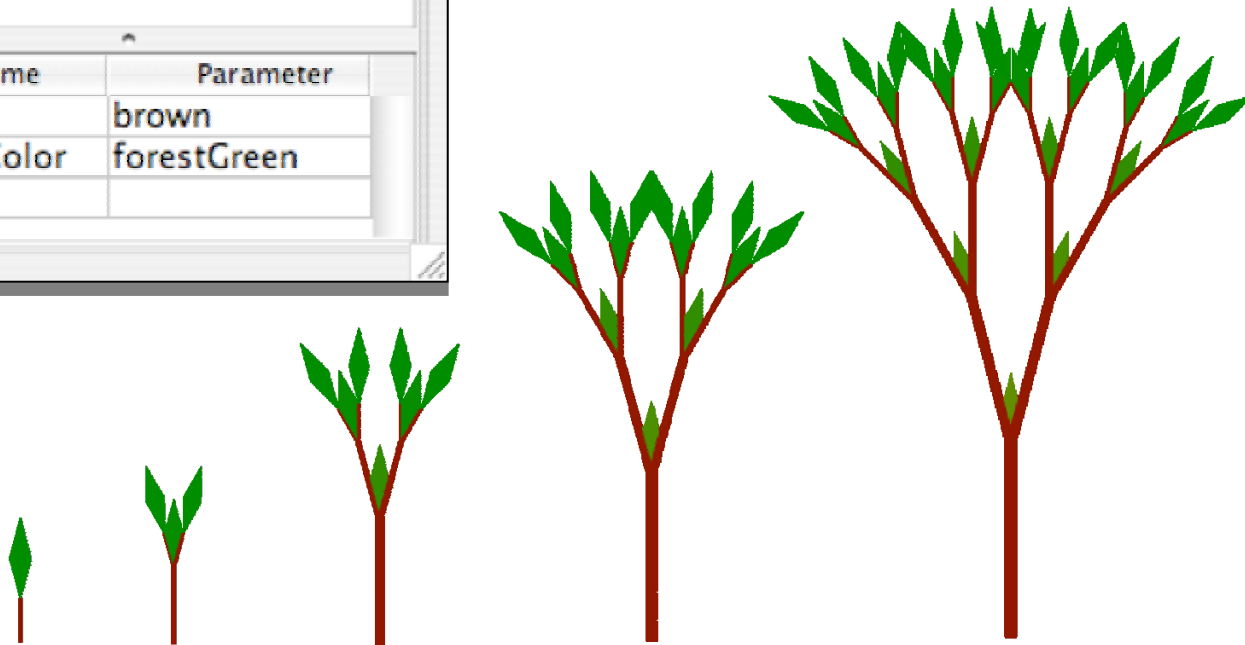
Students love L-Systems
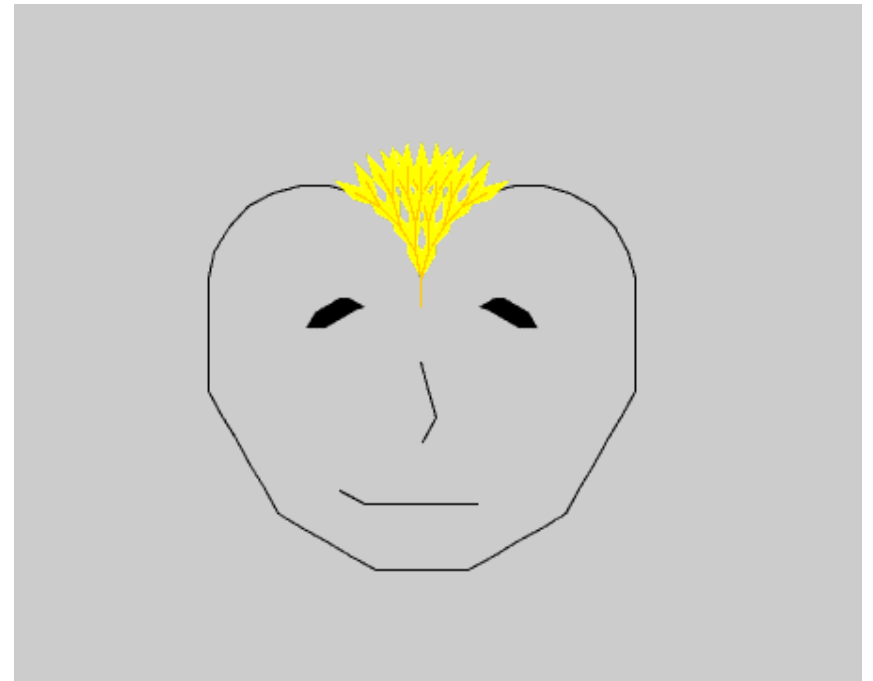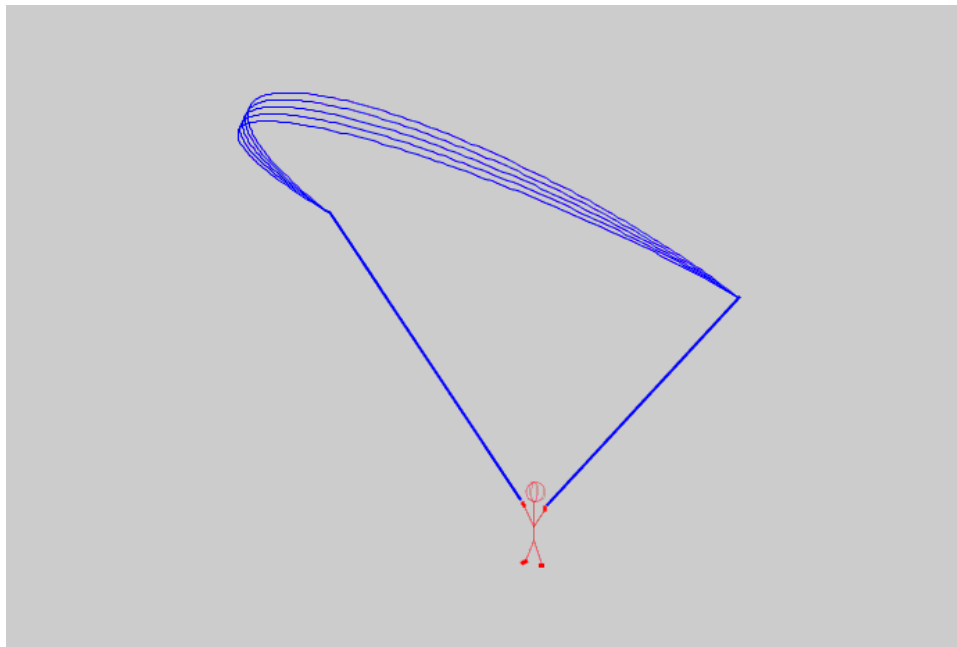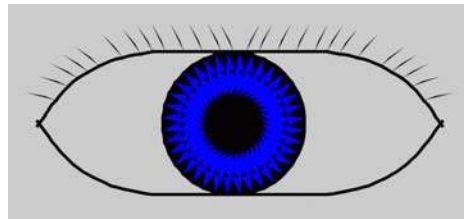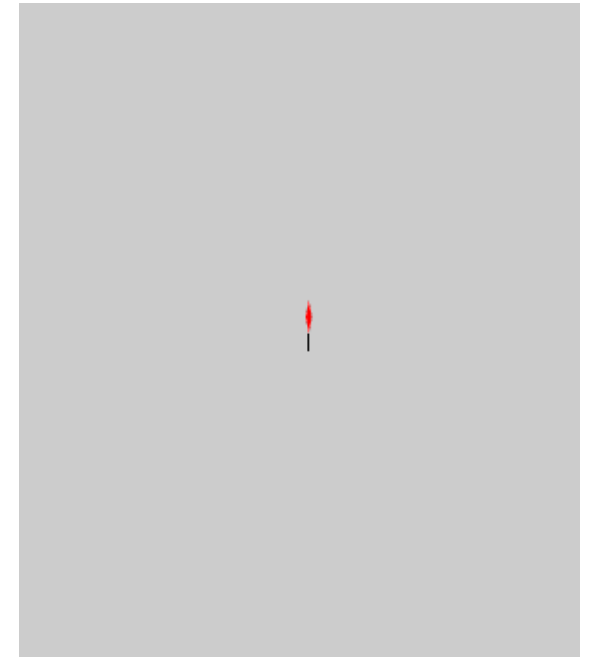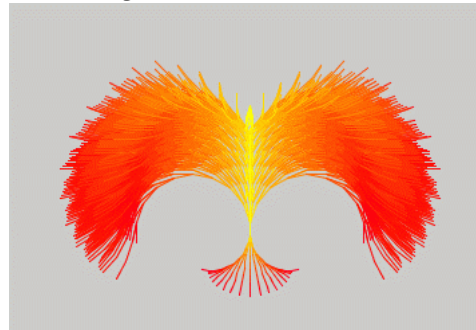
Increasing Interaction in the Course with JFLAP

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
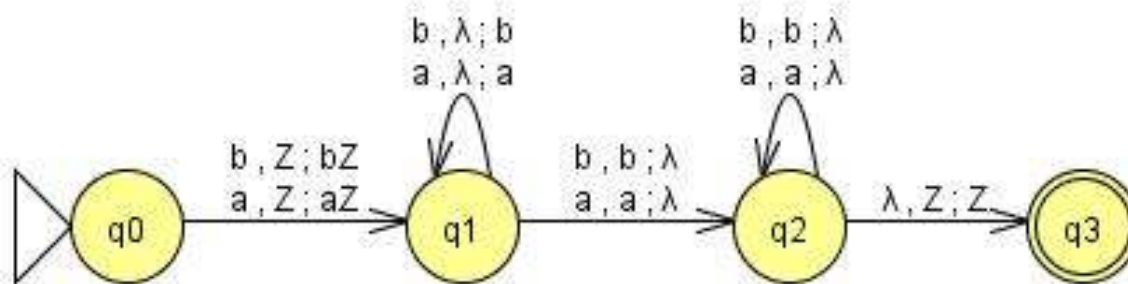
Example 1: JFLAP during Lecture (cont)

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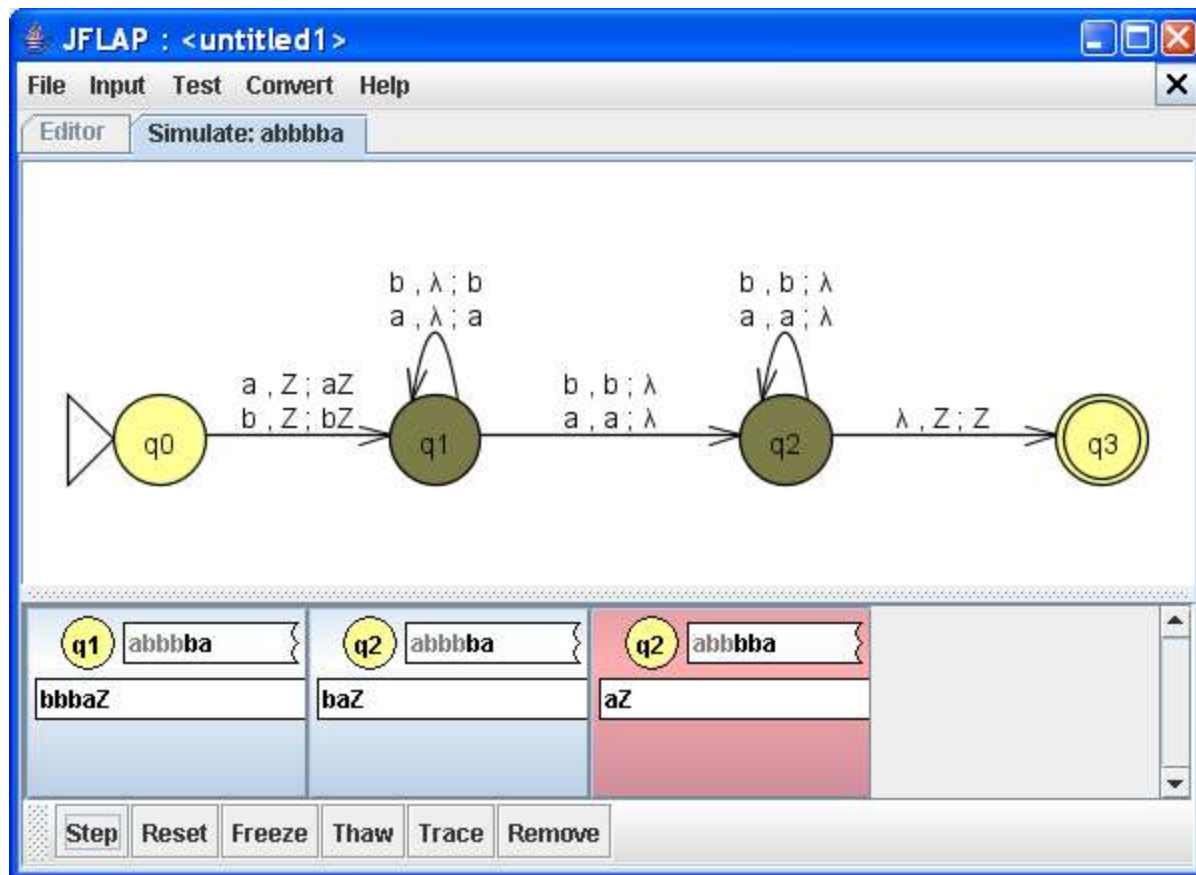
Example 2: JFLAP during Lecture

- **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!

S	$\longrightarrow$	aB
B	$\longrightarrow$	BB
B	$\longrightarrow$	aBa
B	$\longrightarrow$	b
B	$\longrightarrow$	λ

S	$\longrightarrow$	aB
B	$\longrightarrow$	BB
B	$\longrightarrow$	aBa
B	$\longrightarrow$	b
S	$\longrightarrow$	a
B	$\longrightarrow$	B
B	$\longrightarrow$	aa

Example 2 (cont)
Parse Tree Results

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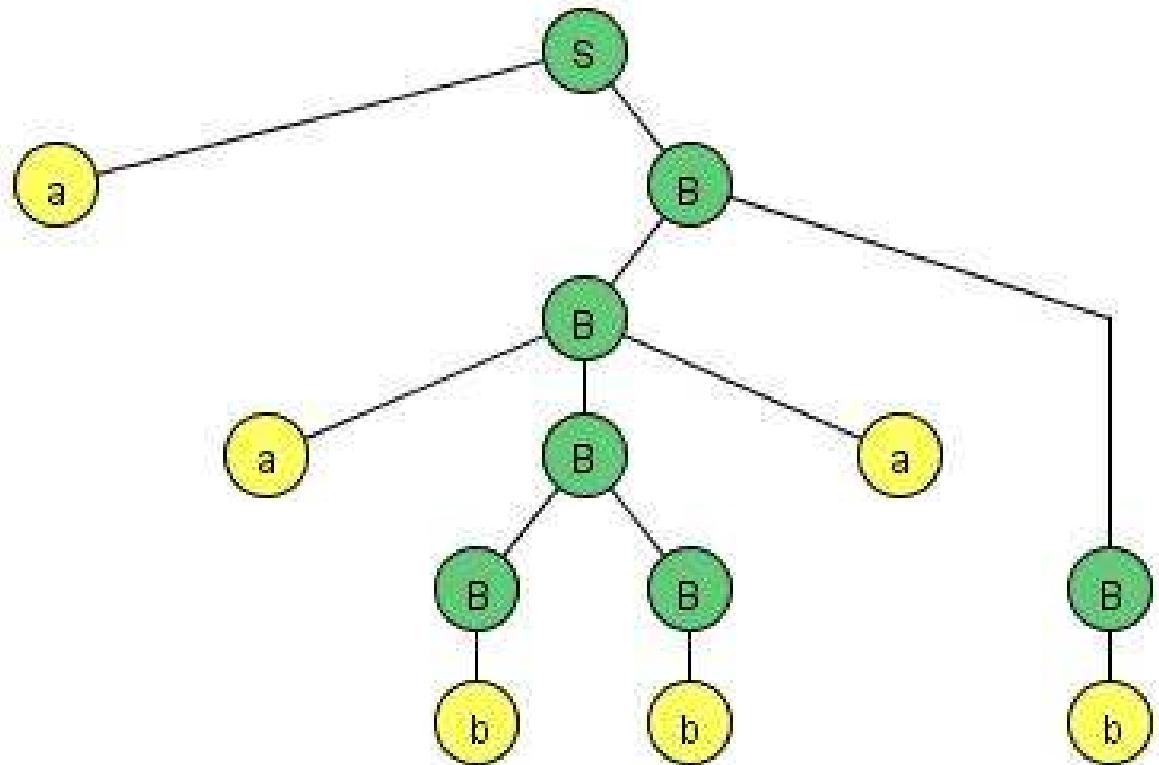
With JFLAP, Exploring Concepts too tedious for paper

- 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
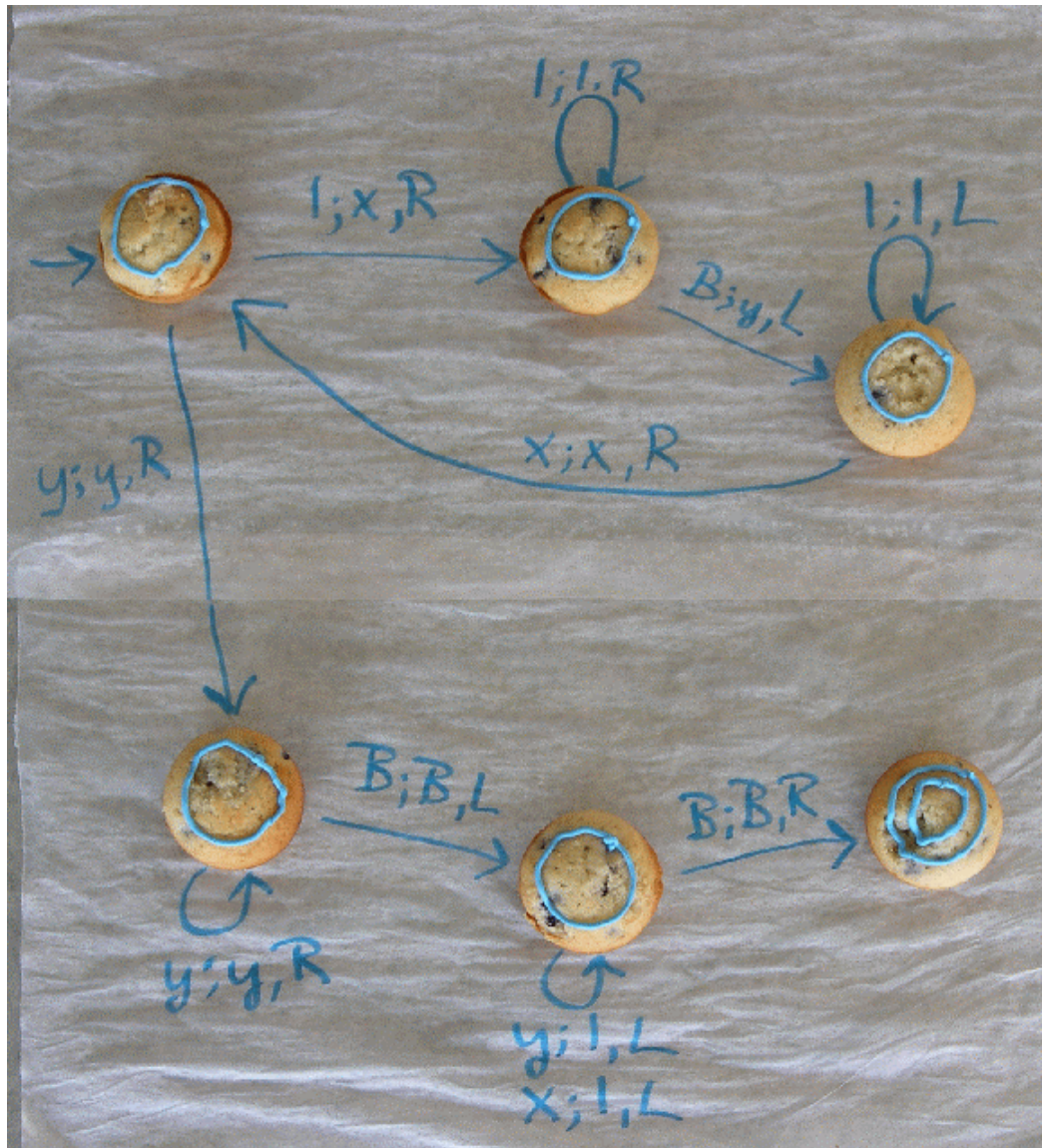
Students building DFA with cookies and icing
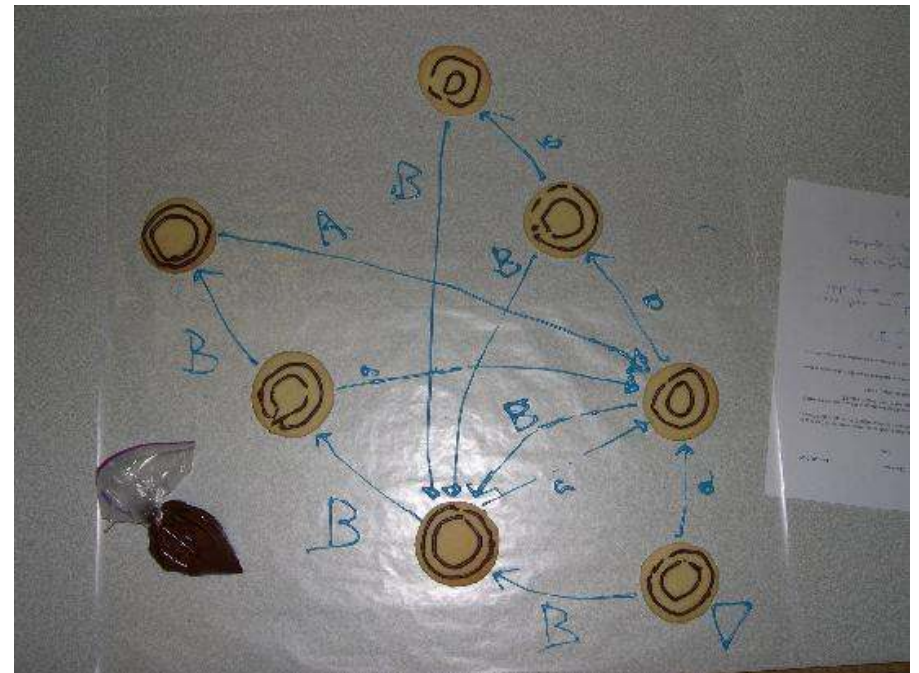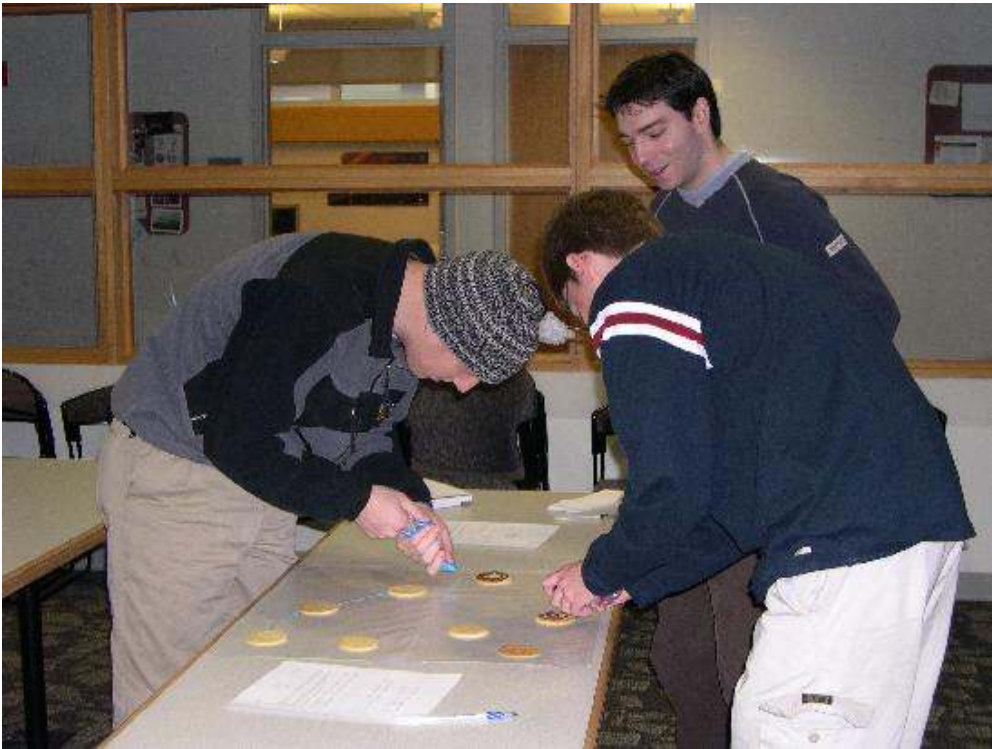
The Smart Waitress vs Customer

- 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
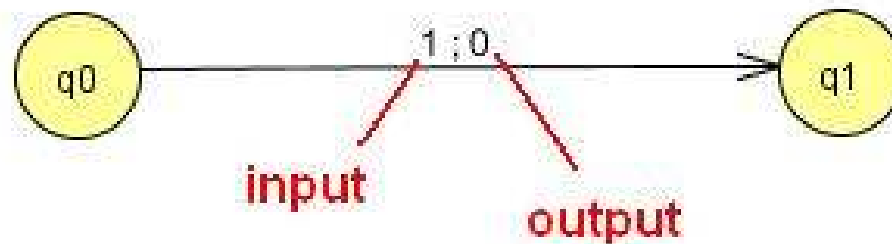
Example – Vending Machine

- 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
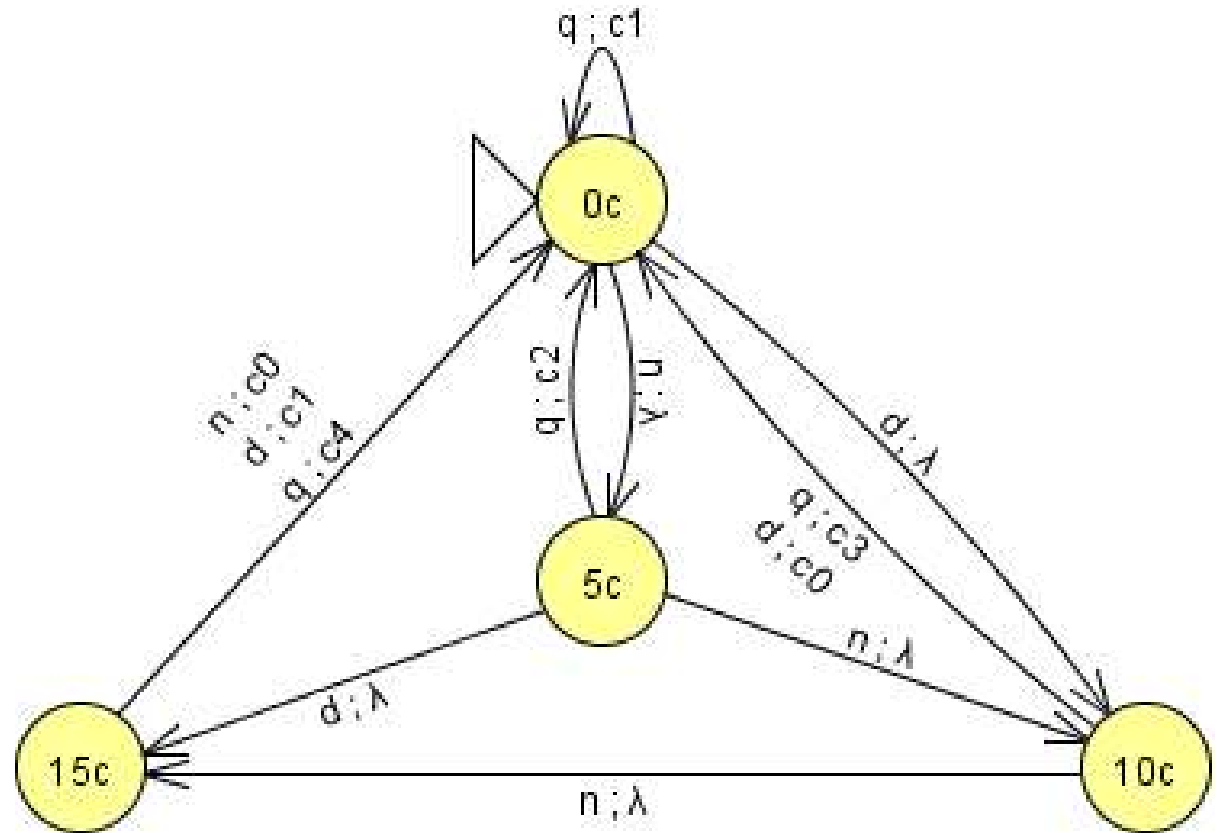
Input	Result
ndq	c4
qqq	c1c1c1
nq	c2
dq	c3
dd	c0
ddn	c0

New in JFLAP: Moore Machine

- Similar to Mealy Machine
  - No final state
  - Output is produce by states, not transitions
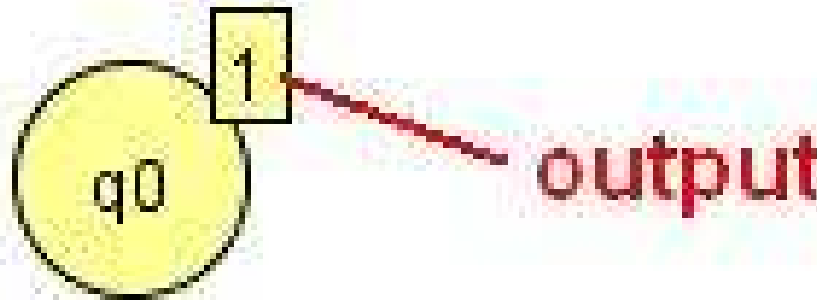
Example – Halve a Binary Number
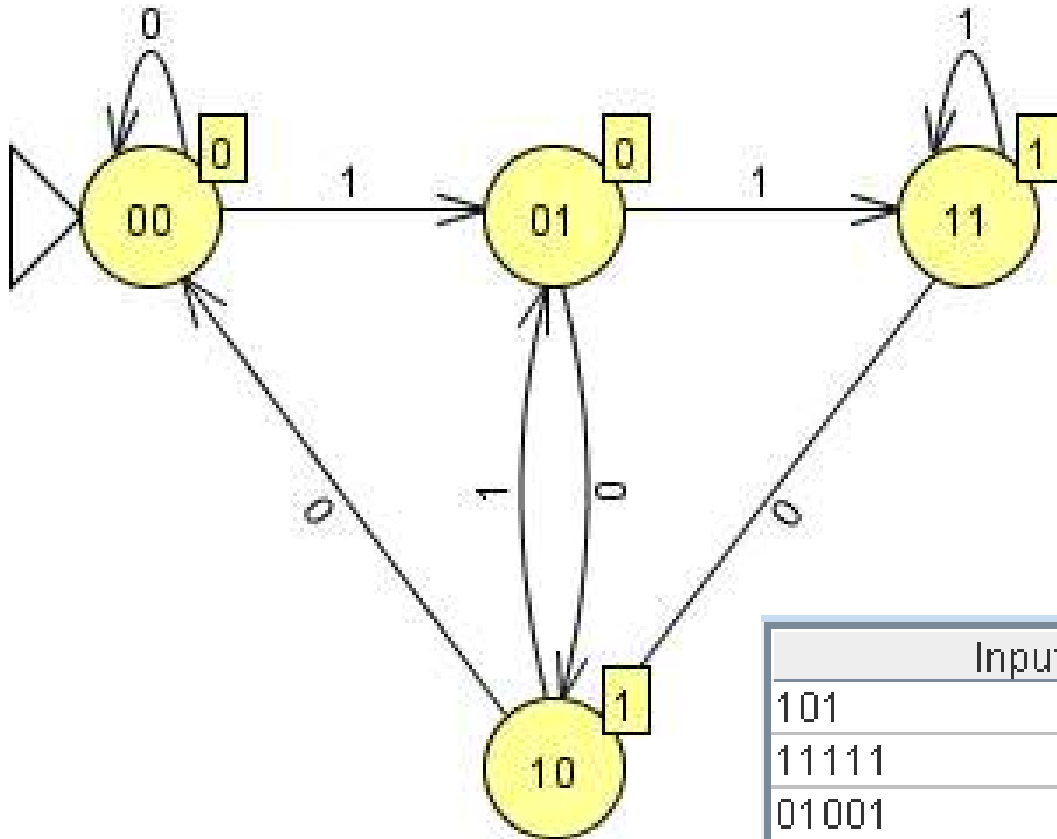
Input	Result
101	0010
11111	001111
01001	000100
100	0010
110	0011
1	00
0	00

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**

$$|xy| \leq m$$
$$|y| \geq 1$$
$$xy^i z \in L \text{ for all } i \geq 0$$

Pick an Example

Select a Pumping Lemma

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

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

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

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

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

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

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

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

$L = \{a^n : n \text{ is even}\}$ Select

JFLAP Pump lemma Game

User enters in steps 1 and 3

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

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

1. Select integer m

6   **Start over**

2. Given integer m, here's string w such that |w| >= m

aaaaaabbbbbb

3. Select decomposition of w into xyz

x: aaa |x|: 3

y: aa |y|: 2   **Set xyz**

z: abbbbbb |z|: 7

| a | a | a | a | a | a | b | b | b | b | b | b |

4. A choice of i to give contradiction

i: 0   pumped string: aaaabbbbbb

5. Animation

$$\begin{array}{ccc} x & y & z \end{array}$$

$$w = aaa\ \ aa\ \ abbbbbb$$

$xy^0z = a^4b^6$ = aaaabbbbbb is NOT in the language: CONTRADICTION.   **Step**   Restart

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 $i \geq 0$, $uv^i xy^i z \in L$

Similar CFL pump lemma game

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

Messages

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

1. Select integer m

5   [Start over]

2. Given integer m, here's string w such that |w| >= m

aaaaabbbbbccccc

3. Select decomposition of w into uvxyz

u: aa |u|: 2

v: a |v|: 1

x: a |x|: 1   [Set uvxyz]

y: abb |y|: 3

z: bbbccccc |z|: 8

| a | a | a | a | a | b | b | b | b | b | c | c | c | c | c |

4. A choice of i to give contradiction

i: 2   pumped string: aaaaaabbabbbbbccccc

Cases:

#	Description

CFL pump lemma (cont)

- Last step shows the contradiction

5. Animation

$$\begin{array}{cccccc} & u & v & x & y & z \\ w = & aa & a & a & abb & bbbccccc \end{array}$$

$uv^2xy^2z = a^6b^2ab^5c^5 =$

aaaaaabbabbbbbccccc is NOT in the language: CONTRADICTION.

Step Restart

- In CFL – there are lots of cases to consider

#	Description
1	v is a string of "a"s and y is a string of "a"s
2	v is a string of "a"s and y is a string of "a"s followed by "b"s
3	v is a string of "a"s and y is a string of "b"s
4	v is a string of "a"s followed by "b"s and y is a string of "b"s
5	v is a string of "b"s and y is a string of "b"s
6	v is a string of "b"s and y is a string of "b"s followed by "c"s
7	v is a string of "b"s and y is a string of "c"s
8	v is a string of "b"s followed by "c"s and y is a string of "c"s
9	v is a string of "c"s and y is a string of "c"s
10	v is an empty string and y is a non-empty string
11	v is a non-empty string and y is an empty string

Cases:

New in JFLAP:
Batch Testing Mode

- Select several files for testing

- Then select input file
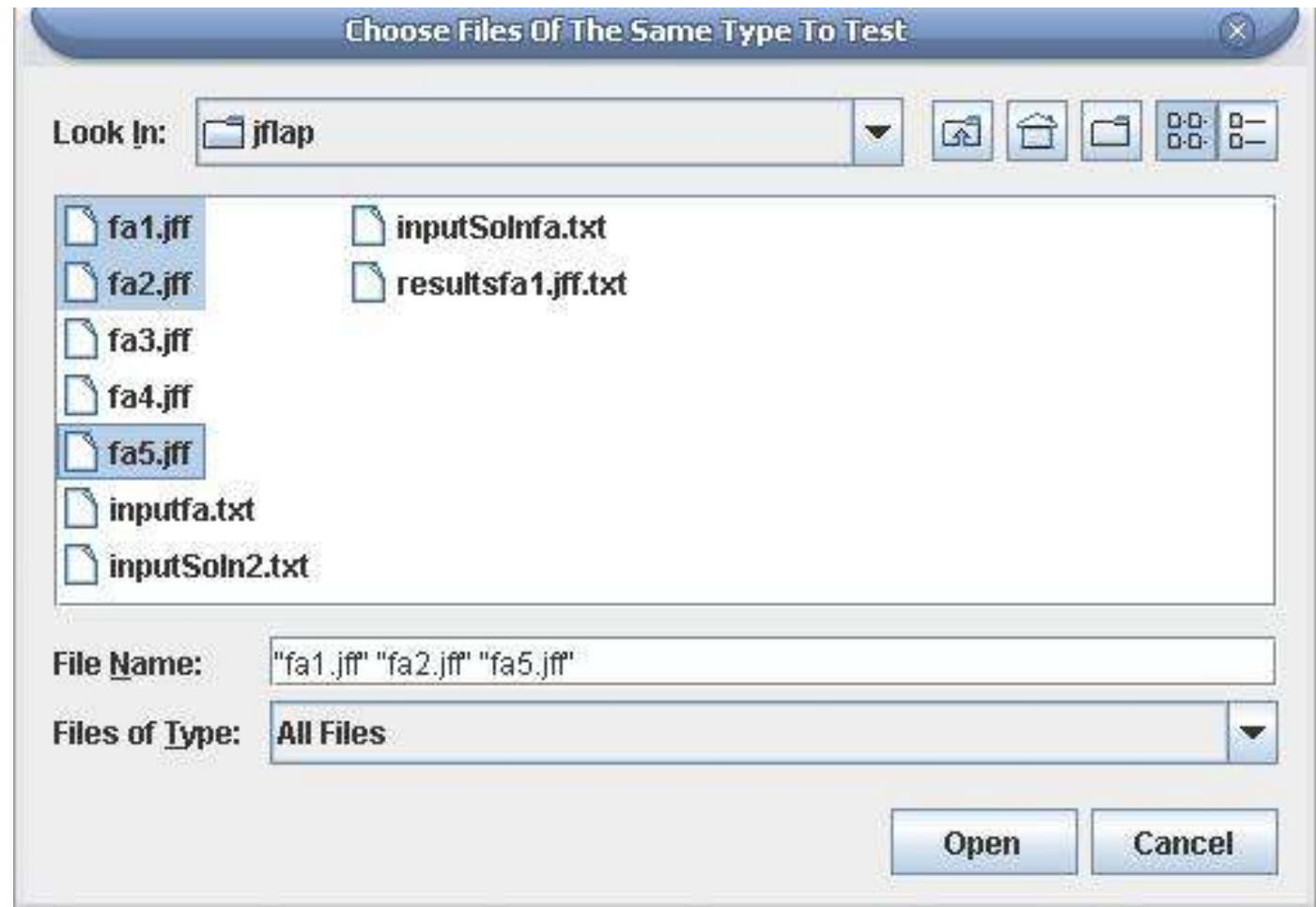
Batch Testing Mode (cont)

File	Input	Result
fa1.jff	aabbaabb	Reject(Accept)
fa1.jff	aaaa	Reject(Accept)
fa1.jff	aab	Accept
fa1.jff	aabaaab	Accept(Reject)
fa2.jff	aabbaabb	Reject(Accept)
fa2.jff	aaaa	Reject(Accept)
fa2.jff	aab	Accept
fa2.jff	aabaaab	Accept(Reject)
fa5.jff	aabbaabb	Accept
fa5.jff	aaaa	Accept
fa5.jff	aab	Accept
fa5.jff	aabaaab	Accept(Reject)

| Run Inputs | View Trace | Edit File | Add input string | Add file | Remove file | Save |

JFLAP Study

- Study of JFLAP's effectiveness in learning
  - Runs 2005-2007
  - 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
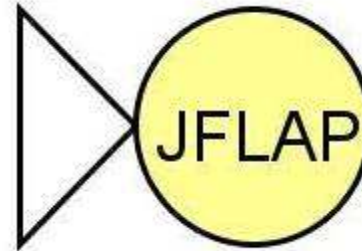
JFLAP's Use Around the World

- 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