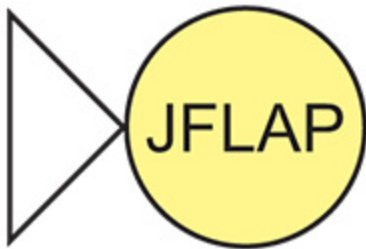


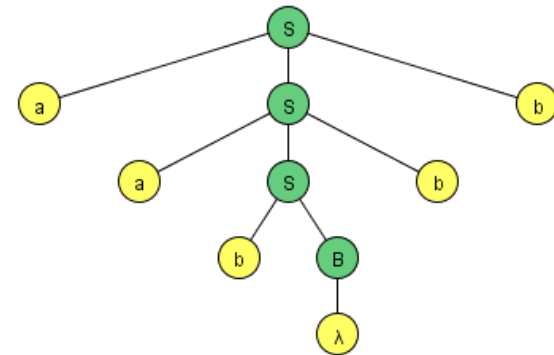
An Interactive Approach to Formal Languages and Automata with JFLAP



NSF Grant DUE CCLI-EMD 0442513



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

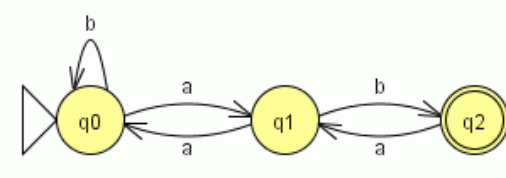
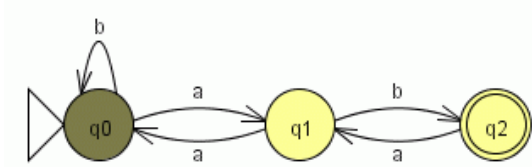


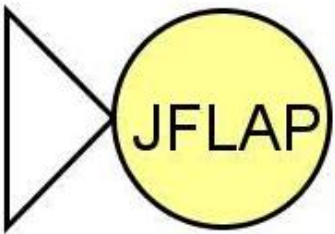
SIGCSE 2011 AlgoViz Workshop
March 9, 2011

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	<table><tr><td></td><td>a</td><td>b</td></tr><tr><td>q_0</td><td>q_1</td><td>q_0</td></tr><tr><td>q_1</td><td></td><td>q_2</td></tr><tr><td>q_2</td><td></td><td></td></tr></table>		a	b	q_0	q_1	q_0	q_1		q_2	q_2		
	a	b											
q_0	q_1	q_0											
q_1		q_2											
q_2													
Visual													
Interactive													

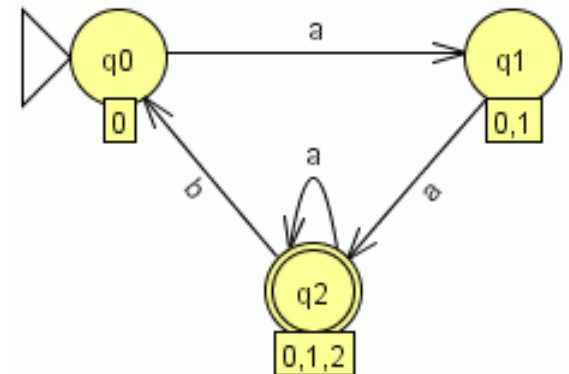
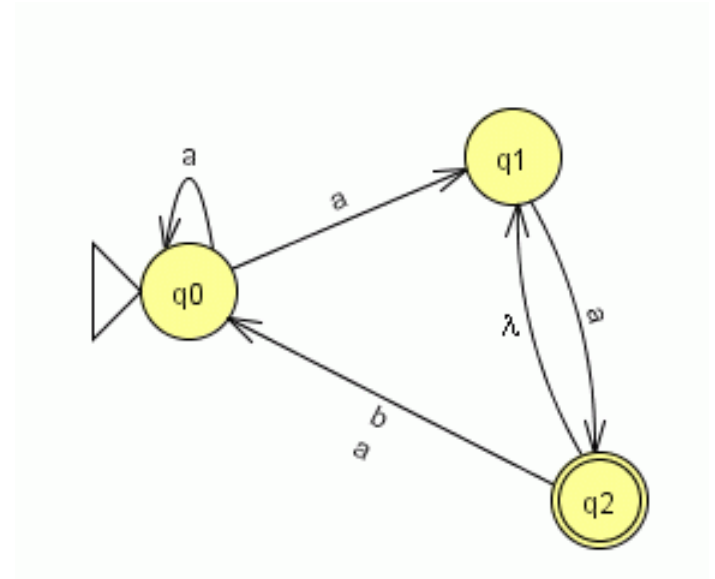


Overview of JFLAP

- **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!**

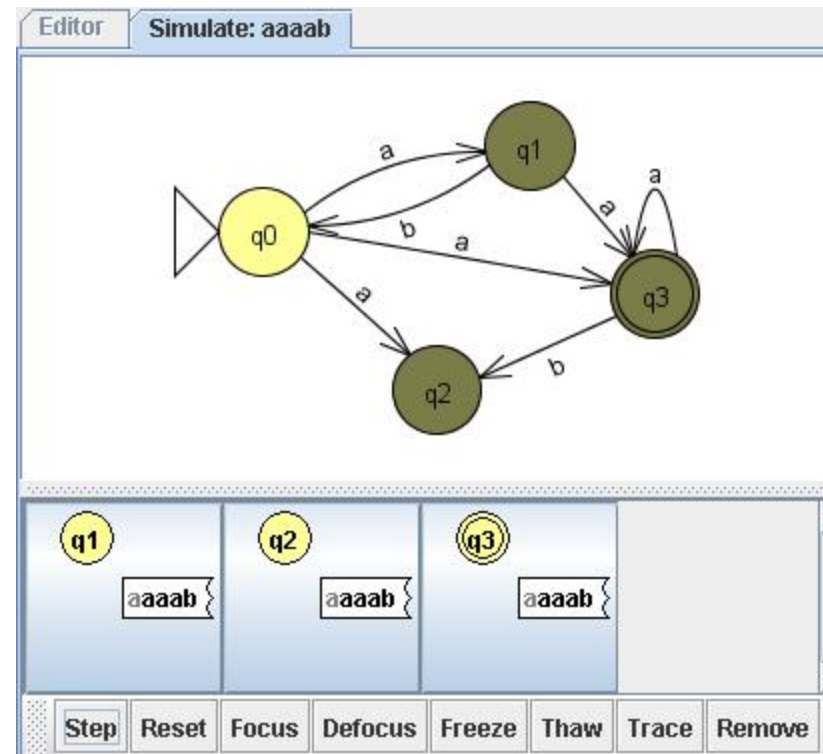
JFLAP – Regular Languages

- 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



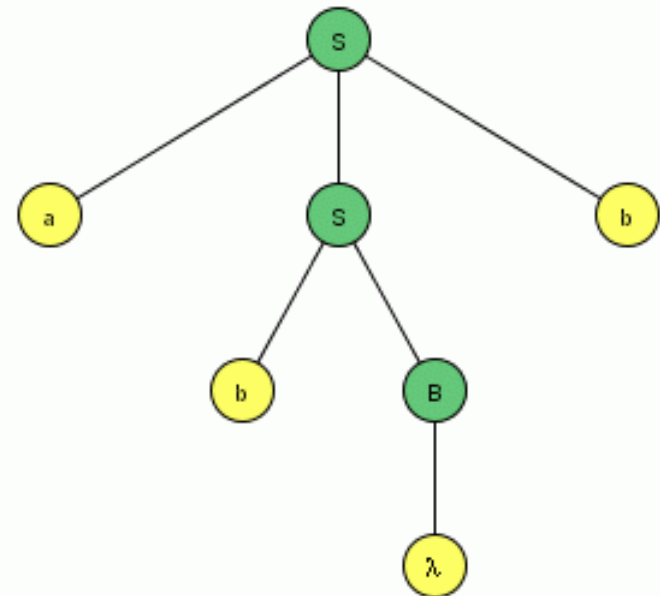
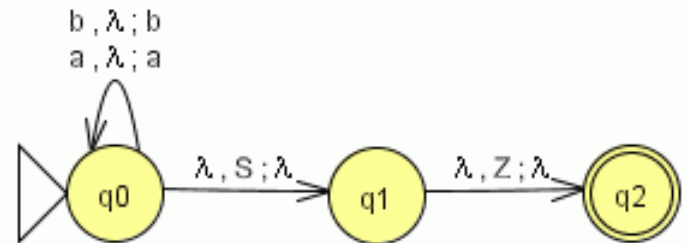
JFLAP – Regular languages (more)

- 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



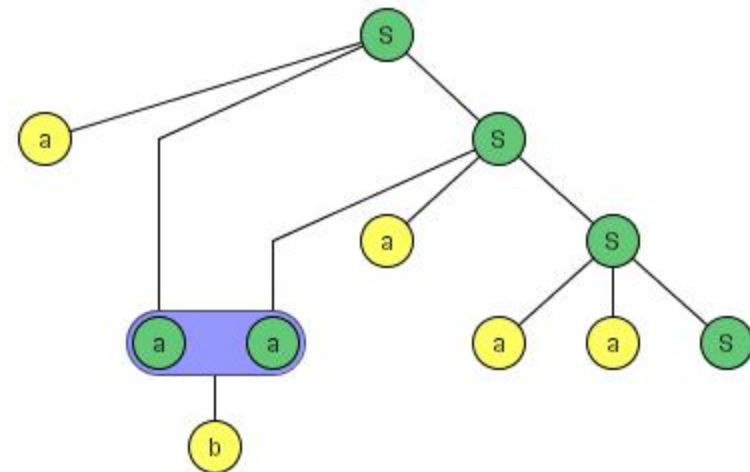
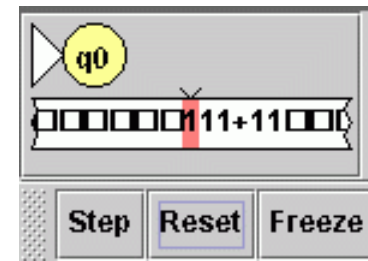
JFLAP – Context-free Languages

- 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

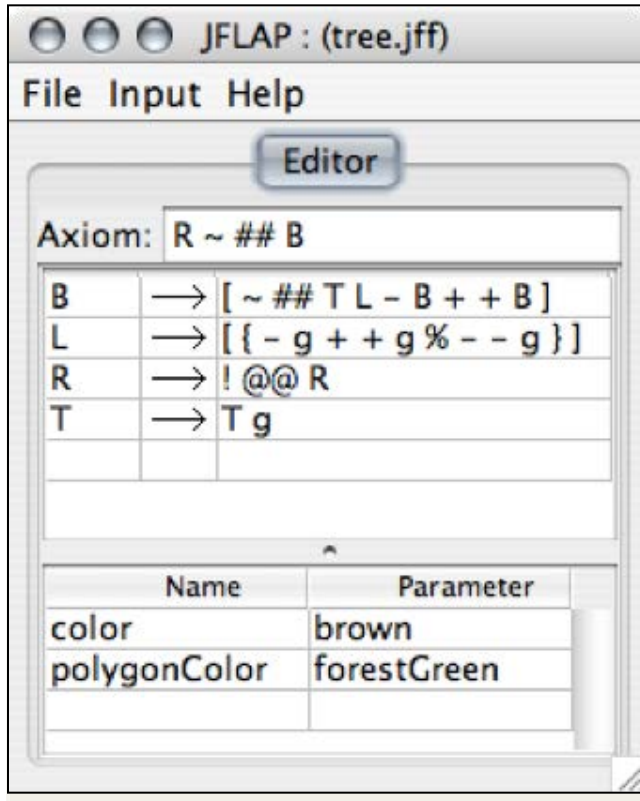


JFLAP – Recursively Enumerable Languages

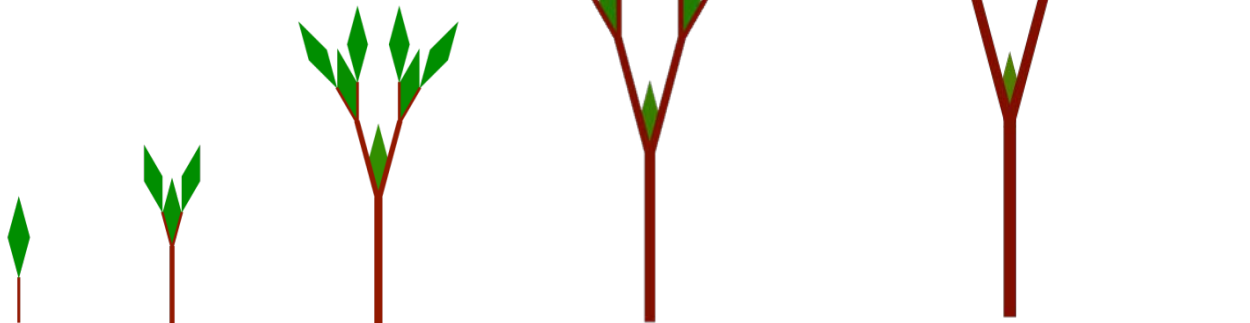
- Create
 - Turing Machine (1-Tape)
 - Turing Machine (multi-tape)
 - Building Blocks
 - Unrestricted grammar
- Parsing
 - Unrestricted grammar with brute force parser



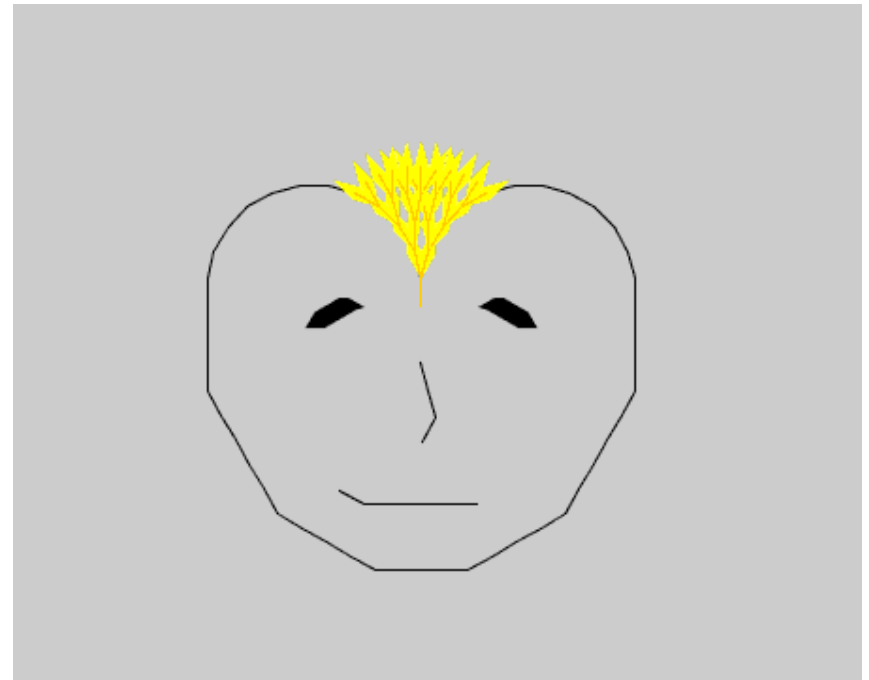
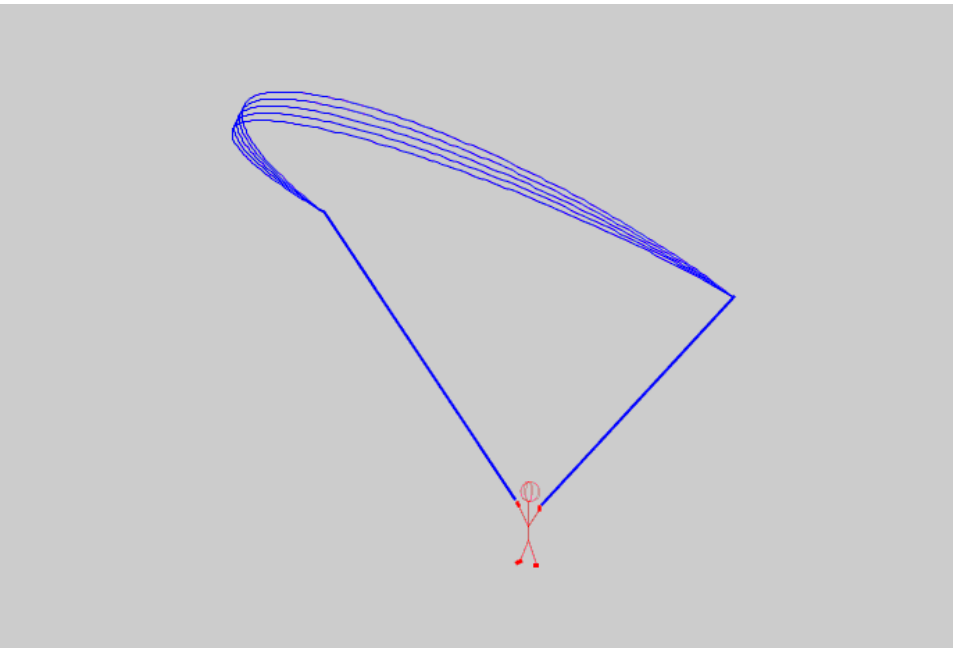
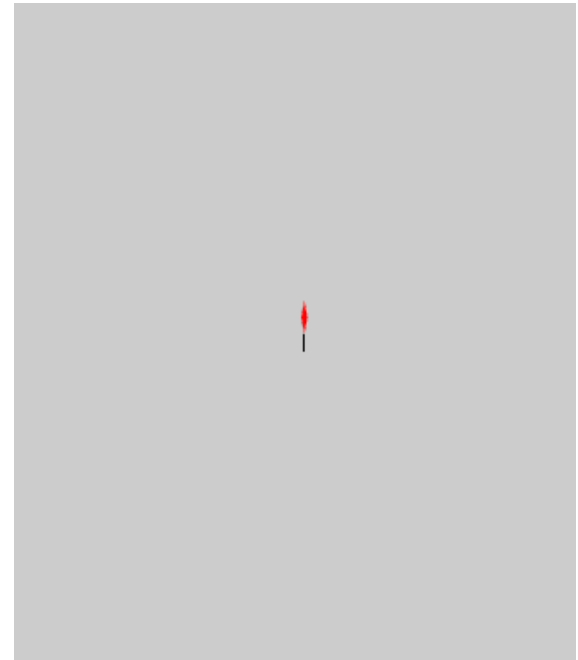
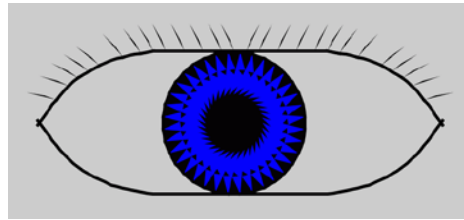
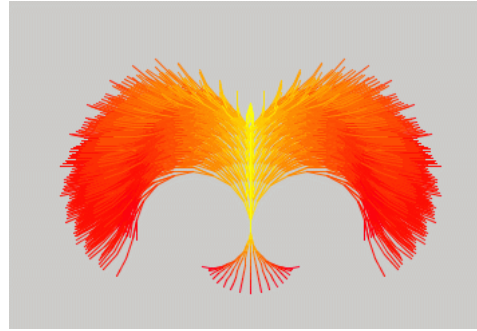
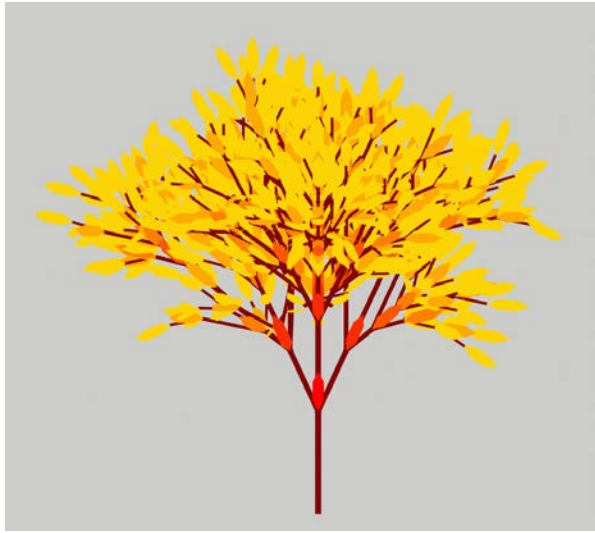
JFLAP - L-Systems



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



Students love L-Systems



Two-year JFLAP Study

Fourteen Faculty Adopter Participants

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

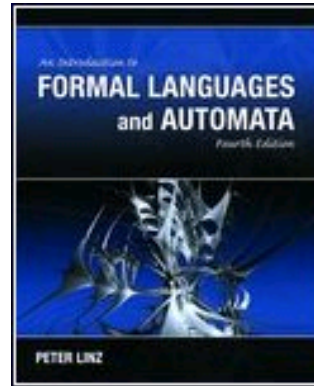
- Duke
- UNC-Chapel Hill
- Emory
- Winston-Salem State University
- United States Naval Academy
- Rensselaer Polytechnic Institute
- UC Davis
- Virginia State University
- Norfolk State University
- University of Houston
- Fayetteville State University
- University of Richmond
- San Jose State University
- Rochester Institute of Technology

Key Findings

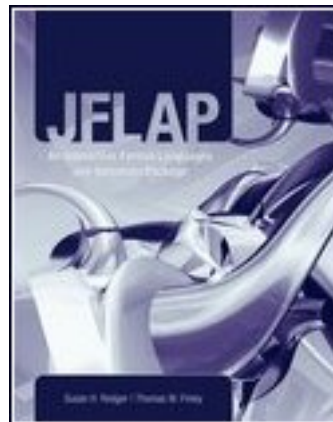
- All the faculty used JFLAP in their courses
 - They used it mostly for homework, some used it for class demonstrations.
- Students had a high opinion of JFLAP
- Four-fifths of the students thought JFLAP was easy to use to draw automata, simulate and interpret the results.
- The majority of students felt that having access to JFLAP made learning course concepts easier, made them feel more engaged in the course and made the course more enjoyable.
- Over half of the students used JFLAP to study for exams, and thought that the time and effort spent using JFLAP helped them get a better grade in the course.
- There was a control group in the second year, but the difference in knowledge between the control group and the JFLAP group was not statistically significant.

JFLAP Materials

JFLAP works well with
Linz book

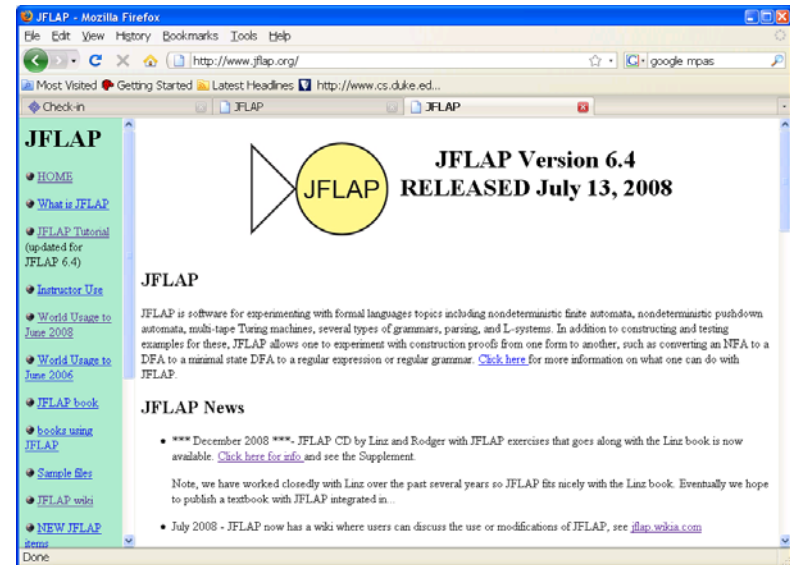


New CD supplement
with JFLAP
exercises to go
with this book

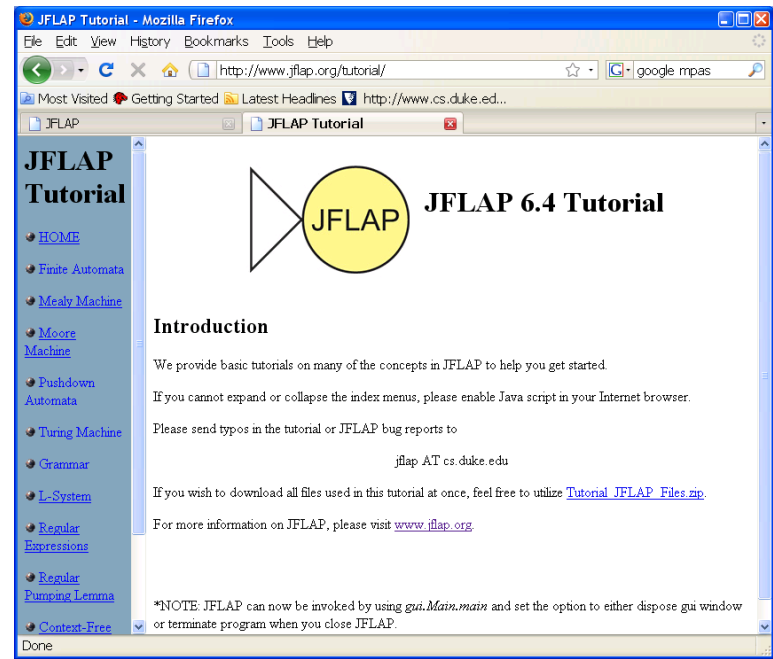


JFLAP book

www.jflap.org



JFLAP online tutorial



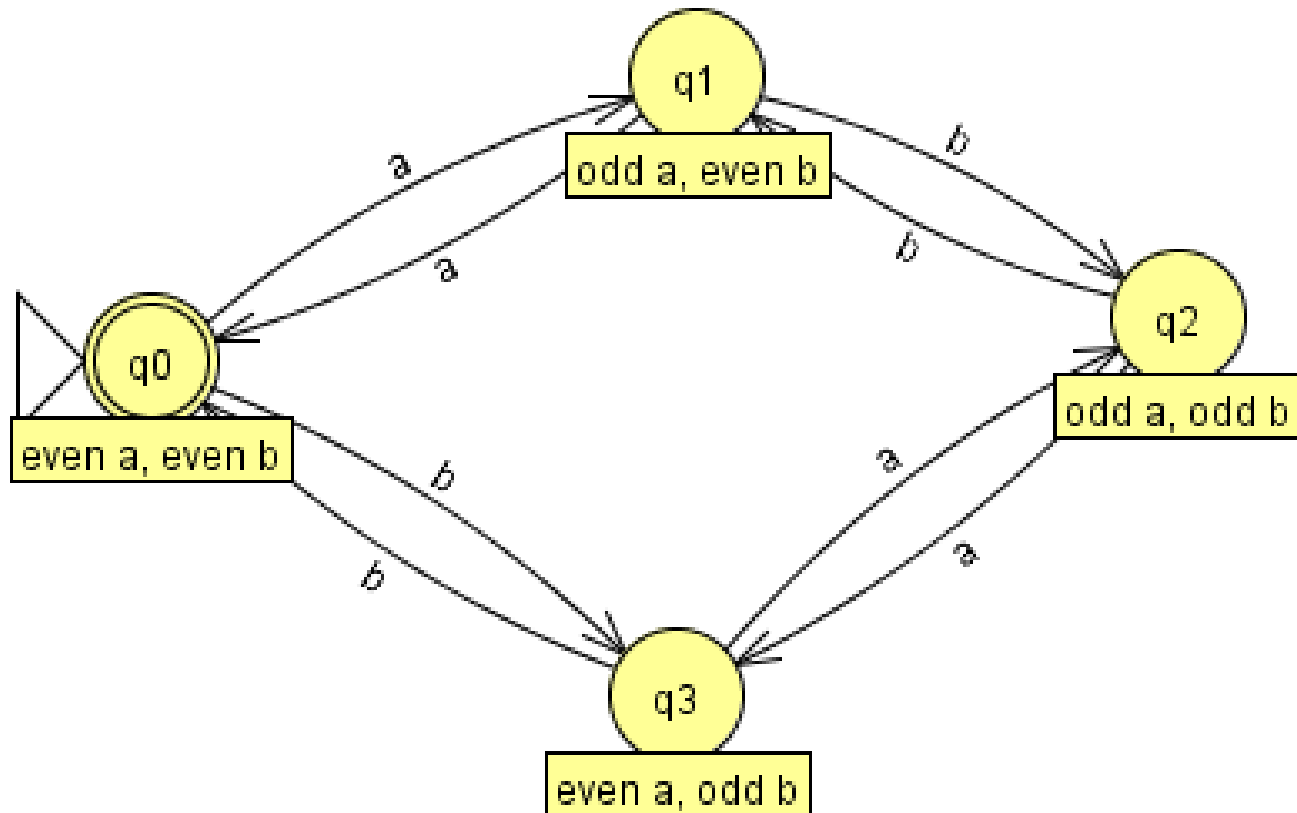
JFLAP Examples in Lecture

Example

- Create a DFA that recognizes strings with an even number of a's and an even number of b's

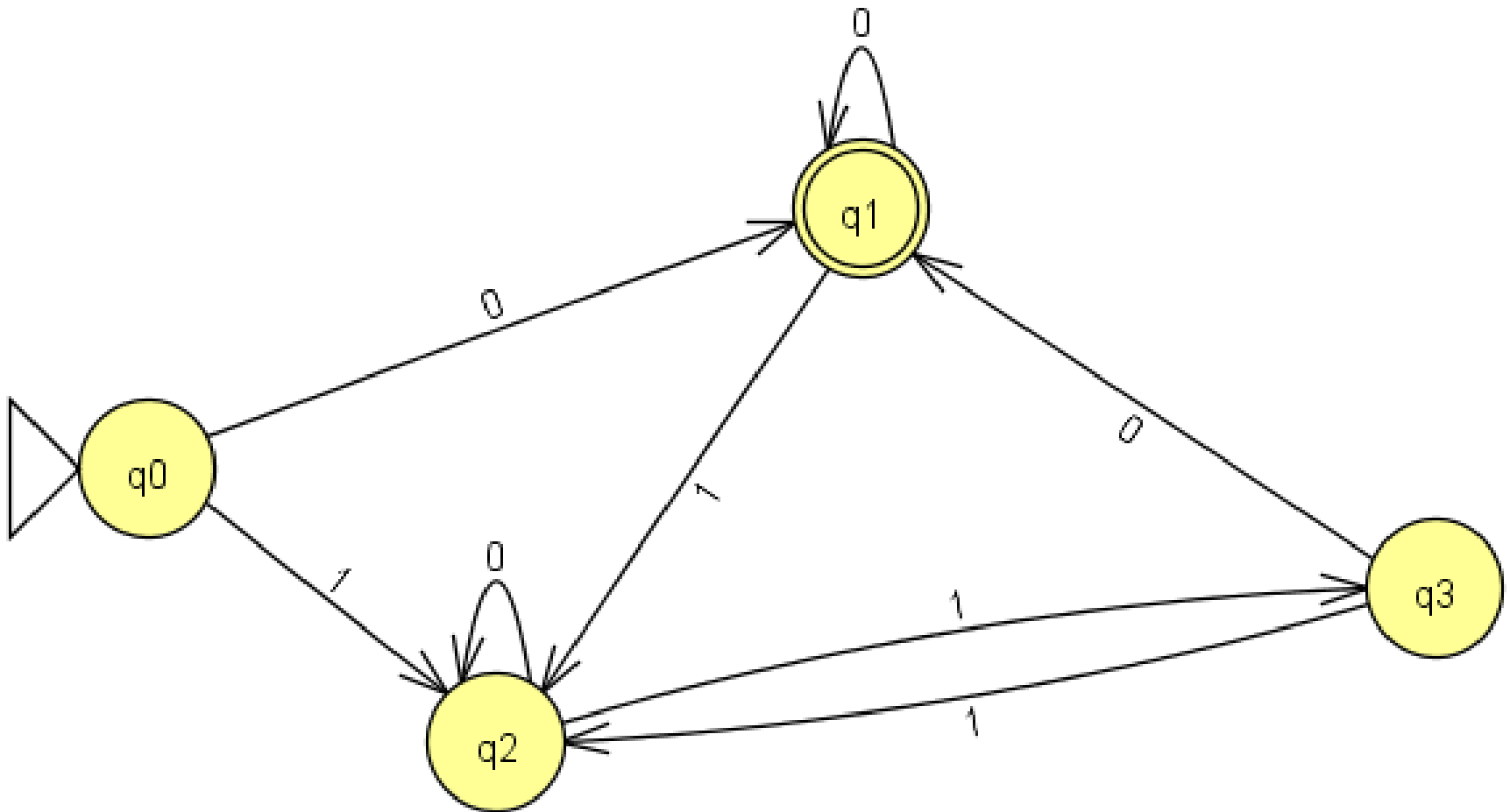
Example

- Create a DFA that recognizes strings with an even number of a's and an even number of b's



Example – DFA for even binary numbers with an even number of ones

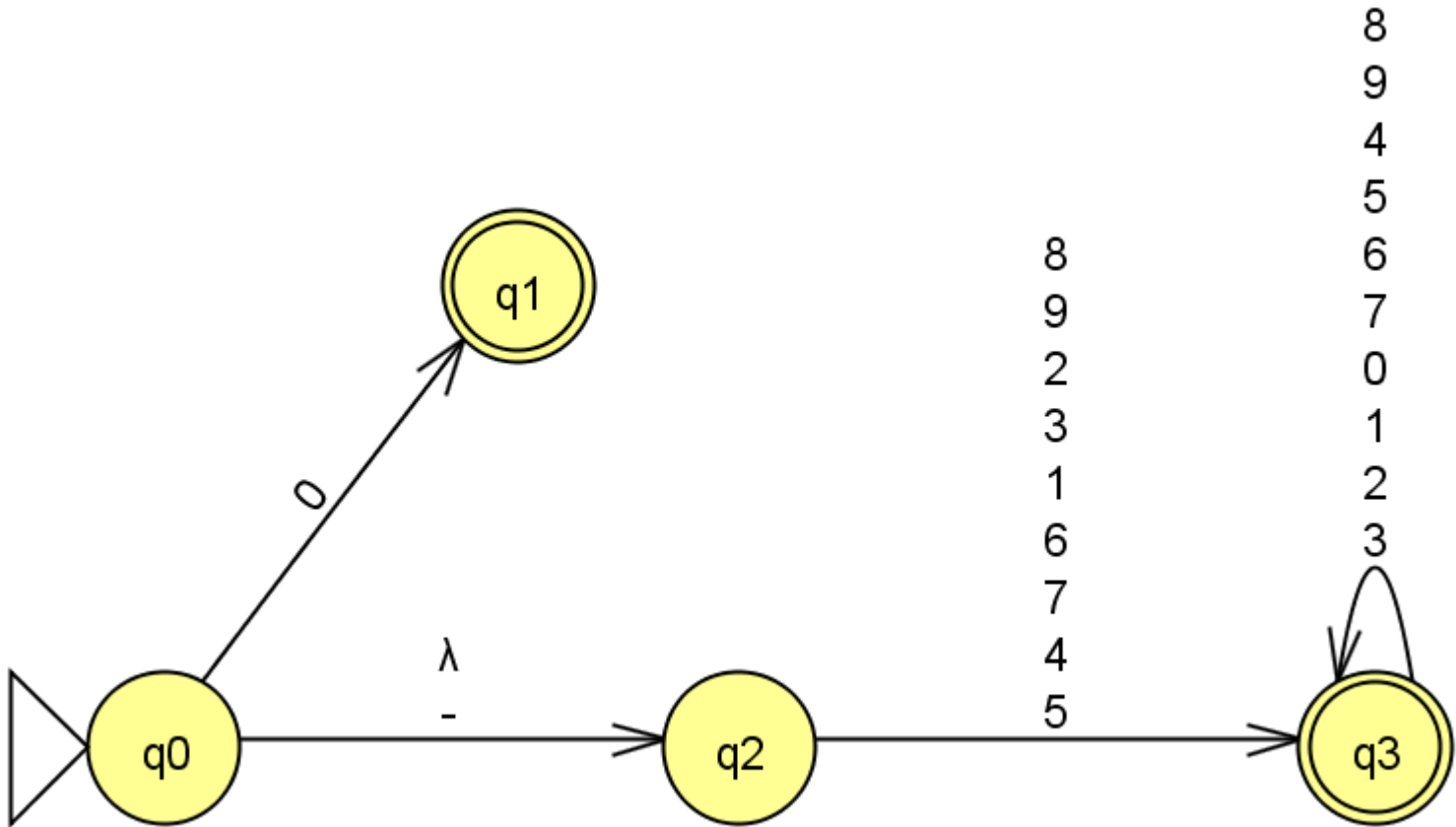
Example – DFA for even binary numbers with an even number of ones



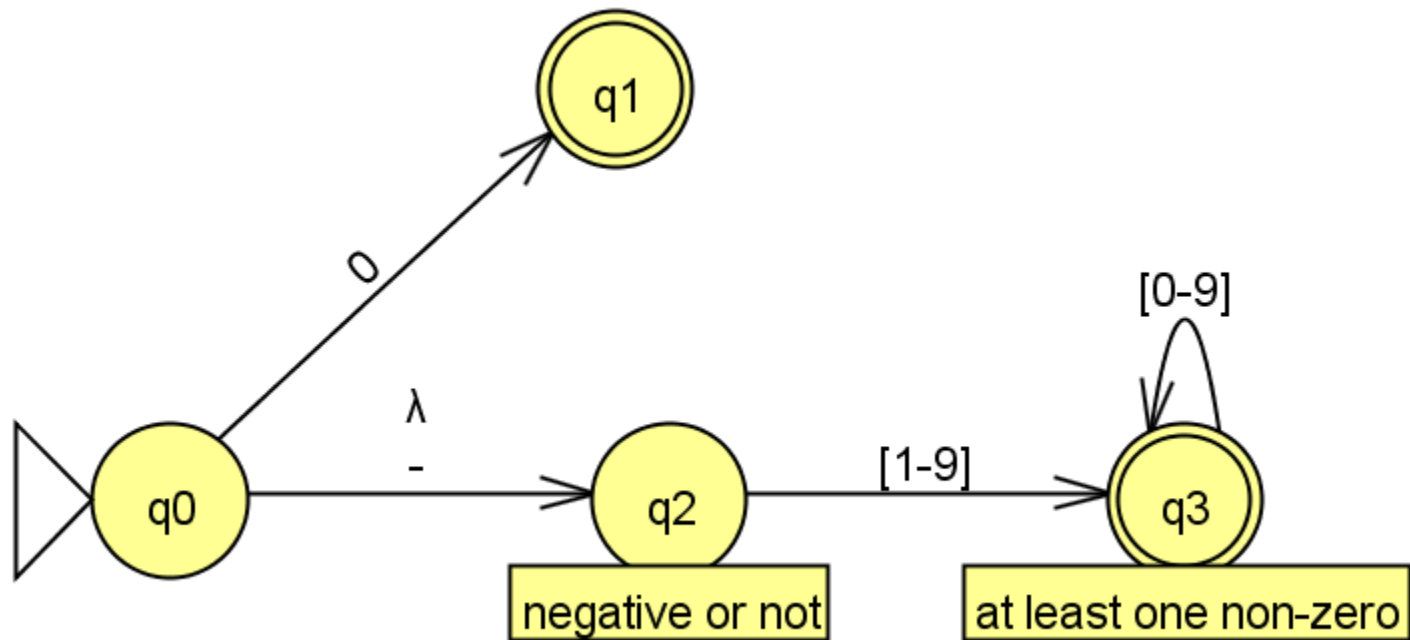
Example: Build an NFA for valid integers

- Example:
 - Valid integers $\{-3, 8, 0, 456, 13, 500, \dots\}$
 - Not valid: $\{006, 3-6, 4.5, \dots\}$

NFA for all valid integers

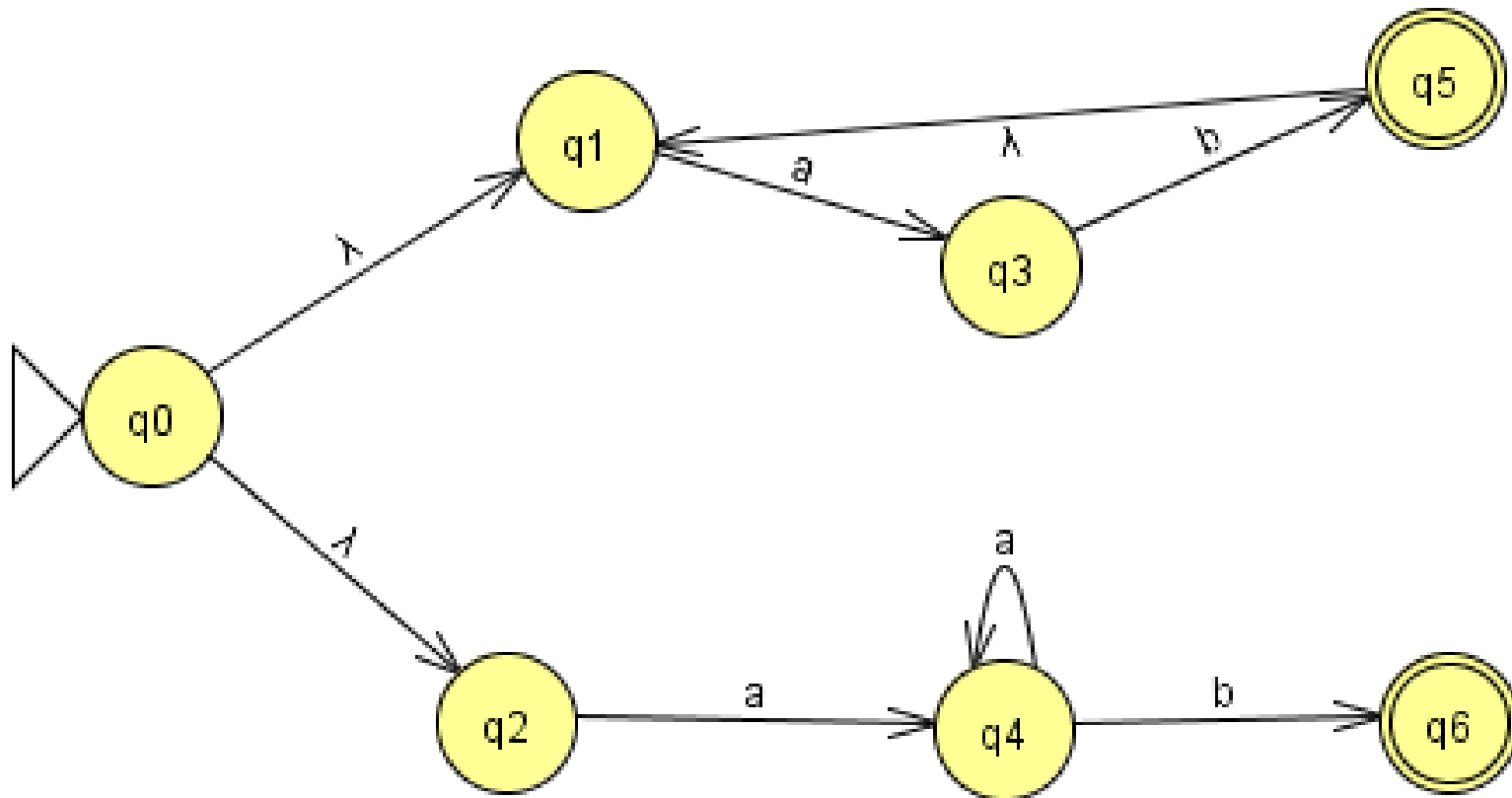


DFA annotated and w/shortcut

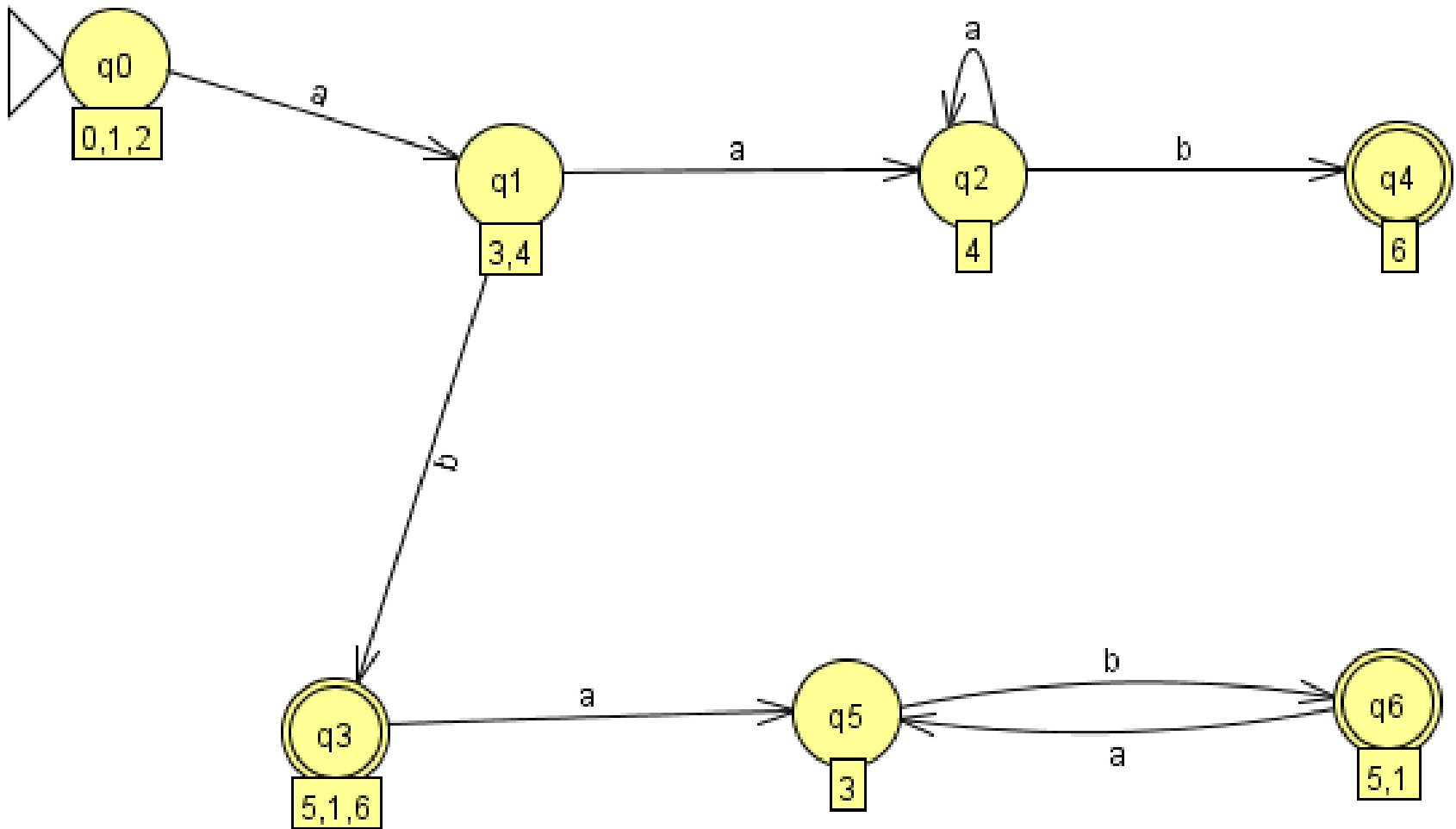


Example: NFA

run and convert to DFA



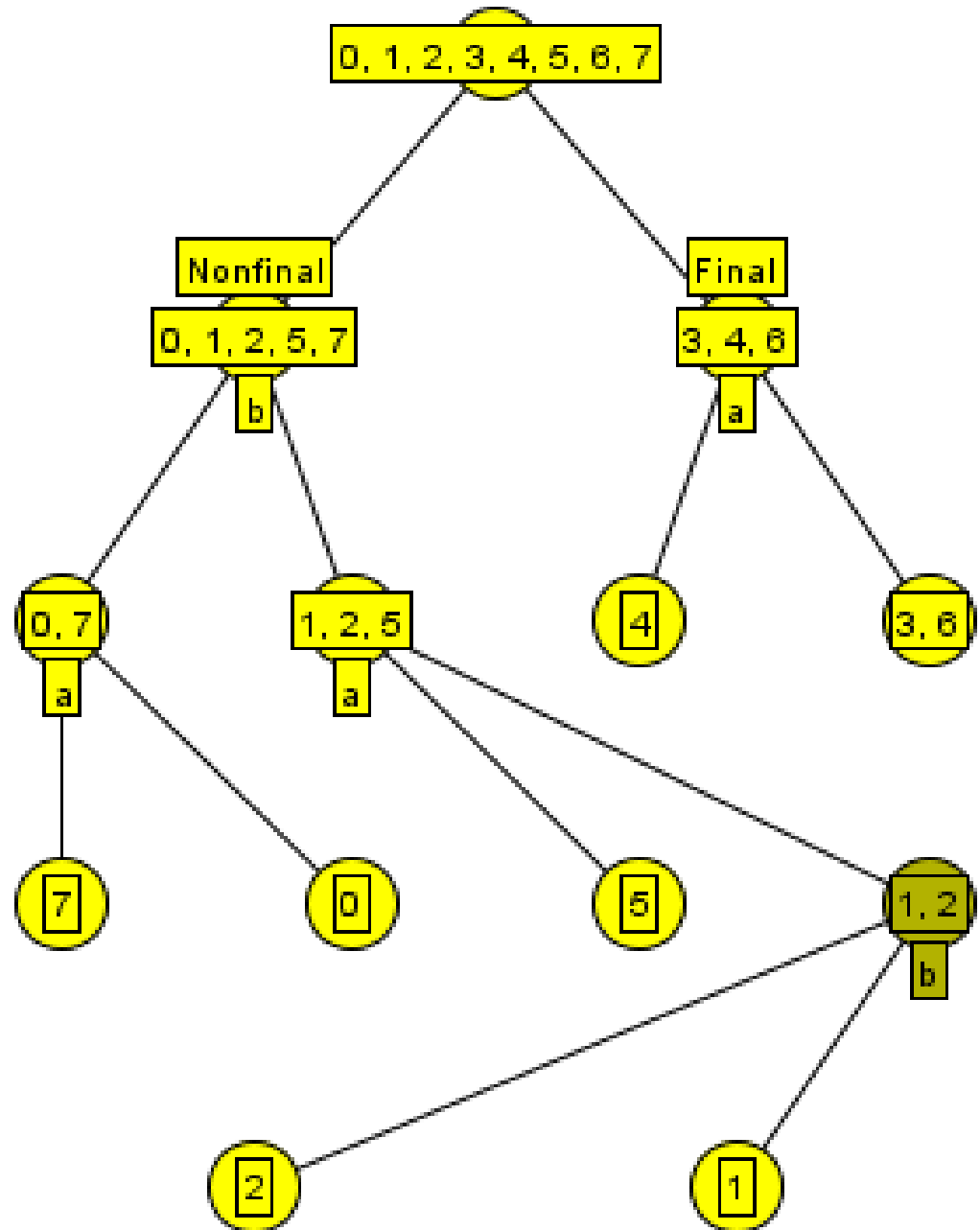
Corresponding DFA



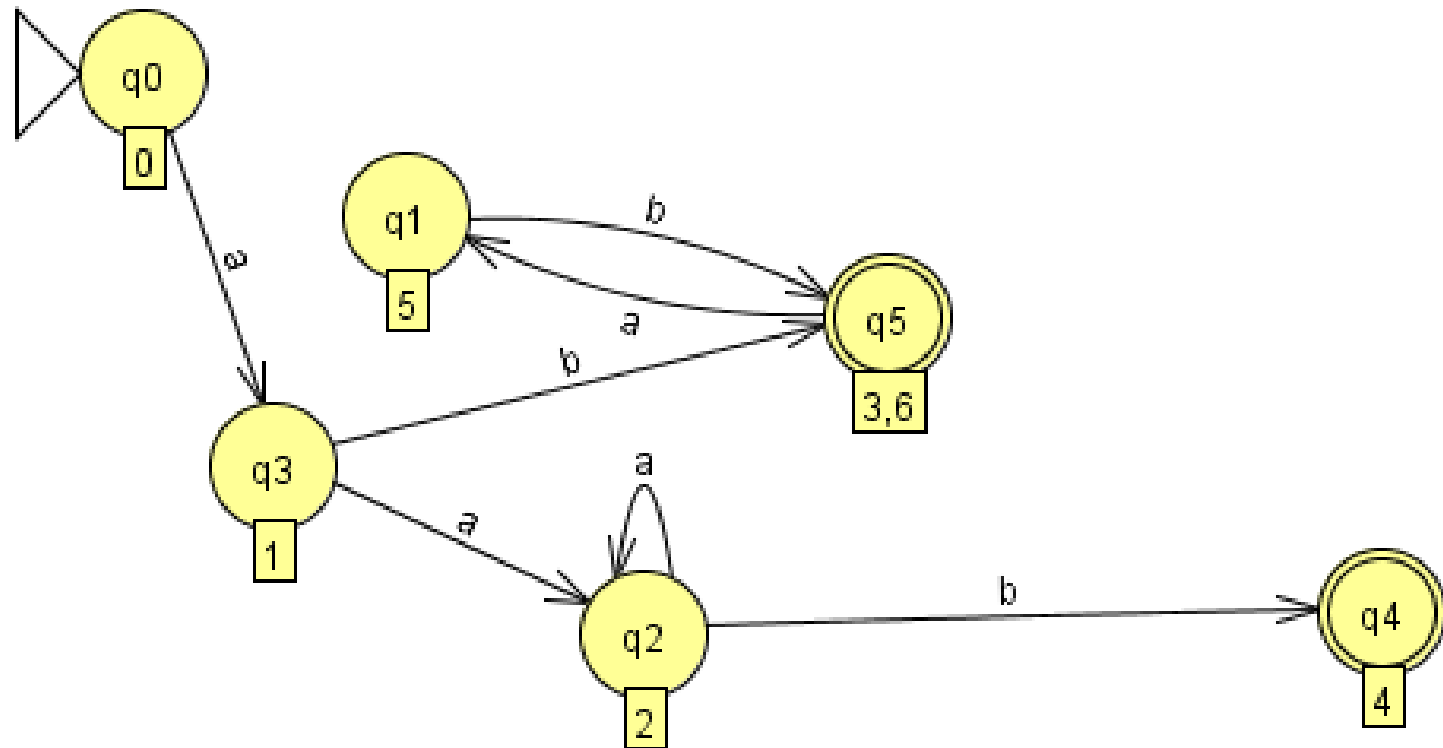
Minimize DFA

First add trap
state q7

then build
tree of
distinguished
states



Final Minimal State DFA



What next?

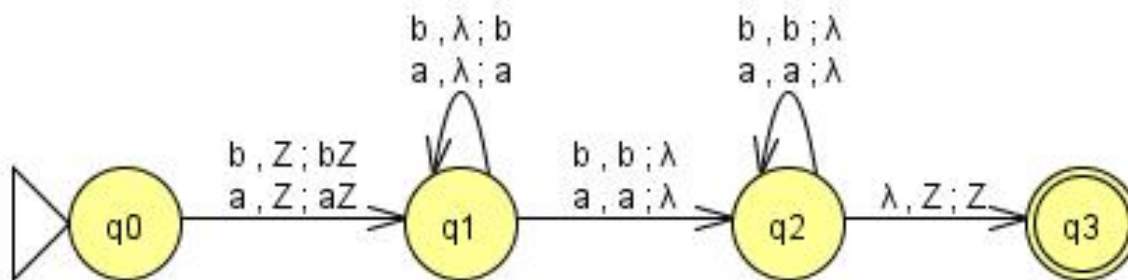
- Can convert to a regular expression ...
- Can convert to an NFA...

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.

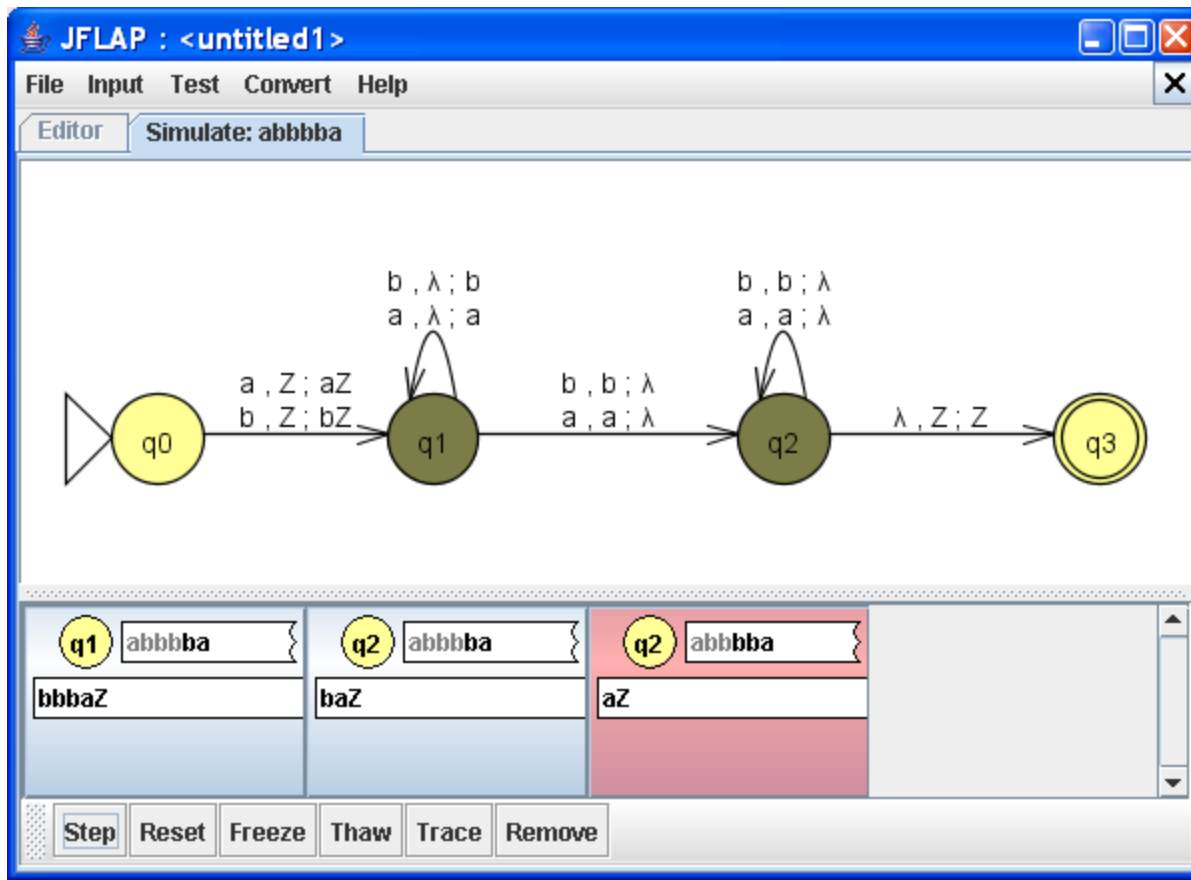
Example : JFLAP during Lecture

- 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 : 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!

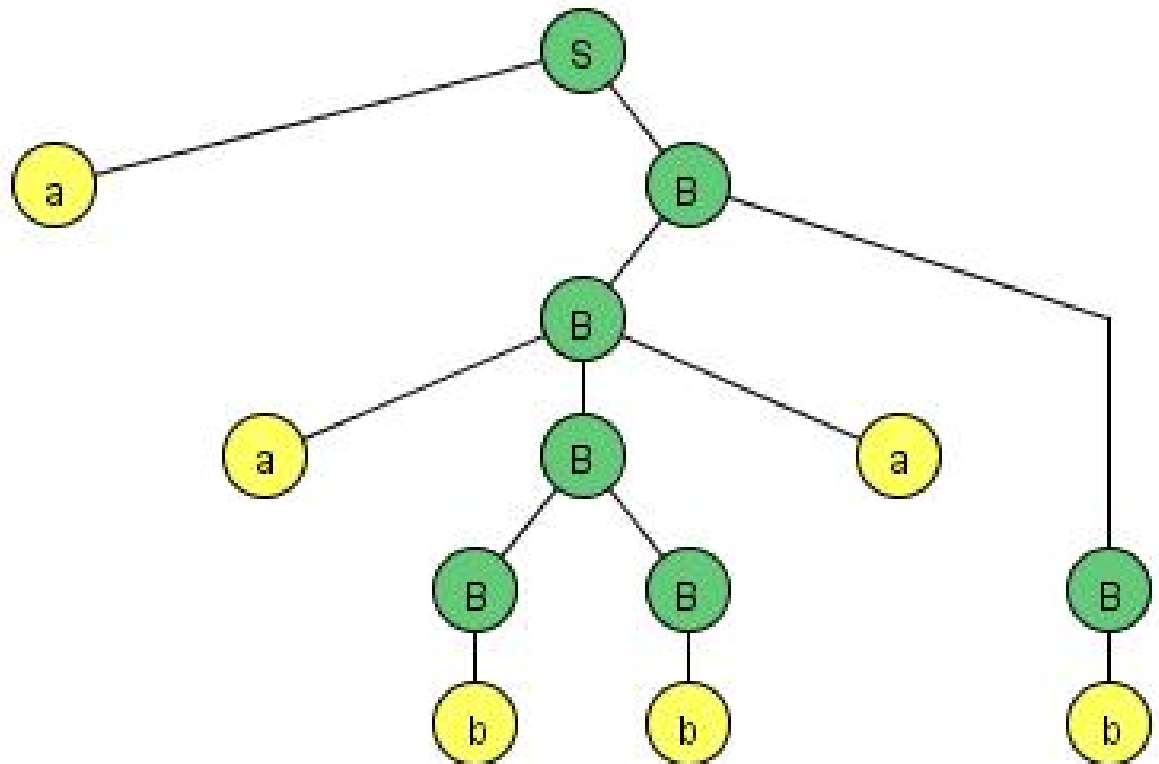
S	→	aB
B	→	BB
B	→	aBa
B	→	b
B	→	λ

S	→	aB
B	→	BB
B	→	aBa
B	→	b
S	→	a
B	→	B
B	→	aa

Example 2 (cont)

Parse Tree Results

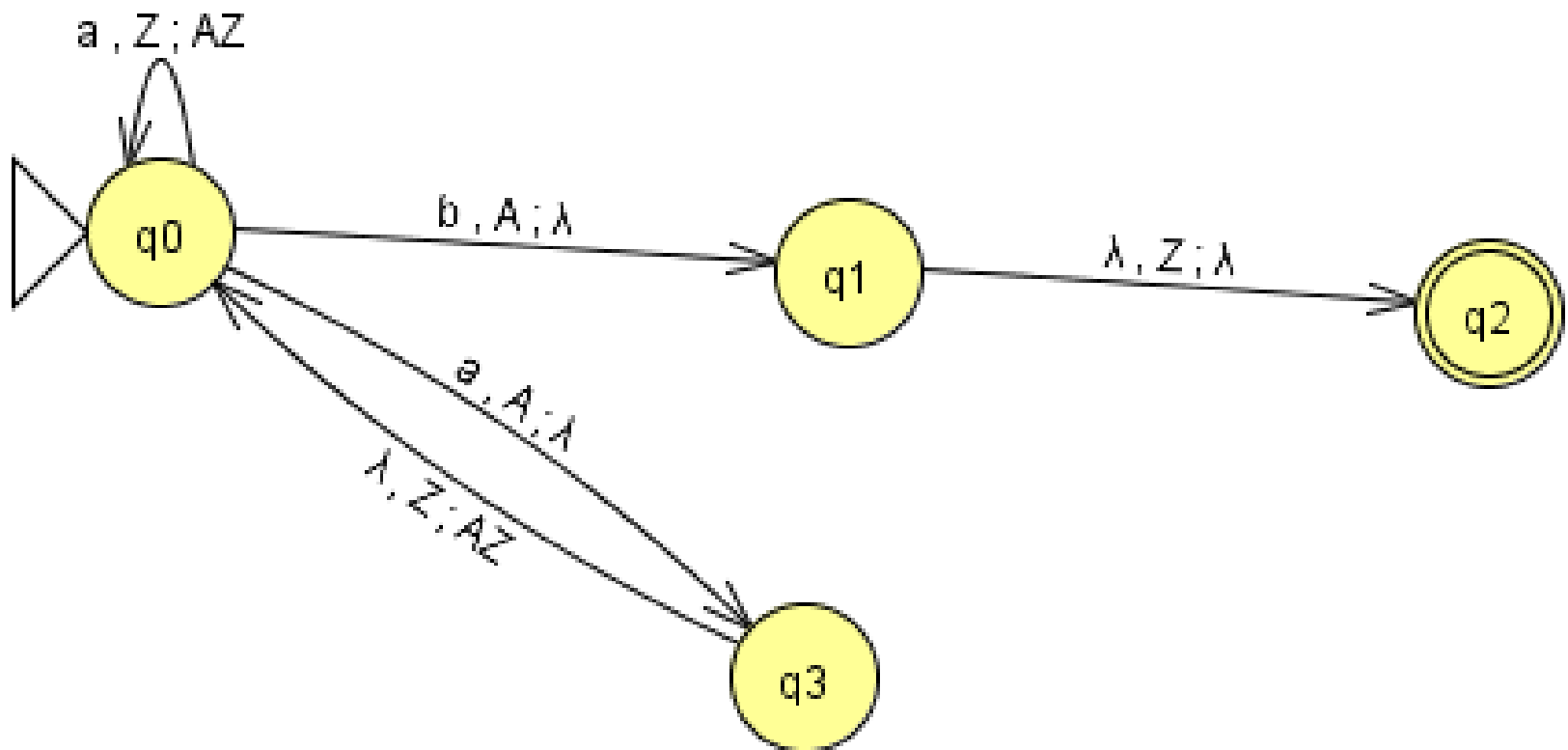
- First Grammar – 1863 nodes generated
- Second Grammar – 40 nodes generated
- Parse tree is the same.



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)

NPDA to CFG



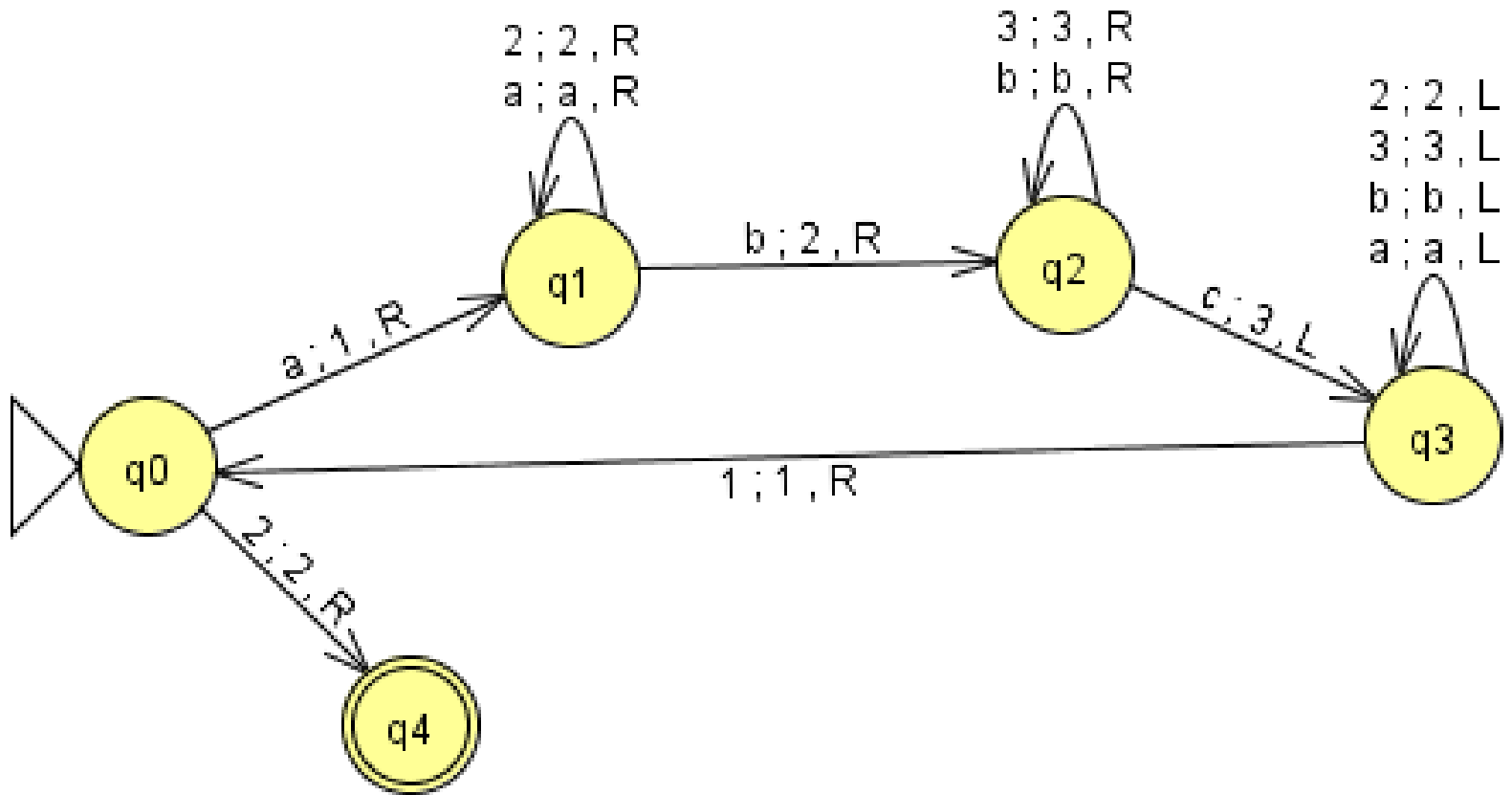
JFLAP's use Outside of Class

- 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

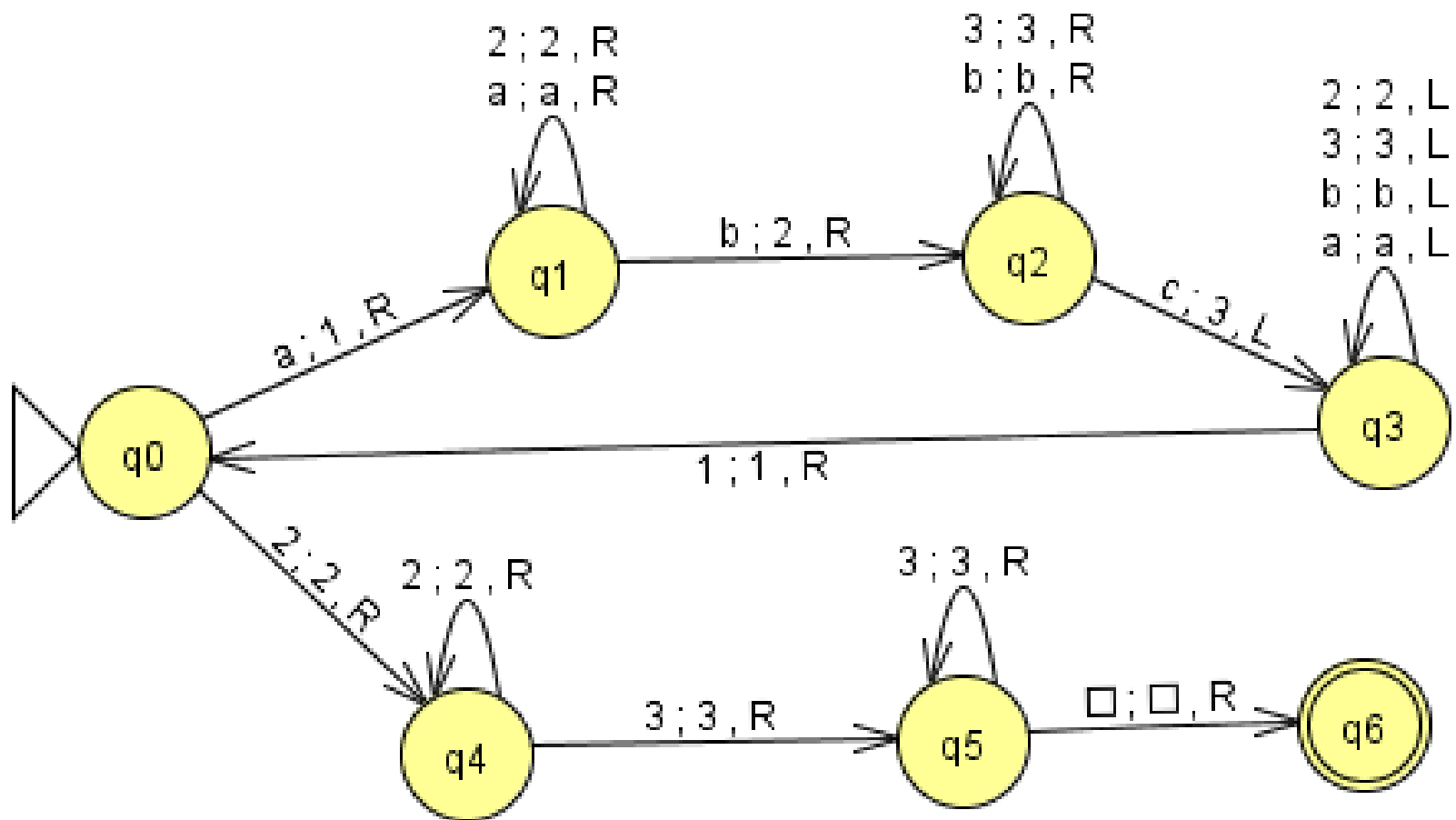
Ordering of Problems in Homework

- 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.

Is this a TM for $anbncn$?



Here is the correct TM for $anbn^n$



Why study finite automata?

- Application: Compiler
- Compiler identifies your syntax errors
- Can write a big DFA to identify all words in a Java program
 - integers, doubles, boolean
 - keywords, variable names
 - arithmetic operators, punctuation symbols
- Example LR Parser...

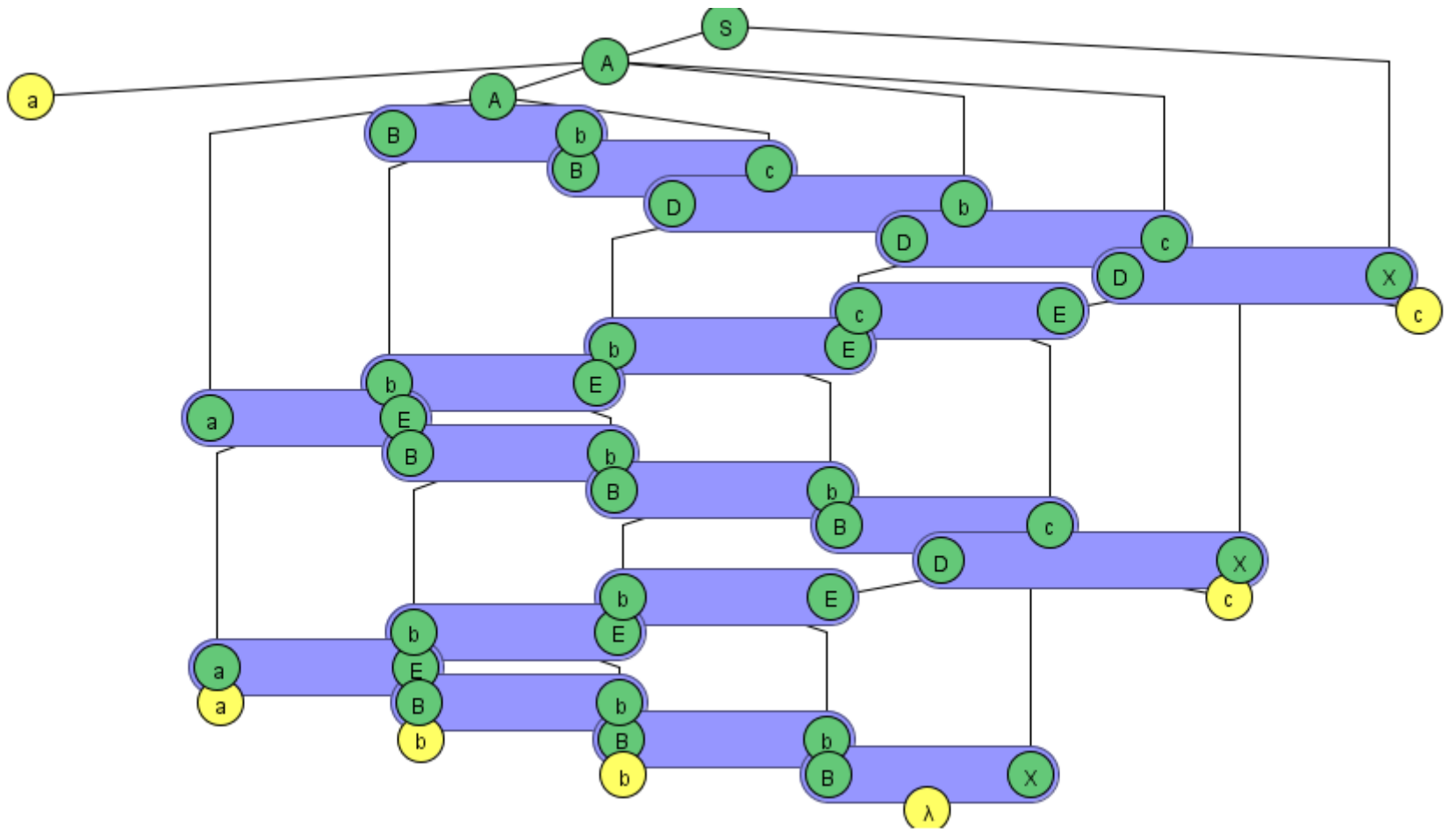
Lsystems

- Another type of grammar
 - Show a simple L-System
 - Show a tree
 - Show a fractal

Unrestricted Grammar - $anbncn$

S	\rightarrow	AX
A	\rightarrow	$aAbc$
A	\rightarrow	$aBbc$
Bb	\rightarrow	bB
Bc	\rightarrow	D
Dc	\rightarrow	cD
Db	\rightarrow	bD
DX	\rightarrow	EXc
BX	\rightarrow	λ
cE	\rightarrow	Ec
bE	\rightarrow	Eb
aE	\rightarrow	aB

Trace aabbcc

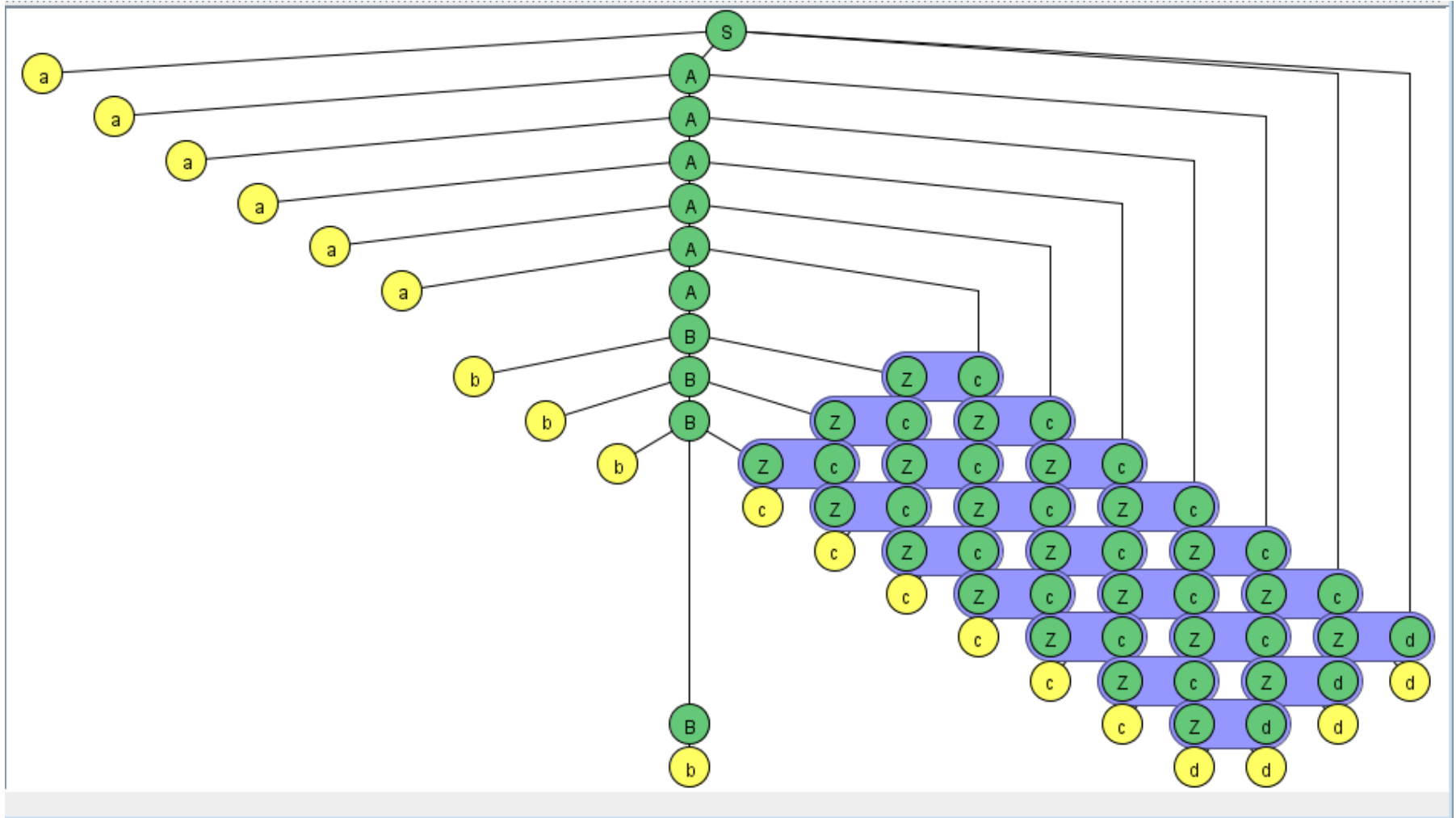


Example - Unrestricted Grammar

$anbmcndm$

S	→	aAcd
A	→	aAc
A	→	B
B	→	bBZ
B	→	b
Zc	→	cZ
Zd	→	dd

Example – Unrestricted Grammar (cont)

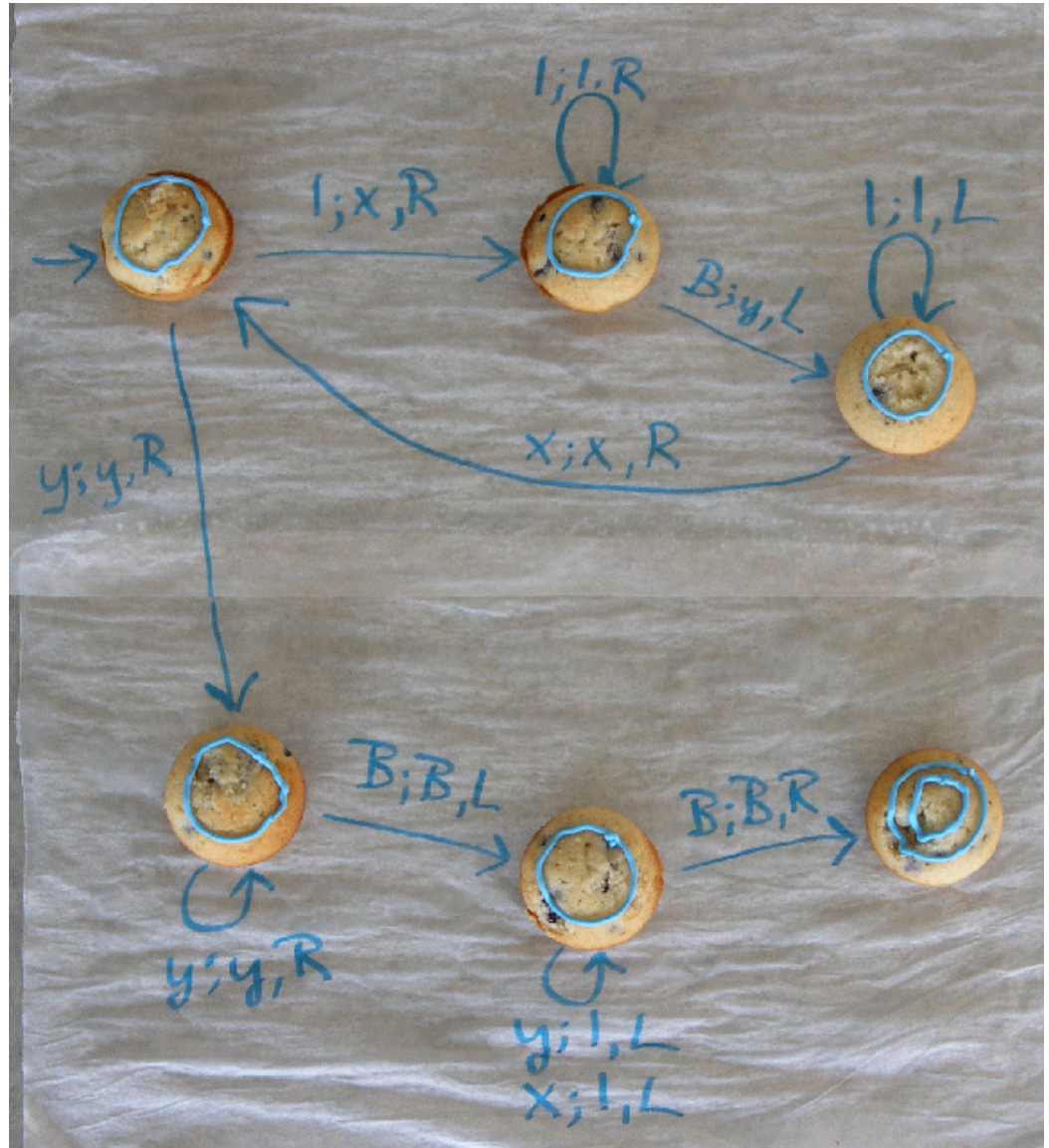


There are other ways to get
interaction in this course besides
software...

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

