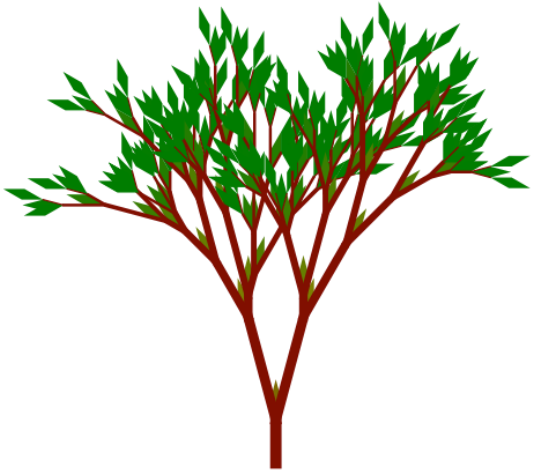
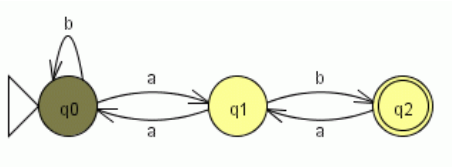
Through Visualization and Interaction, Computer Science Concepts Come Alive



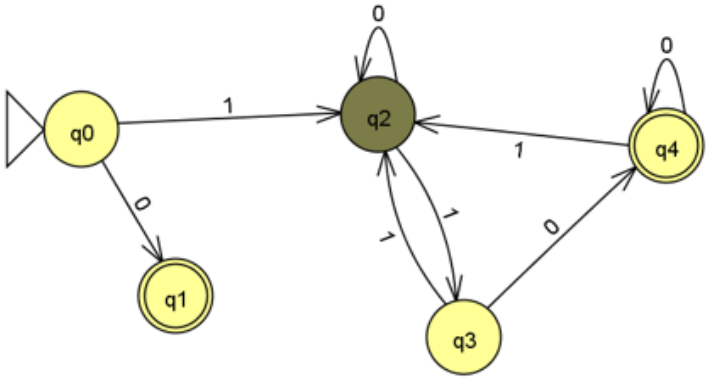
Susan H. Rodger
Duke University
May 11, 2016
Evergreen State College



Supported by NSF Grants DRL-1031351, DUE 1044191, CRA-W, and IBM Faculty Awards

About Me

- Professor of the Practice of Computer Science
- Area: Visualization and Animation, Computer Science Education



- Passionate about education/diversity
  - SIGCSE Chair
  - ACM Education Policy Committee
  - CRA-W Board Member

SIG CSE

CRA-W
Computing Research Association
Women

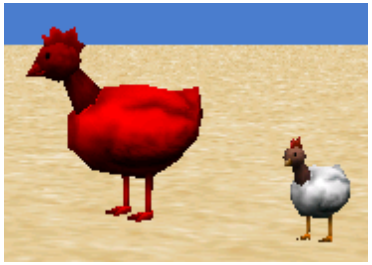
About Me - Hobby – Baking Shape cakes

How do you make those cakes?

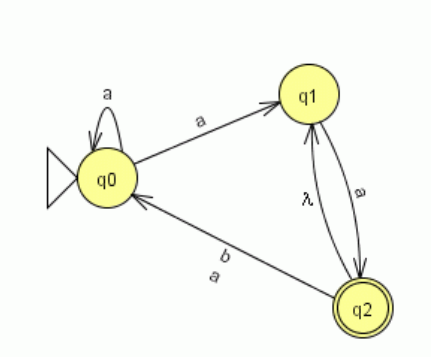
CS 1 Sorting Cookies

Graduate School

- PhD Purdue University 1989
  - Computational Geometry
  - Parallel Scheduling Algorithms



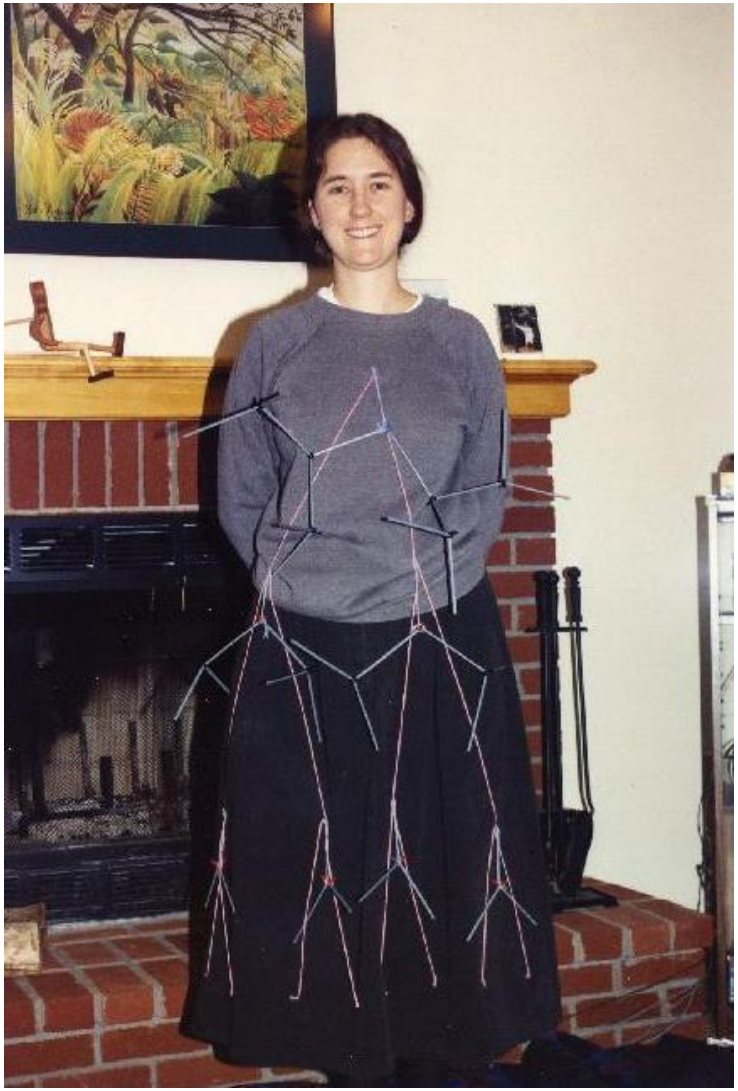
Outline



- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Additional Ways to Engage with CS
- Diversity Efforts

CS Concepts Coming Alive

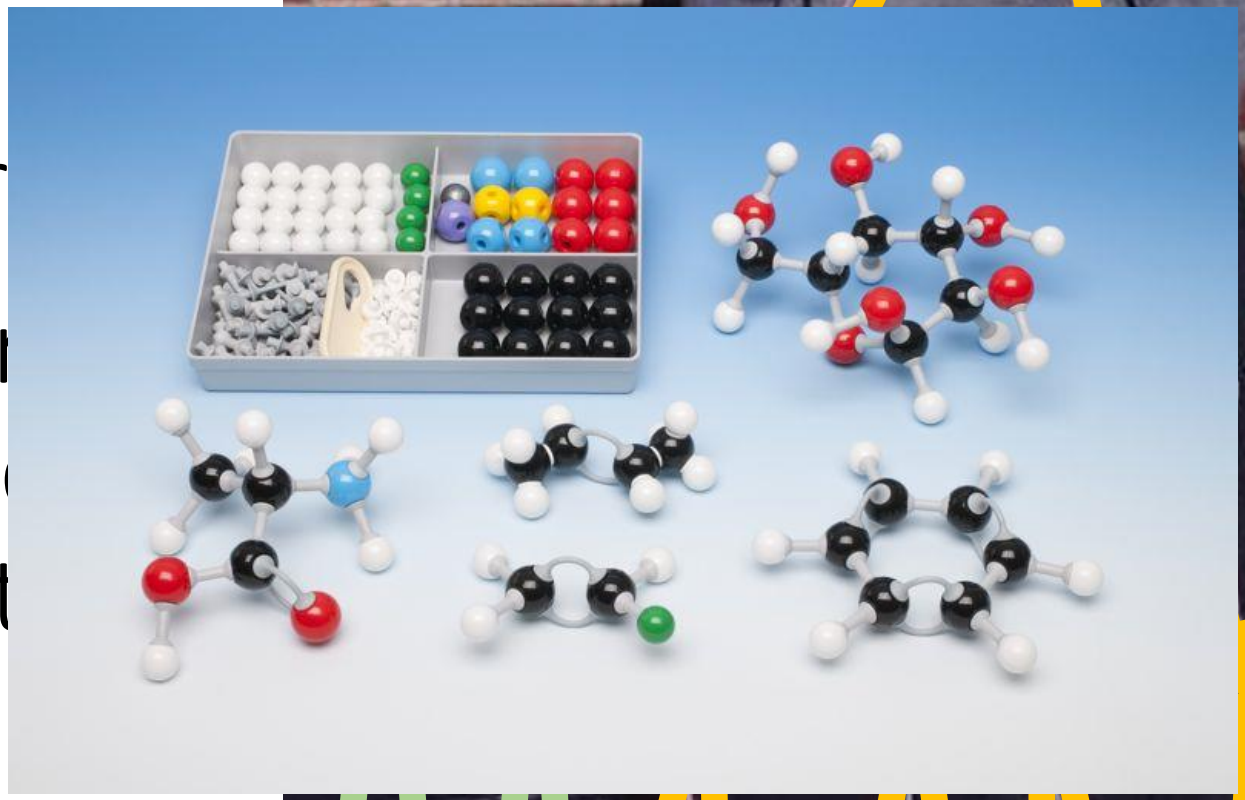
- What data structure is this?



YARN,
in the
shap
binar

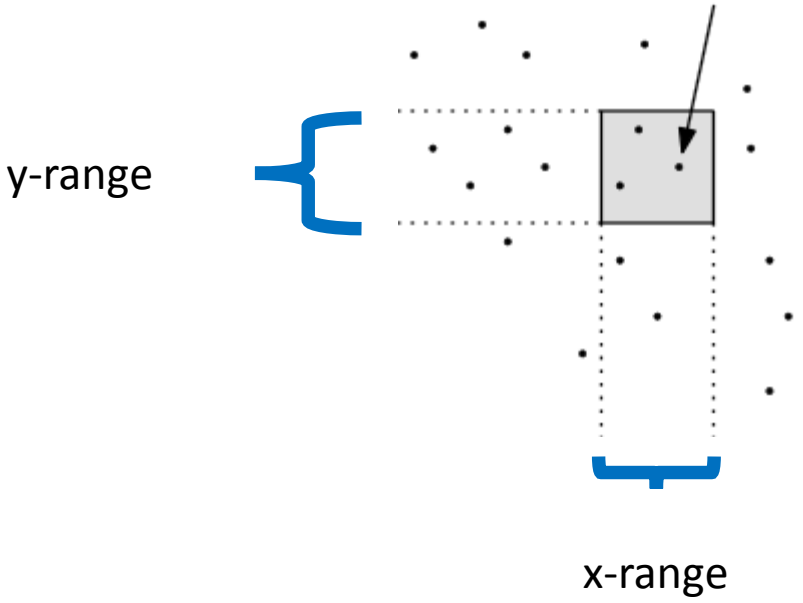
Subtr
ma
wit
mole
kit

What is it?

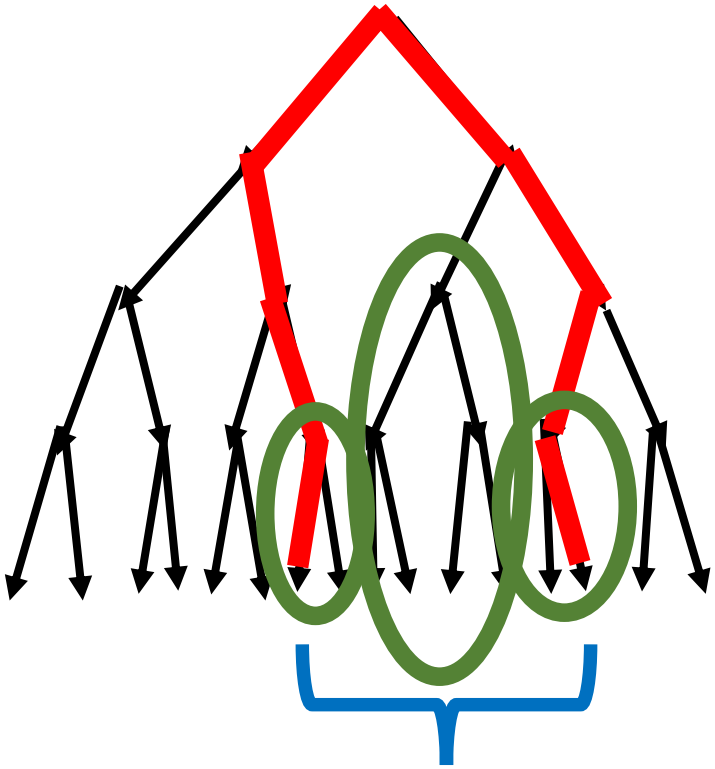


2D-range tree

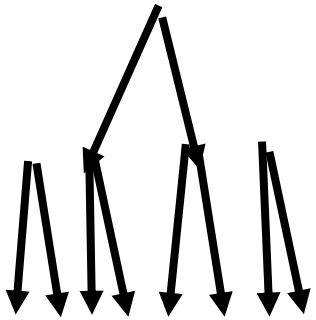
- Search in x-y plane
- Main tree organized by x-values
- Subtree organized by y values



Binary Search tree of points in the plane – sorted by X-value



In the x-range



Each subtree organized by y-value

Search each subtree by y-value

Different Types of Learners

- Learning Styles
  - Visual Learners
    - Learn through seeing
    - Learn best from visual displays
  - Auditory Learners
    - Learn through listening
    - Learn best through verbal lectures, discussions
  - Kinesthetic Learners
    - Learn through moving, doing and touching
    - Learn best through hands-on approach

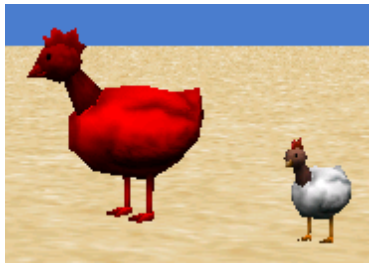
How do you reach all three types?

- You must do all three!
  - Provide pictures, diagrams
  - Discuss what you are doing
  - Provide activities for trying it

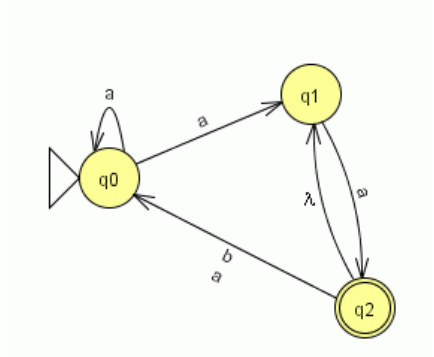
Learner Engagement Taxonomy with visualization software

- Different forms of Learner engagement
  - No Viewing
  - Viewing
  - Responding
  - Changing
  - Constructing
  - Presenting

- ITiCSE Working Group Report 2002 (Naps et al.)



Outline



- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Additional Ways to Engage with CS
- Diversity Efforts

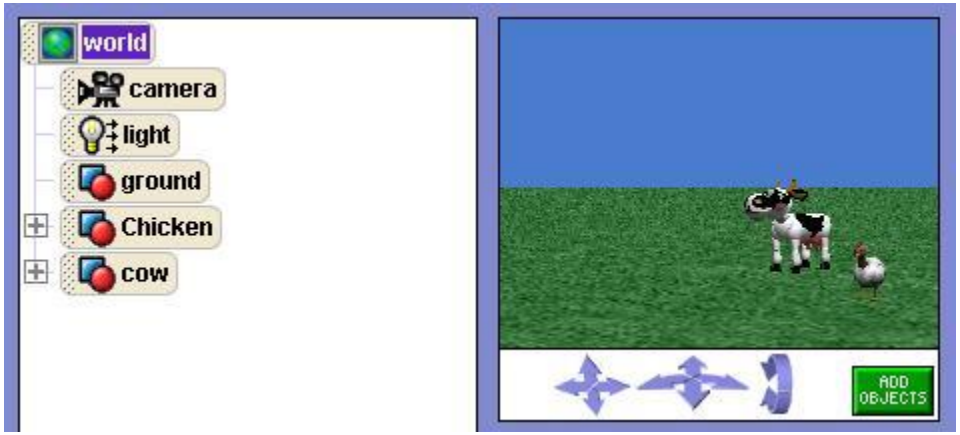
Alice Programming Language

- Create interactive stories or games
- Learn programming in an easy way, drag-and-drop your code
- Problem solving with visual feedback
  - Objects are visual!
- Alice is free: www.alice.org
- Developed by Randy Pausch

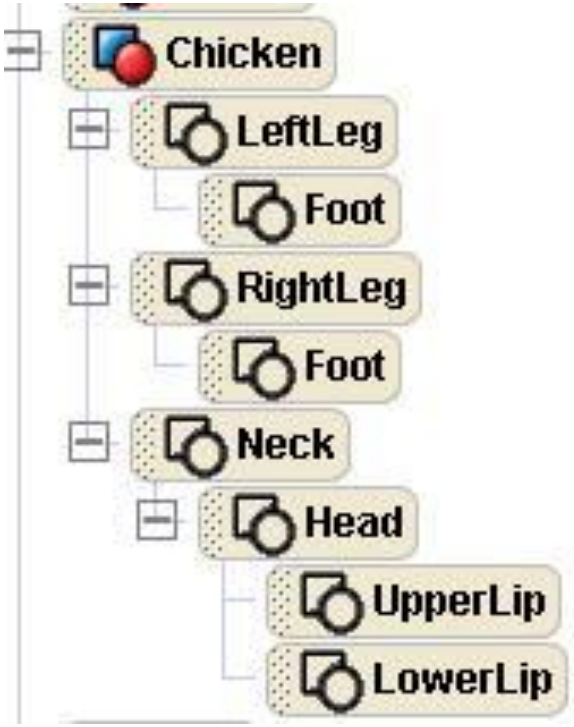
More on … Alice Programming Language

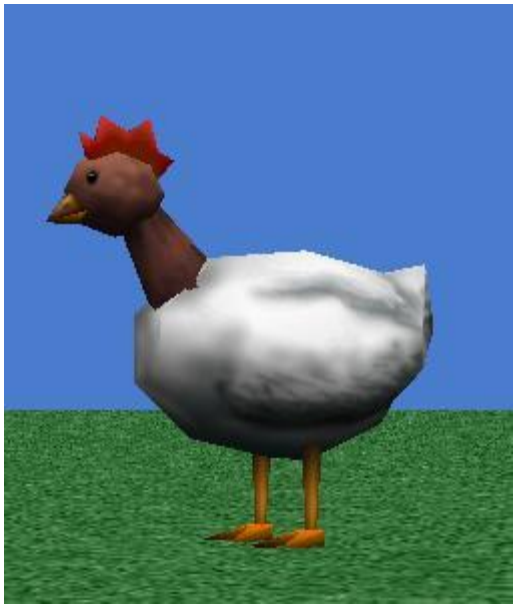
- Has libraries of 3D objects

- Keeps Track of objects you select



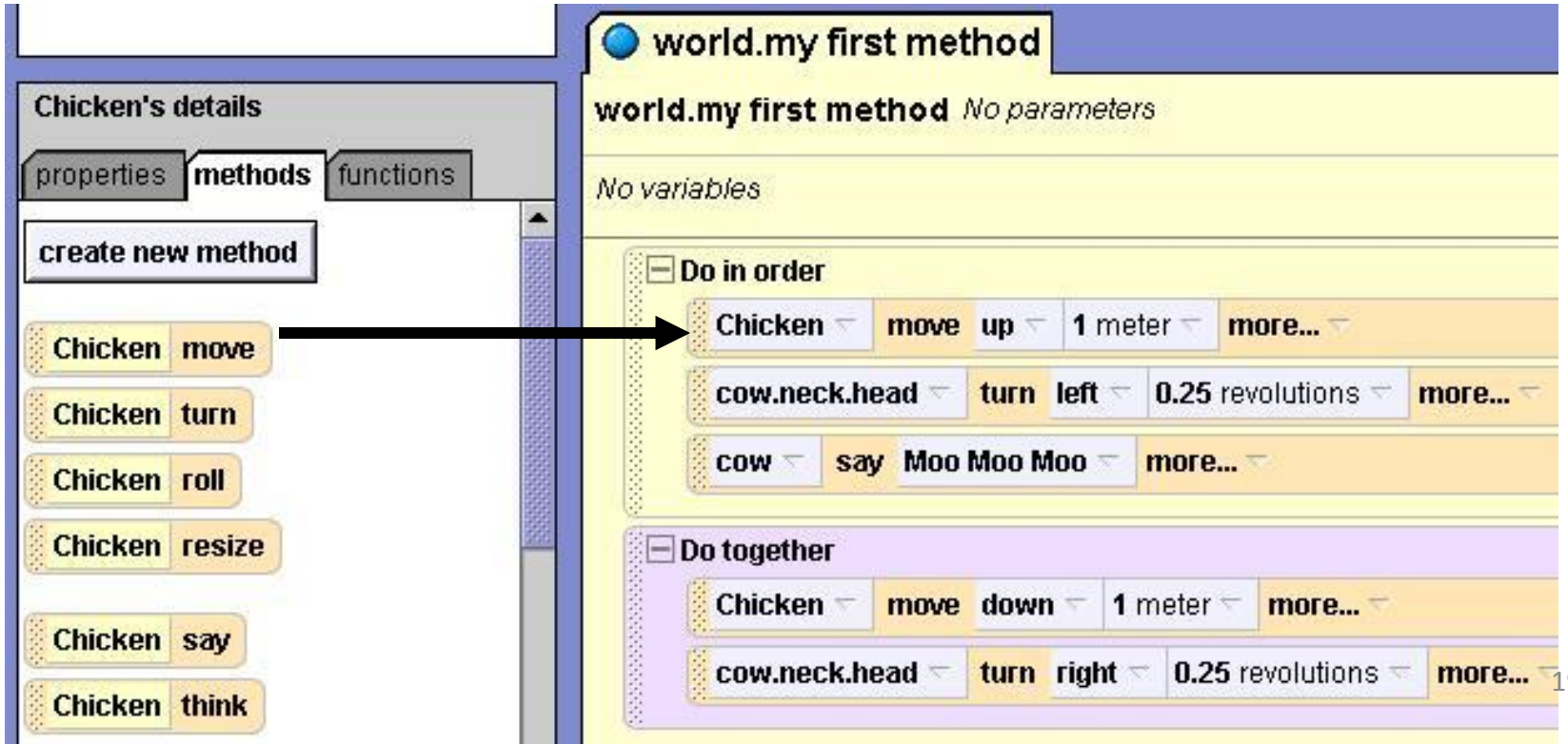
Objects Have Multiple Parts that are moveable





Alice Code is Easy to Learn

Select Code, Drag-and-Drop code in program



Play Alice Animation

- Chicken rises, cow turns head and talks

Why Alice?

- Lots of other great tools for teaching programming

Greenfoot

SCRATCH

LEGO

Snap!

- Alice is easy to use, drag-and-drop, objects already exist
- Storytelling - Attractive to both girls and boys

Success - Alice attracts diverse group

- At Duke
  - CompSci 4 Spring 2005
    - 22 preregister, 30 enroll (12 female + 3 African Amer.)
  - CompSci 4 Fall 2005
    - 20 preregister, 31 enroll (17 female + 1 African Amer.)
  - CompSci 4 Fall 2006 – 2 sections
    - 64 students, 33 female, 7 African Amer.
  - CompSci 4 Fall 2007 – 2 sections
    - 84 students - > 50% female
  - CompSci 4 Fall 2008 – 2 sections
    - 100 students - > 50% female
  - Same for Spring 2009, Fall 2009…
  - Advertised in school paper
    - picture of ice skater
    - Web site of animations
  - This course is now CompSci 94

Computer Science Concepts come alive with Alice - Examples

- Objects are visible
- Variables
- Inheritance
- Lists
- Array

Example: Objects are visible

Getting Started Tutorial teaches

- Placing objects
- Moving objects
- Setting up Camera tripods and moving between views
- Using built in methods and writing your own
  - Dragon flapWings
- Gluing objects together
- Adding sound, 2D pictures to enhance world

Getting Started Tutorial – 3 part

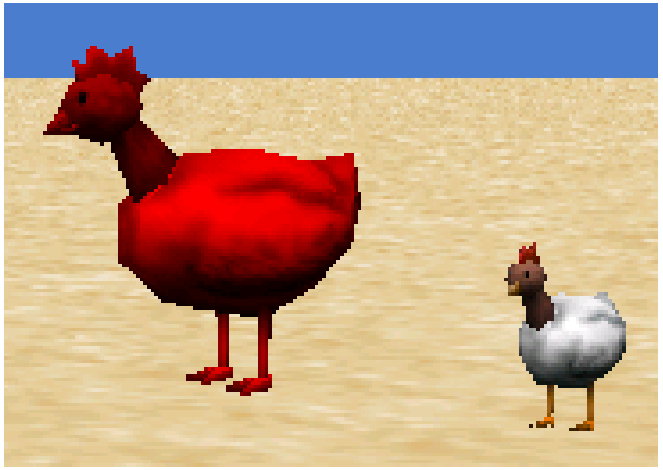
Variables – Timer and Score

Variables – Scores/Timers Game: Eragon

4 tasks to win the game

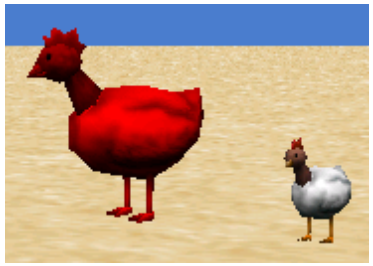
Example - Inheritance

- Start with a chicken object
- Rename it to TalentedChicken
  - Change its color
  - Resize it larger
  - Add new methods (jump, fly, scurry)
  - Add events for this chicken
- Save this new class TalentedChicken that inherits from the Chicken class

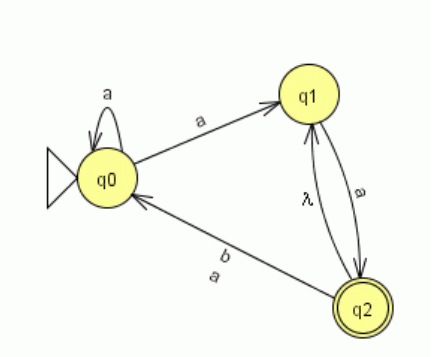


Example list

Example – Arrays
Shuffle, then Selection Sort



Outline



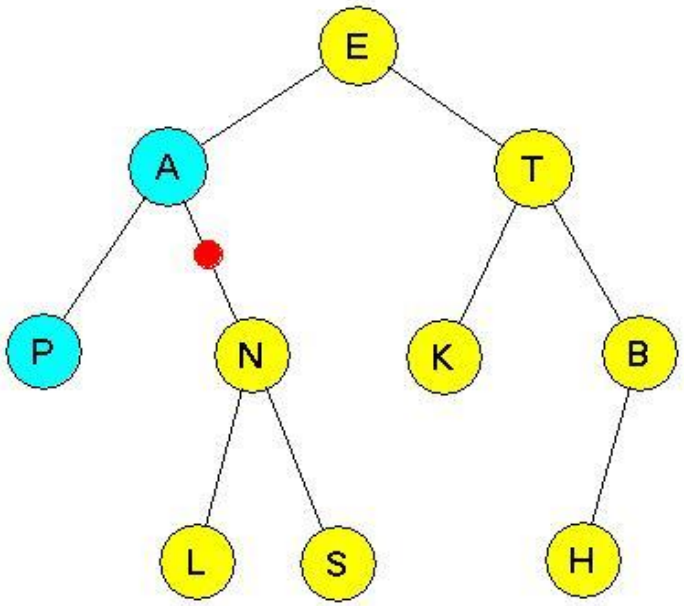
- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Additional Ways to Engage with CS
- Diversity Efforts

Algorithm Visualization/Animation Software/Aps/Videos

- Tango, Xtango, Samba, JSamba - Stasko (Georgia Tech)
- AnimalScript – Roessling (Darmstadt Univ of Tech, SIGCSE 2001)
- JHAVE – Naps (U. Wisc. Oshkosh, SIGCSE 2000)
- TRAKLA2 – Software Visualization Group – TKK Finland
- JAWAA – Rodger et al (Duke, SIGCSE 2003)
- Lots of animations and systems on the web!
- Lots of videos of algorithm animations on the web!

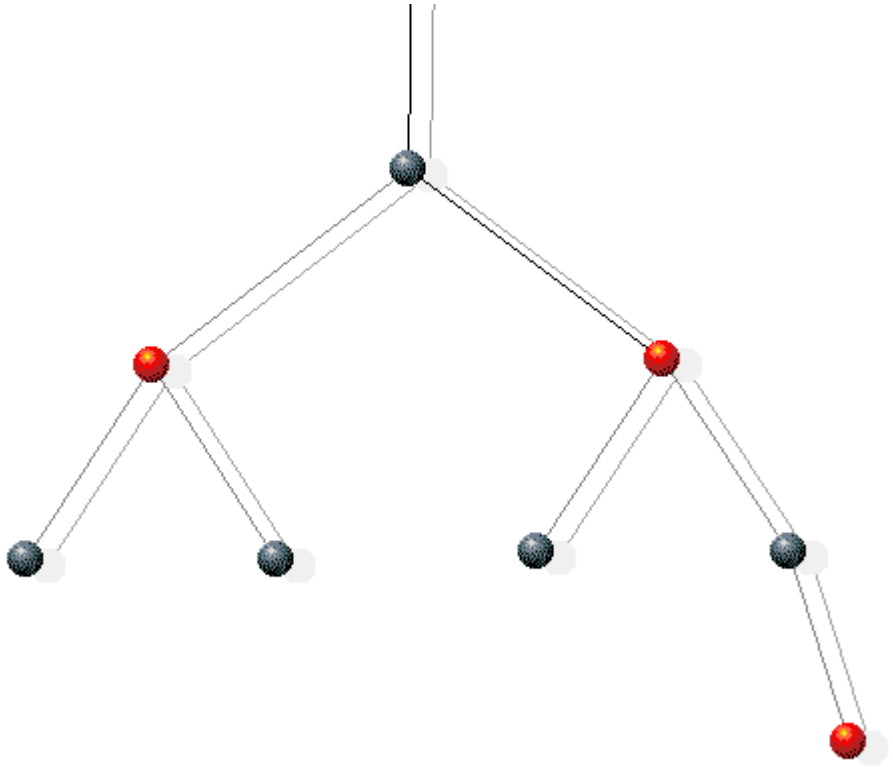
Use of Algorithm Animation in CS 1/2

- Instructor
  - Make/Use animations for lecture
  - Stop/Pause – ask what will happen next
  - must be interactive
- Student
  - Create animations
  - Replay animations from lecture with same or new inputs



Lots of other software/programs for algorithm animation

- Red Black Tree – animation on web page

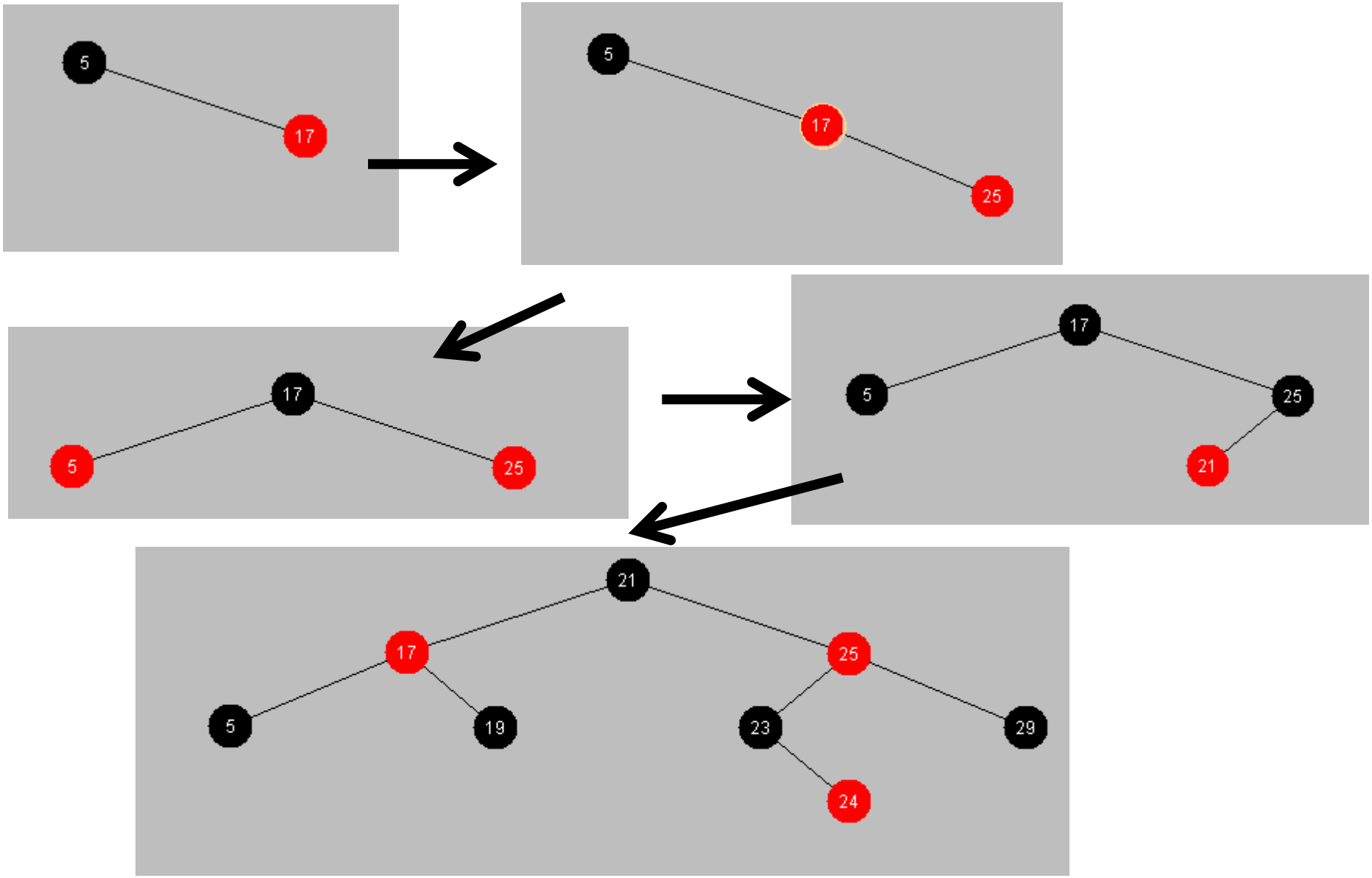


Student must have graduated. Link no longer works!

http://aleph0.clarku.edu/~achou/cs102/examples/bst_animation/RedBlackTree-Example.html

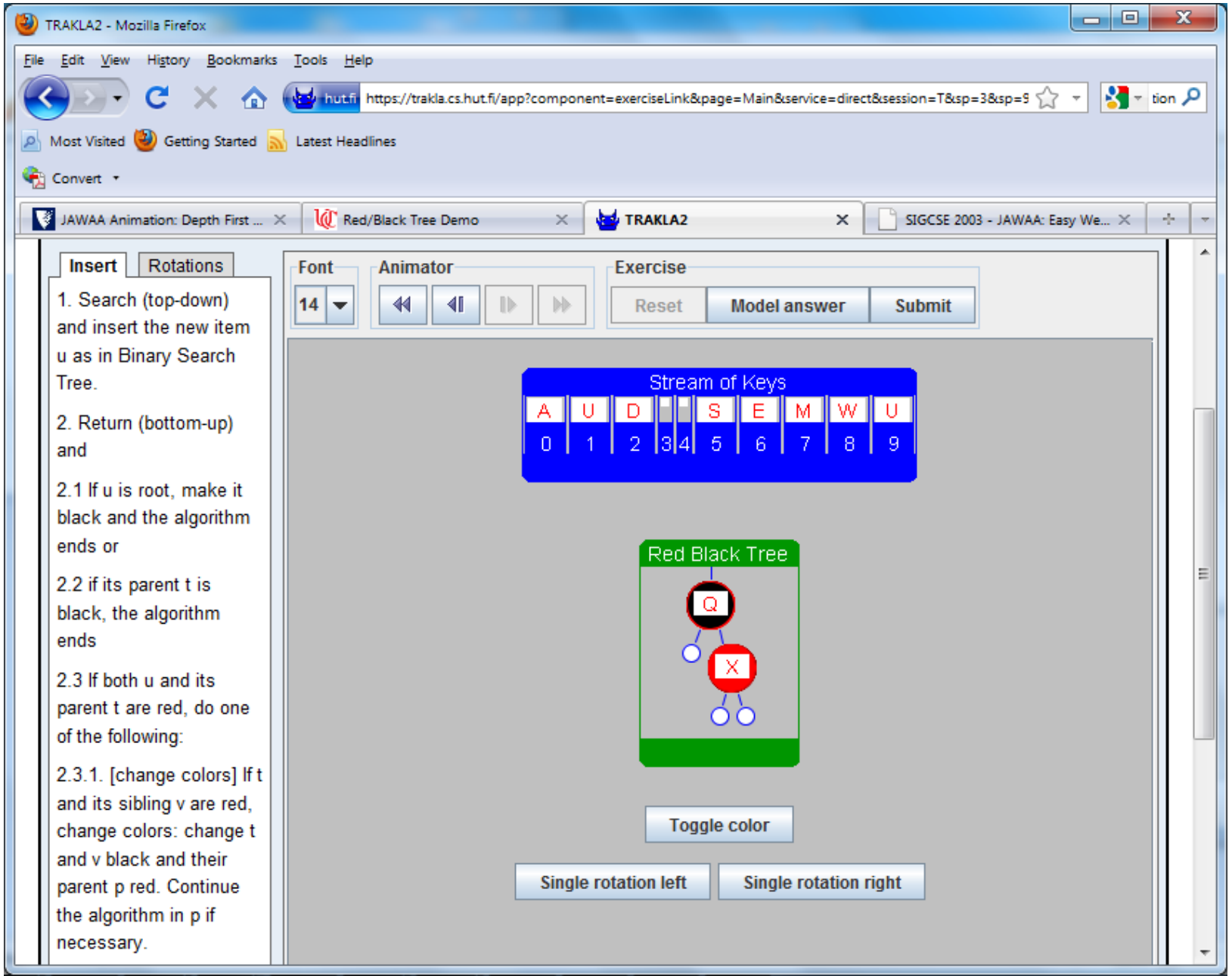
X

Another red-black tree animation



http://www.ece.uc.edu/~franco/C321/html/RedBlack/redblack.html

TRAKLA2 http://www.cse.hut.fi/en/research/SVG/TRAKLA2/video/



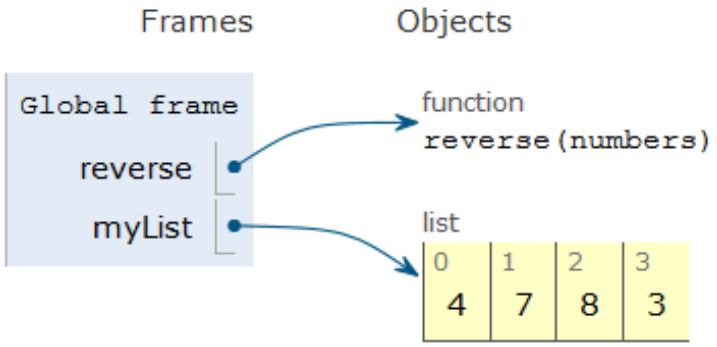
Python Tutor
Compute reverse of a list

Python 2.7

```
1  def reverse(numbers):
2      answer = []
3      for num in numbers:
4          answer.insert(0, num)
5      return answer
6
7  myList = [4, 7, 8, 3]
8  reversed = reverse(myList)
```

Edit code

<< First | < Back | Step 3 of 16 | Forward > | Last >>



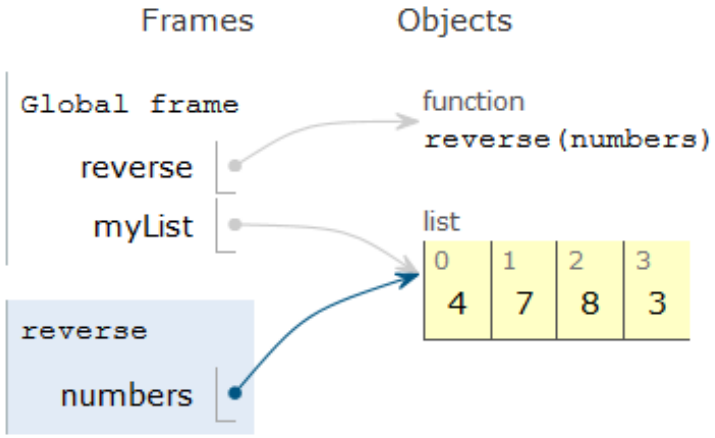
Python Tutor
Compute reverse of a list

Python 2.7

```
1  def reverse(numbers):
2      answer = []
3      for num in numbers:
4          answer.insert(0, num)
5      return answer
6
7  myList = [4, 7, 8, 3]
8  reversed = reverse(myList)
```

Edit code

<< First | < Back | Step 4 of 16 | Forward > | Last >>



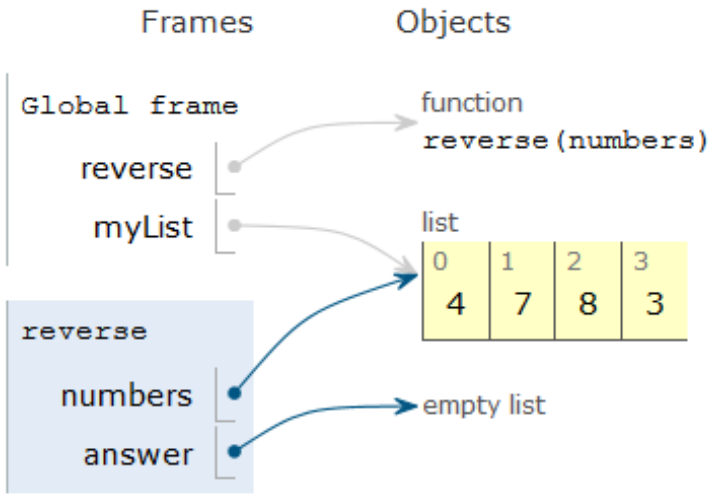
Python Tutor
Compute reverse of a list

Python 2.7

```
1  def reverse(numbers):
2      answer = []
3      for num in numbers:
4          answer.insert(0, num)
5      return answer
6
7  myList = [4, 7, 8, 3]
8  reversed = reverse(myList)
```

Edit code

<< First | < Back | Step 6 of 16 | Forward > | Last >>



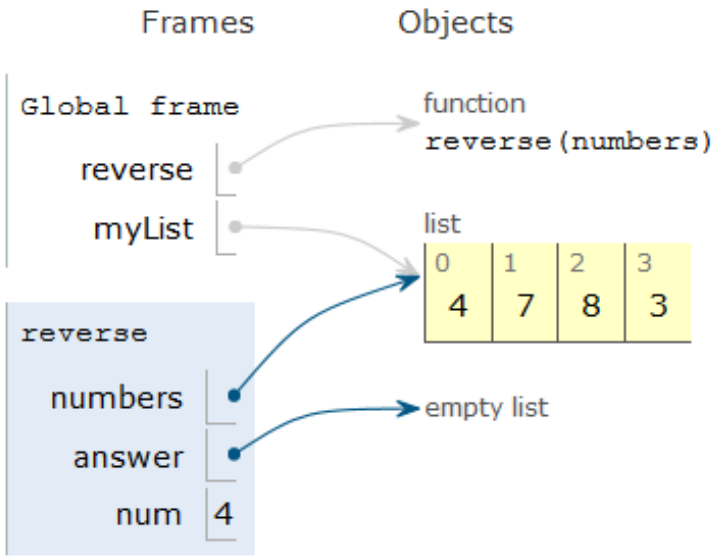
Python Tutor
Compute reverse of a list

Python 2.7

```
1  def reverse(numbers):
2      answer = []
3      for num in numbers:
4          answer.insert(0, num)
5      return answer
6
7  myList = [4, 7, 8, 3]
8  reversed = reverse(myList)
```

Edit code

<< First | < Back | Step 7 of 16 | Forward > | Last >>



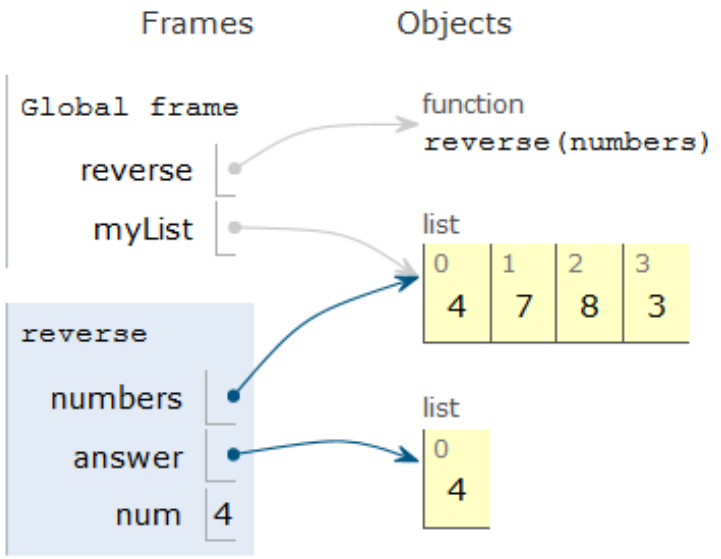
Python Tutor
Compute reverse of a list

Python 2.7

```
1  def reverse(numbers):
2      answer = []
3      for num in numbers:
4          answer.insert(0, num)
5      return answer
6
7  myList = [4, 7, 8, 3]
8  reversed = reverse(myList)
```

Edit code

<< First | < Back | Step 8 of 16 | Forward > | Last >>



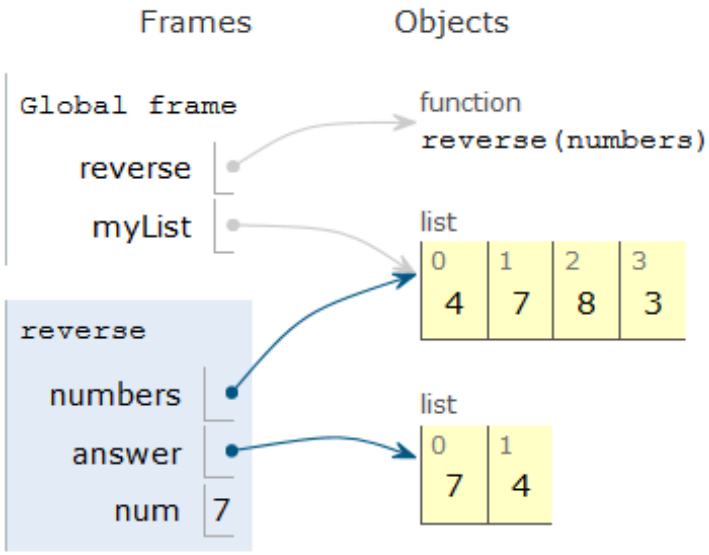
Python Tutor
Compute reverse of a list

Python 2.7

```
1  def reverse(numbers):
2      answer = []
3      for num in numbers:
4          answer.insert(0, num)
5      return answer
6
7  myList = [4, 7, 8, 3]
8  reversed = reverse(myList)
```

Edit code

<< First | < Back | Step 10 of 16 | Forward > | Last >>



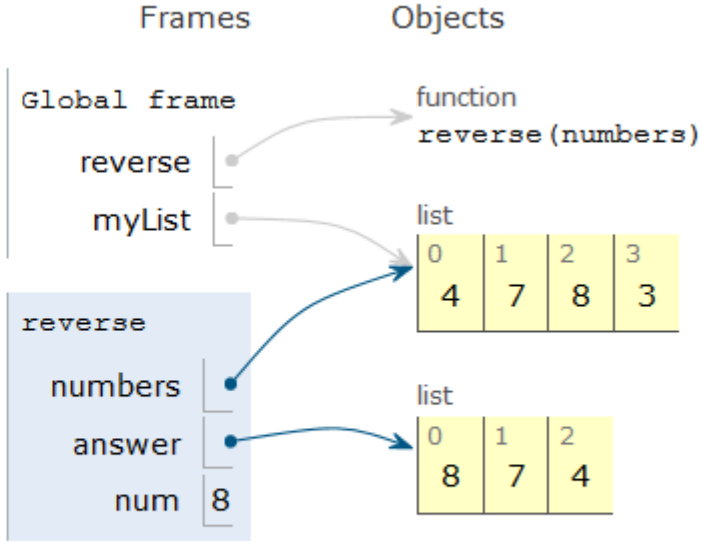
Python Tutor
Compute reverse of a list

Python 2.7

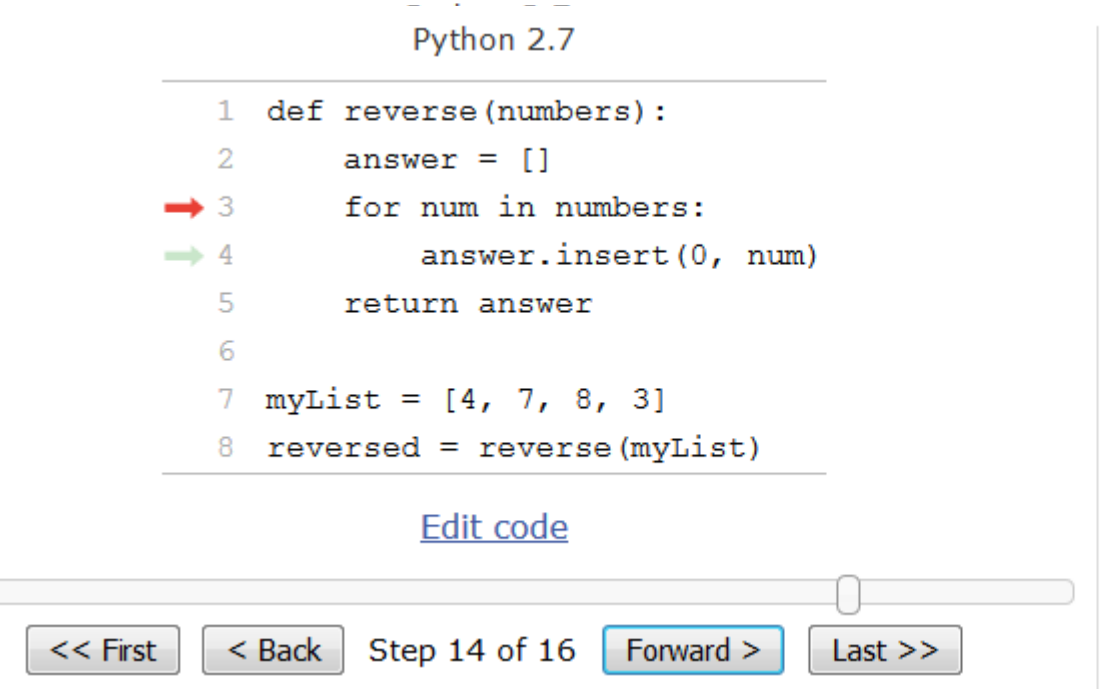
```
1  def reverse(numbers):
2      answer = []
3      for num in numbers:
4          answer.insert(0, num)
5      return answer
6
7  myList = [4, 7, 8, 3]
8  reversed = reverse(myList)
```

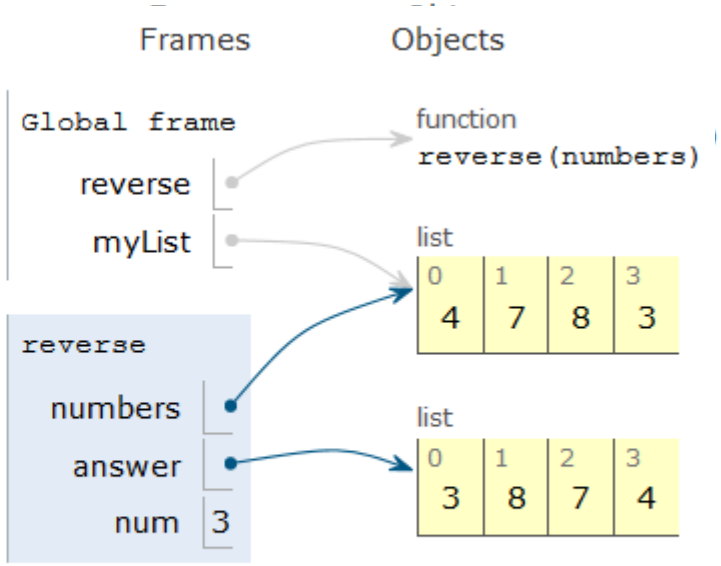
Edit code

<< First | < Back | Step 12 of 16 | Forward > | Last >>



Python Tutor
Compute reverse of a list





AlgoViz – Repository of Algorithm Visualizations

Browse the Catalog

Linear Structures
Search Structures
Spatial Search Structures
Misc. Data Structures
Quadratic Sorts
N log N sorts
Misc. Sorts
Search Algorithms
Compression Algorithms
Computational Geometry
Graph Algorithms
NP Completeness
Numerical Algorithm
Systems and Languages
Algorithmic Techniques
Misc. Topics

Electronic Textbooks (ebooks) engage students

- OpenDSA (Shaffer, Virgina Tech)
  - Algorithm animations built in
- runestoneinteractive.org (Brad Miller)
  - Several books (Python)
    - Python - try and run code built in
    - Quizzes
- Zyante.com – interactive textbooks
- Track student progress
- Requirements and design strategies for open source interactive computer science eBooks
  - ITiCSE 2013 Working Group (Korhonen, Naps, et al)

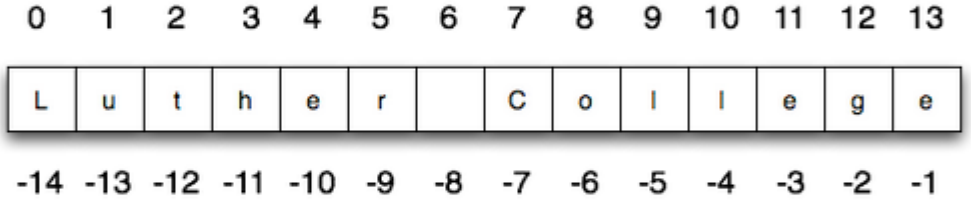
How to Think Like a Computer Scientist

Learning with Python: Interactive Edition 2.0

How To Think Like a Computer Scientist

Index Operator: Working with the Characters of a String

The **indexing operator** (Python uses square brackets to enclose the index) selects a single character from a string. The characters are accessed by their position or index value. For example, in the string shown below, the 14 characters are indexed left to right from postion 0 to position 13.

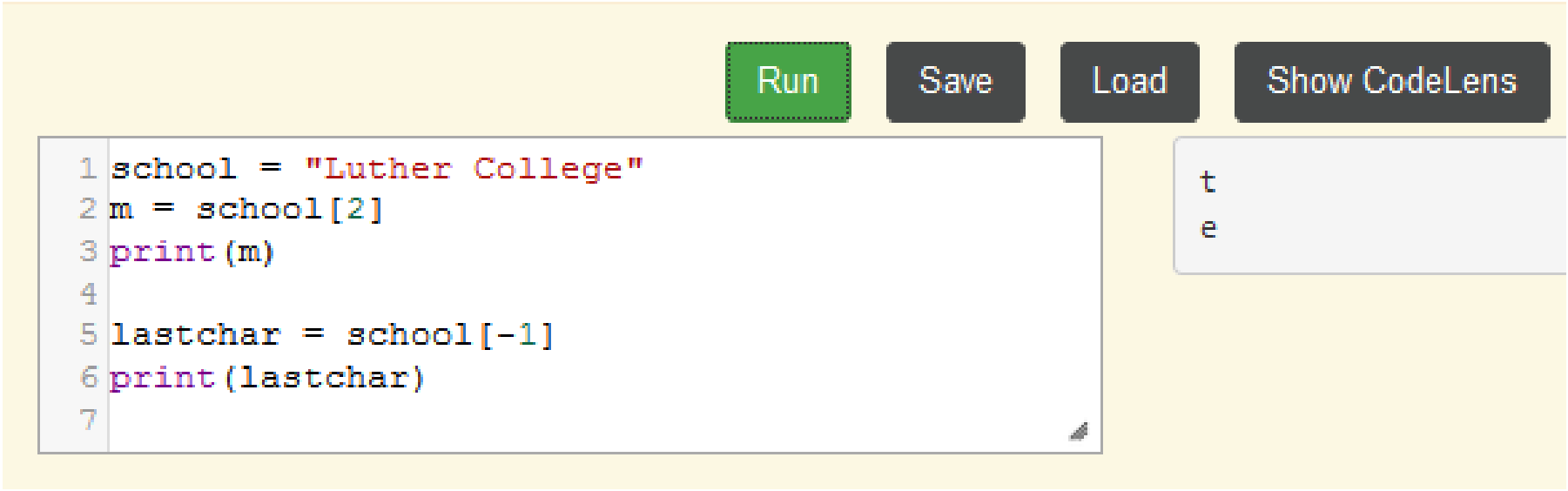


It is also the case that the positions are named from right to left using negative numbers where -1 is the rightmost index and so on. Note that the character at index 6 (or -8) is the blank character.

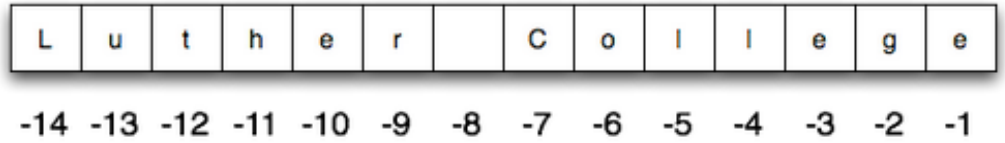
Run Save Load Show CodeLens

```
1 school = "Luther College"
2 m = school[2]
3 print(m)
4
5 lastchar = school[-1]
6 print(lastchar)
7
```

Run and edit code in the book



Integrates in Python Tutor



It is also the case that the positions are named from right to left using negative numbers where rightmost index and so on. Note that the character at index 6 (or -8) is the blank character.

Run | Save | Load | Hide Codelens

```
1 school = "Luther College"
2 m = school[2]
3 print(m)
4
5 lastchar = school[-1]
6 print(lastchar)
7
```

```
t
e
```

Python 2.7

```
1 school = "Luther College"
2 m = school[2]
3 print(m)
4
5 lastchar = school[-1]
6 print(lastchar)
```

< Back | Program terminated | Forward >

line that has just executed

next line to execute

Visualized using Online Python Tutor by Philip Guo

Frames | Objects

Global frame

school	"Luther College"
m	"t"
lastchar	"e"

Program output:

```
t
e
```

Questions for feedback

Check your understanding

strings-4-1: What is printed by the following statements?

```
s = "python rocks"
print(s[3])
```

- (•) t
- ( ) h
- ( ) c
- ( ) Error, you cannot use the [ ] operator with a string.

Check Me | Compare me

Incorrect. Index locations do not start with 1, they start with 0.

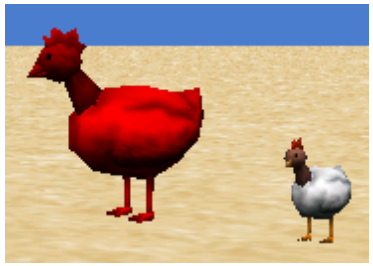
strings-4-2: What is printed by the following statements?

```
s = "python rocks"
print(s[2] + s[-5])
```

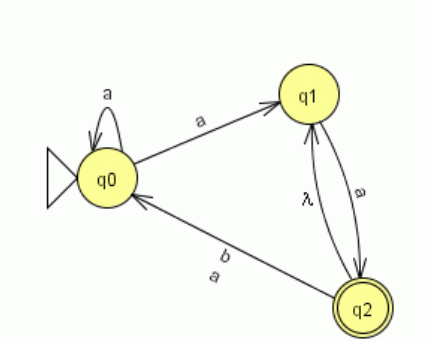
- (•) tr
- ( ) ps
- ( ) nn
- ( ) Error, you cannot use the [ ] operator with the + operator.

Check Me | Compare me

Correct! Yes, indexing operator has precedence over concatenation.



Outline



- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Additional Ways to Engage with CS
- Diversity Efforts

Java Program to count number of words in a phrase

```
public static void main() {

    String phrase = "baby fish should be fed three times a day";
    int numWords = 0;
    int pos = phrase.indexOf(" ");
    while (pos > 0)
    {
        numWords += 1;
        phrase = phrase.substring(pos+1);
        pos = phrase.indexOf(" ");
    }
    numWords += 1;
    System.out.println("Number of words in phrase is " + numWords);
}
```

- Is this program syntactically correct?

First identify each word/token

```
public static void main() {
```

TOKEN	TYPE
public	keyword
static	keyword
void	keyword
main	variable
(	leftparen
)	rightparen
{	leftbrace

First identify each word/token (cont)

```
String phrase = "baby fish should be fed three times a day";
```

TOKEN	TYPE
String	keyword
phrase	variable
=	equals
”	doublequote
baby fish … day	string
”	doublequote
;	semicolon

First identify each word/token (cont)

```
int numWords = 0;
```

TOKEN	TYPE
int	keyword
numWords	variable
=	equals
0	int
;	semicolon

Define Rules for a valid program (Grammar)

<program> := <procDef> ( ) { <body> }
<program> := <procDef> (<arglist>) { <body> }
<procDef> := public static void main
<procDef> := <pubtype> <returntype> <variable>
<body> := <decllist> <stmtlist>
<declist> := <decl> ; <decllist>
<declist> := <decl> ;
<stmtlist> := <stmt> ; <stmtlist>
<stmtlist> := <stmt> ;

Define Rules for a valid program (Grammar)

<decl> := int <variable> = <integer> ;

<decl> := String <variable> = “ <string> “ ;

<stmt> := while (<cond>) { <stmtlist> }

<stmt> := <variable> += <integer> ;

<stmt> := <variable> = <integer> ;

etc

Now derive the program using the rules (grammar)

<program> := <procDef> ( ) { <body> }

:= public static void main () { <body>}

:= public static void main () {
<declist> <stmtlist> }

:= public static void main() {
<decl> ; <declist> <stmtlist> }

And so on until you derive the program

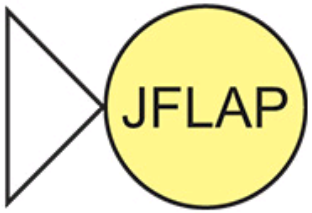
…

:= public static void main() { String
phrase = “baby fish should be fed …” ;
int numWords = 0; int pos =
phrase.indexOf(“ “);
while (pos > 0) { numWords += 1;
…
}

Determining if a Java program is syntactically correct

- Finite state machine (or determinisitic finite automaton - DFA) – to identify the words or tokens of the program
- Context-free grammar – to write the rules of the programming language
- LR Parsing determining if the program fits the rules – trying to derive the program.

- This area is known as Formal languages and Automata theory



Overview of JFLAP

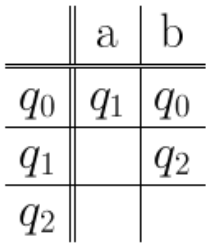
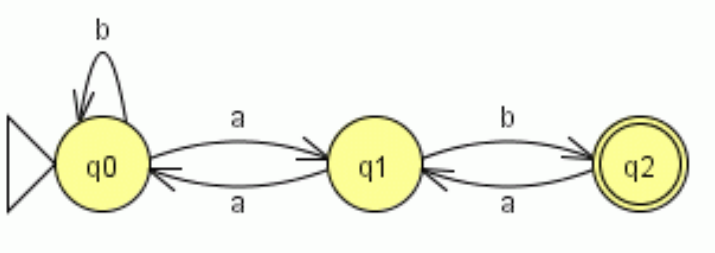
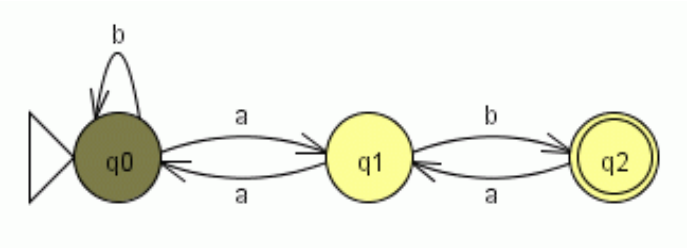
- **J**ava **F**ormal **L**anguages and **A**utomata **P**ackage
- Instructional tool to learn concepts of Formal Languages and Automata Theory
- Topics:
  - Regular Languages
  - Context-Free Languages
  - Recursively Enumerable Languages
  - Lsystems
- With JFLAP your creations come to life!

Thanks to Students - Worked on JFLAP and Automata Theory Tools

- 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
- JFLAP 7.0 - 2009 Henry Qin, Jonathan Su
- JFLAP 8.0Beta – 2011-14 Julian Genkins, Ian McMahon, Peggy Li, Lawrence Lin, John Godbey
- JFLAP in OpenDSA – 2015 Sung-Hoon Kim and Martin Tamayo

Over 25 years!

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	

DFA Example

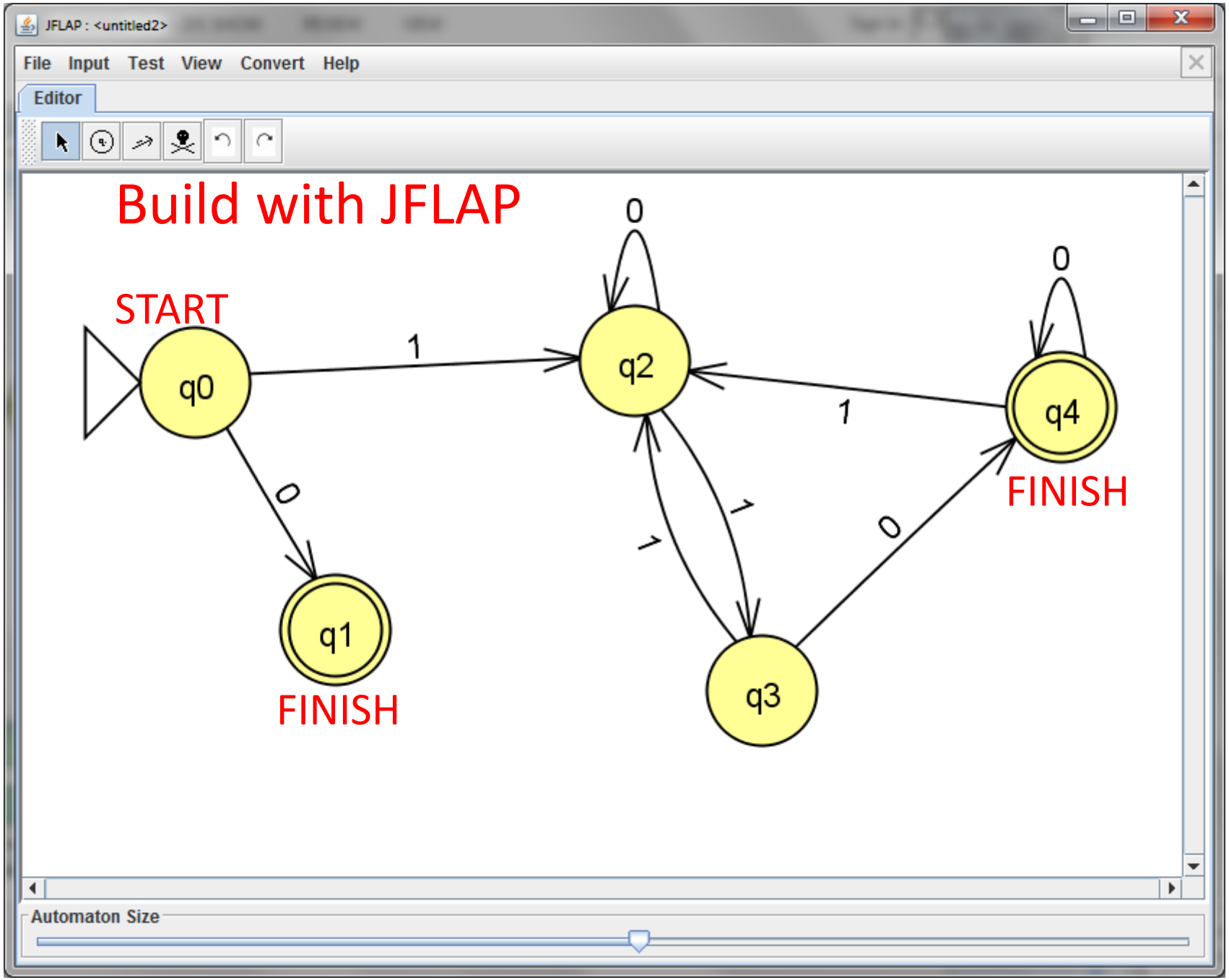
- Build a deterministic finite automaton(DFA) to recognize even binary numbers with an even number of 1s.
- Only use symbols 0 and 1
- Binary numbers: 0, 1, 10, 11, 100, 101, 110, 111, …
- When is a binary number an even number?
  - Ends in 0
- Which strings should be accepted?
- 11010, 10010, 1111, 10100

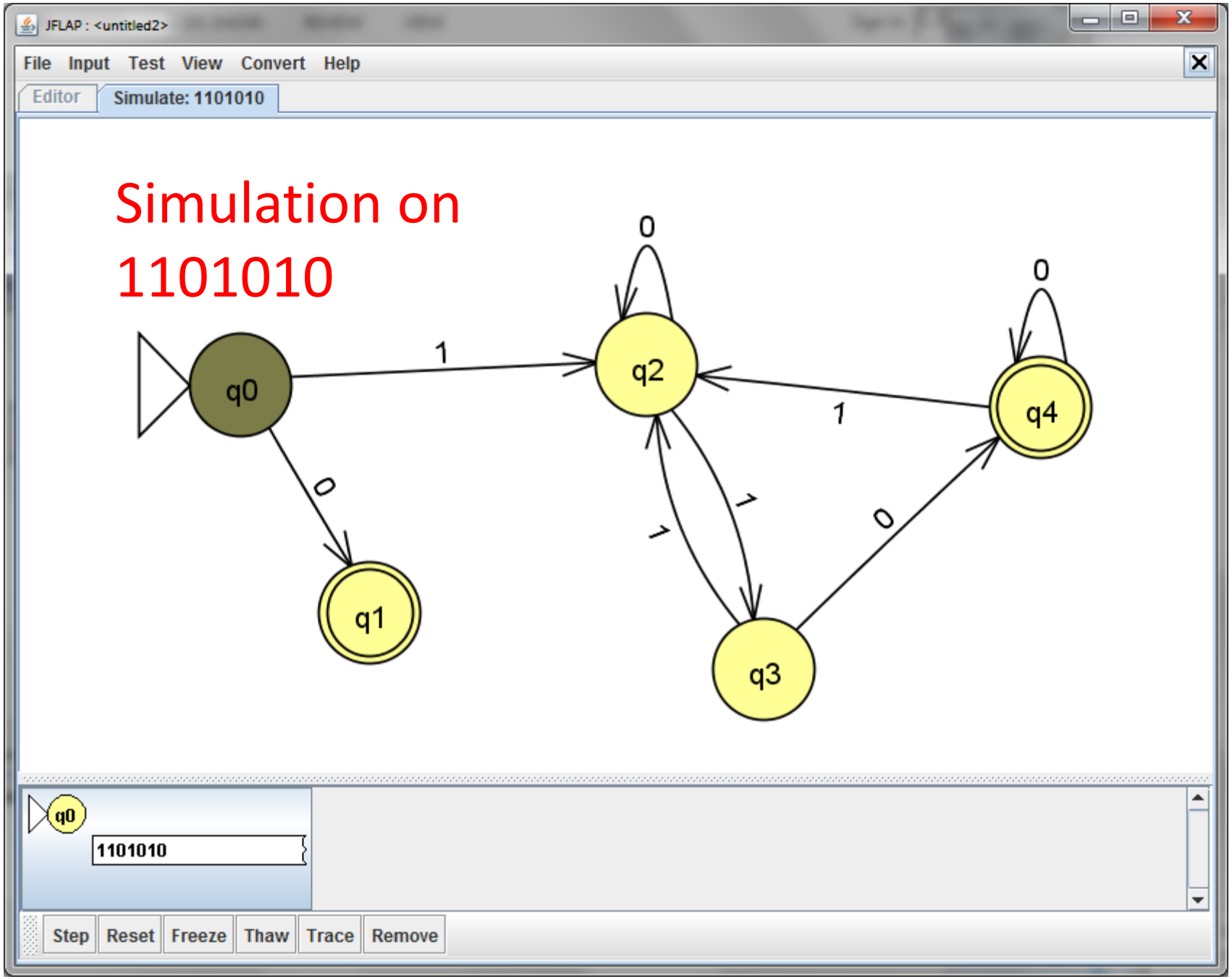
No, odd no. of 1's

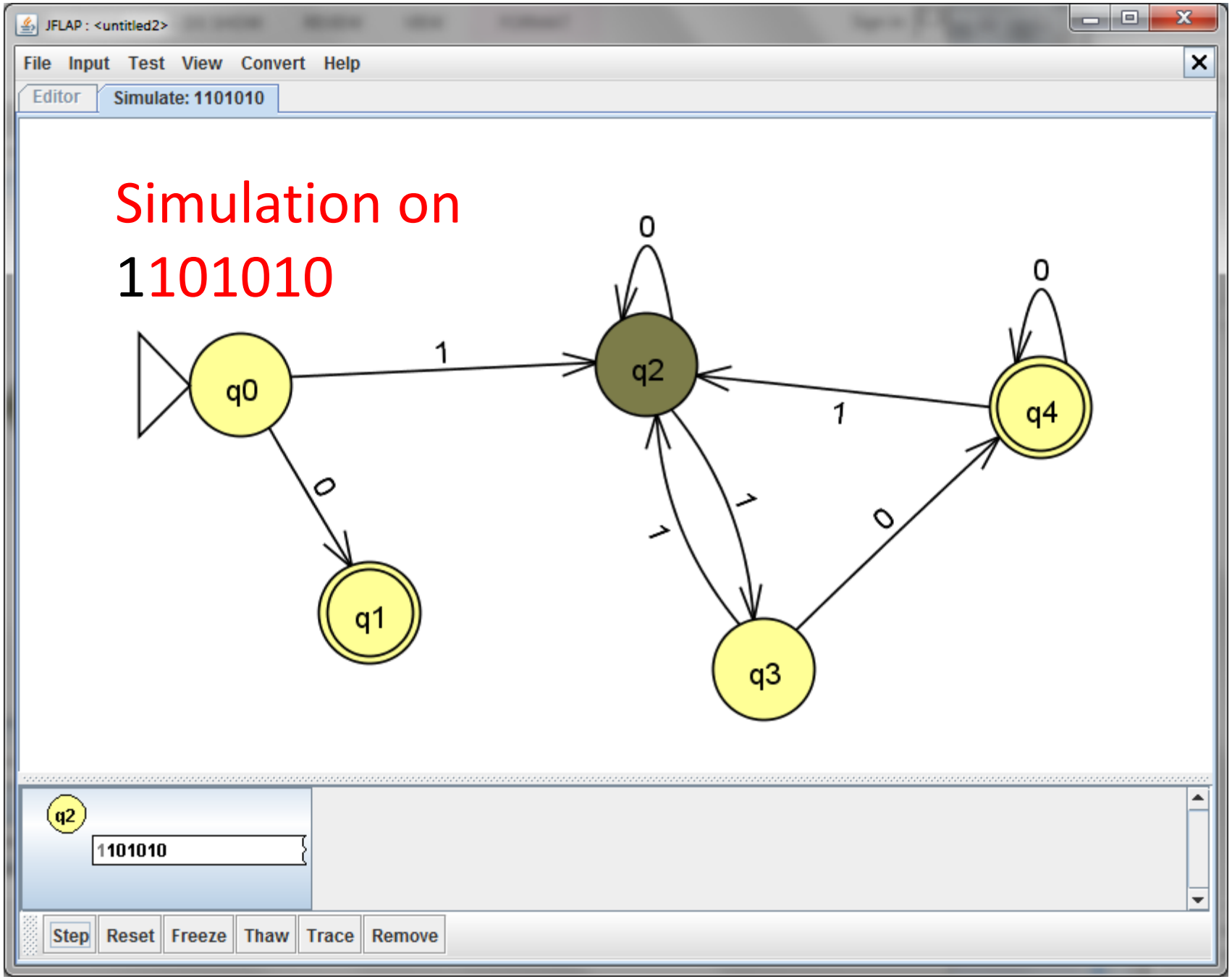
Yes

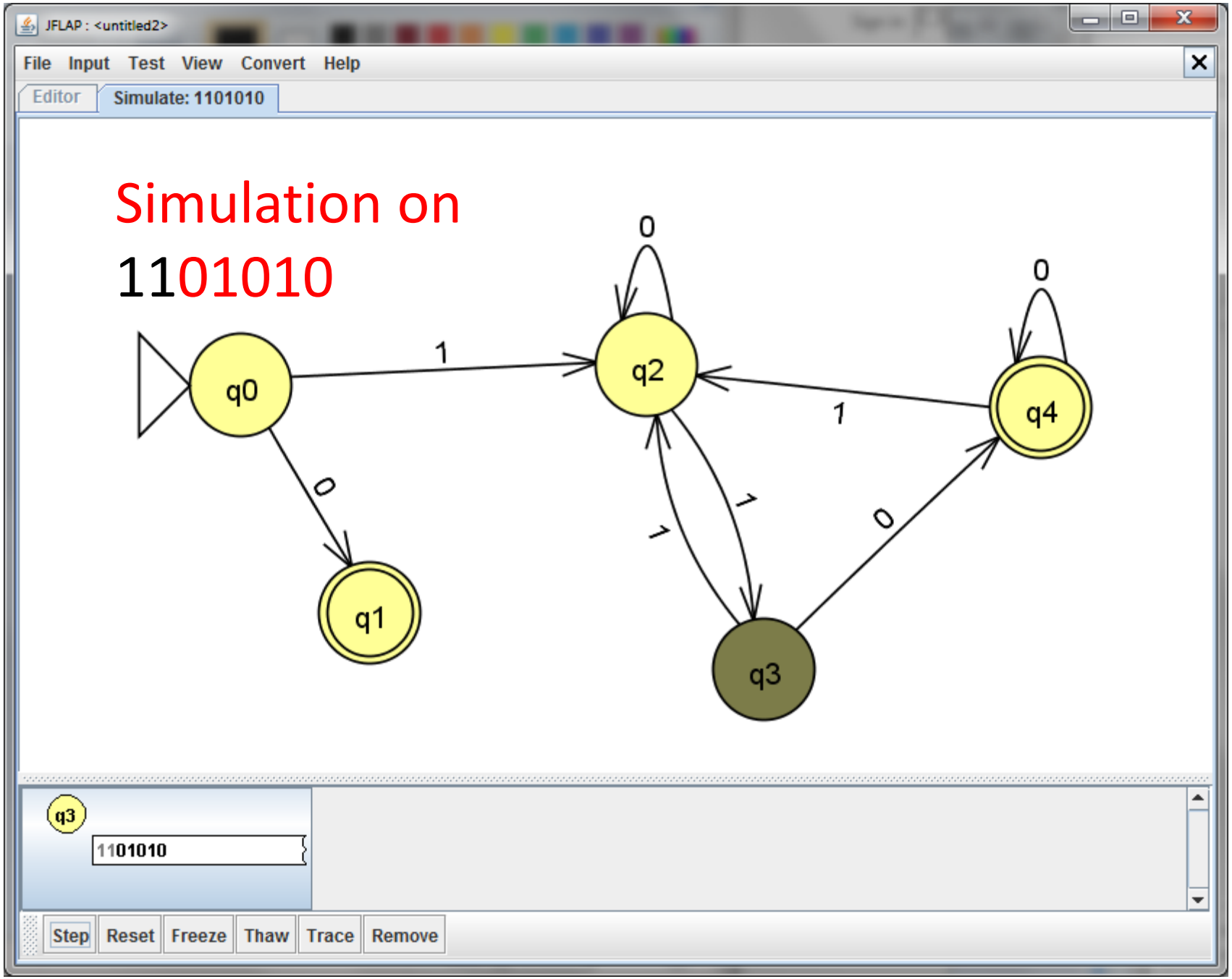
No, ends In 1

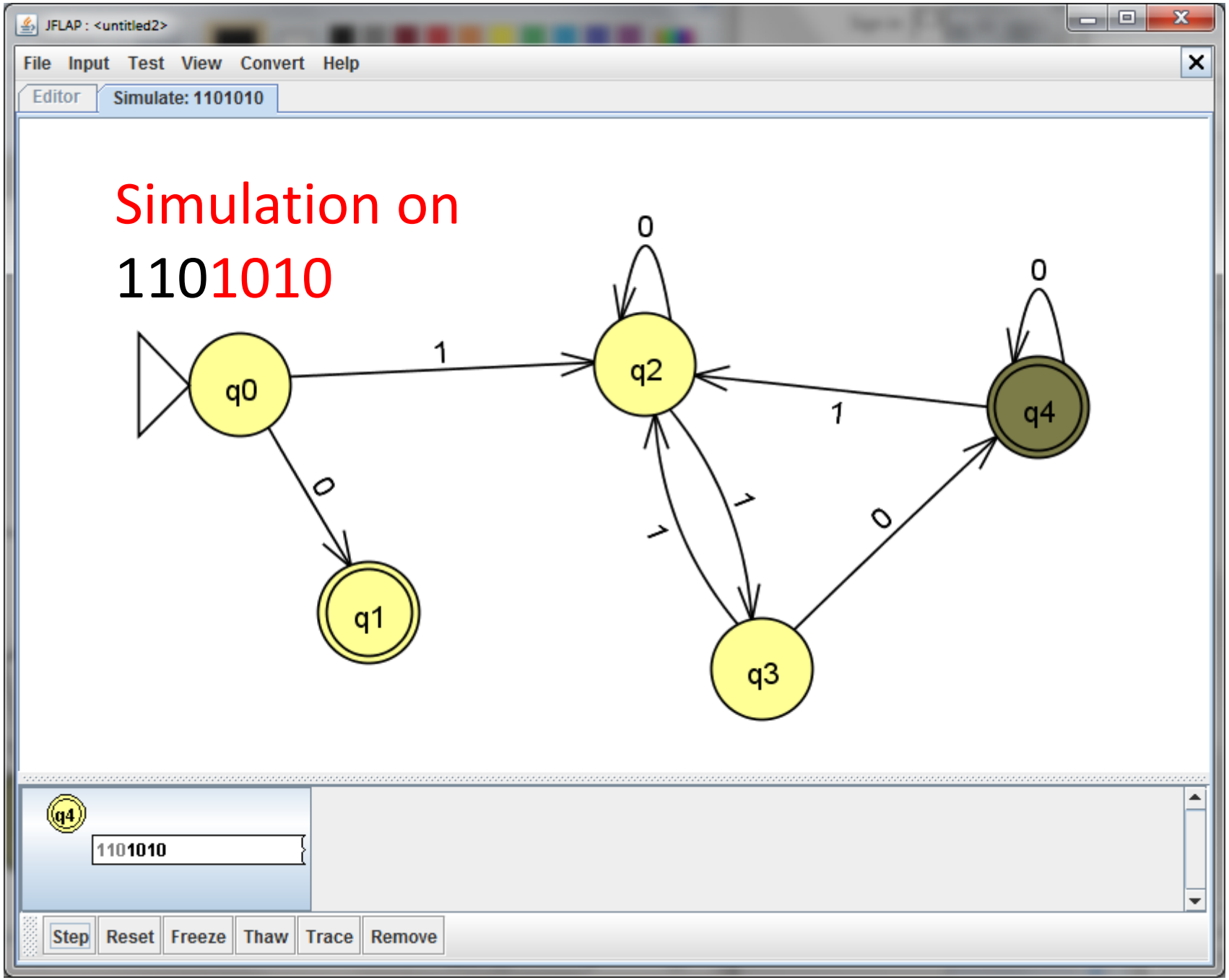
Yes

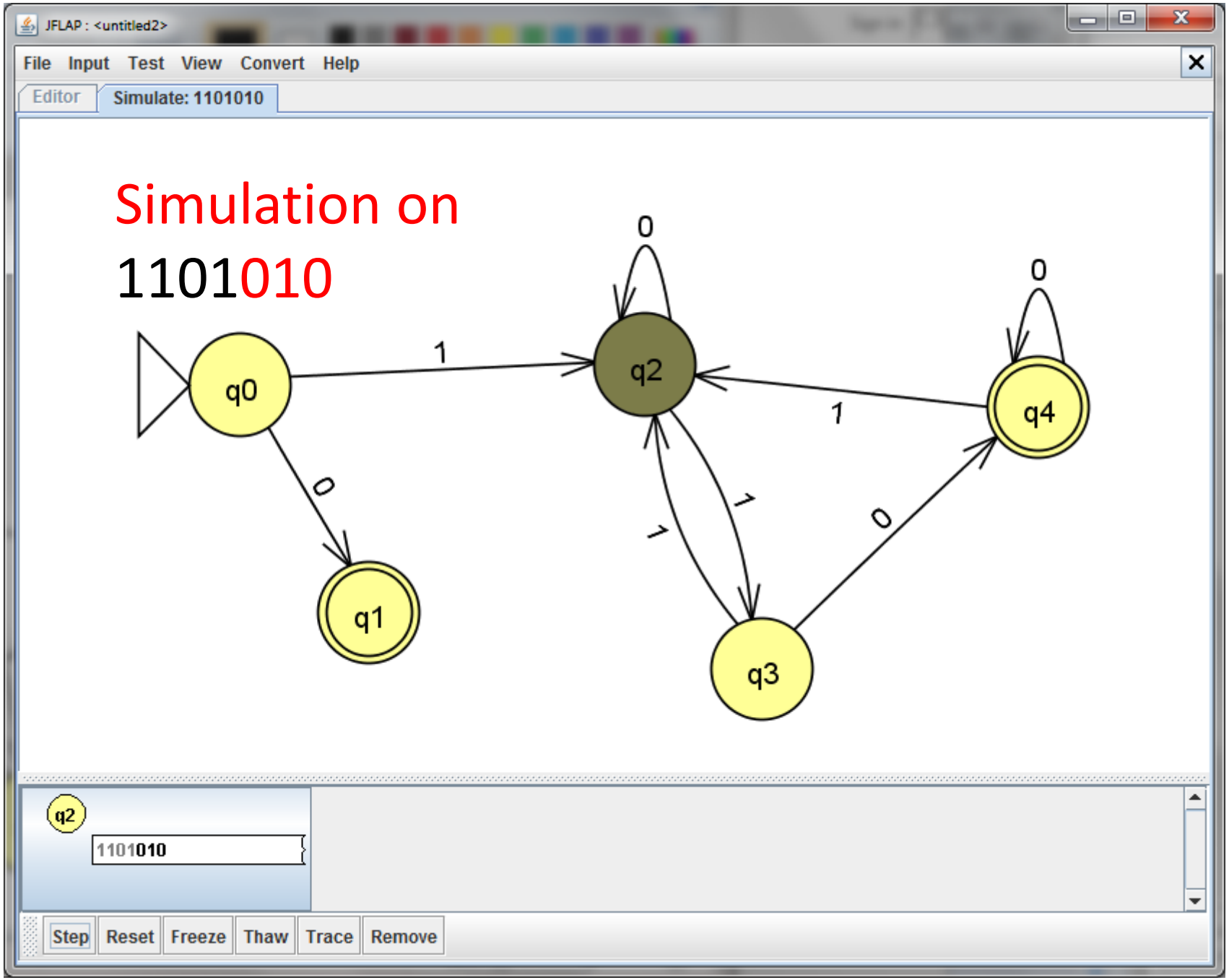

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Build with JFLAP
START
q0
1
q2
0
0
q4
1
FINISH
0
1
1
0
q1
FINISH
q3
Automaton Size

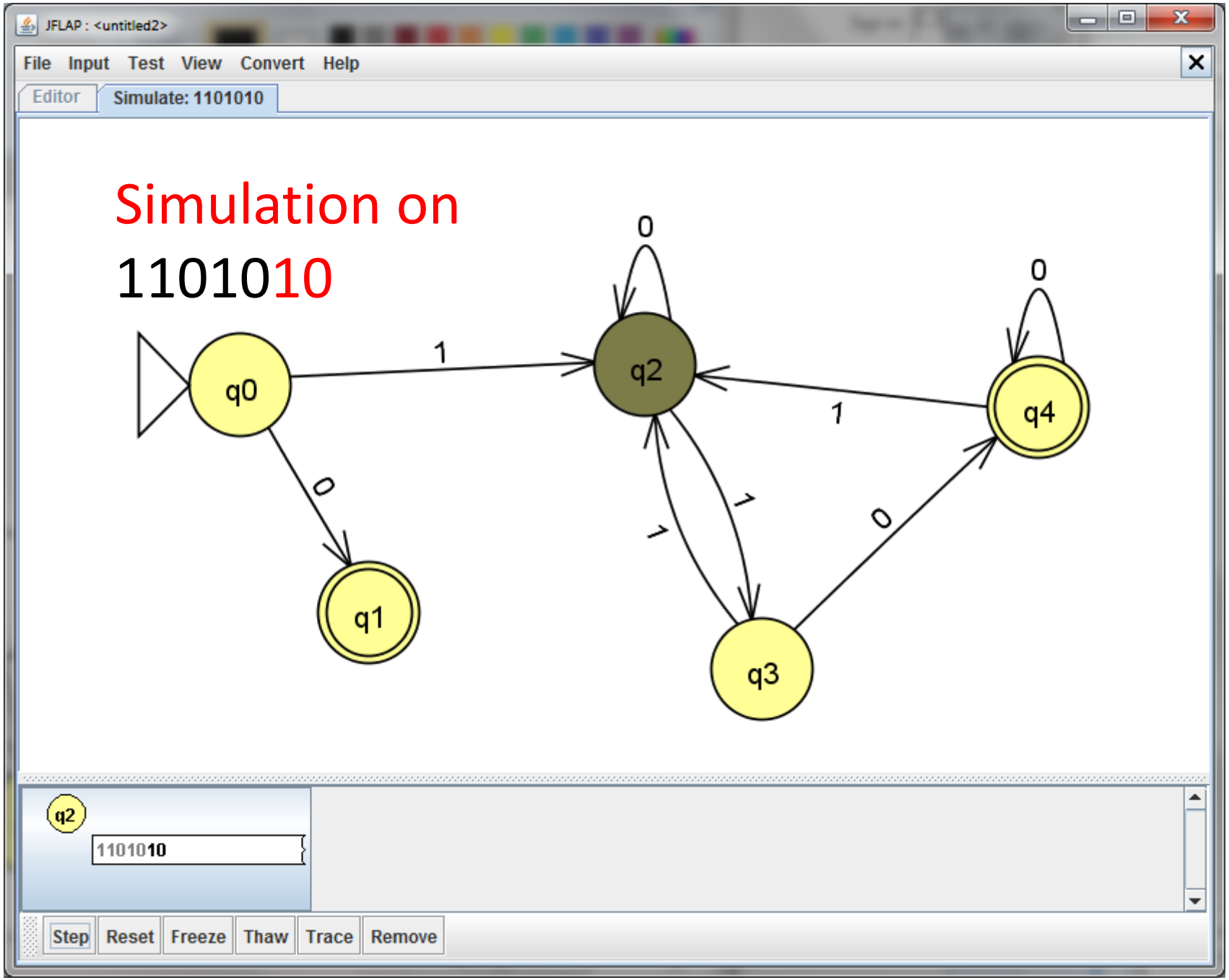

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on 1101010
q0
q1
q2
q3
q4
0
1
0
1
1
1
0
0
q0
1101010
Step
Reset
Freeze
Thaw
Trace
Remove

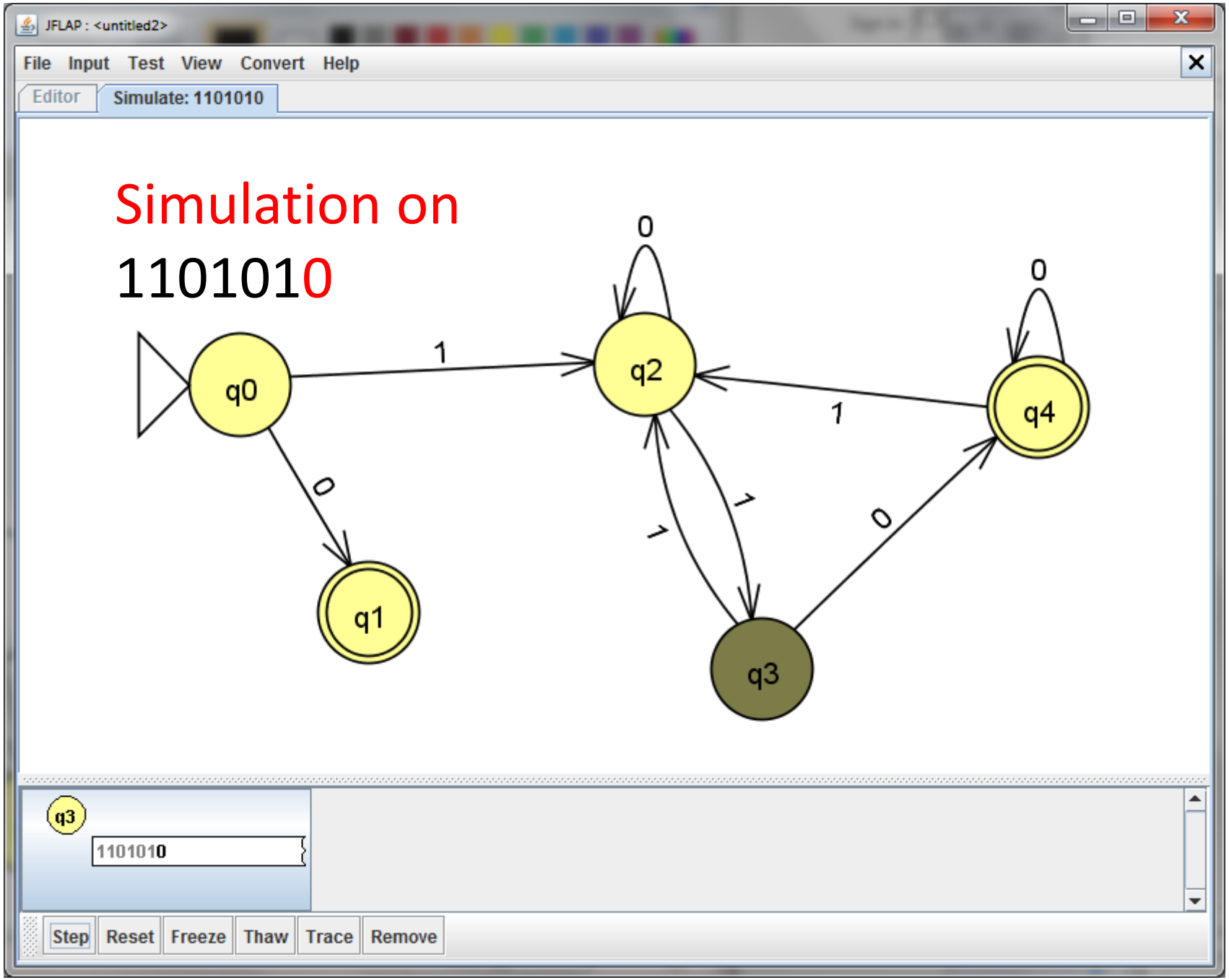

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on
1101010
q0
q1
q2
q3
q4
0
1
0
1
1
1
0
0
q2
1101010
Step
Reset
Freeze
Thaw
Trace
Remove

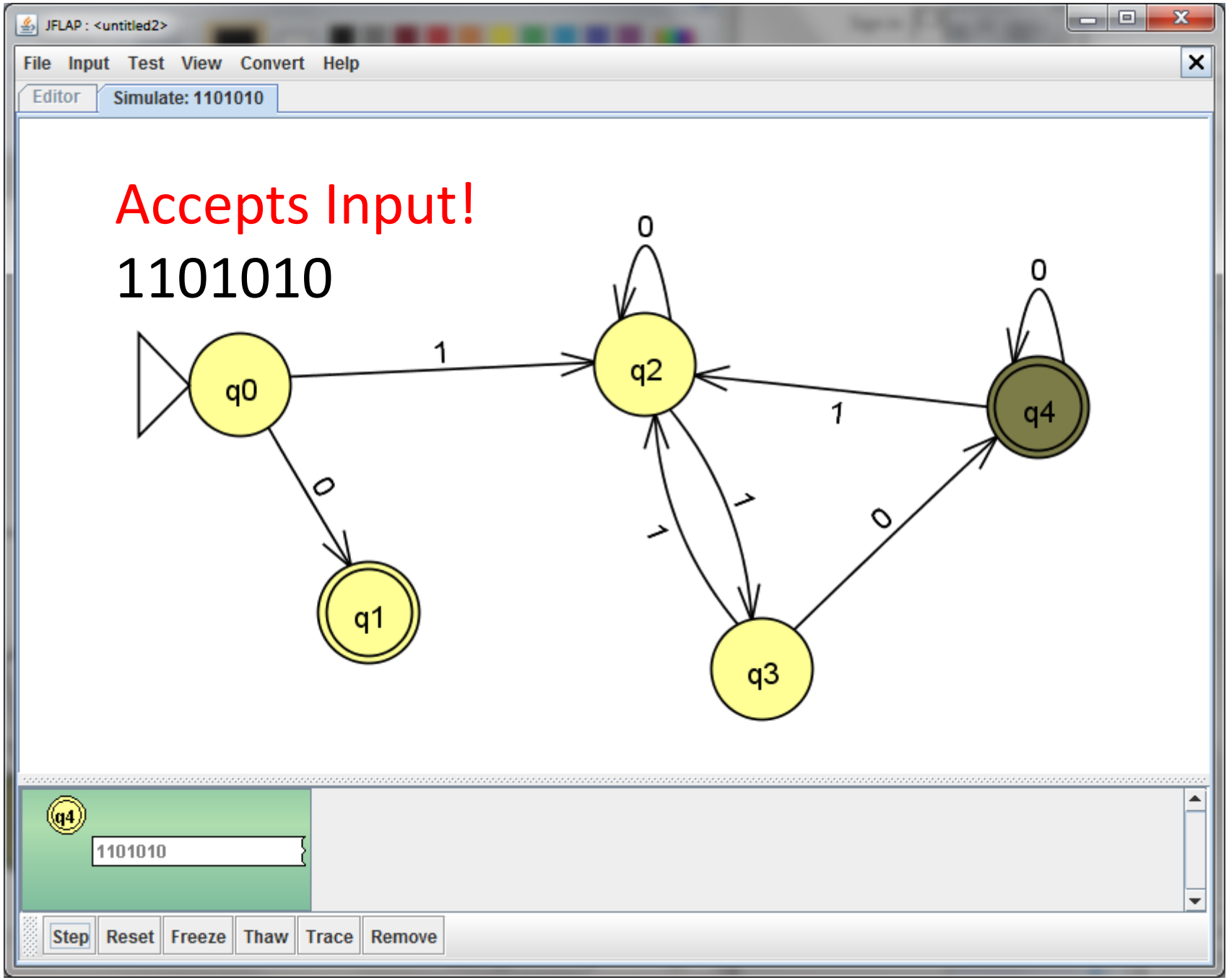

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on
1101010
q0
q1
q2
q3
q4
0
1
q3
1101010
Step
Reset
Freeze
Thaw
Trace
Remove

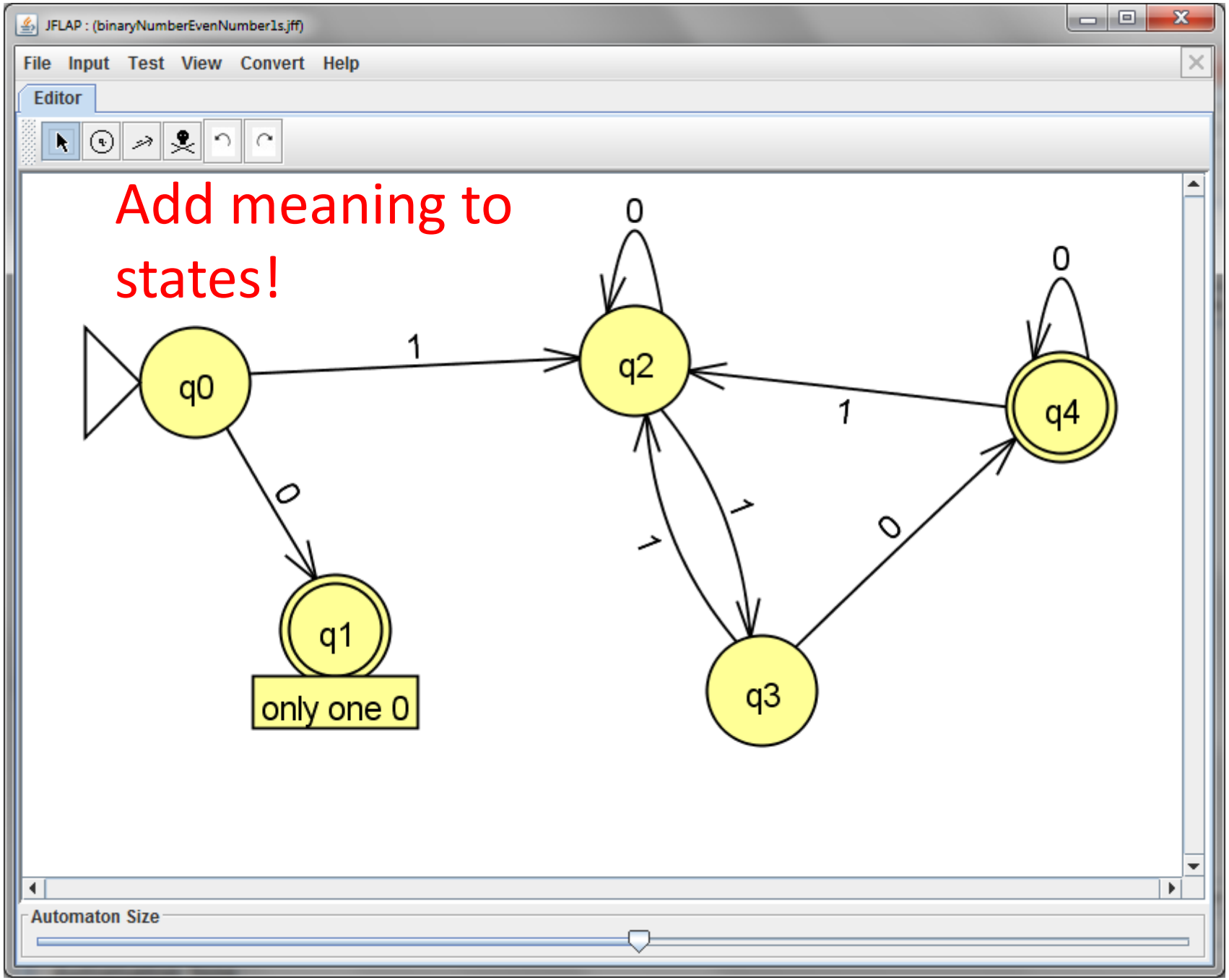

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on
1101010
q0
q1
q2
q3
q4
0
1
0
1
1
0
1
0
(q4)
1101010
Step
Reset
Freeze
Thaw
Trace
Remove

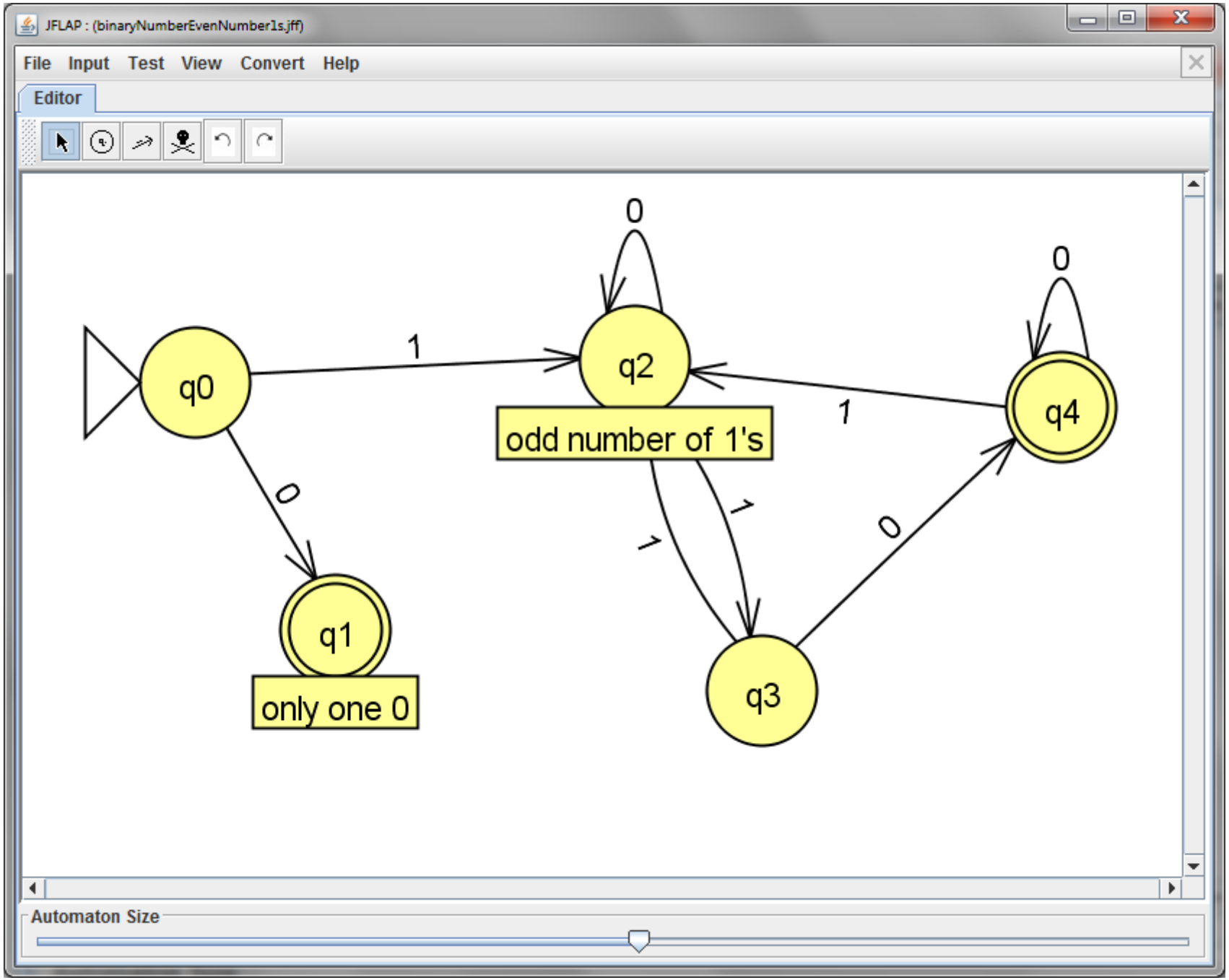

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on
1101010
q0
q1
q2
q3
q4
0
1
0
1
1
0
0
q2
1101010
Step
Reset
Freeze
Thaw
Trace
Remove

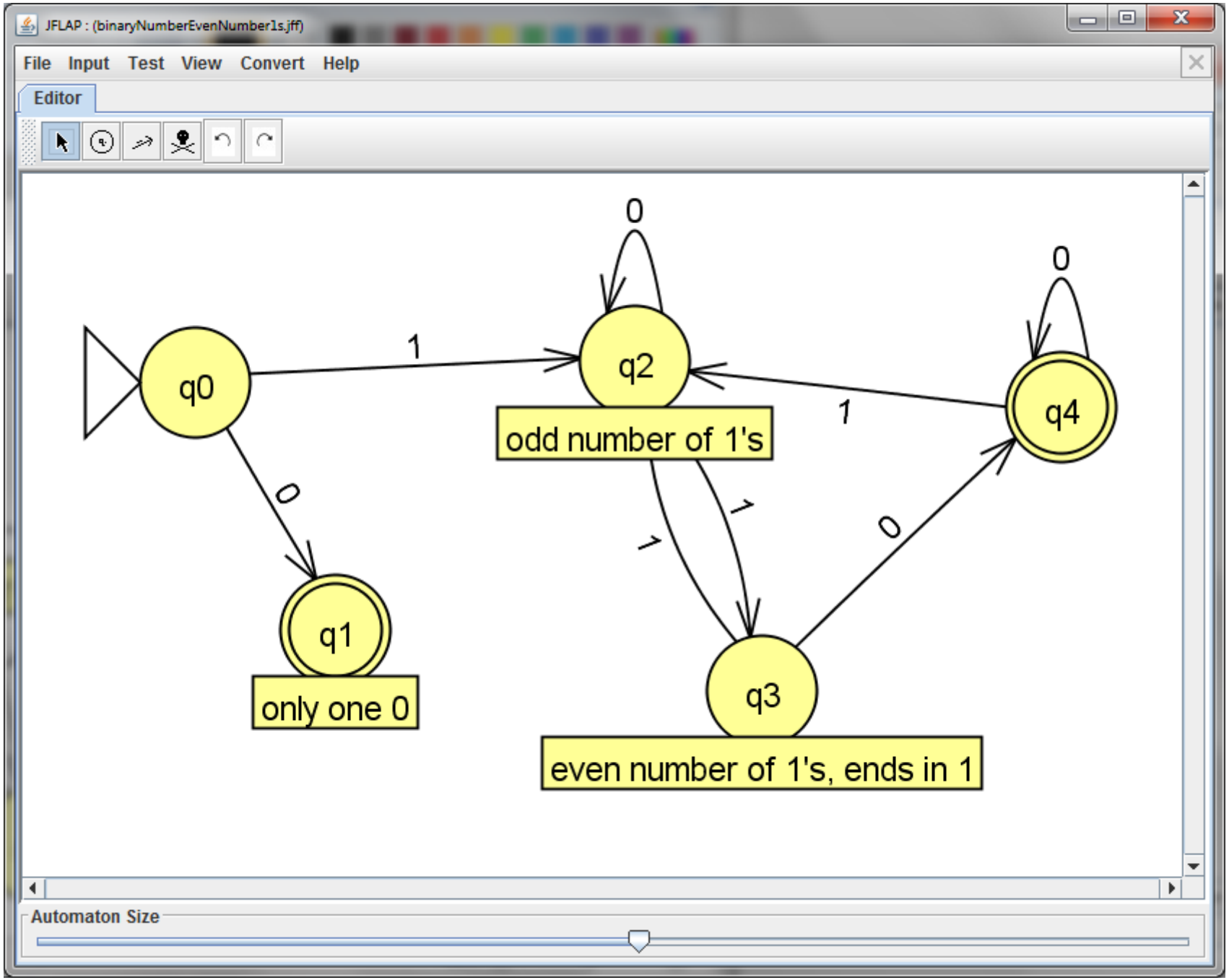

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on
1101010
q0
q1
q2
q3
q4
0
1
0
1
1
1
0
0
q2
1101010
Step
Reset
Freeze
Thaw
Trace
Remove

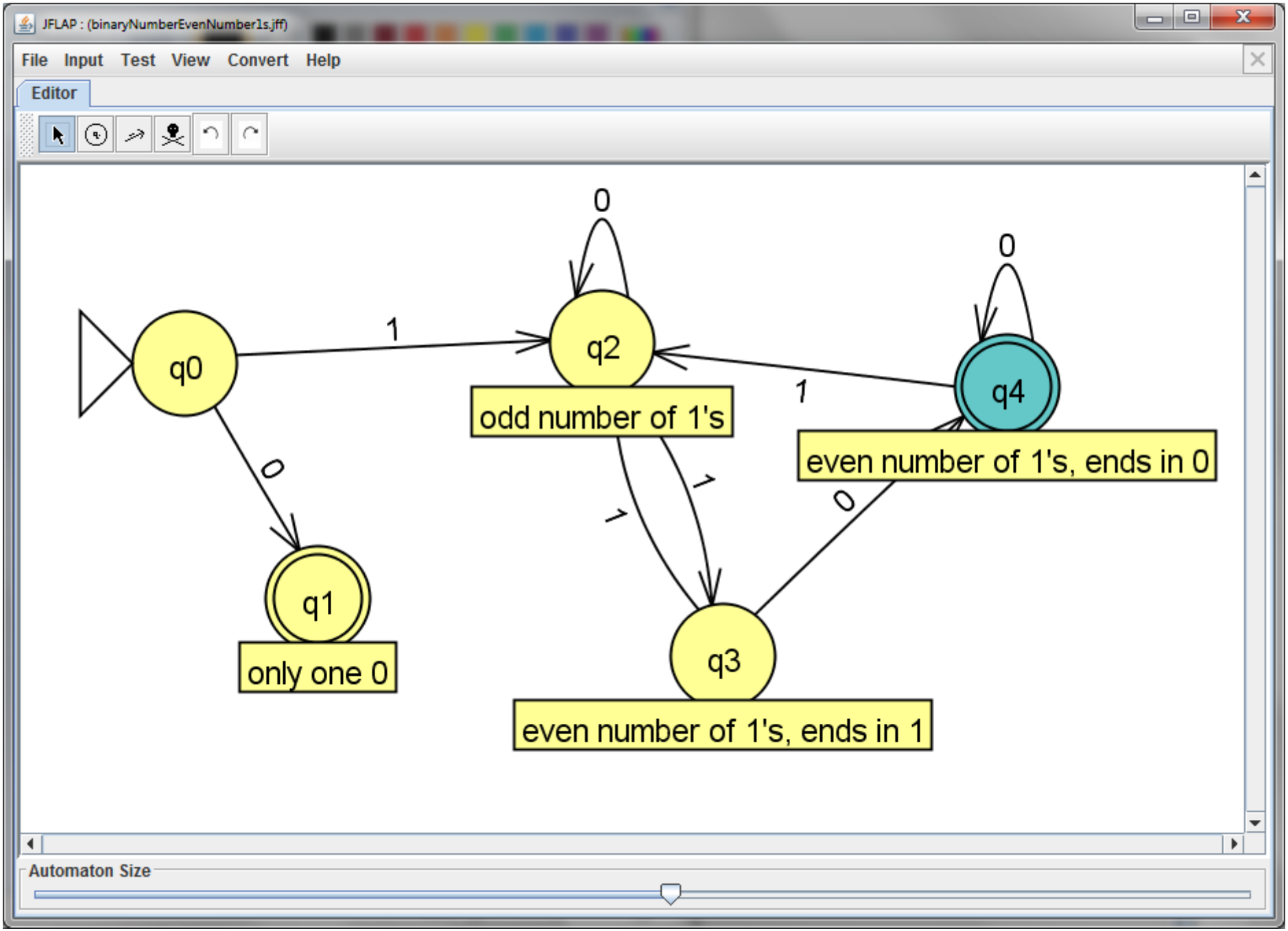

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on
1101010
q0
q1
q2
q3
q4
0
1
0
1
1
0
0
q3
1101010
Step
Reset
Freeze
Thaw
Trace
Remove


JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Accepts Input!
1101010
q0
q1
q2
q3
q4
0
1
0
1
1
1
0
0
q4
1101010
Step
Reset
Freeze
Thaw
Trace
Remove

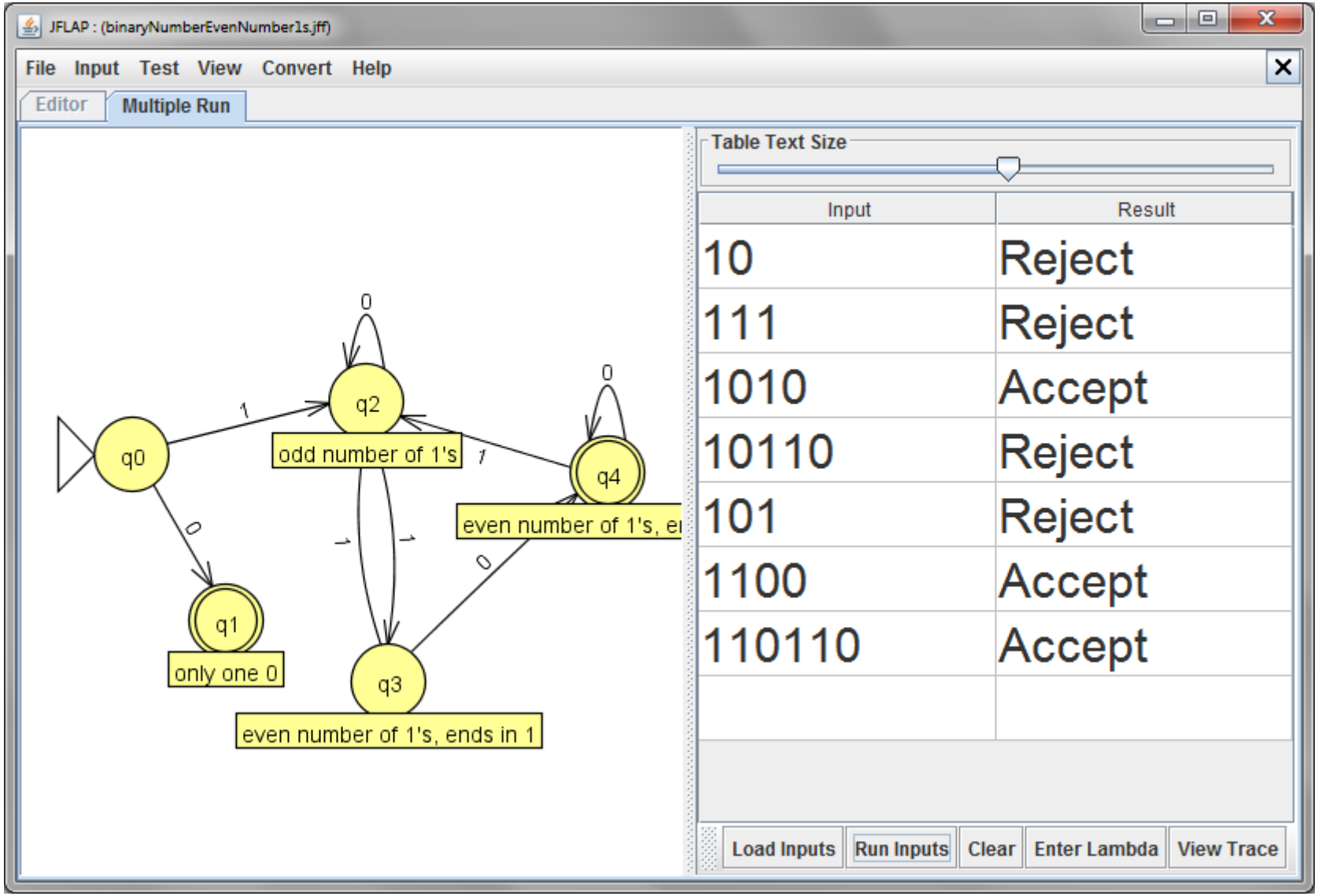

JFLAP : (binaryNumberEvenNumber1s.jff)
File Input Test View Convert Help
Editor
Add meaning to states!
q0
q1
only one 0
q2
q3
q4
0
1
0
1
1
0
0
Automaton Size


JFLAP : (binaryNumberEvenNumber1s.jff)
File Input Test View Convert Help
Editor
0
0
1
q0
q2
q4
1
odd number of 1's
0
1
1
0
q1
only one 0
q3
Automaton Size


JFLAP : (binaryNumberEvenNumber1s.jff)
File Input Test View Convert Help
Editor
0
0
1
q0
q2
q4
1
odd number of 1's
0
1
1
0
q1
q3
only one 0
even number of 1's, ends in 1
Automaton Size


JFLAP : (binaryNumberEvenNumber1s.jff)
File Input Test View Convert Help
Editor
0
0
1
q0
q2
q4
1
odd number of 1's
even number of 1's, ends in 0
0
1
1
0
q1
q3
only one 0
even number of 1's, ends in 1
Automaton Size

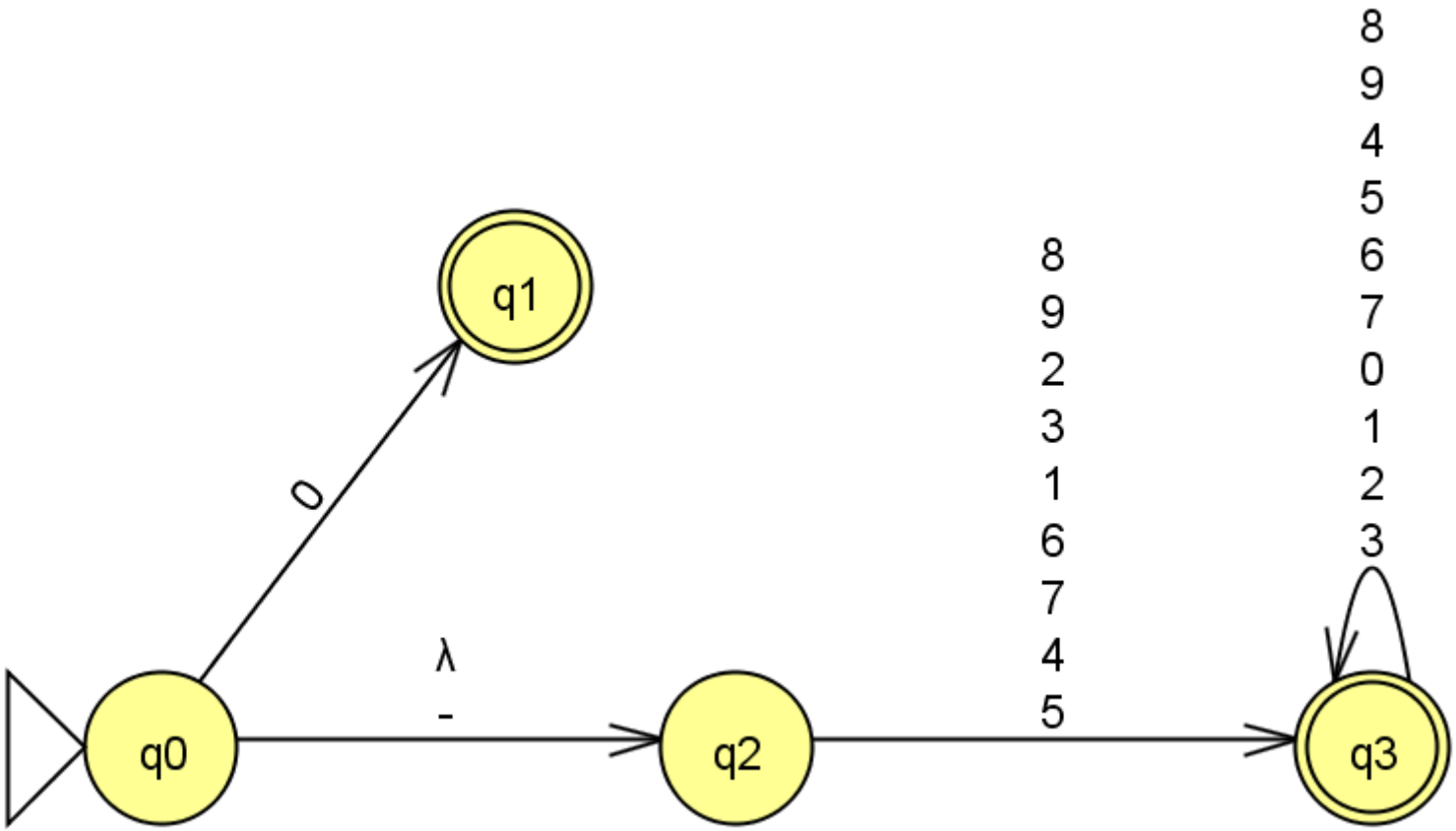
Test Multiple Inputs



Example: Build an NFA for valid integers

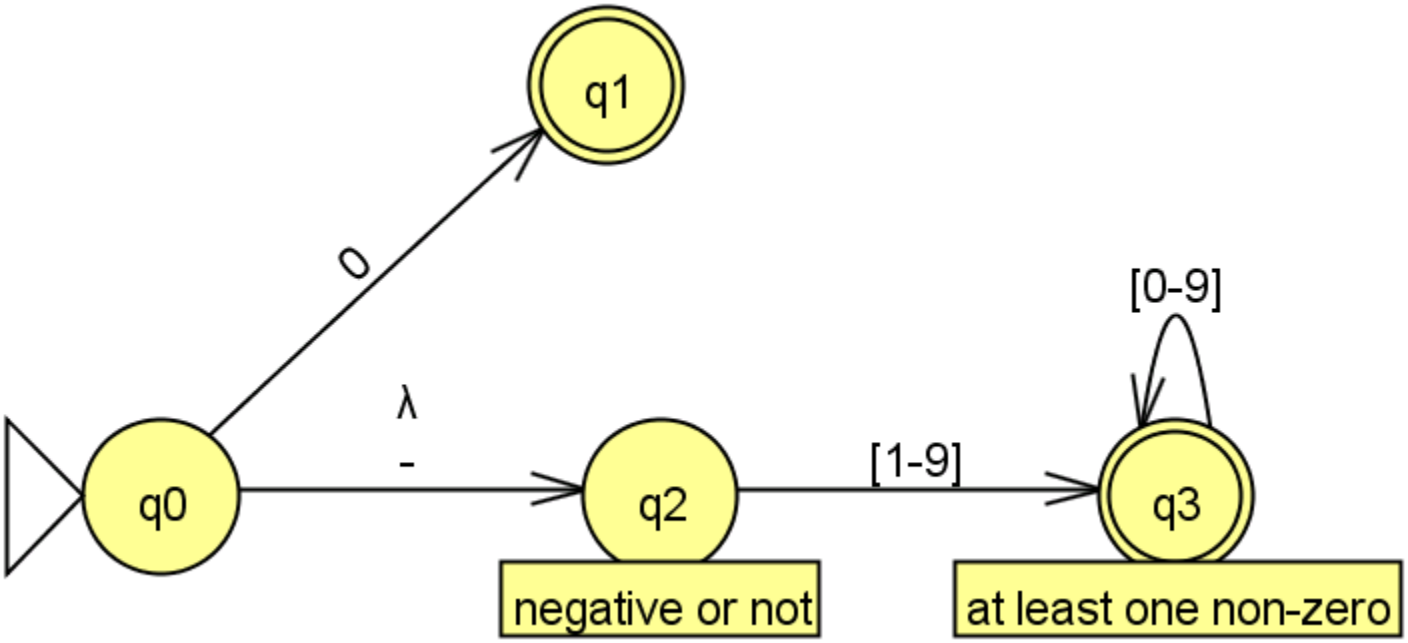
- Example:
  - Valid integers {-3, 8, 0, 456, 13, 500, …}
  - Not valid: {006, 3-6, 4.5, …}

Example: NFA for all valid integers



NFA annotated and shortcut

- Shortcut: [1-9] on labels



Back to Recognizing whether a Java Program is syntactically correct or not…

- You would need a DFA to recognize all valid words in a program
  - An integer
  - A variable name
  - All keywords
  - All special symbols ; + - ( ) { }
  - etc

Another Example: Grammar

- Grammar – set of replacement rules to define a language
- Previously looked at grammar for Java (very small part of it!)
- Grammar for a formal language (simpler)
- Consider representing underlined words in a text file (to be interpreted later):
  - cookie&&&&&&_____ cookie

    & = go back one

Grammar for $a^n b^n c^n$

S	→	A X
A	→	a A b c
A	→	a B b c
B X	→	λ
B b	→	b B
B c	→	D
D X	→	E X c
D b	→	b D
D c	→	c D
a E	→	a B
b E	→	E b
c E	→	E c

- Unrestricted grammar
- Generates strings with an equal number of a's, b's, c's
- a's first, then b's, then c's
- Example strings can derive:

  abc

  aabbcc

  aaabbbccc

  aaaabbbbcccc

  aaaaabbbbbccccc

  …

Example Derivation for aabbcc

$S \rightarrow AX$ rule: S -> AX

Example Derivation for aabbcc

S → AX rule: S -> AX

→ aAbcX rule: A -> aAbc

Example Derivation for aabbcc

S → AX	rule: S -> AX
→ aAbcX	rule: A -> aAbc
→ aaBbcbcX	rule: A -> aBbc

NOTE: We have generated the correct symbols, aabcbc, but they are in the wrong order!

Example Derivation for aabbcc

S → AX	rule: S -> AX
→ aAbcX	rule: A -> aAbc
→ aaBbcbcX	rule: A -> aBbc
→ aabBcbcX	rule: Bb -> bB

Example Derivation for aabbcc

S → AX	rule: S -> AX
→ aAbcX	rule: A -> aAbc
→ aaBbcbcX	rule: A -> aBbc
→ aabBcbcX	rule: Bb -> bB
→ aabDbcX	rule: Bc -> D

Note: the D absorbed the c!

Example Derivation for aabbcc

S → AX	rule: S -> AX
→ aAbcX	rule: A -> aAbc
→ aaBbcbcX	rule: A -> aBbc
→ aabBcbcX	rule: Bb -> bB
→ aabDbcX	rule: Bc -> D
→ aabbDcX	rule: Db -> bD

Example Derivation for aabbcc

S → AX	rule: S -> AX
→ aAbcX	rule: A -> aAbc
→ aaBbcbcX	rule: A -> aBbc
→ aabBcbcX	rule: Bb -> bB
→ aabDbcX	rule: Bc -> D
→ aabbDcX	rule: Db -> bD
→ aabbcDX	rule: Dc -> cD

Example Derivation for aabbcc

S → AX
→ aAbcX
→ aaBbcbcX
→ aabBcbcX
→ aabDbcX
→ aabbDcX
→ aabbcDX
→ aabbcEXc

Eventually … → aabbcc

rule: S -> AX

rule: A -> aAbc

rule: A -> aBbc

rule: Bb -> bB

rule: Bc -> D

rule: Db -> bD

rule: Dc -> cD

rule: Dx -> Exc

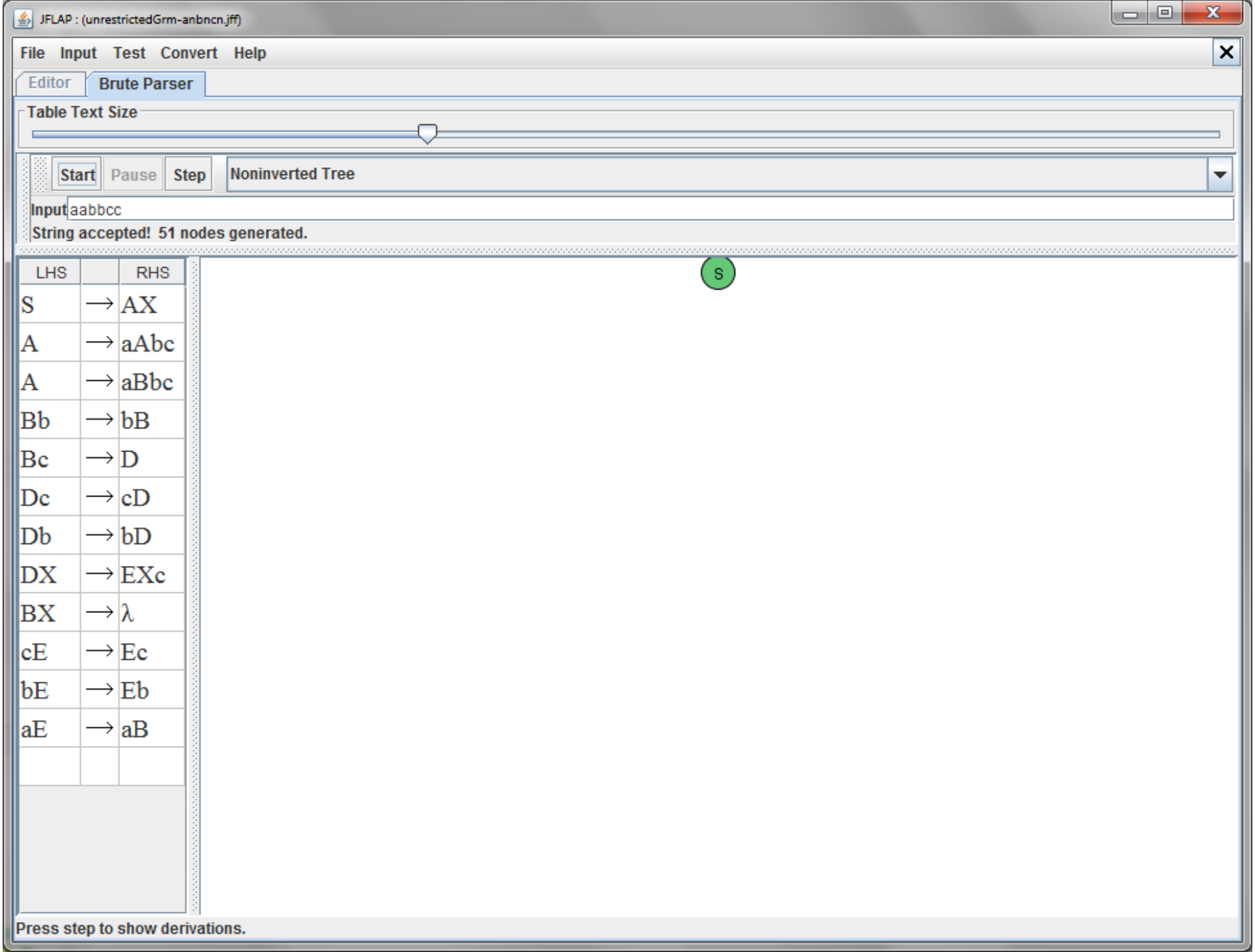
Note the c spit out on right end!

We could have done this derivation of aabbcc with JFLAP.

Now let's see how JFLAP visualizes this derivation with a "parse tree"

XXXXXXXX

Parse DAG


JFLAP : (unrestrictedGrm-anbncn.jff)
File Input Test Convert Help
Editor Brute Parser
Table Text Size
Start Pause Step Noninverted Tree
Input aabbcc
String accepted! 51 nodes generated.
LHS RHS
S → AX
A → aAbc
A → aBbc
Bb → bB
Bc → D
Dc → cD
Db → bD
DX → EXc
BX → λ
cE → Ec
bE → Eb
aE → aB
S
Press step to show derivations.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB

Derived AX from S.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

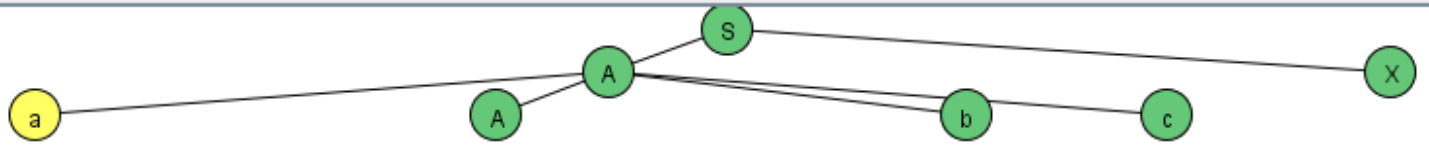
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived aAbc from A.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

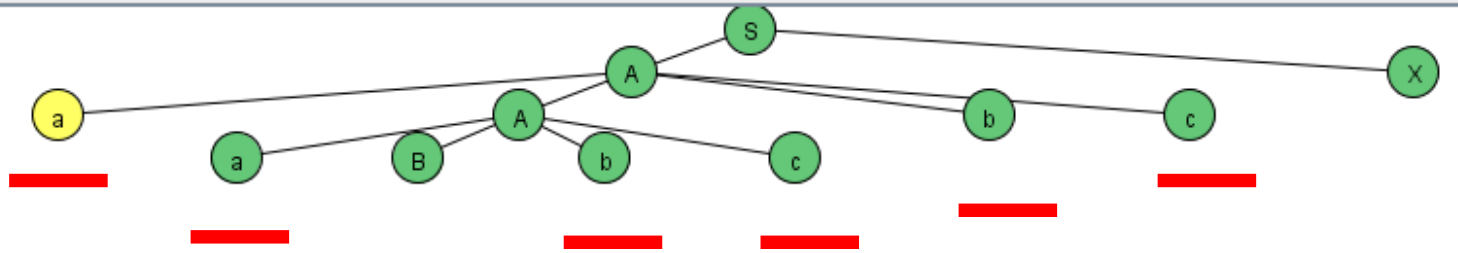
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Note all letters there, but wrong order: aabcbc

Derived aBbc from A.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

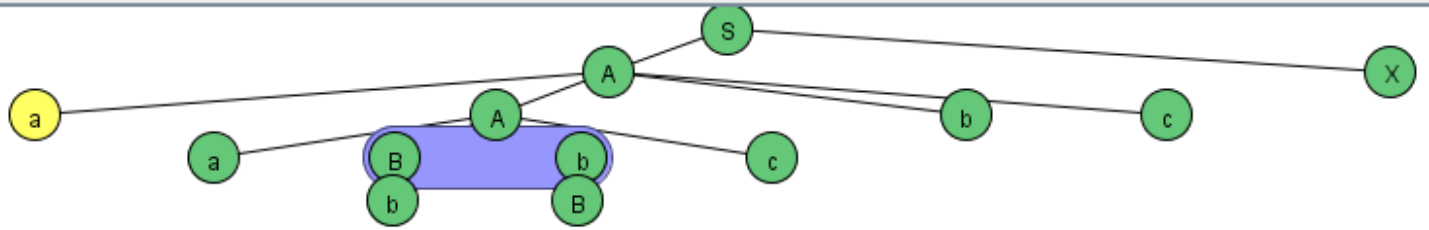
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived bB from Bb.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

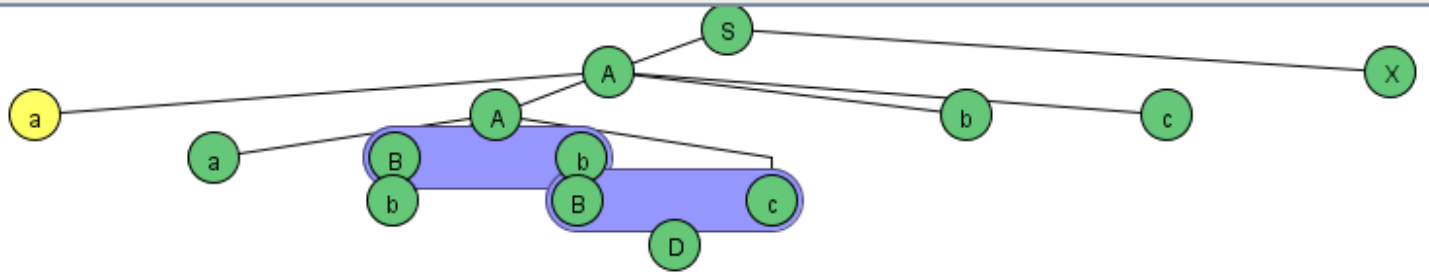
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Absorb the “c”

Derived D from Bc.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

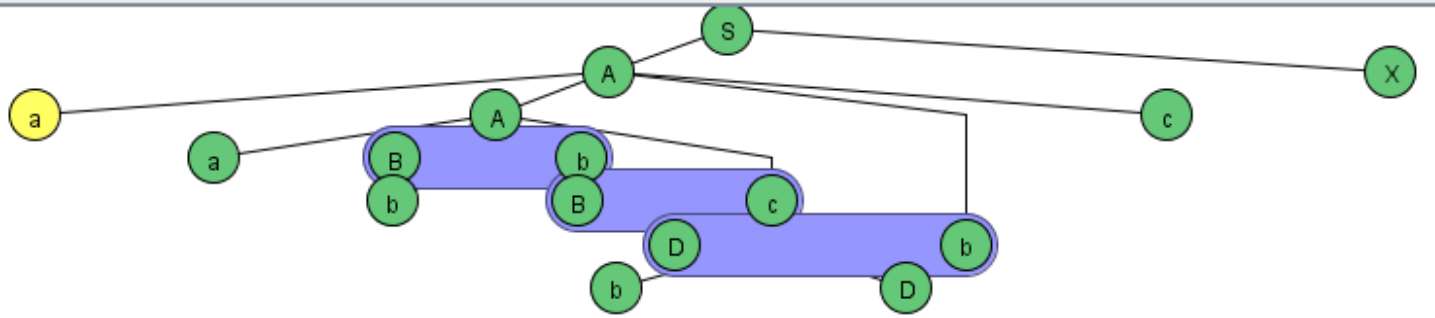
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived bD from Db.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

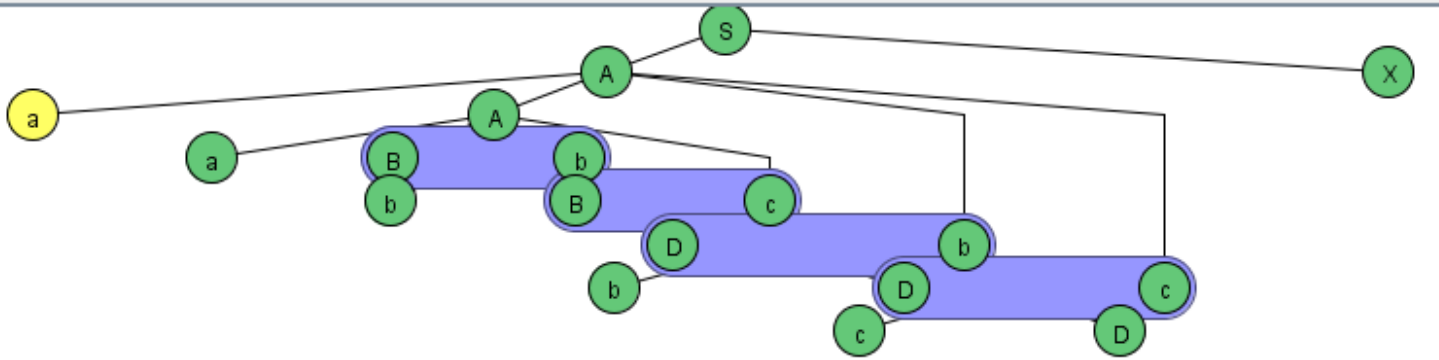
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived cD from Dc.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

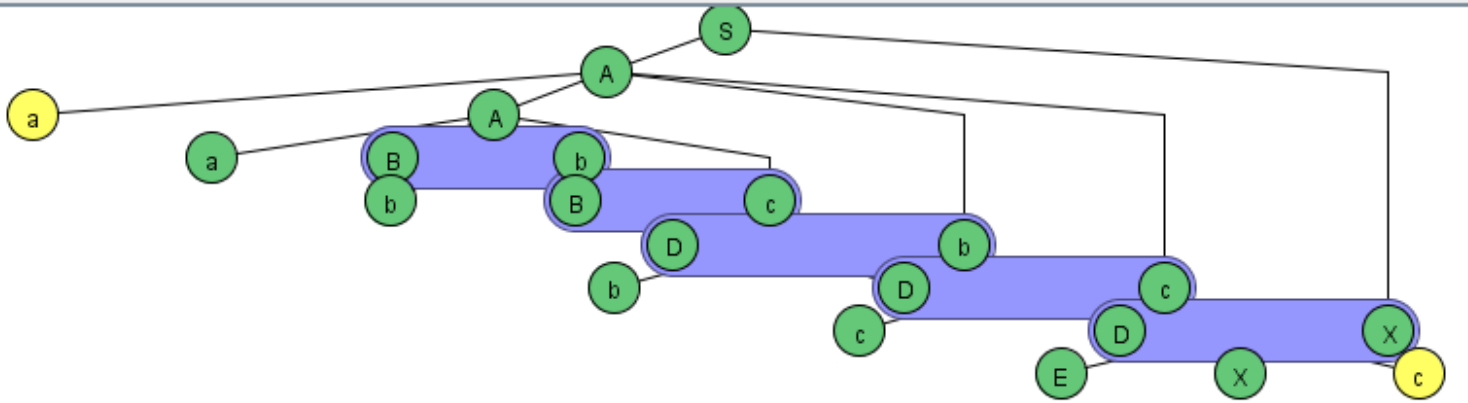
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Spit out the “c” at the right end

Derived EXc from DX.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

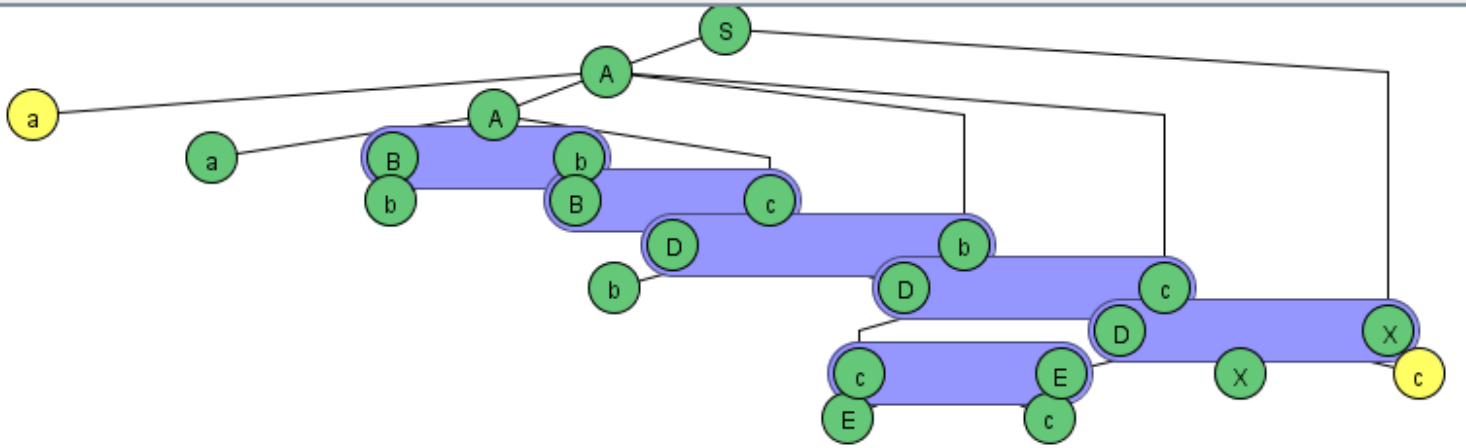
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived Ec from cE.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

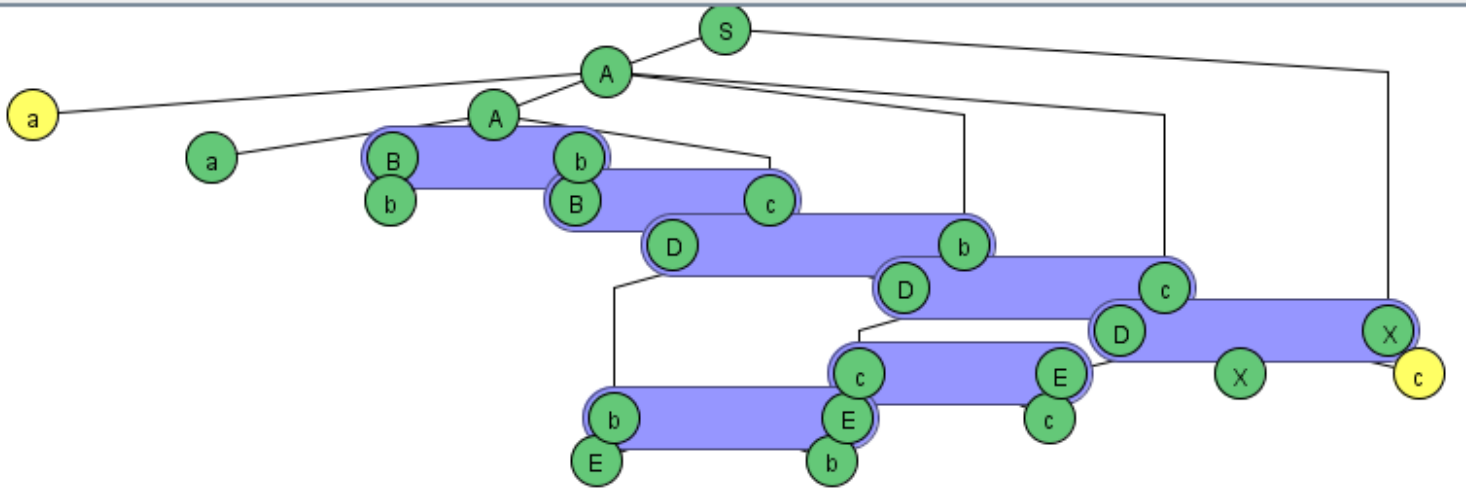
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived Eb from bE.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

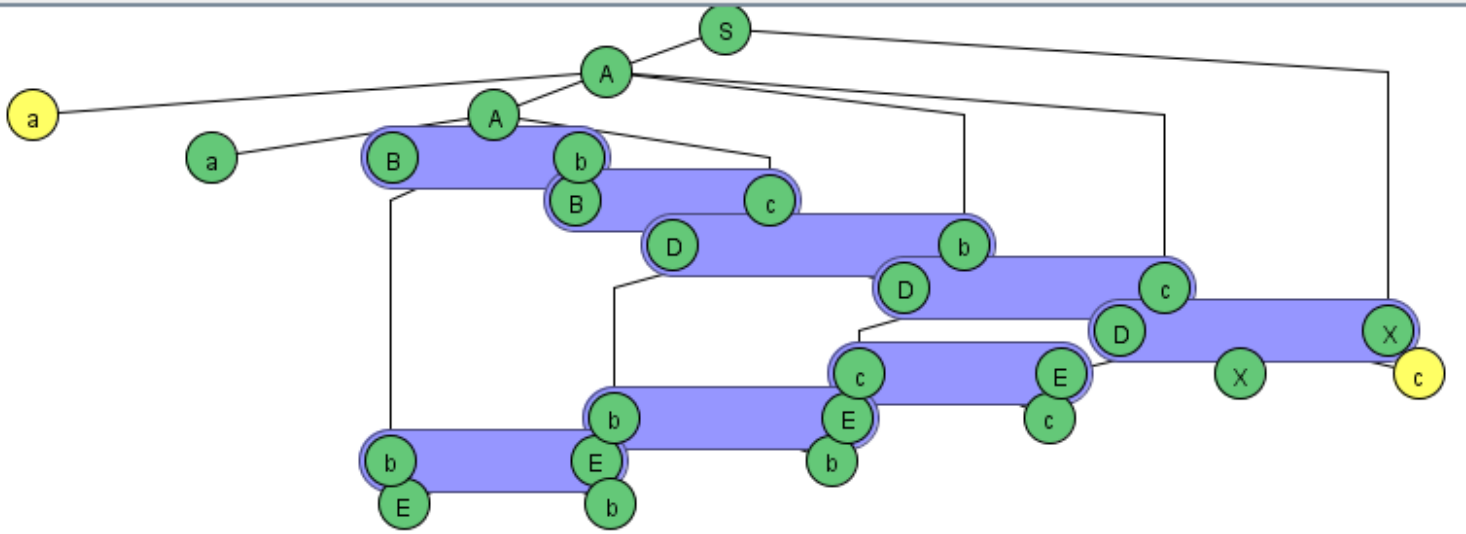
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived Eb from bE.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

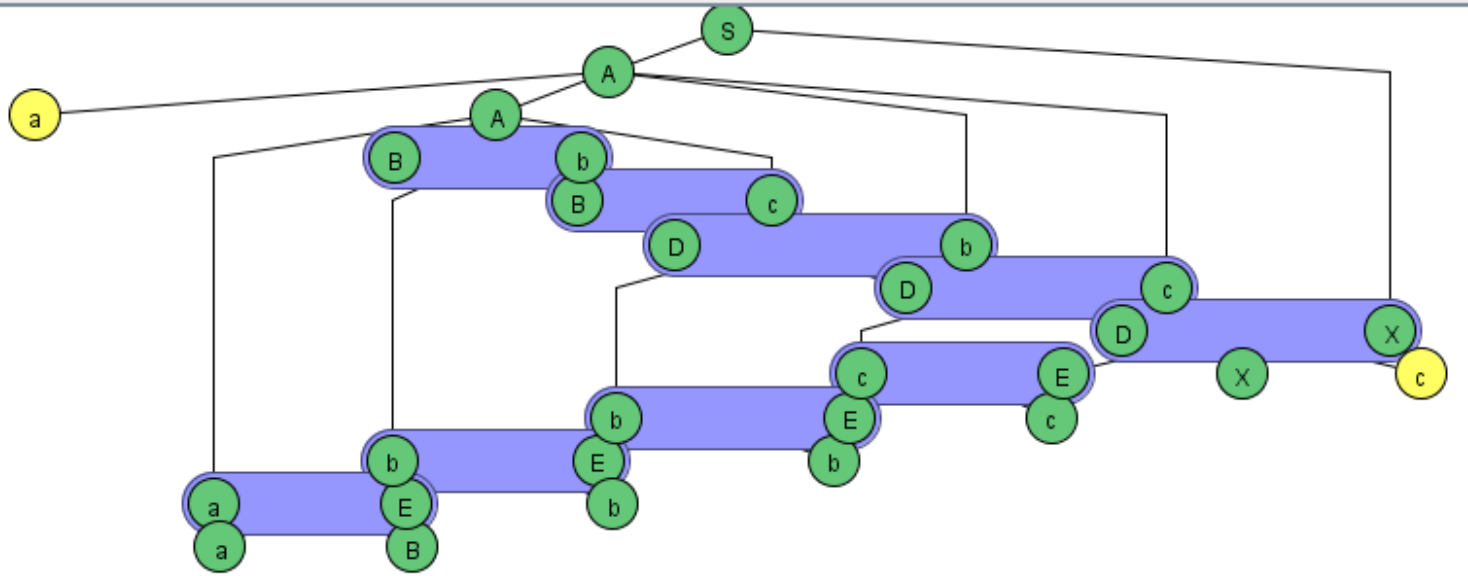
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived aB from aE.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

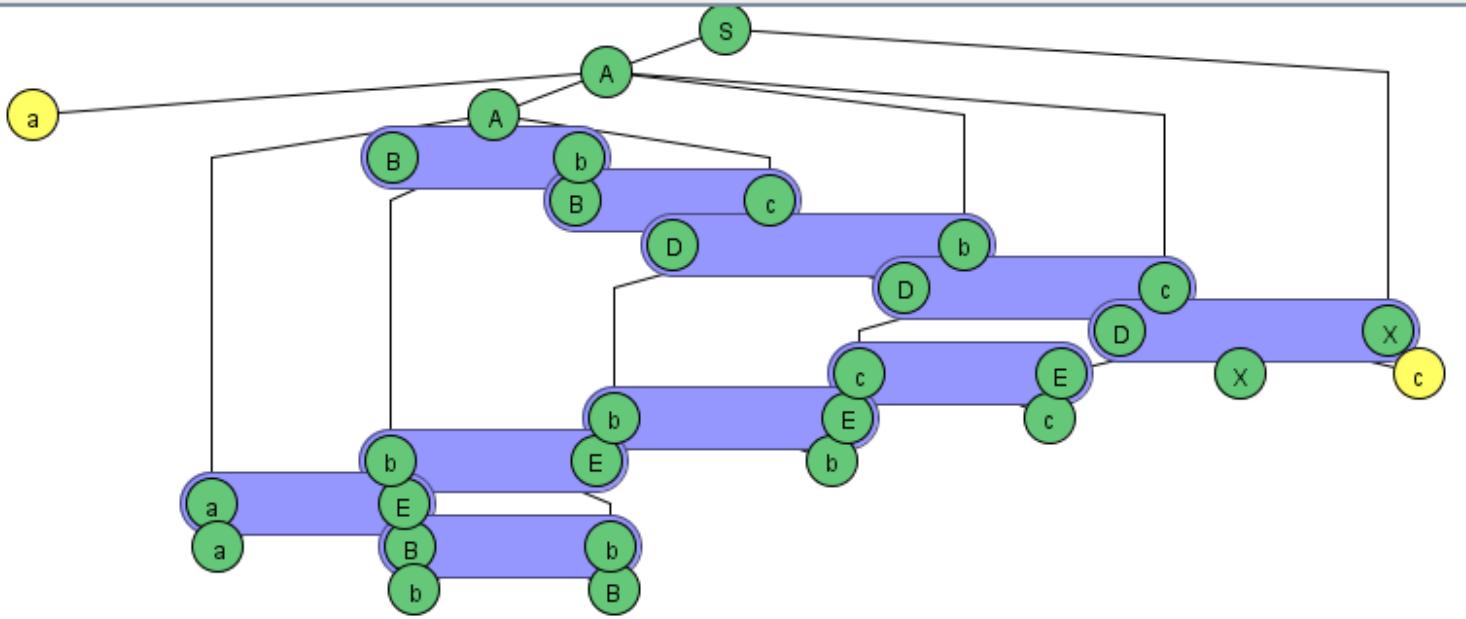
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived bB from Bb.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

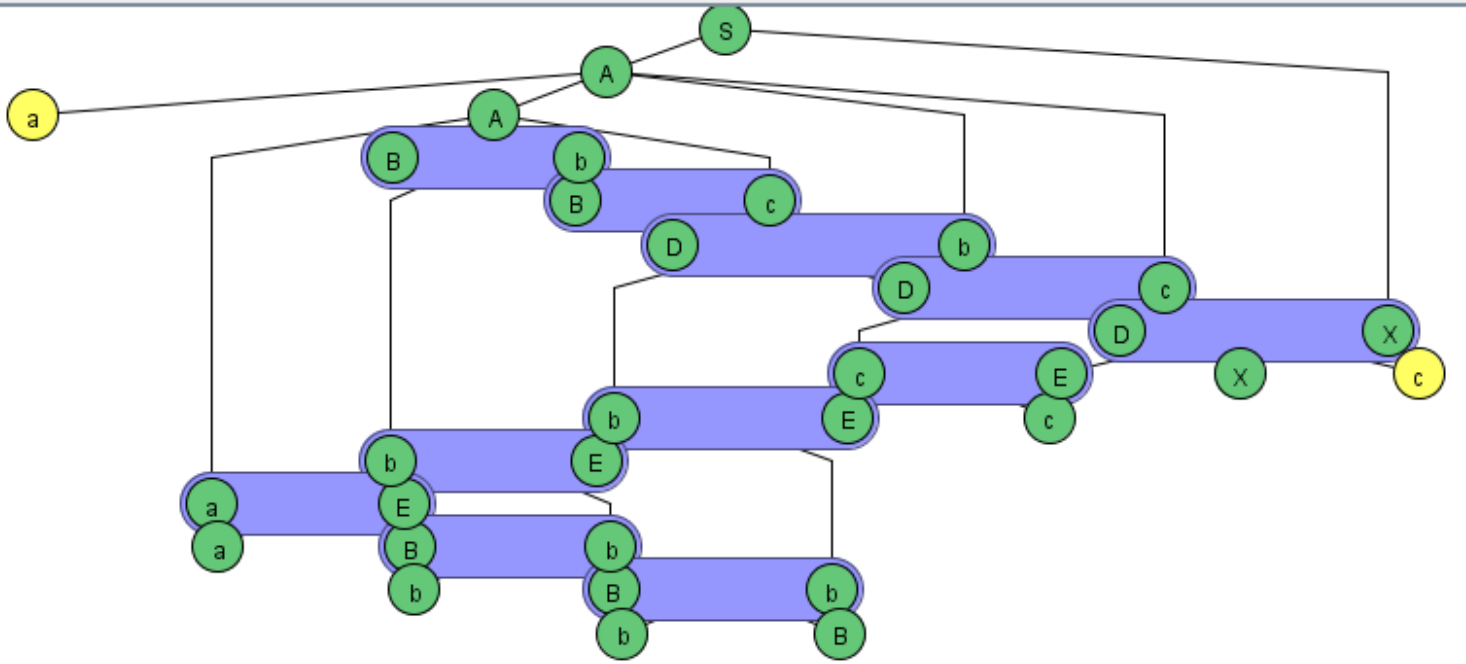
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived bB from Bb.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

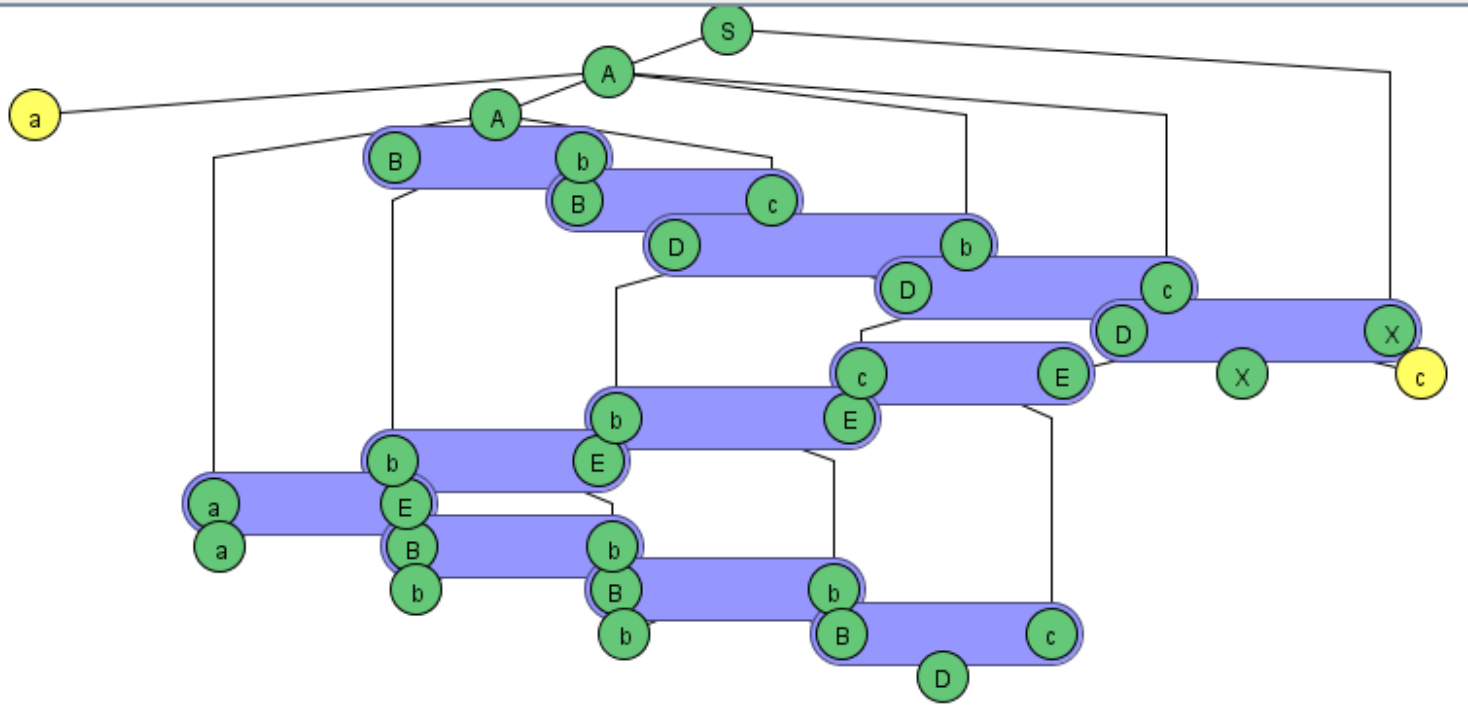
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Absorb second “c”

Derived D from Bc.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

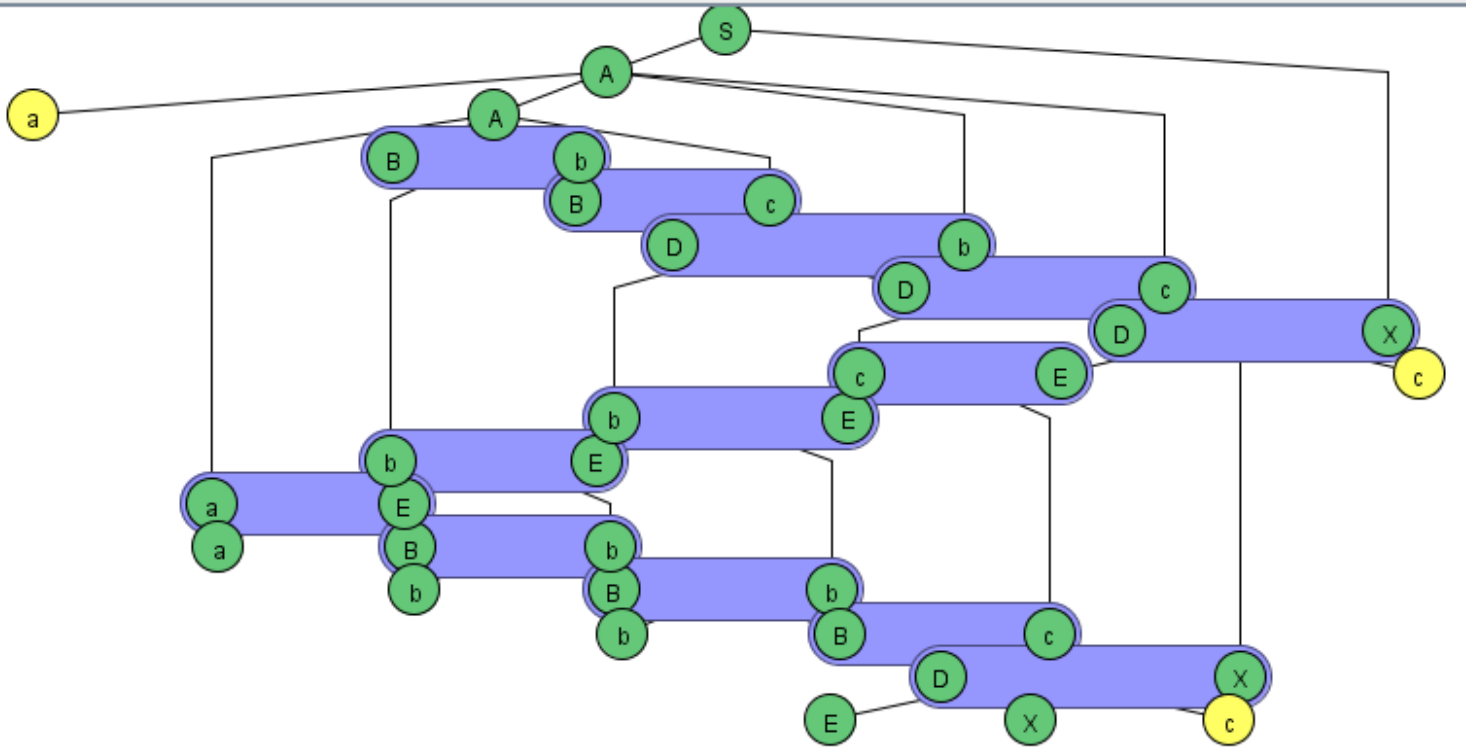
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Spit the “c” out at right end

Derived EXc from DX.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

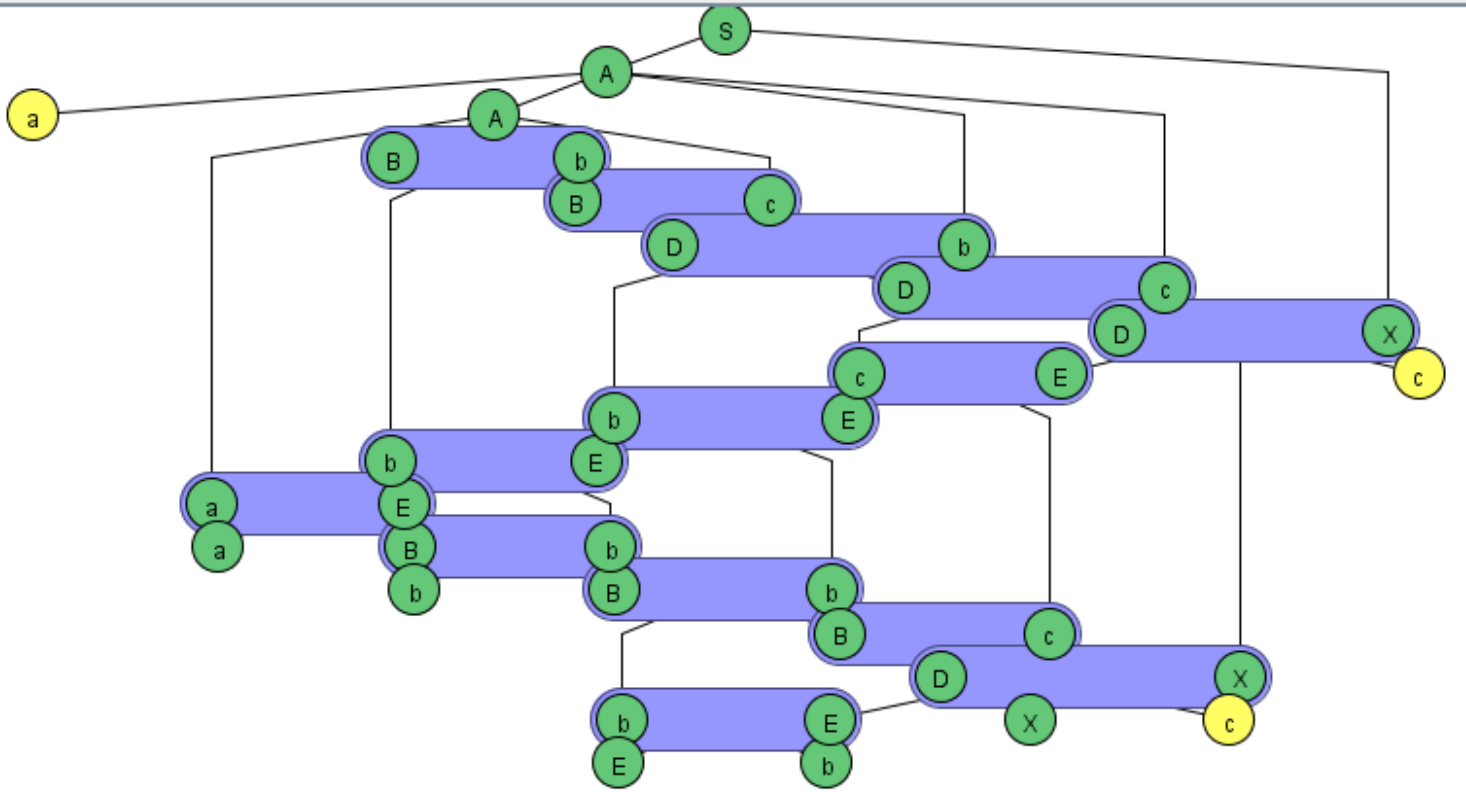
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived Eb from bE.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

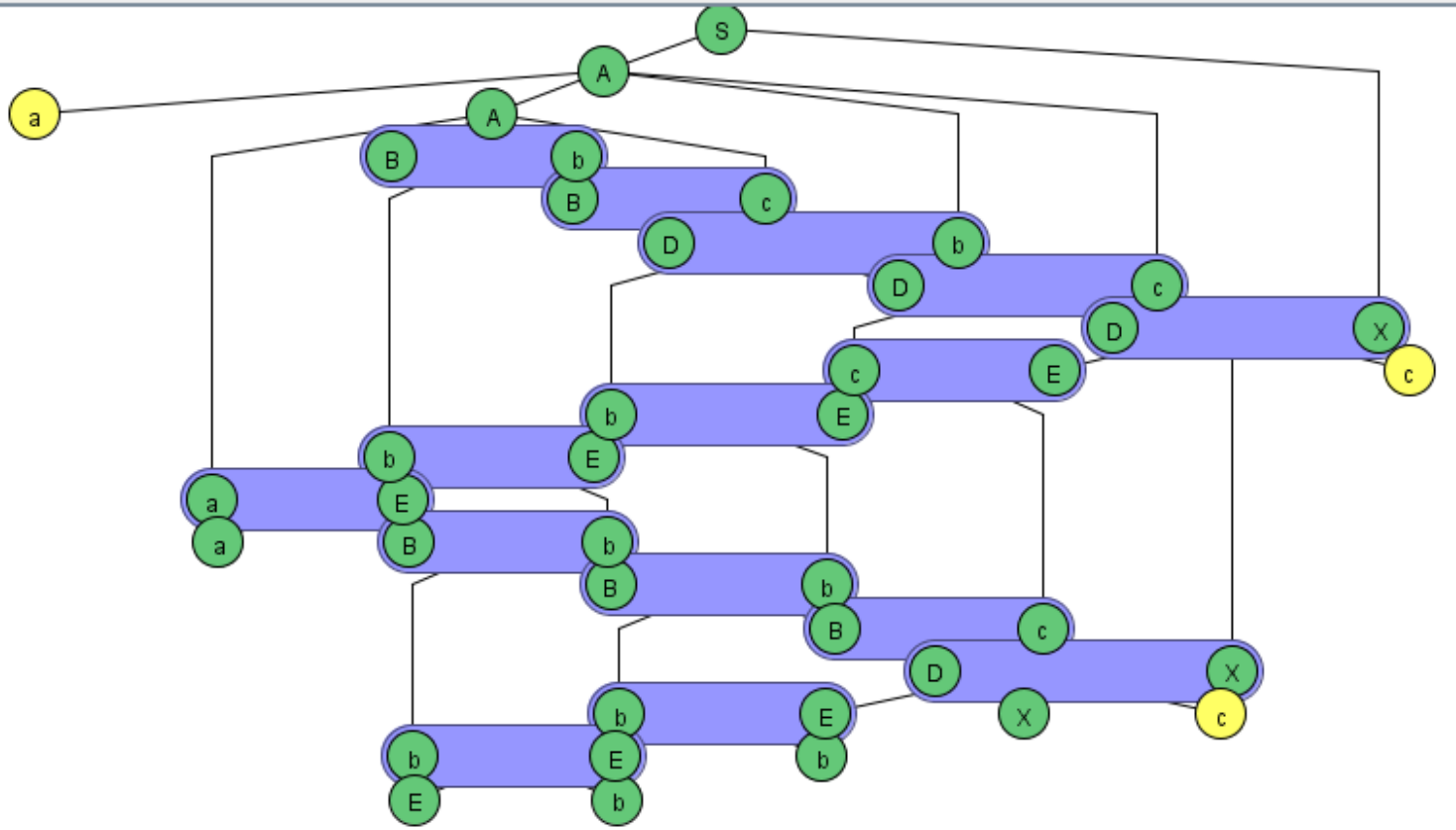
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived Eb from bE.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

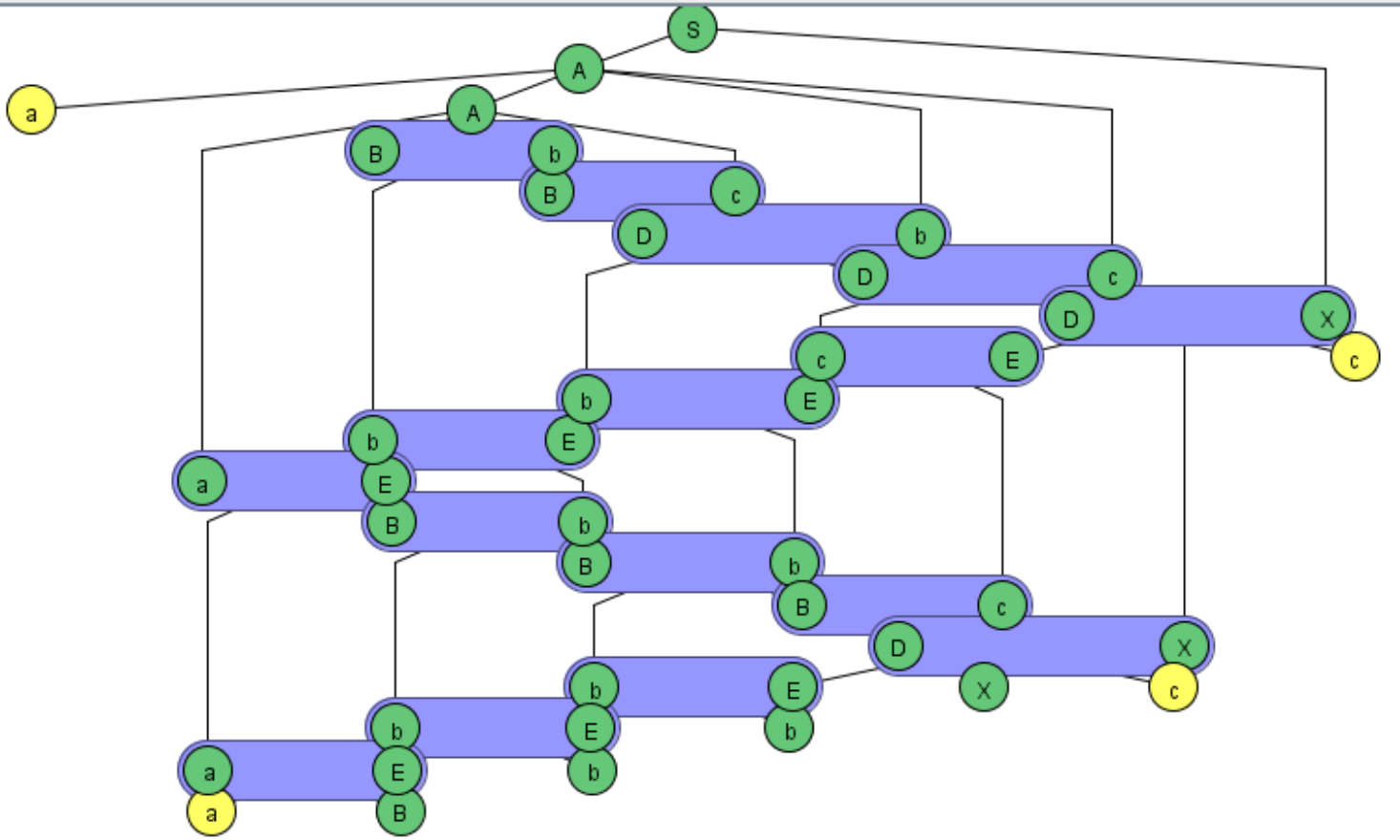
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived aB from aE.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

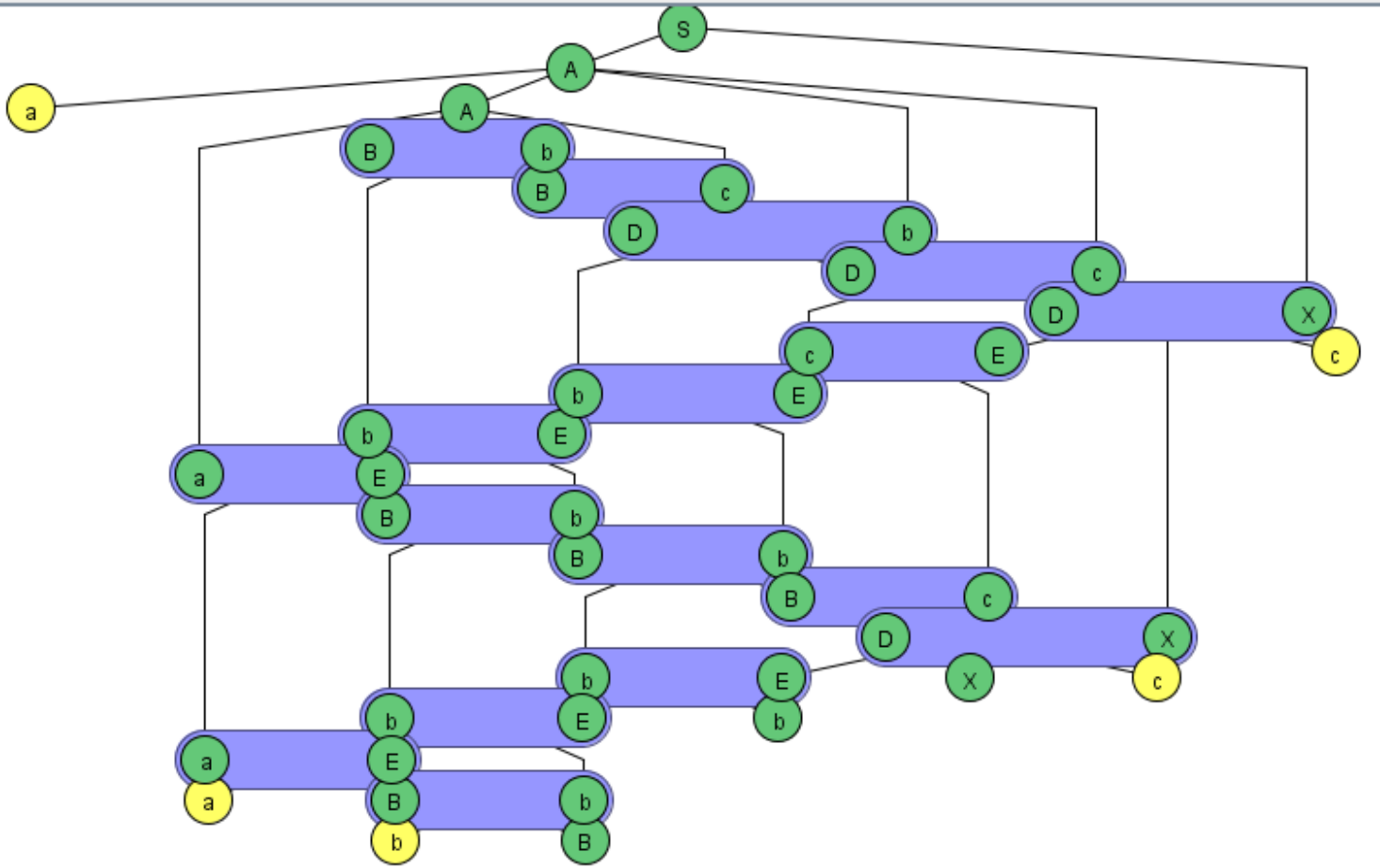
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived bB from Bb.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

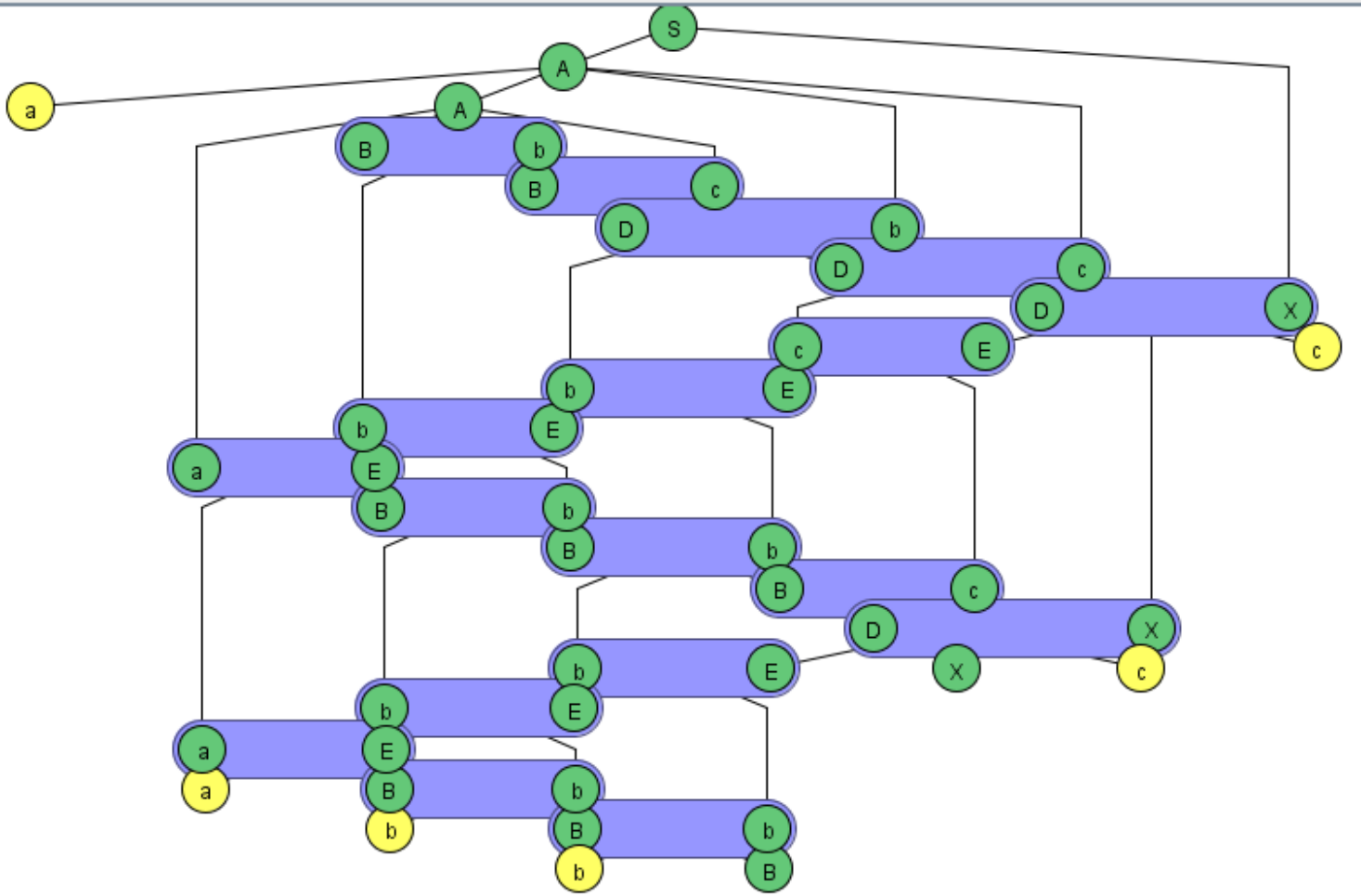
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived bB from Bb.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

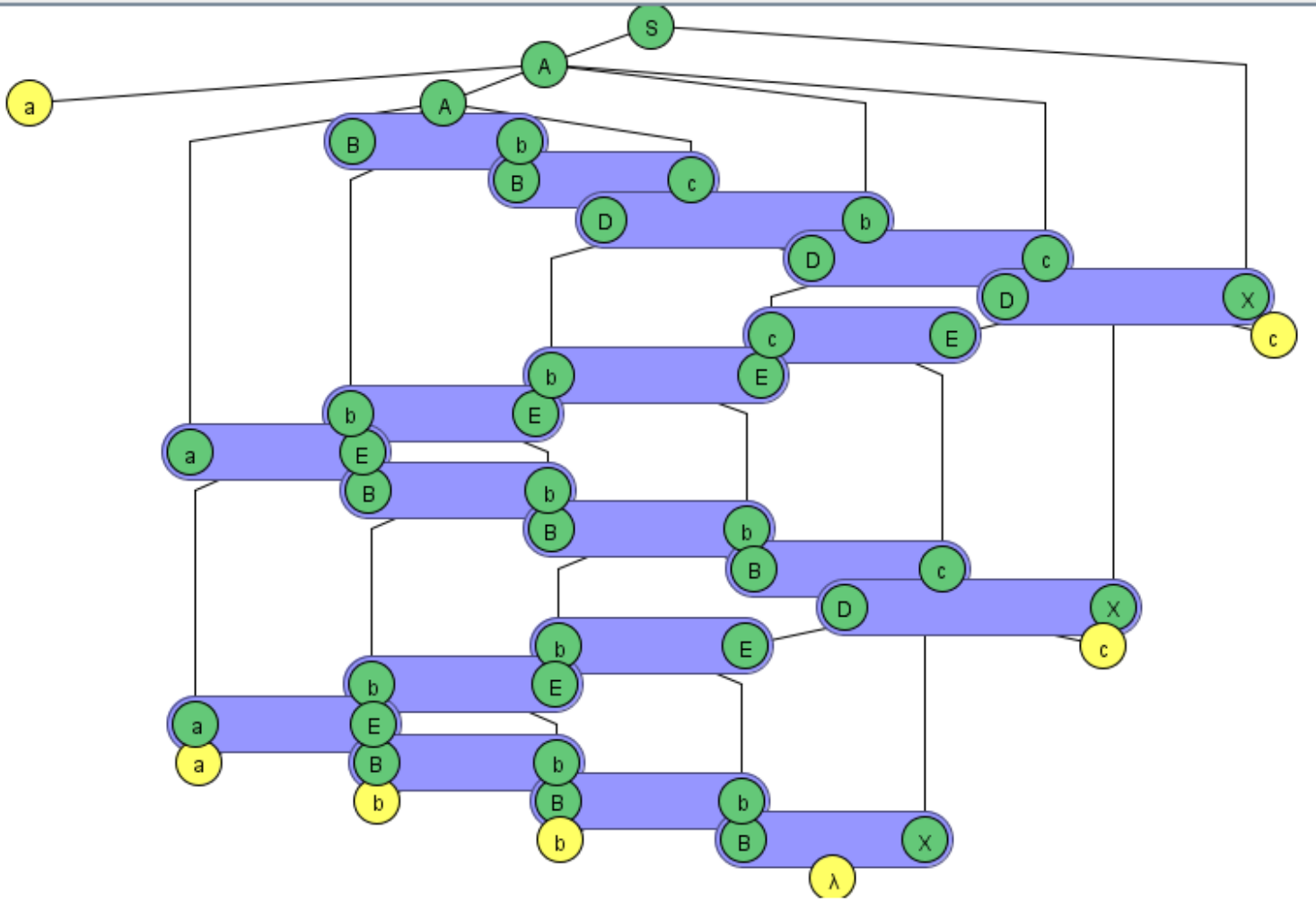
Table Text Size

Start | Pause | Step | Noninverted Tree

Input: aabbcc

String accepted! 51 nodes generated.

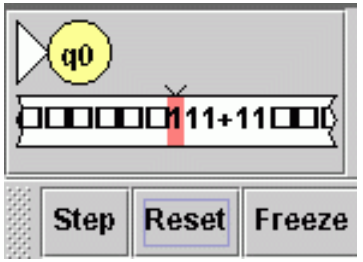
LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB



Derived λ from BX. Derivations complete.

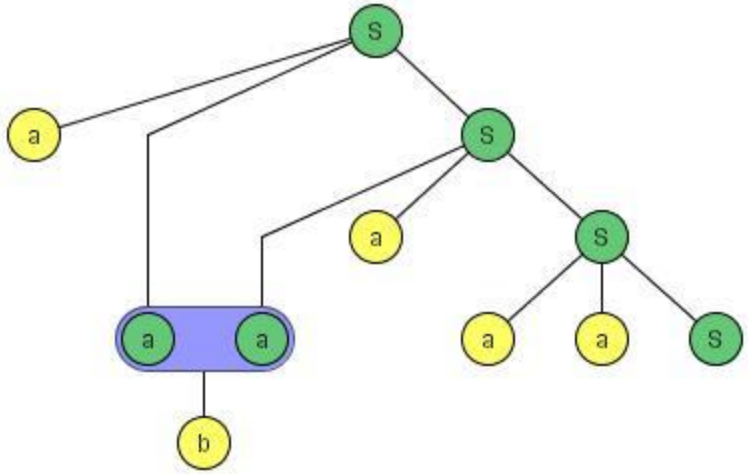
What else can JFLAP do?

- Create other machines
  - Moore and Mealy
  - Pushdown Automaton
  - Turing machine



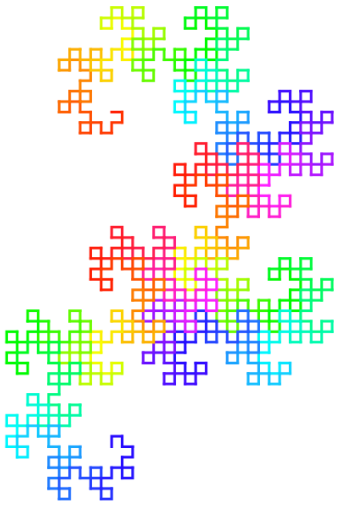
- Parsing of grammars
  - regular, context-free grammars
  - Unrestricted grammar

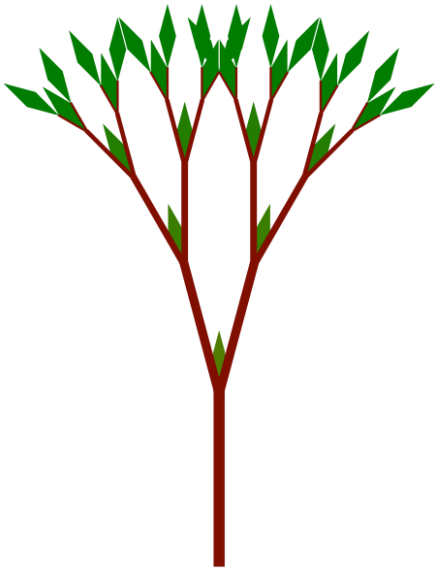
- Conversions for proofs
  - NFA to DFA to minimal DFA
  - NFA ←→ regular expression
  - NFA ←→ regular grammar
  - CFG ←→ NPDA

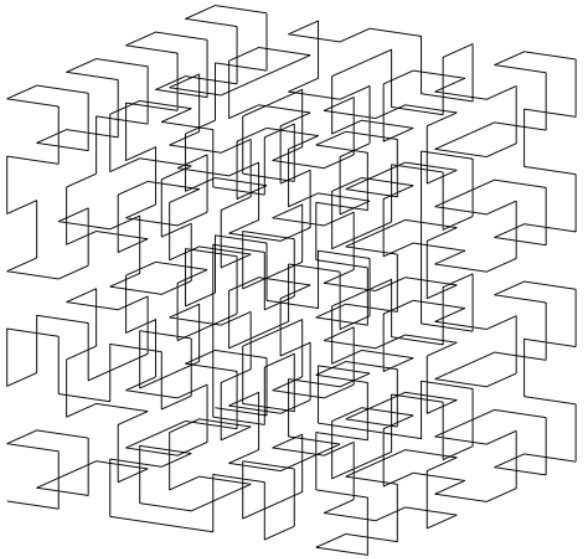


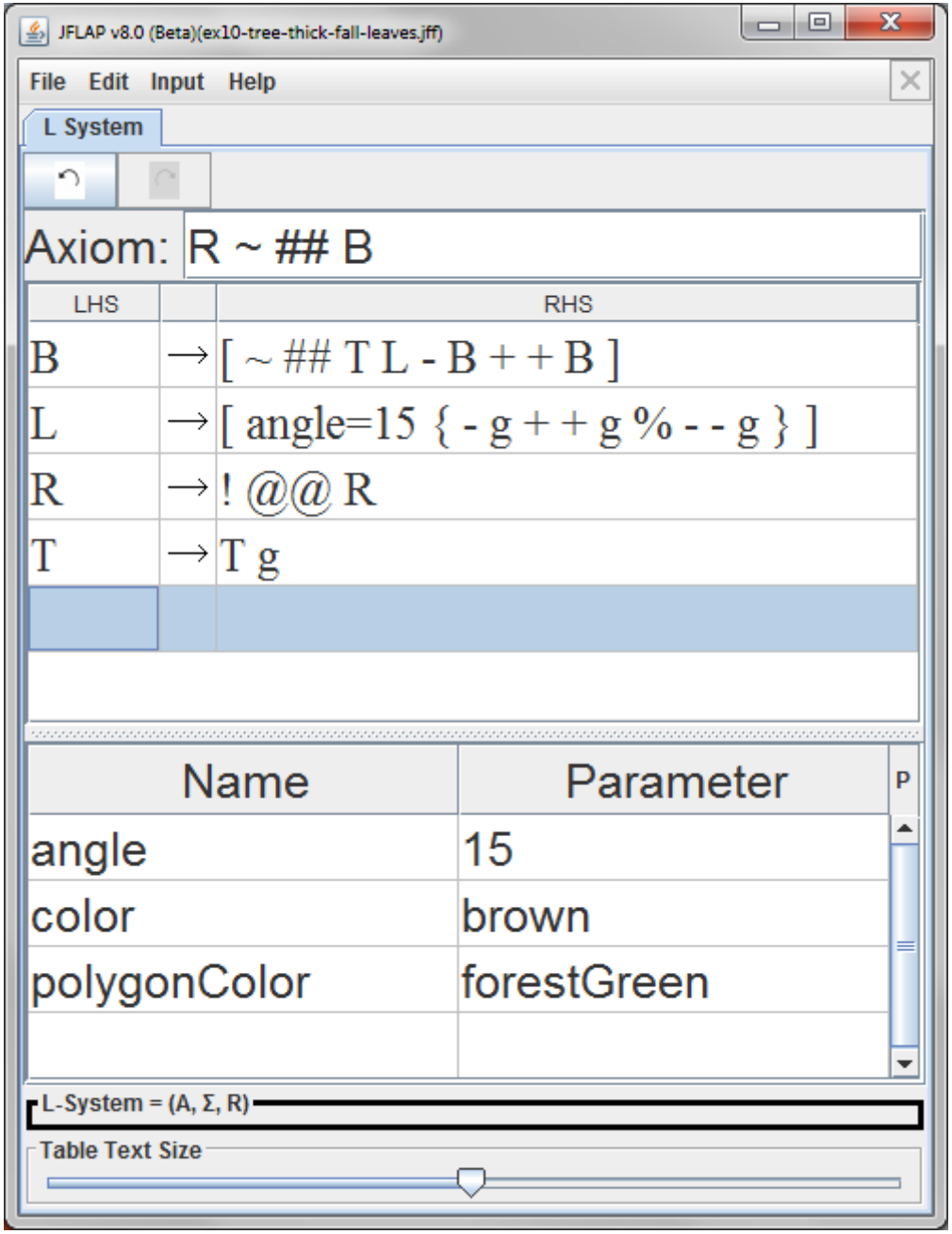
JFLAP - L-Systems

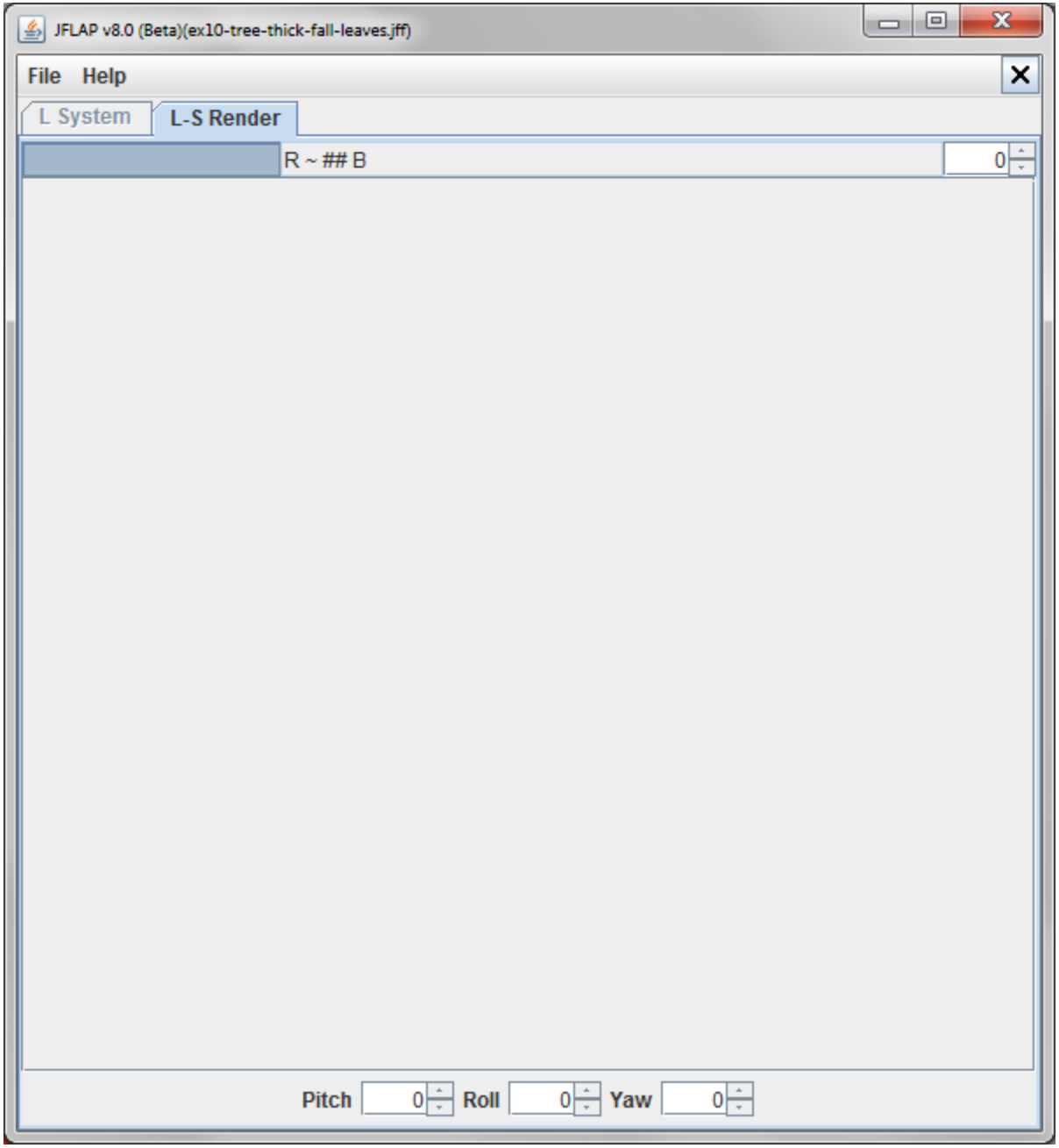
- L-Systems may be used to model biological systems and create fractals.
- Similar to Chomsky grammars, except *all* variables are replaced in each derivation step, not just one!
- Commonly, strings from successive derivations are interpreted as strings of render commands and are displayed graphically.

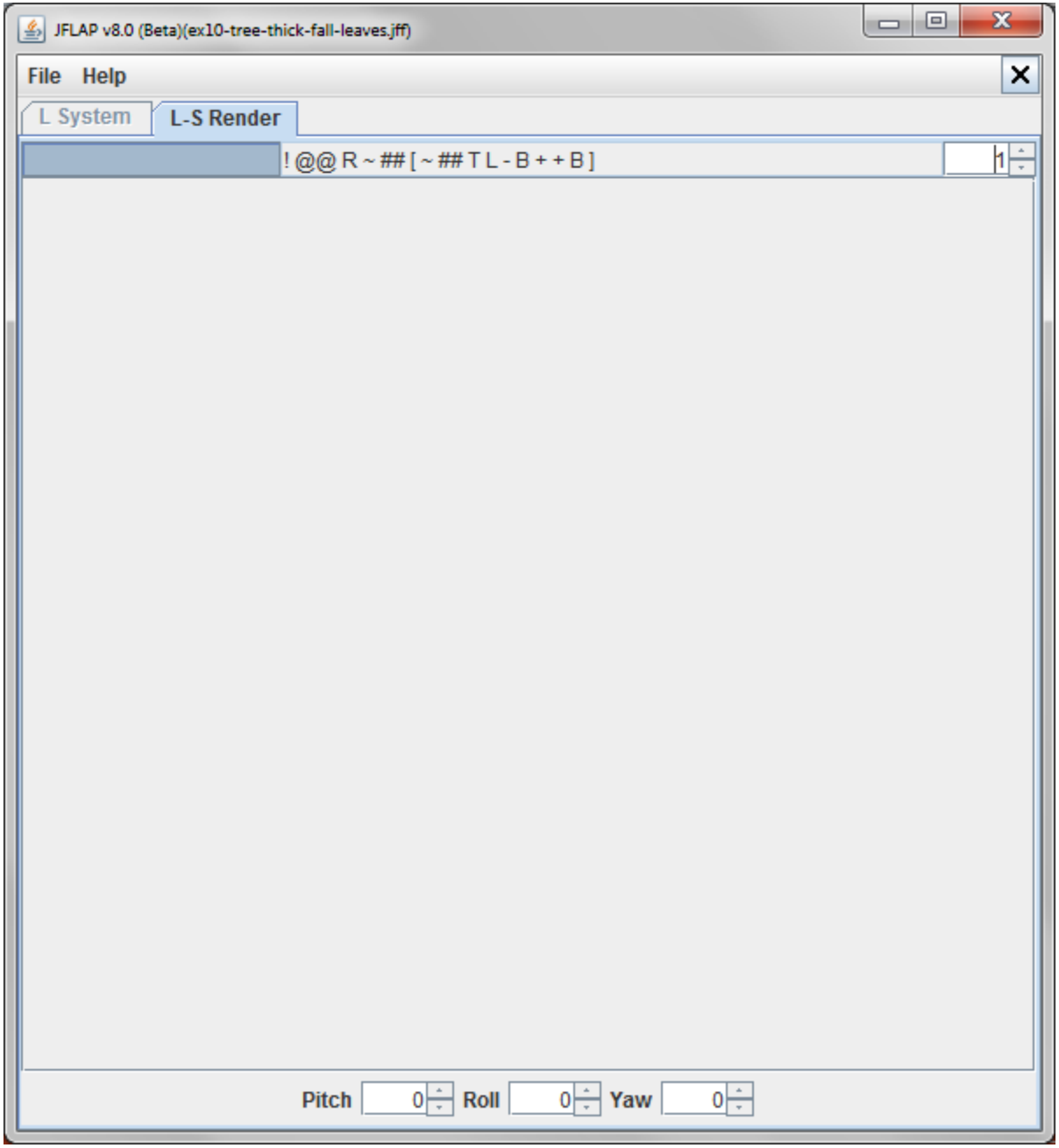


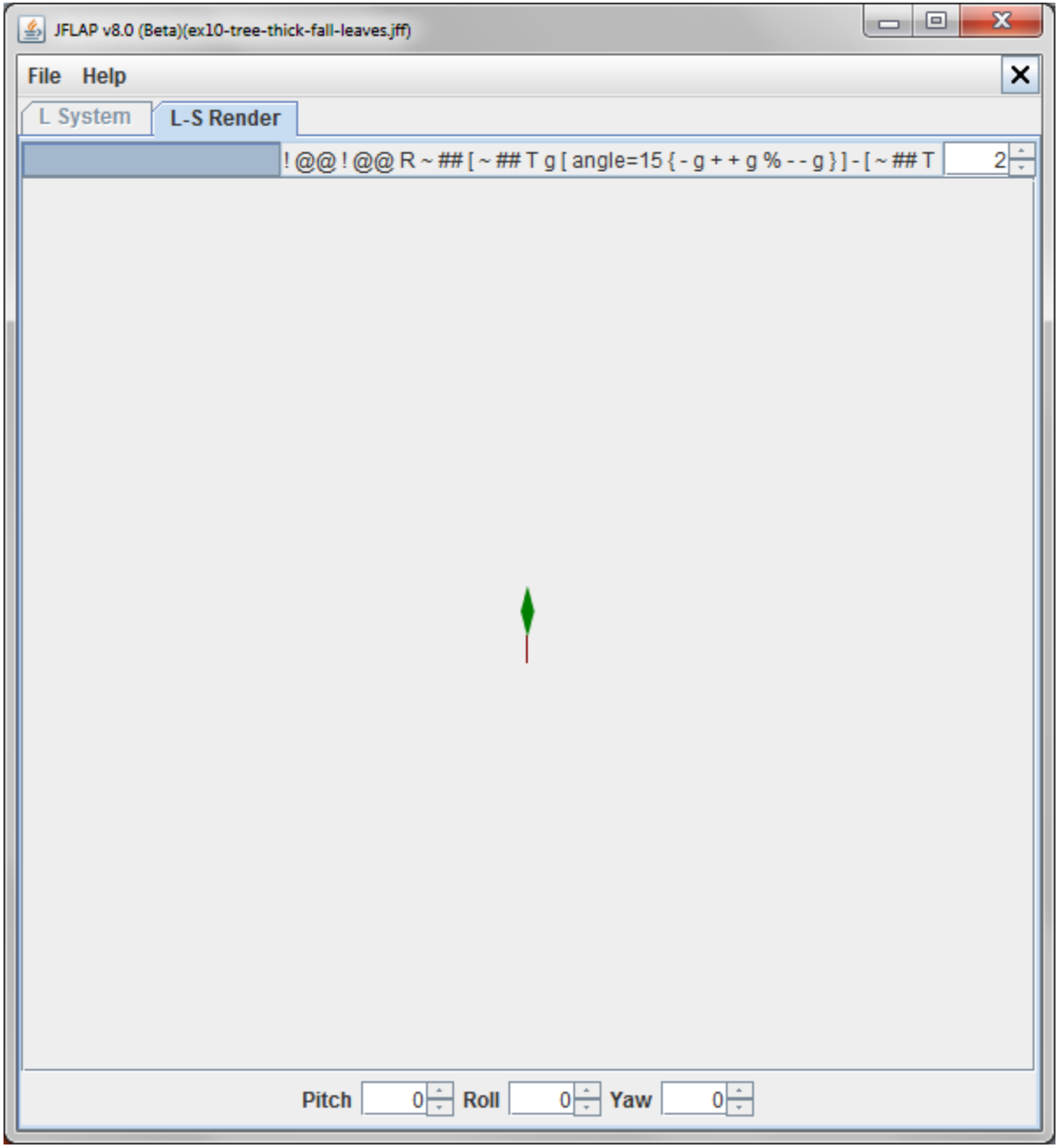


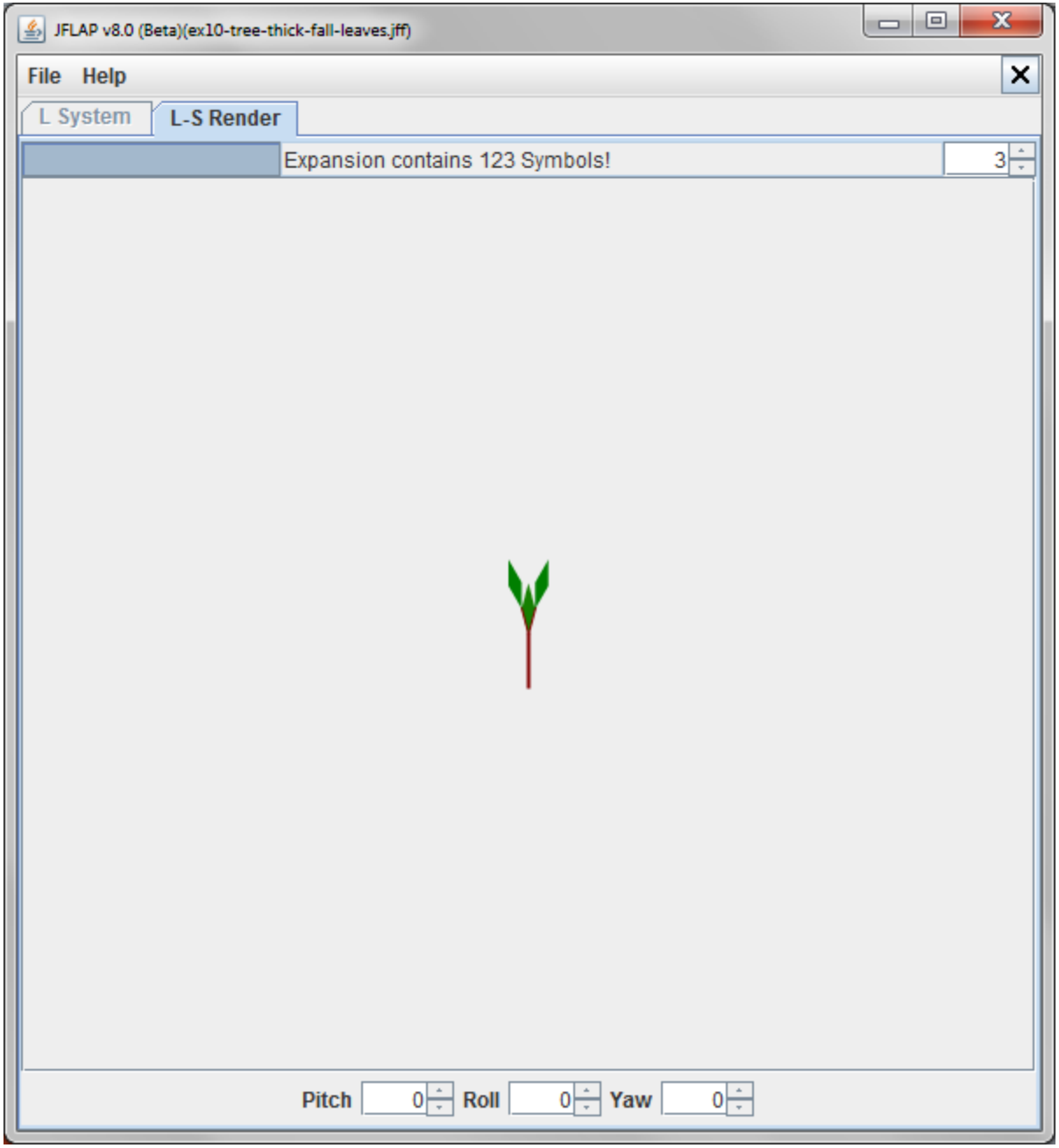


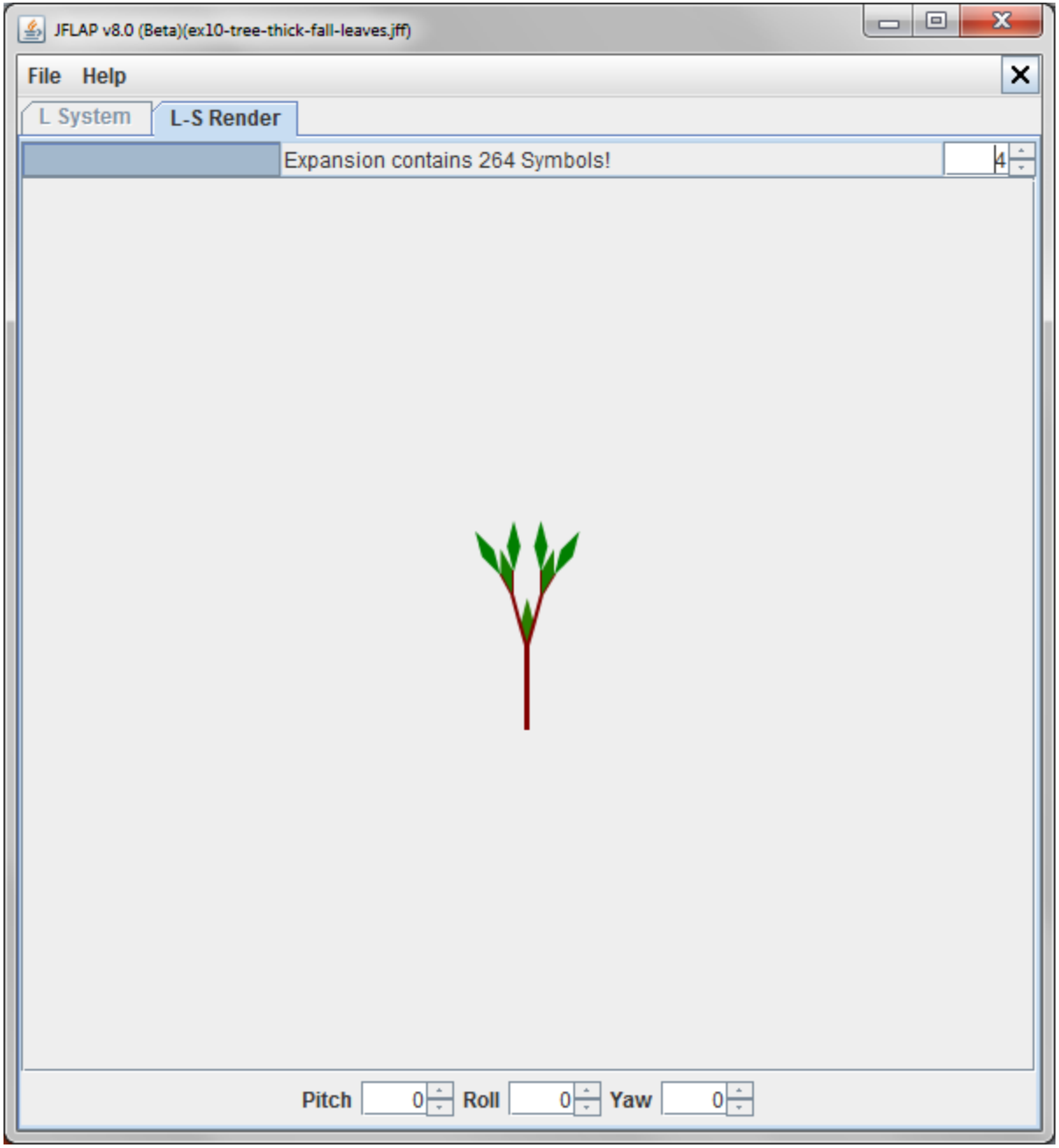

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File Edit Input Help
L System
Axiom: R ~ ## B
LHS
RHS
B → [~ ## T L - B + + B]
L → [angle=15 { - g + + g % - - g }]
R → ! @@ R
T → T g
Name
Parameter
P
angle
15
color
brown
polygonColor
forestGreen
L-System = (A, Σ, R)
Table Text Size

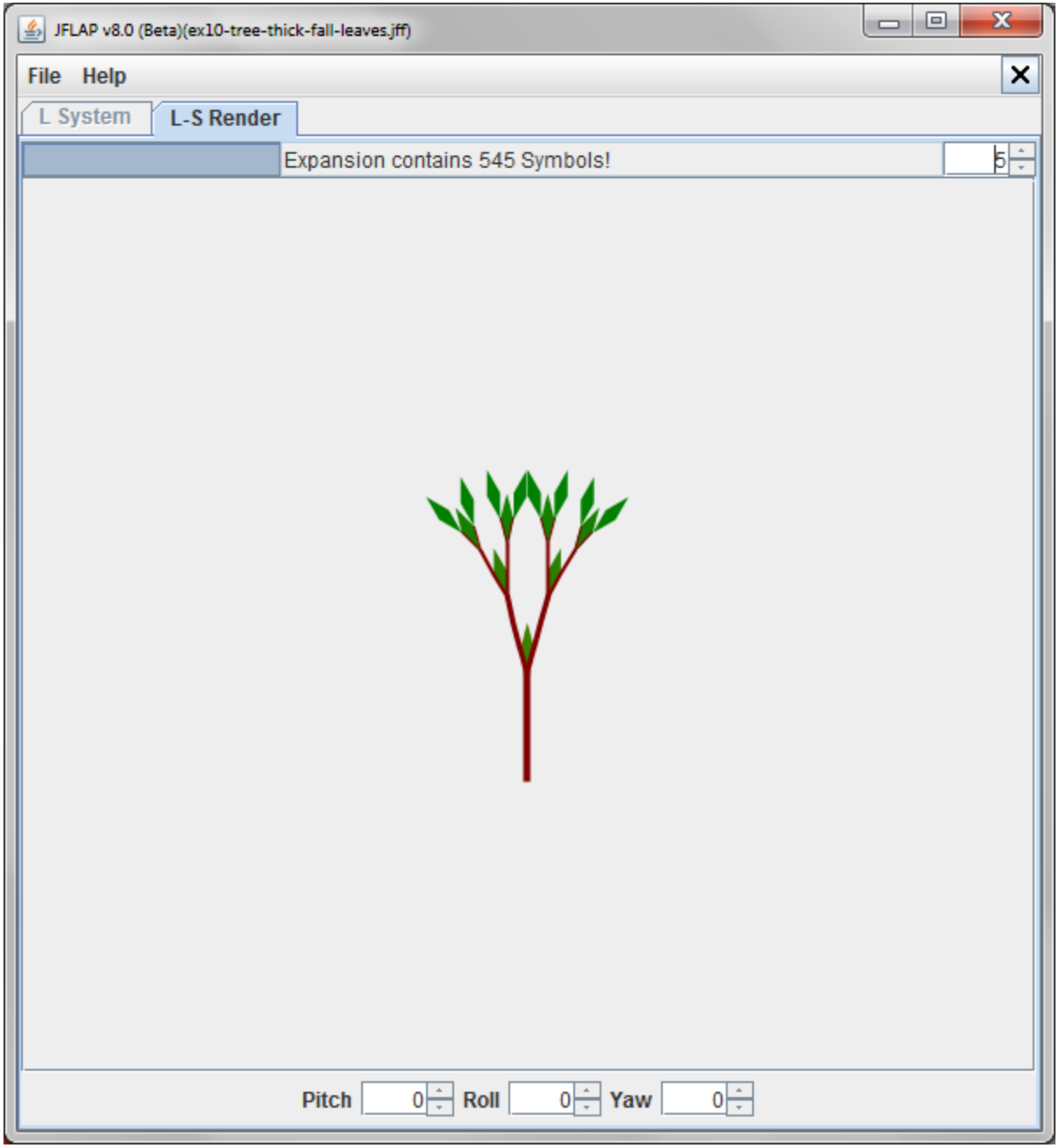

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
R ~ ## B
0
Pitch
0
Roll
0
Yaw
0

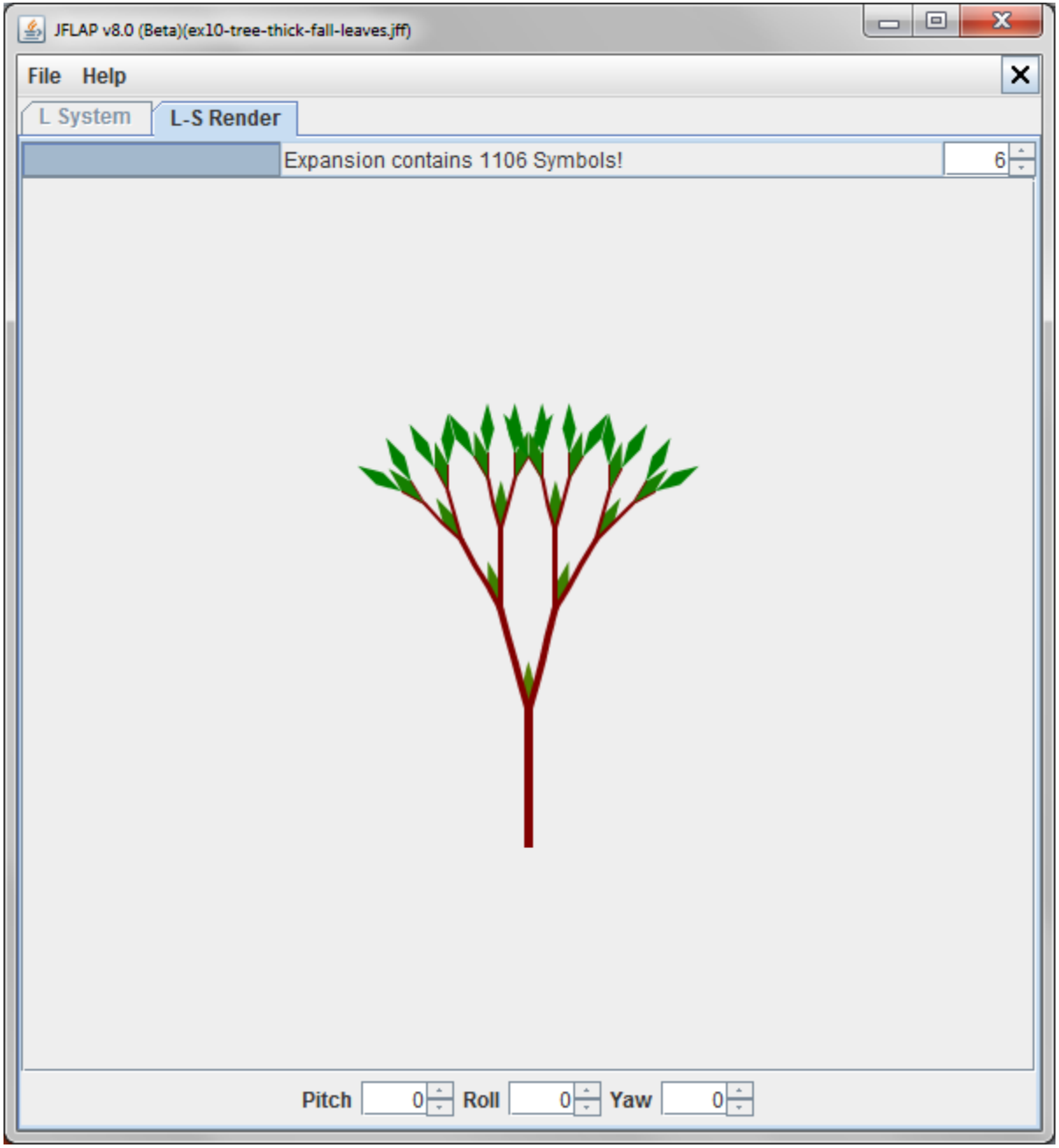

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
! @@ R ~ ## [~ ## T L - B + + B]
1
Pitch
0
Roll
0
Yaw
0

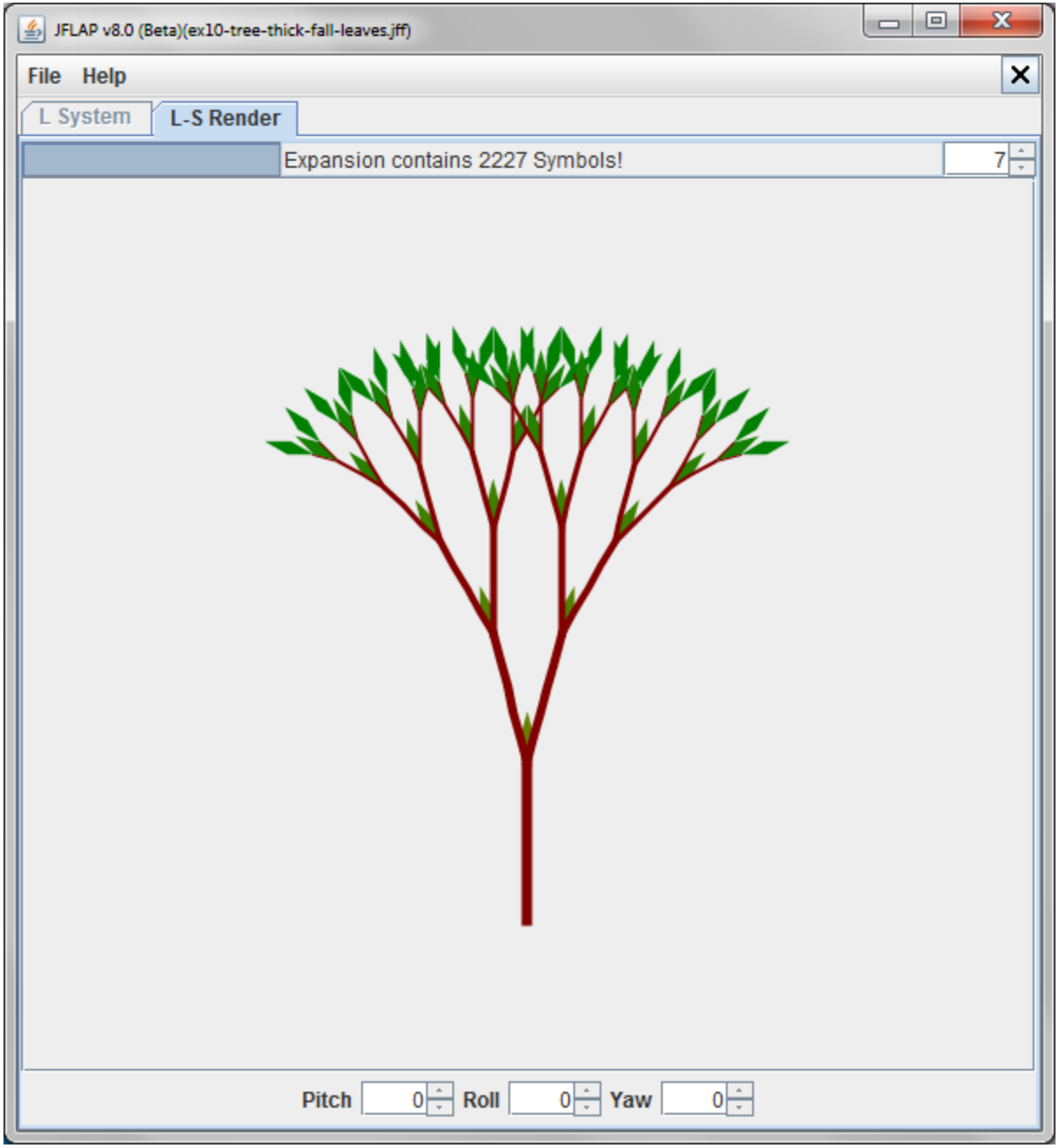

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
! @@ ! @@ R ~ ## [~ ## T g [angle=15 { - g + + g % - - g }] - [~ ## T
2
Pitch
0
Roll
0
Yaw
0


JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 123 Symbols!
3
Pitch
0
Roll
0
Yaw
0

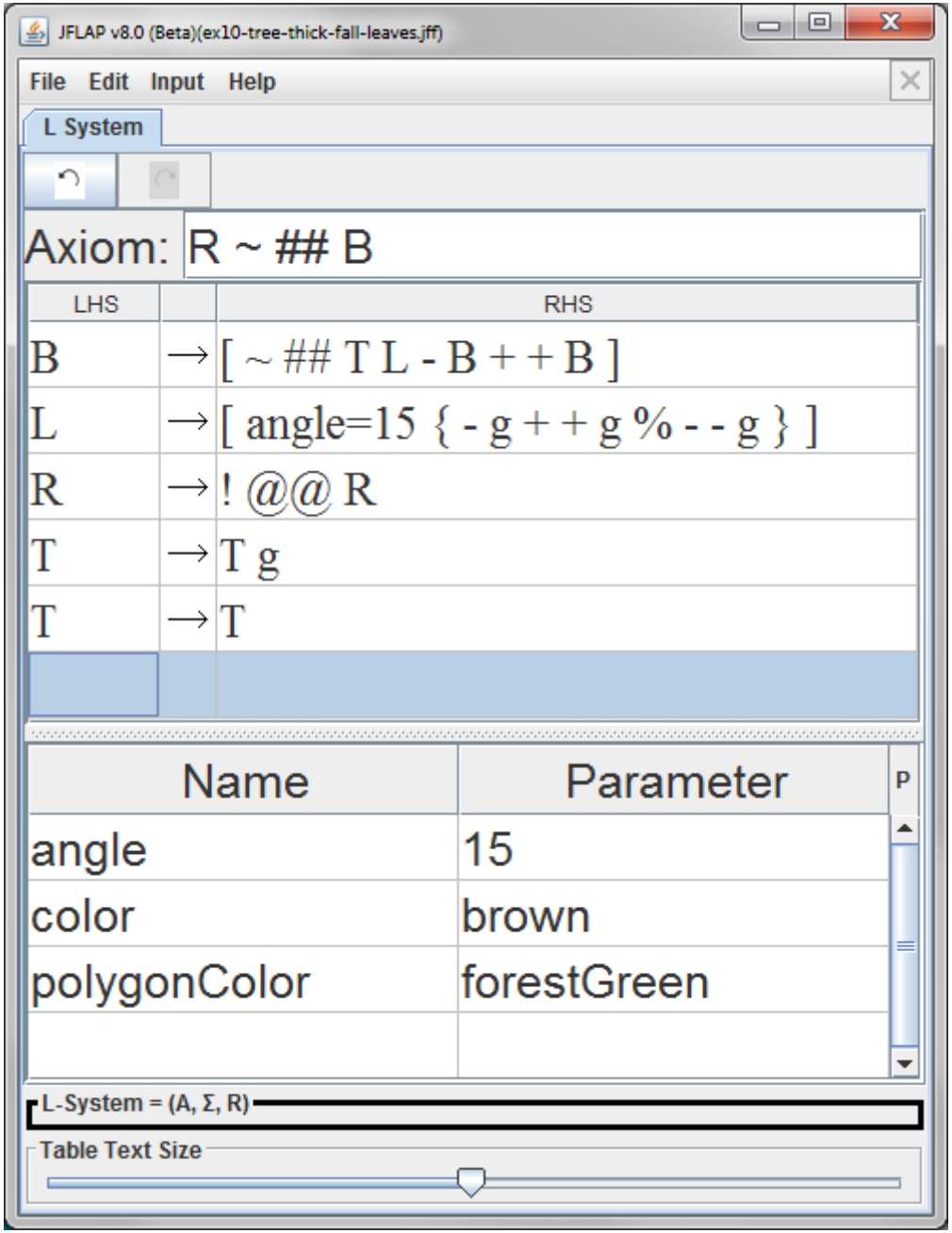

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 264 Symbols!
4
Pitch
0
Roll
0
Yaw
0

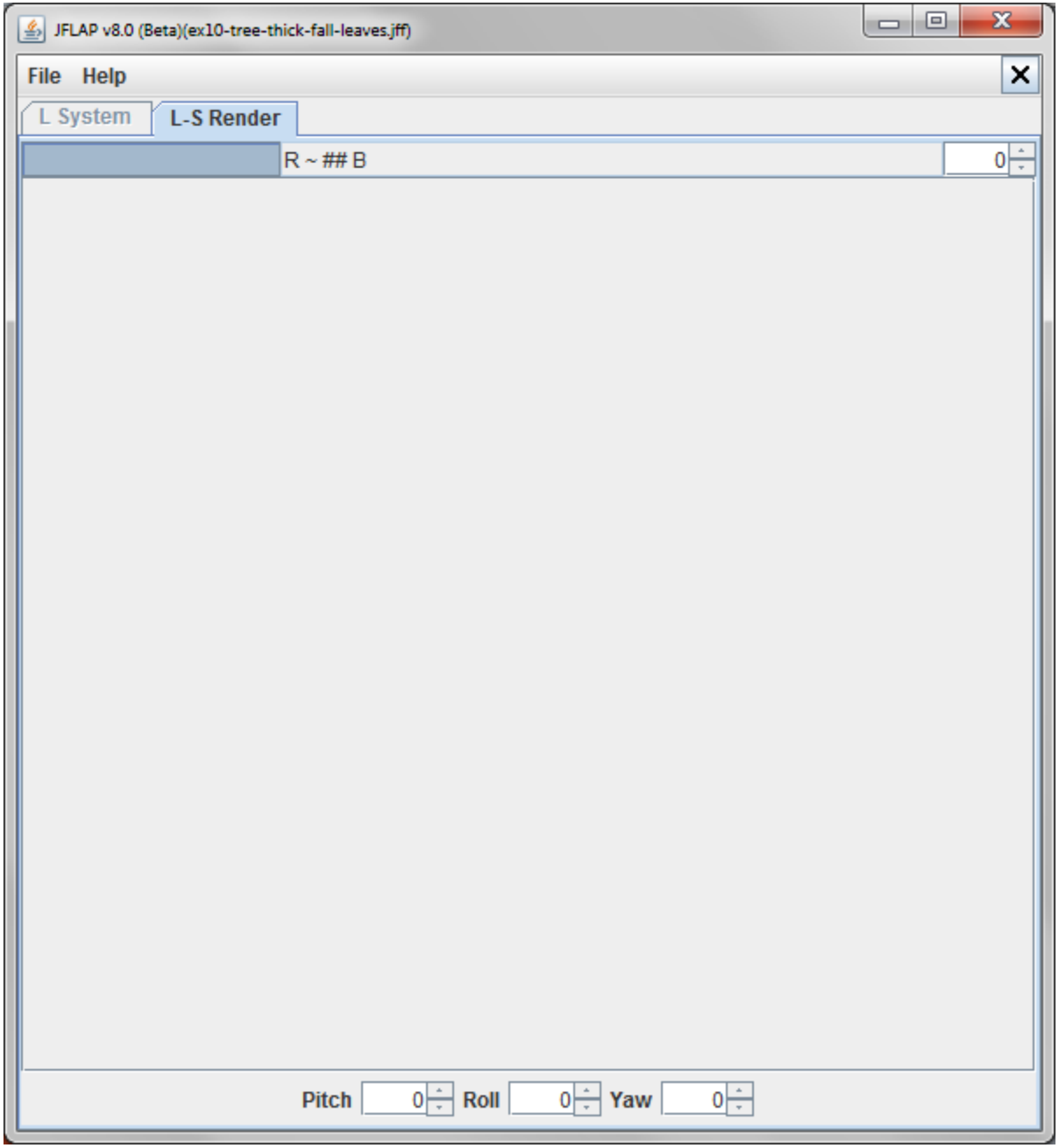

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 545 Symbols!
5
Pitch
0
Roll
0
Yaw
0

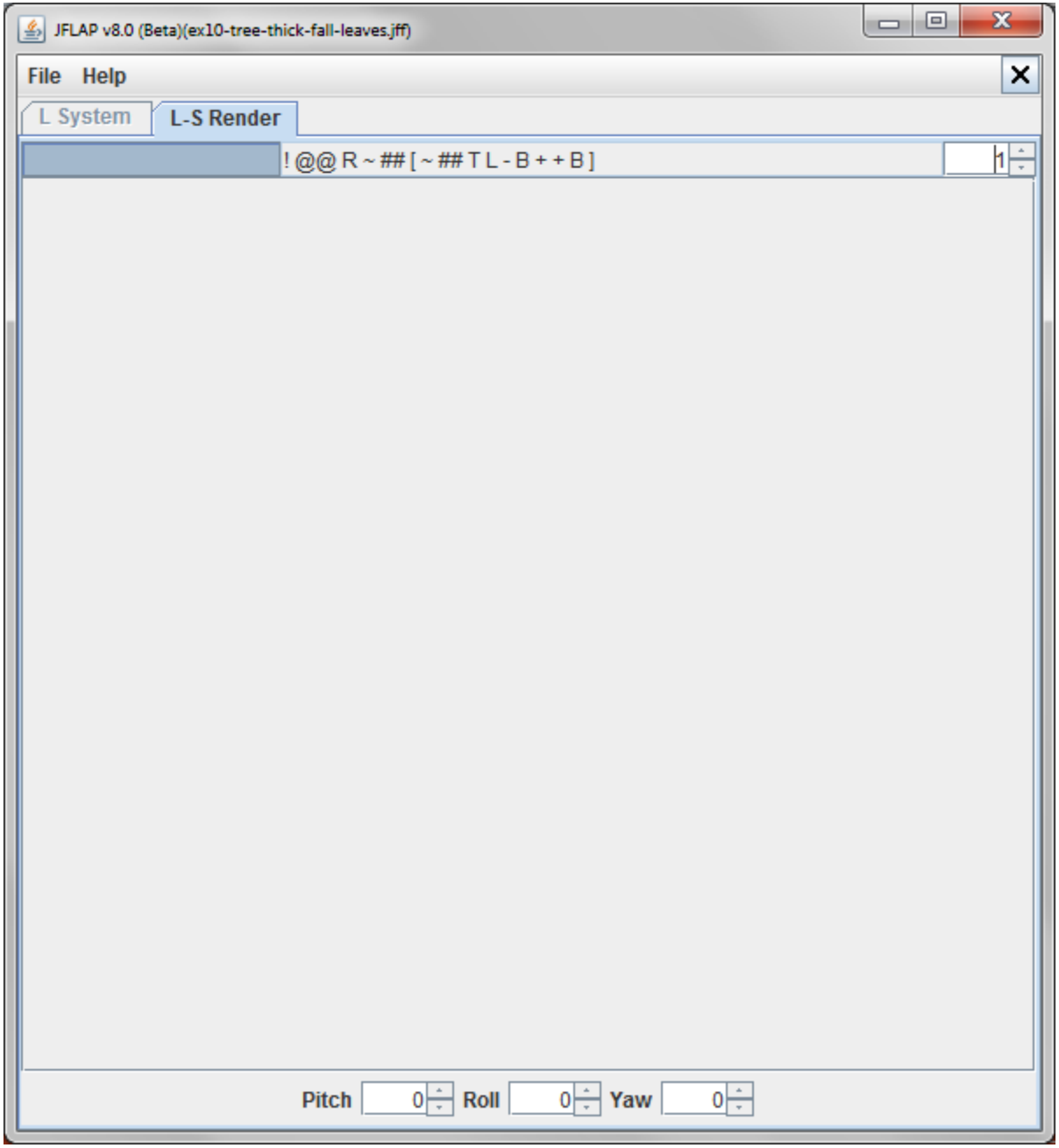

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 1106 Symbols!
6
Pitch
0
Roll
0
Yaw
0

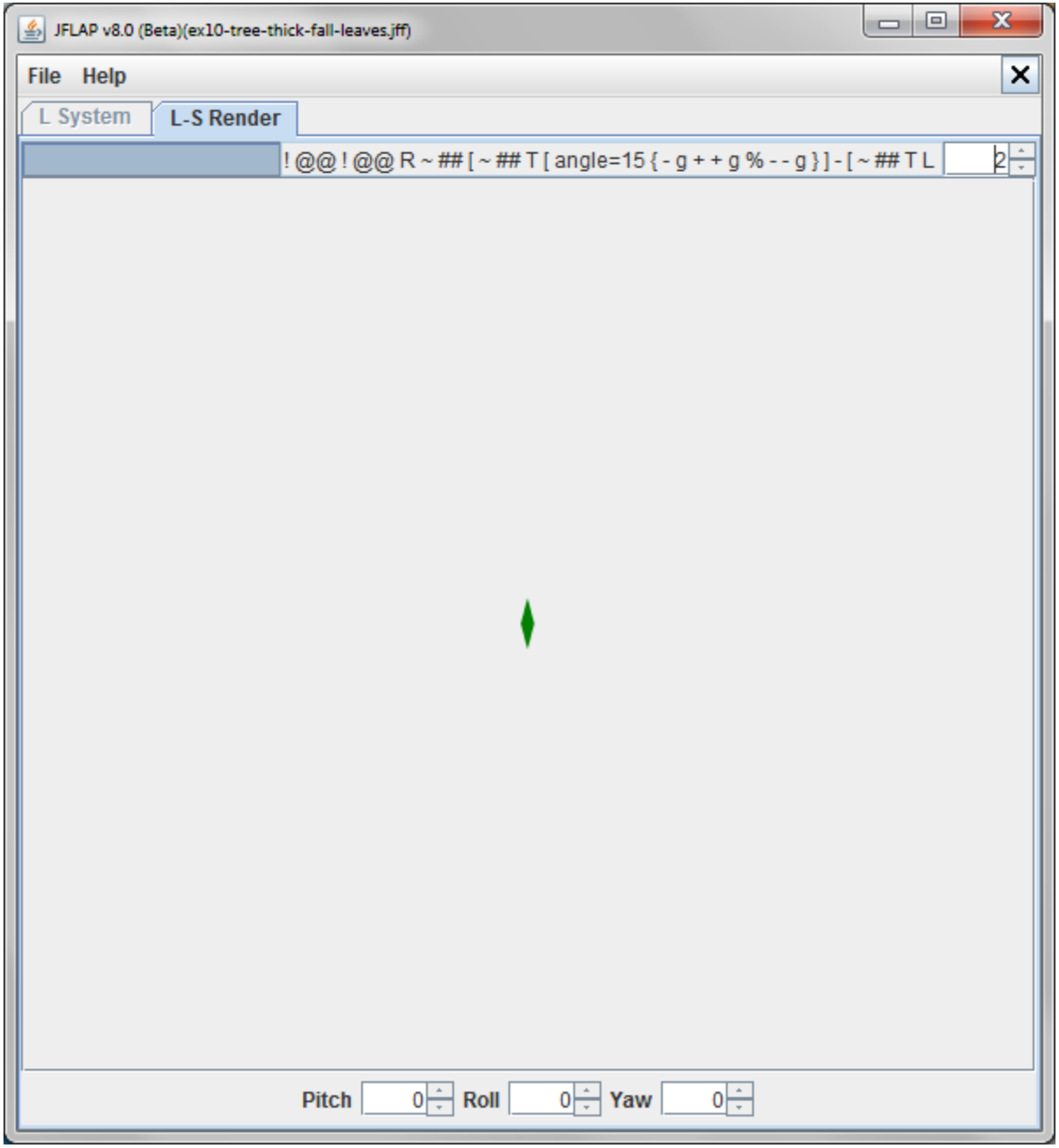

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 2227 Symbols!
7
Pitch
0
Roll
0
Yaw
0

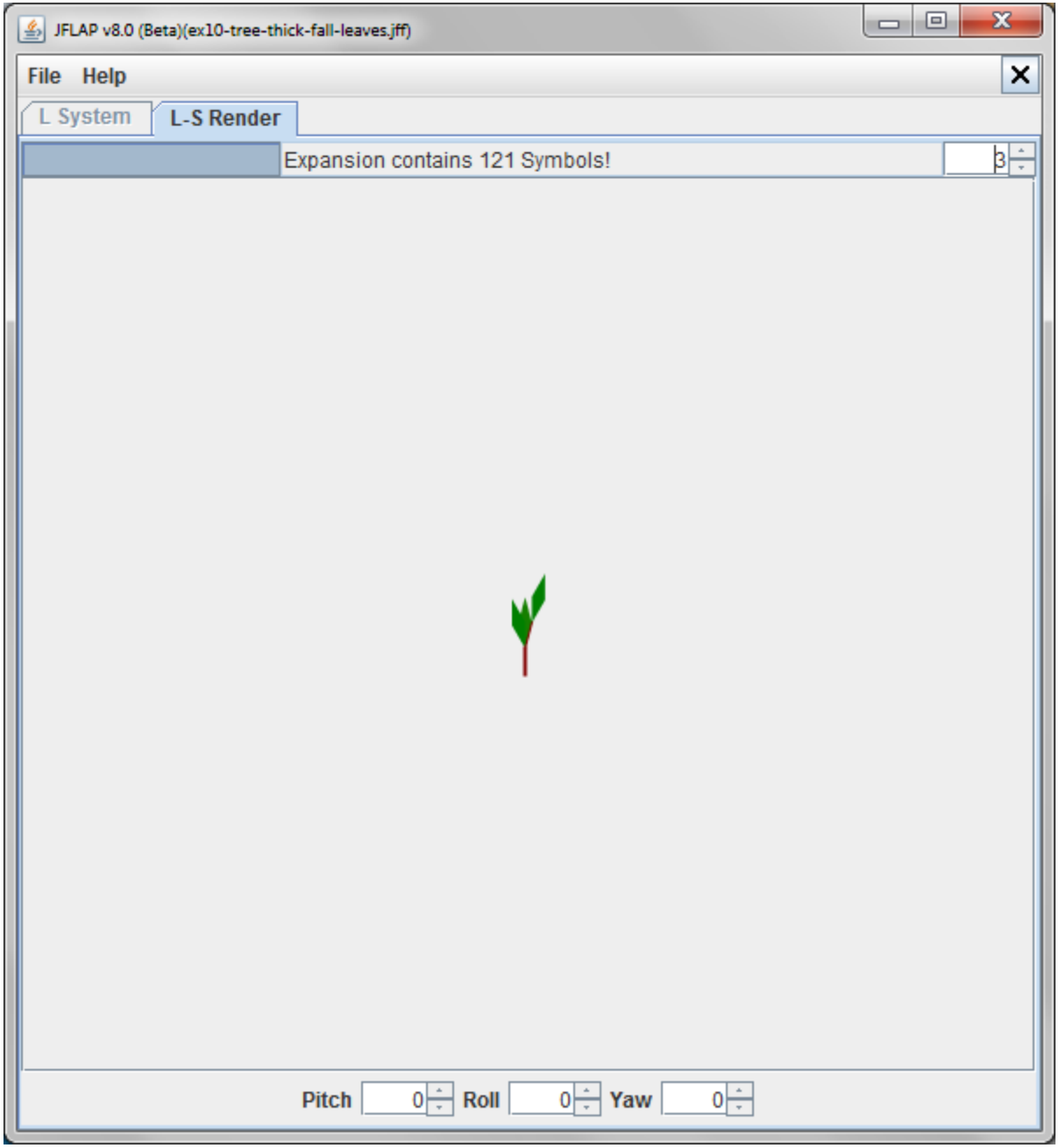
Add second T rule

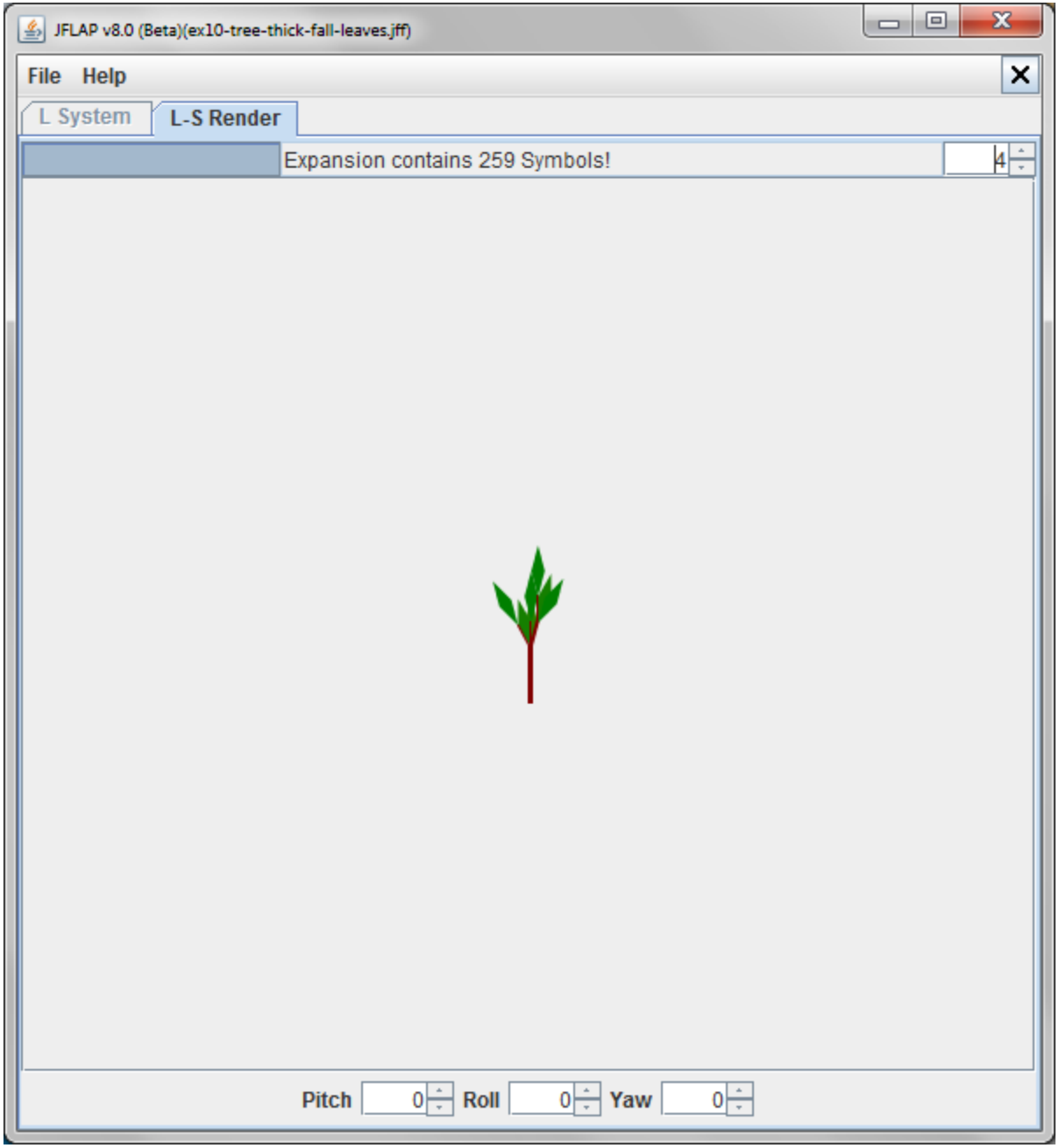


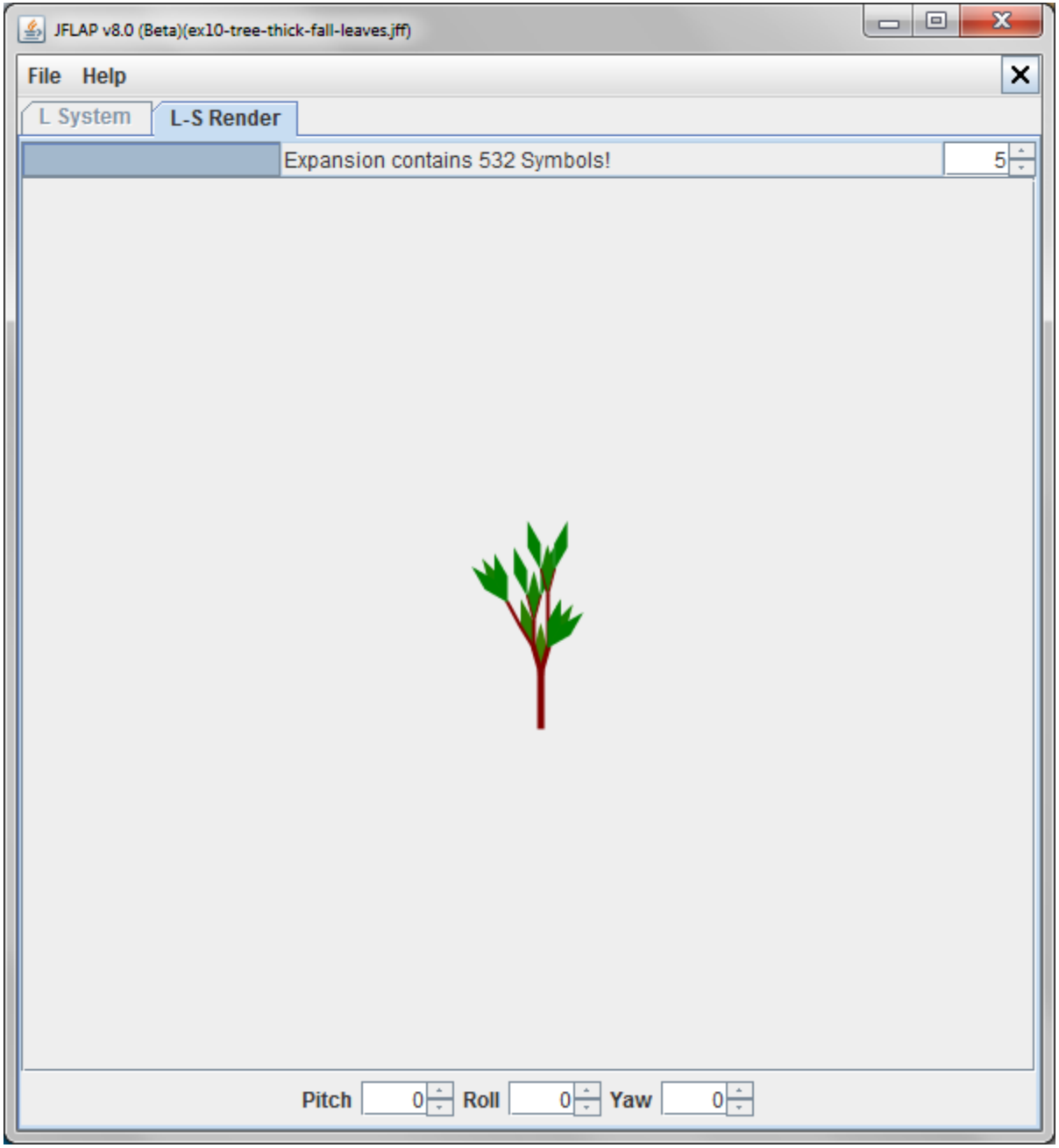

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
R ~ ## B
0
Pitch
0
Roll
0
Yaw
0

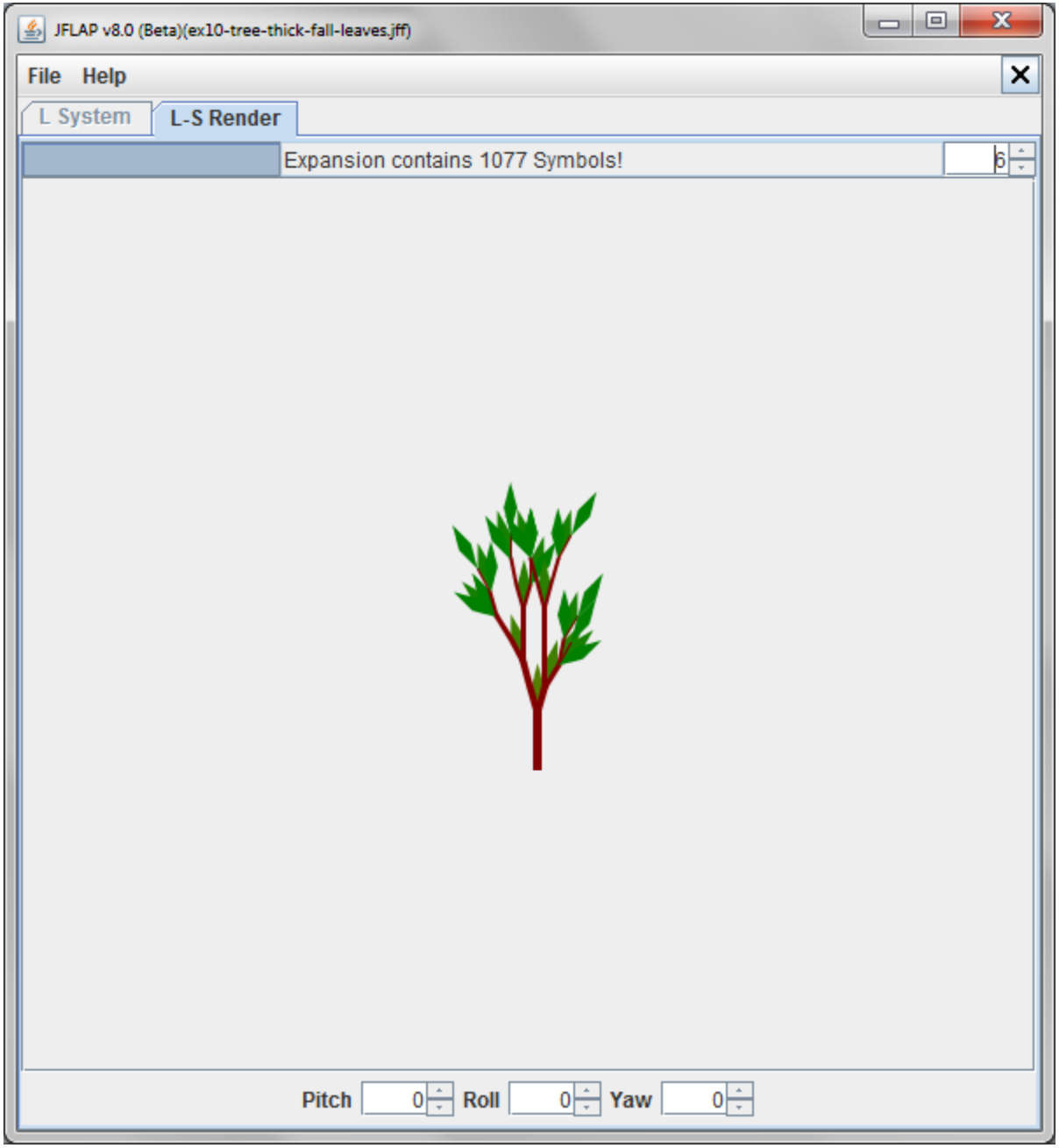

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
! @@ R ~ ## [~ ## T L - B + + B]
1
Pitch
0
Roll
0
Yaw
0

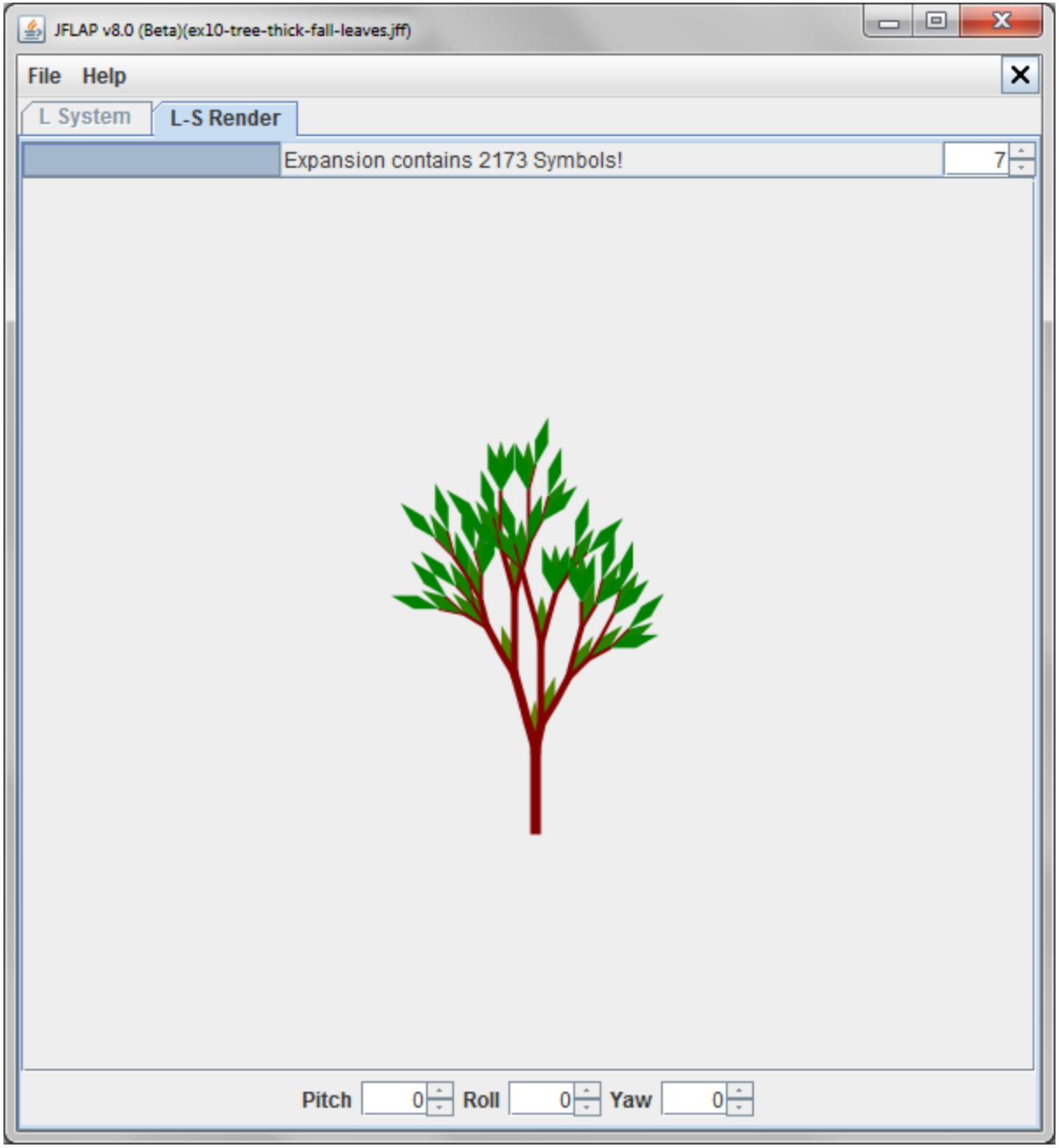

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
! @@ ! @@ R ~ ## [~ ## T [angle=15 { - g + + g % - - g }] - [~ ## T L
2
Pitch
0
Roll
0
Yaw
0

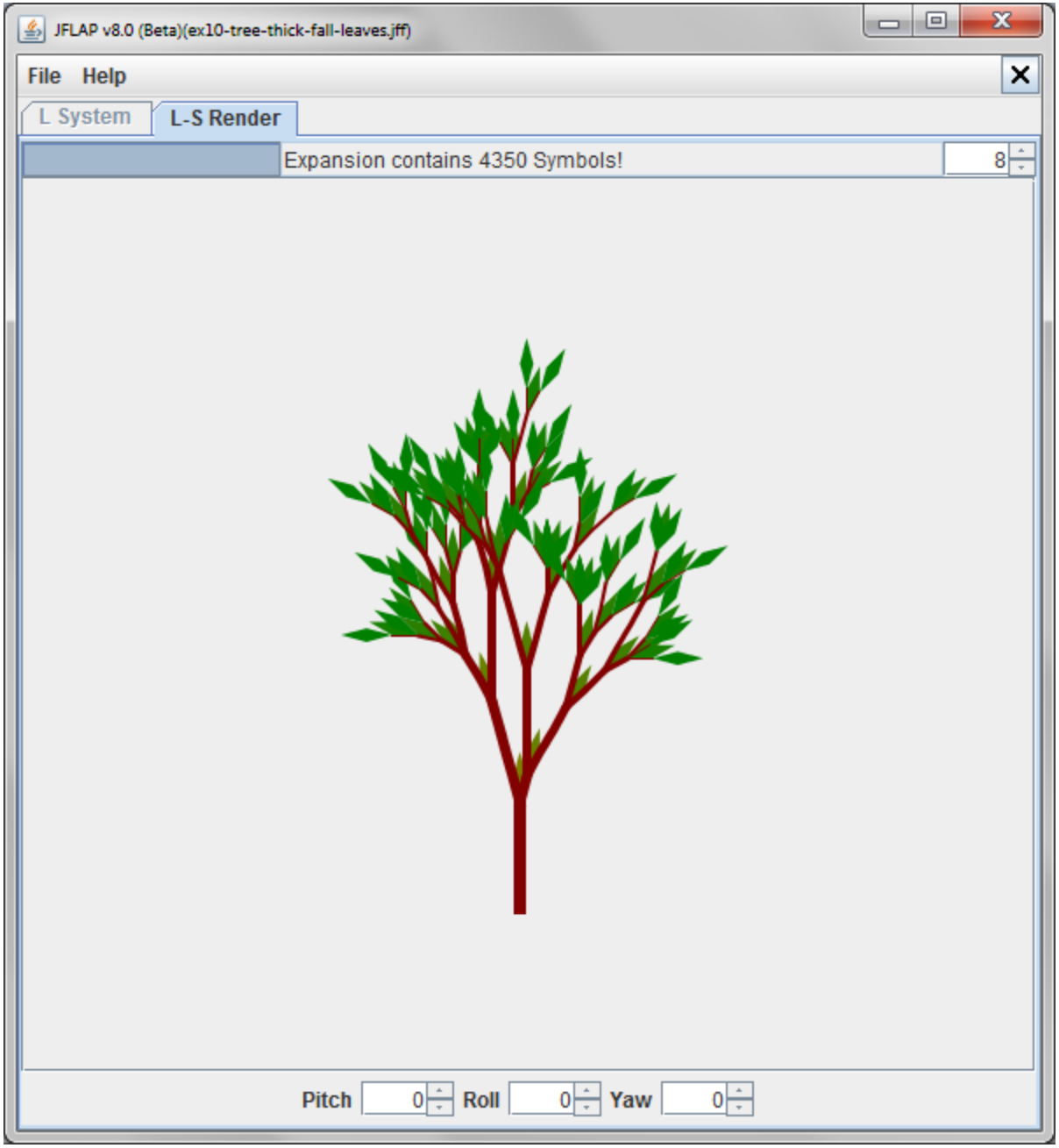

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 121 Symbols!
3
Pitch
0
Roll
0
Yaw
0

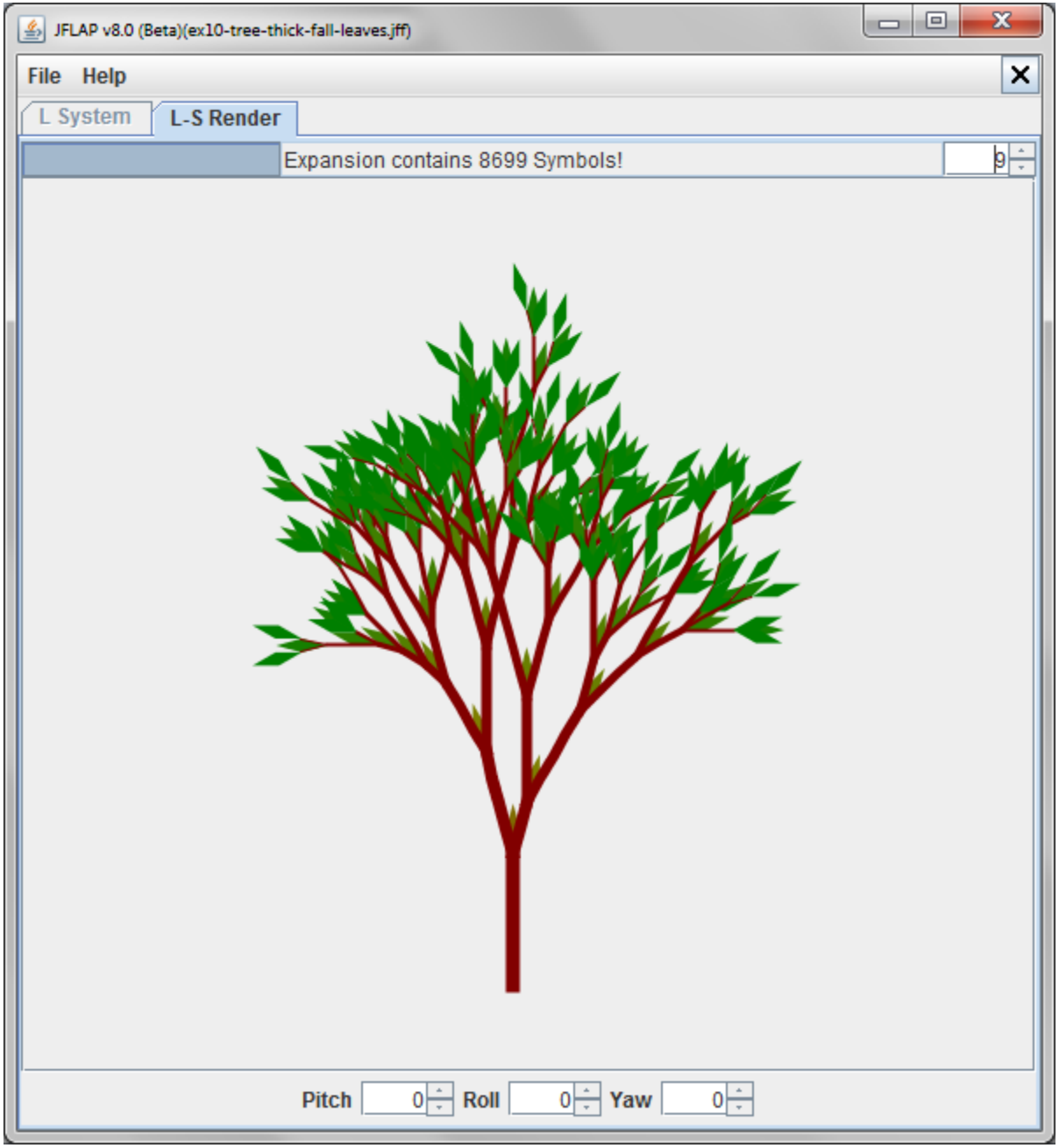

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 259 Symbols!
4
Pitch
0
Roll
0
Yaw
0

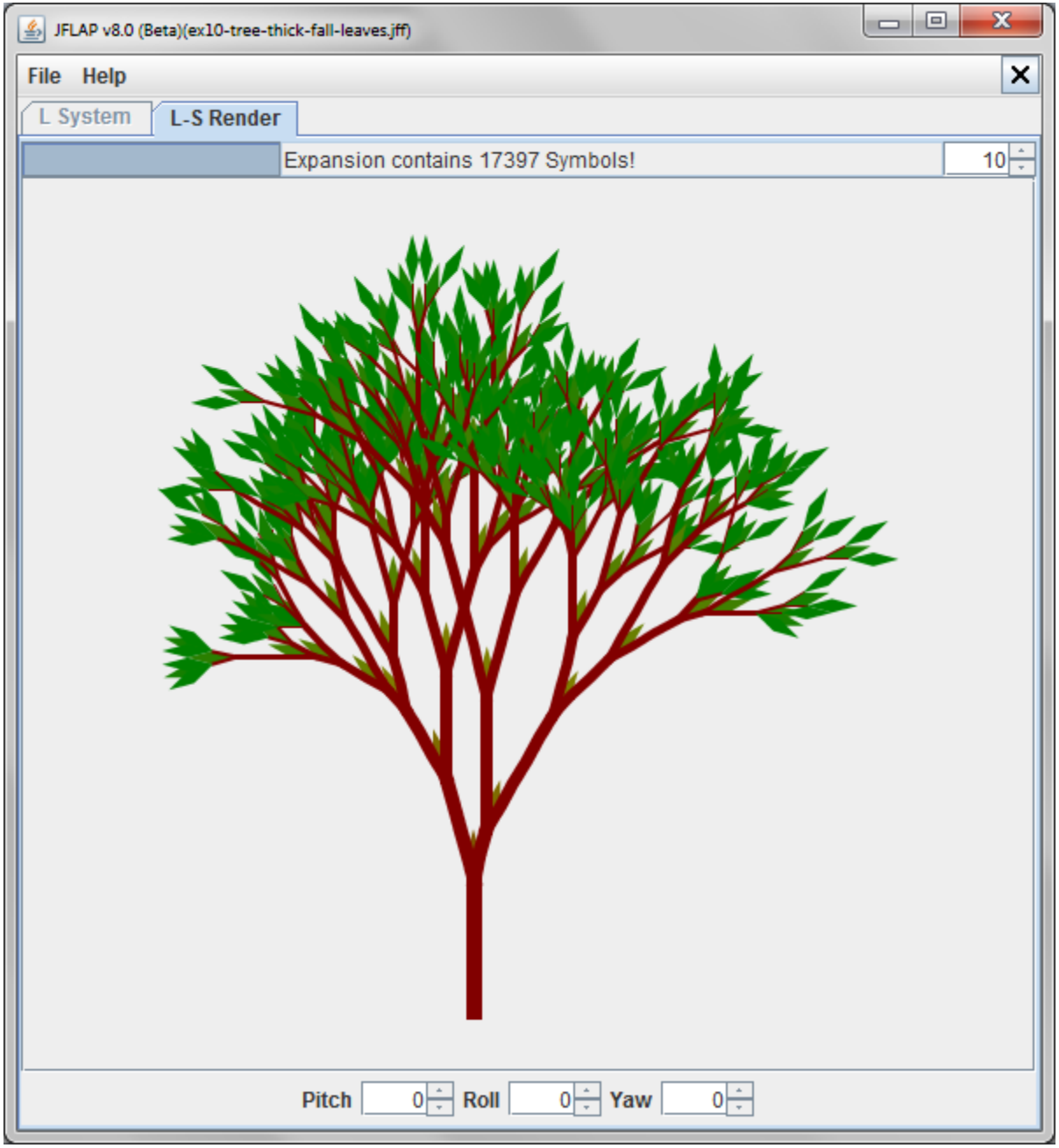

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 532 Symbols!
5
Pitch
0
Roll
0
Yaw
0


JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 1077 Symbols!
6
Pitch
0
Roll
0
Yaw
0


JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 2173 Symbols!
7
Pitch
0
Roll
0
Yaw
0

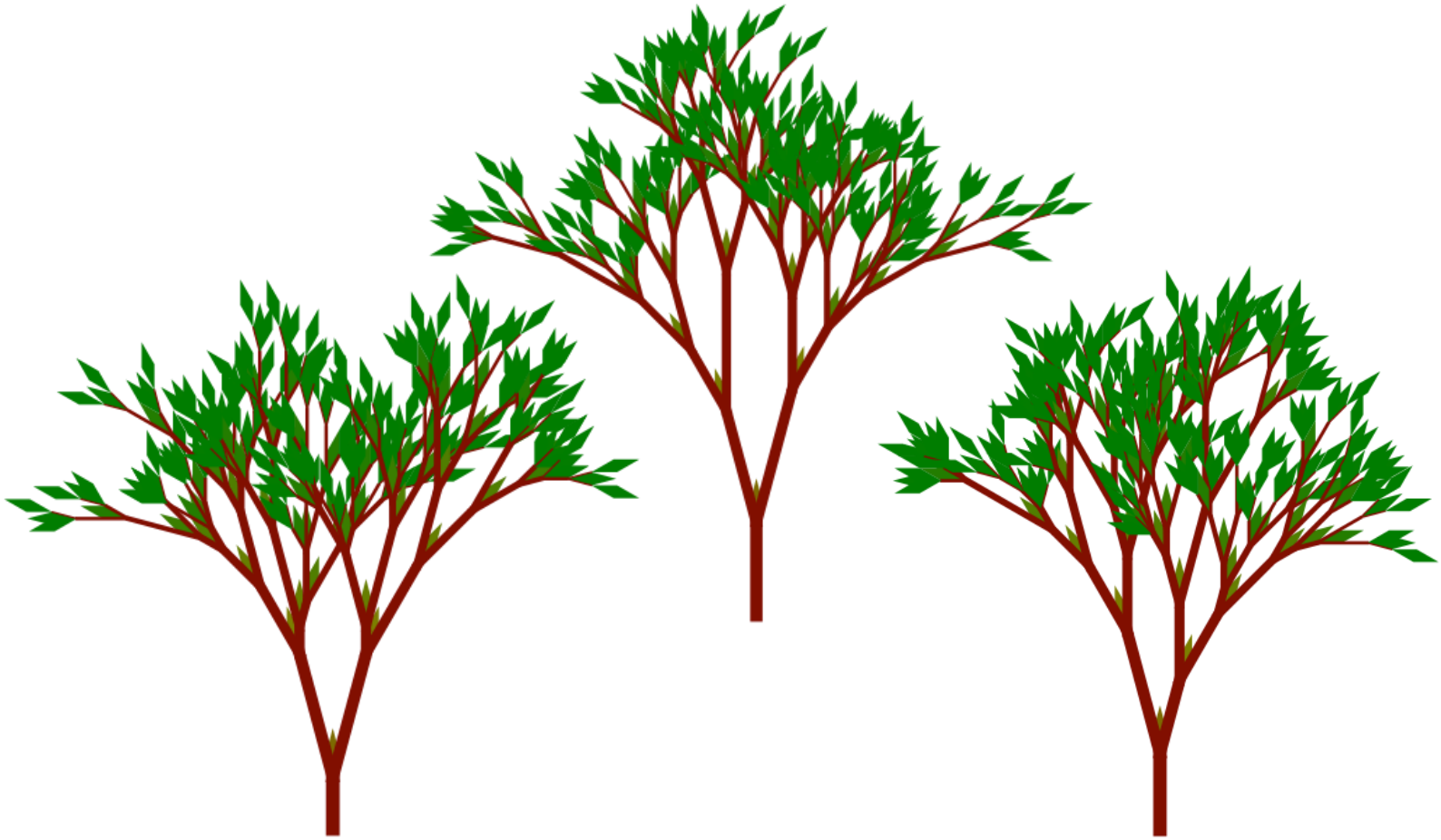

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 4350 Symbols!
8
Pitch
0
Roll
0
Yaw
0


JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 8699 Symbols!
9
Pitch
0
Roll
0
Yaw
0

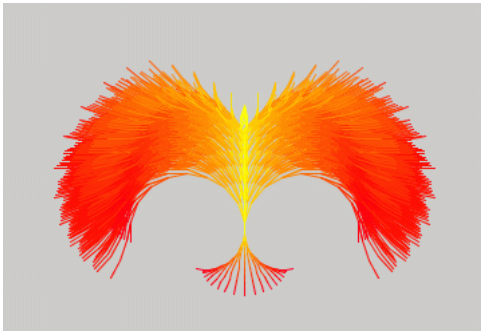

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 17397 Symbols!
10
Pitch
0
Roll
0
Yaw
0

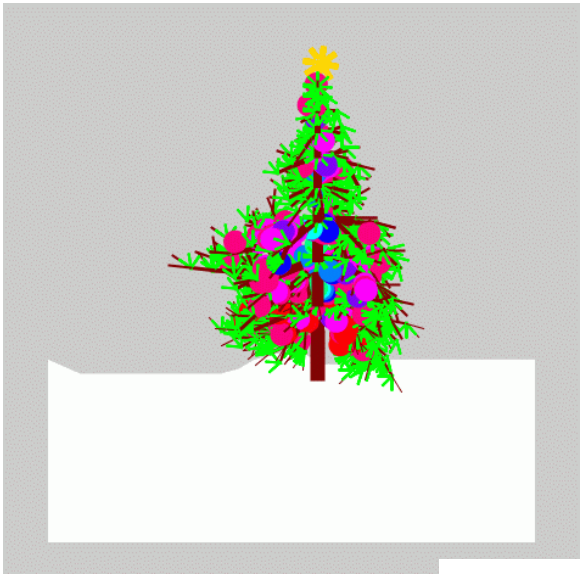
L-Systems

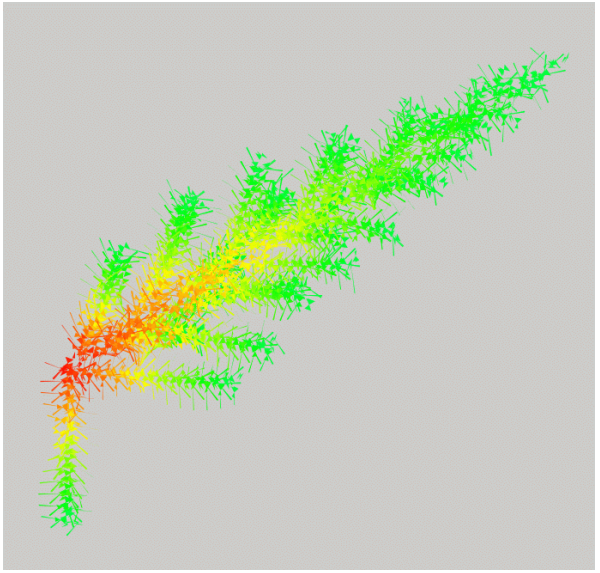
The same stochastic L-system, rendered 3 different times all at the 9th derivation.

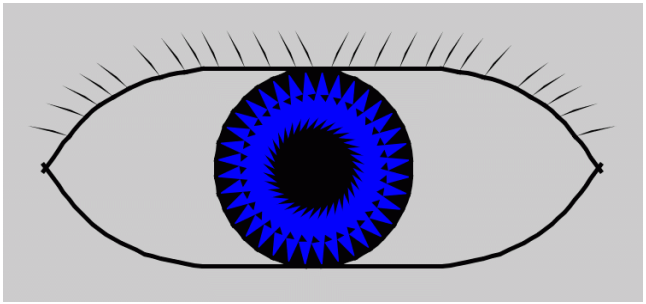


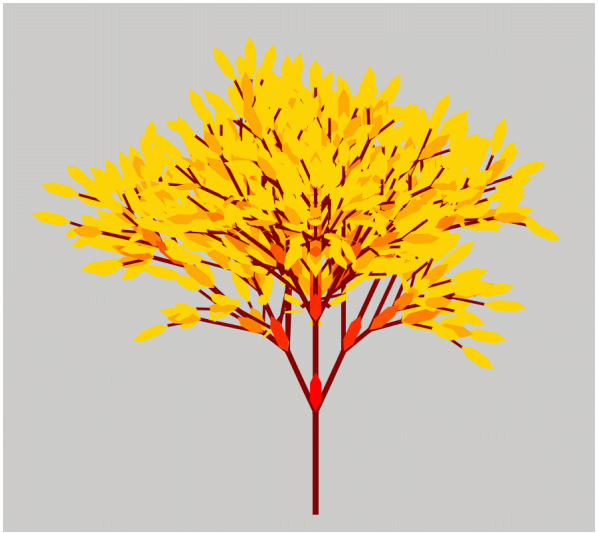
Students like L-systems

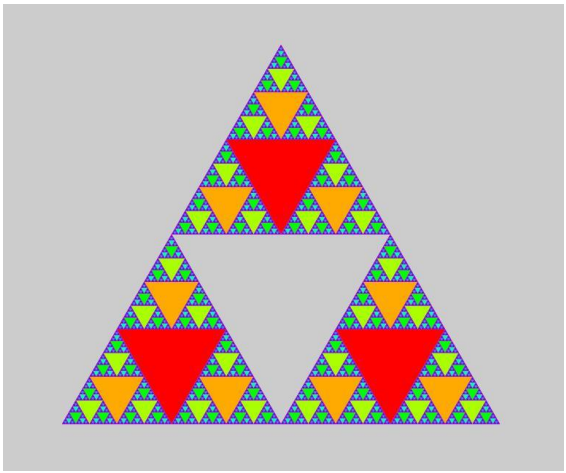


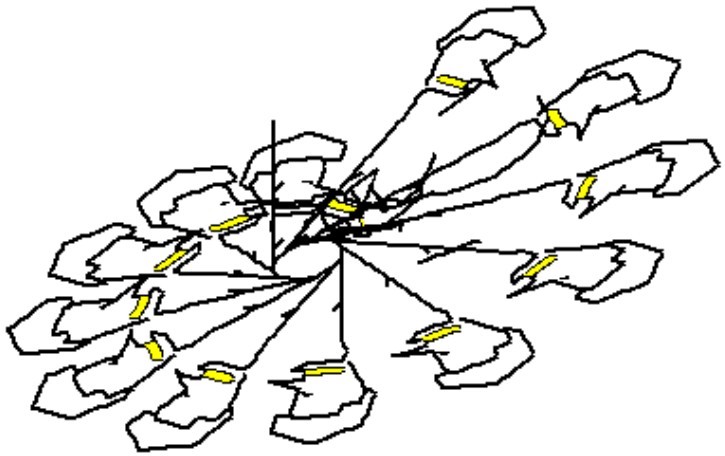












Two-year JFLAP Study
2005-2007

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

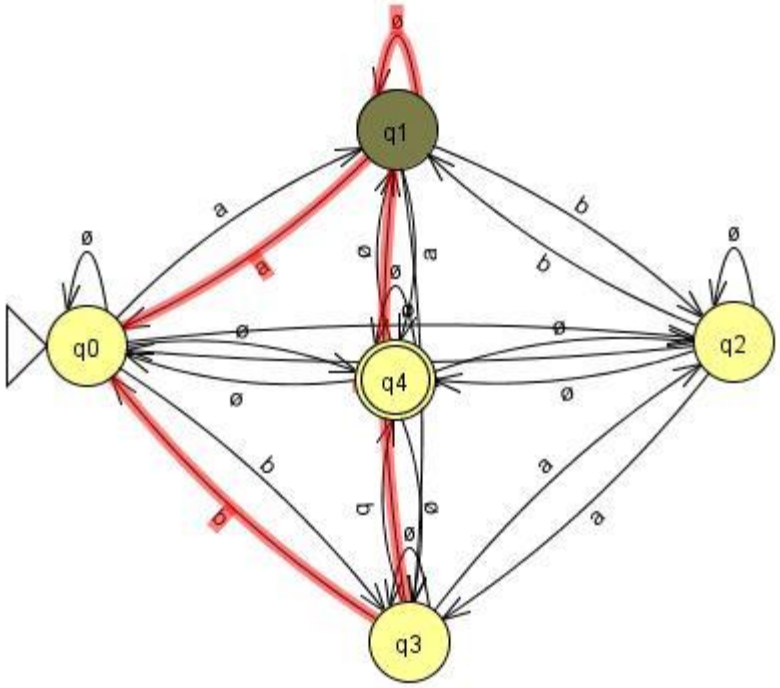
Conclusions From Study

- 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 students thought time and effort using JFLAP helped them get a better grade.

Now a few tips if you ever write educational software…

Make your tool as interactive as possible – but not too tedious!

- User shouldn't type everything
- Sometimes select
- Example: DFA to regular expression in JFLAP



Transitions

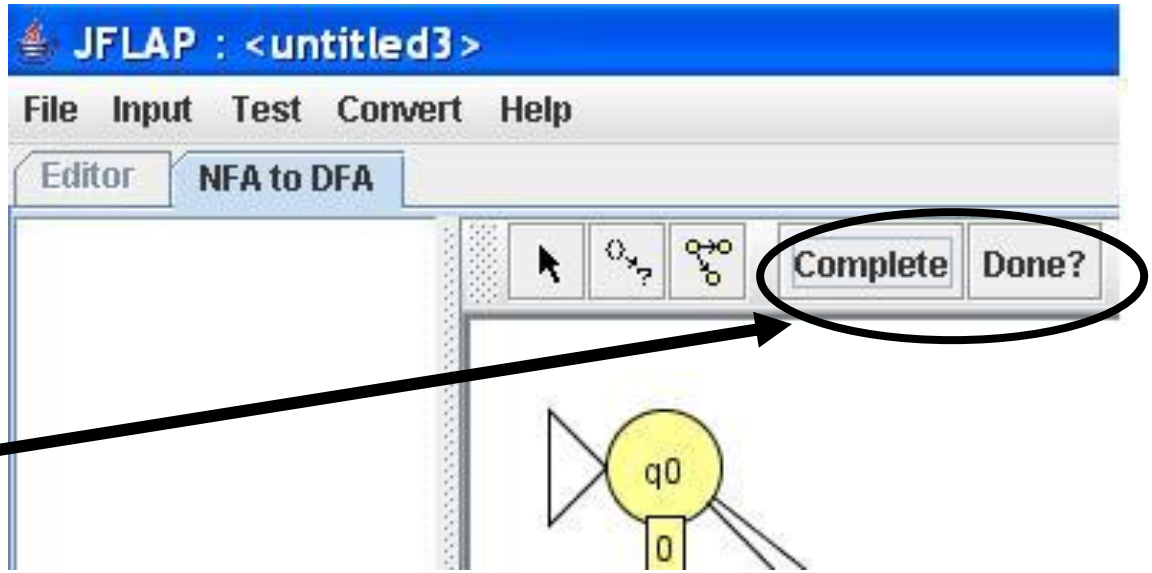
Select to see what transitions were com...

From	To	Label
0	0	aa
0	2	ab
0	3	b
0	4	aa
2	0	ba
2	2	bb
2	3	a
2	4	ba
3	0	b
3	2	a
3	3	ø
3	4	b
4	0	ø
4	2	ø
4	3	ø
4	4	ø

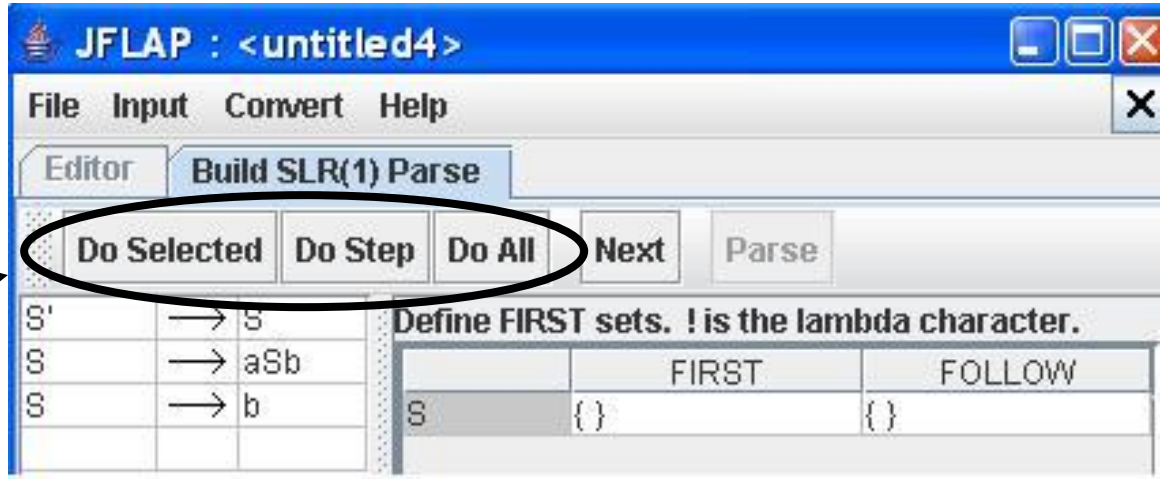
Finalize

Allow user to proceed on if they got it

- Complete the rest for them

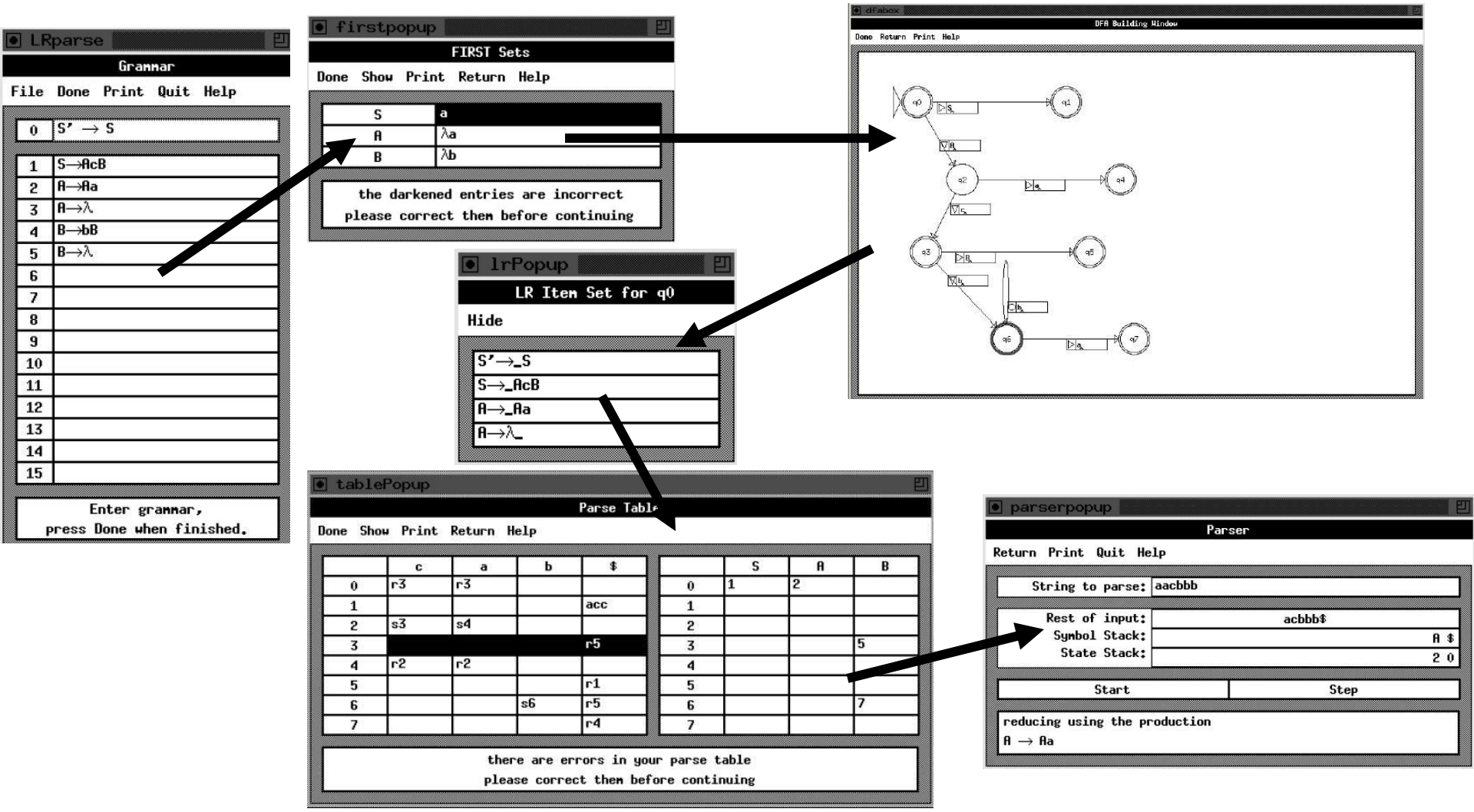


- Complete parts for them



Avoid Too Many Pop up windows

- OLD JFLAP LR PARSE TOOL

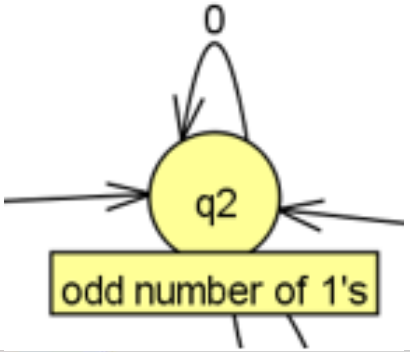


Add Pause/Checkpoint questions

- Allow for pause to think about what comes next
- Undo/go back

- Pop up a quiz question to see if the user understands what he/she just did
  - JHAVE tool does this
  - Can integrate into ebooks

What can make the tool more useable?



- Annotations on states
- Multiple run window
  - Develop test data
  - Easier for grading
- General definitions
  - FA – recognize one or more symbols
  - NPDA – pop or push 0 or more symbols
- Batch processing

Multiple Run

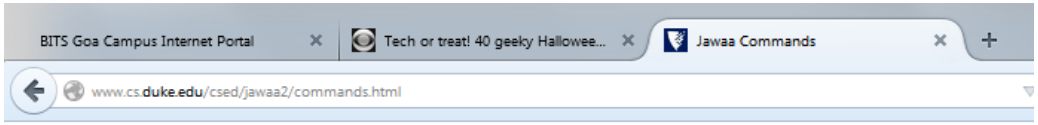
Input	Result
a	Accept
aa	Accept
aab	Accept
aabb	Accept
	Reject
acb	Reject
abcbb	Accept
abbcc	Accept
abcab	Reject
bc	Reject

Naming your software

What is a “good” name for your tool?

Jawaa

- Algorithm Animation tool



Rectangle

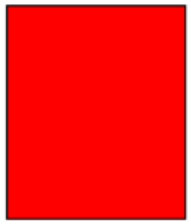
Parameters:

name	a name uniquely identifying this rectangle
x	x-coordinate
y	y-coordinate
width	width of the rectangle
height	height of the rectangle
color	color of the rectangle outline
bkgrd	color of the rectangle's background

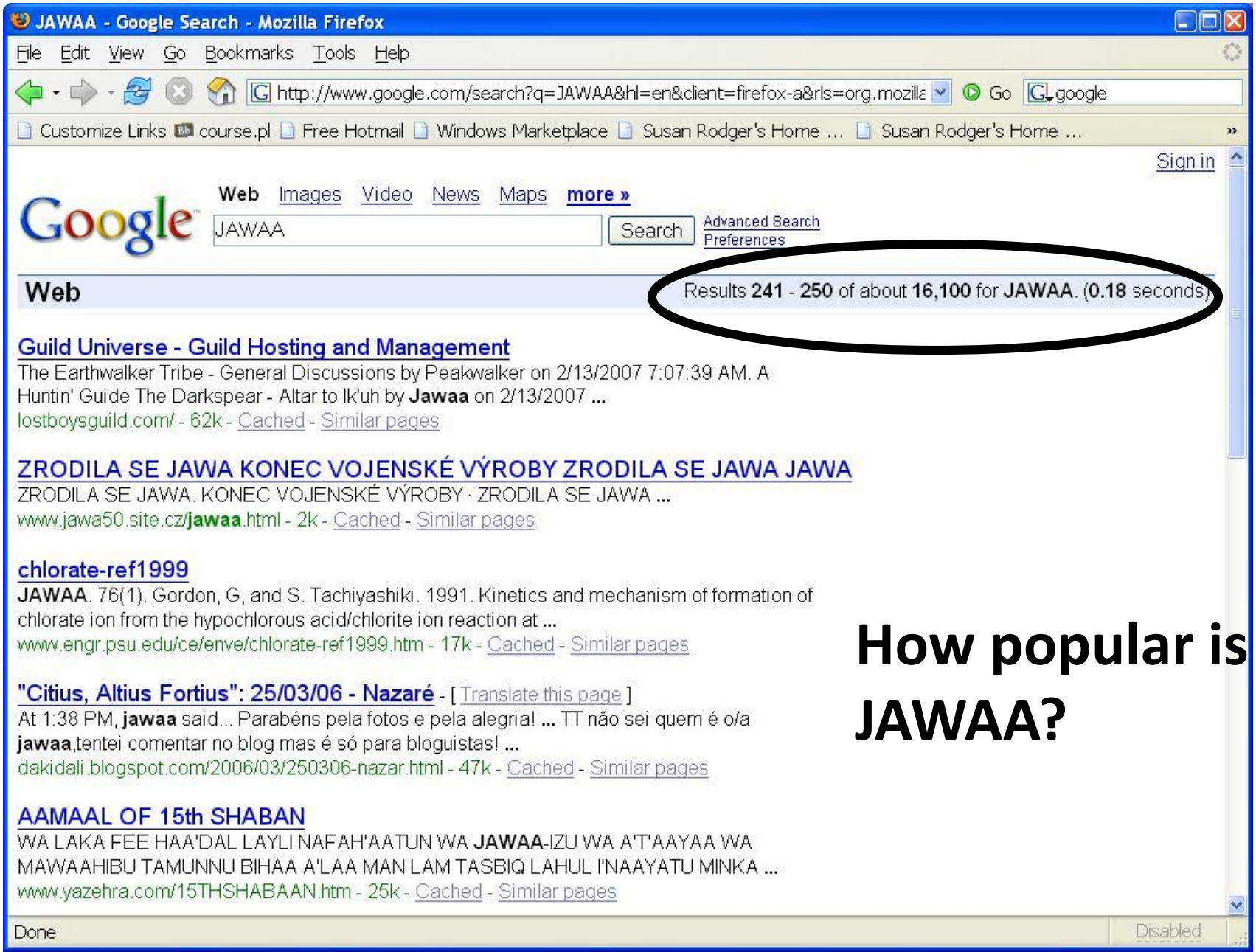
Example:

```
rectangle r1 10 20 100 120 black red
rectangle r2 150 20 180 60 cyan yellow
```

The first example will create a rectangle with its upper left corner at (10,20) and
rectangle will be red with a black outline, as shown in the figure below on the lef
The second example will create a rectangle with its upper corner at (150,20) and
rectangle will be yellow with a cyan outline. This is shown in the figure below or



JAWAA name is not unique



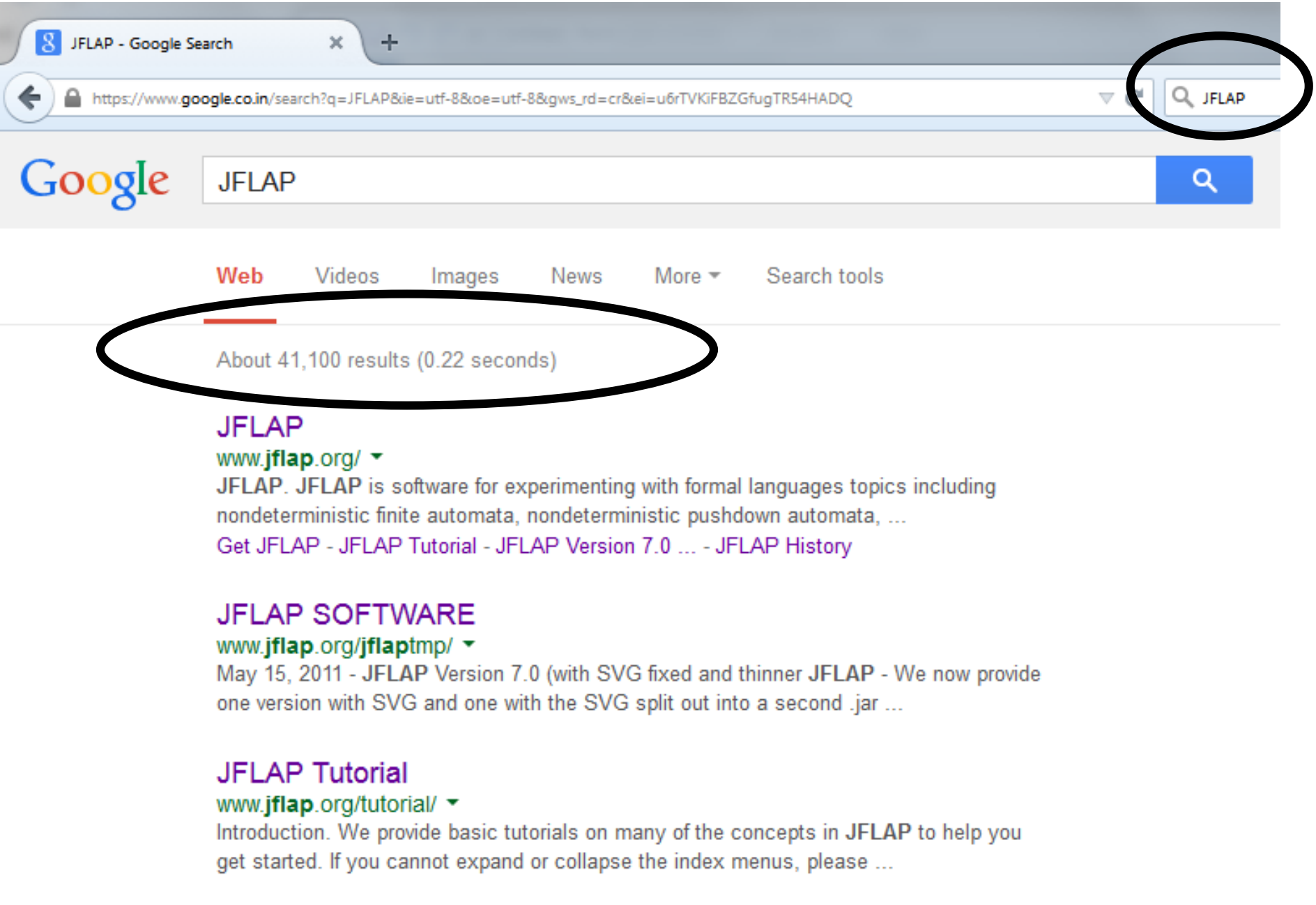
How popular is JAWAA?

FLAP

- **F**ormal **L**anguages and **A**utomata **P**ackage

- 1996 – converted to Java
- FLAP -> JFLAP

JFLAP name is unique



Much more than Google Analytics
Forums, Blogs, Course websites

Newest 'jflap' Questions - Stack Overflow
stackoverflow.com/questions/tagged/**jflap**
We can use small letters for terminals and caps for Non-terminals in **JFLAP** while entering grammar. But this restricts to only 26 options. Can we have more ...

Blog:Recent posts - JFLAP
jflap.wikia.com/wiki/**Blog**:Recent_posts
Watchlist Random page Recent changes · Create **blog** post. Recent posts. **Blog** posts. Retrieved from "http://**jflap**.wikia.com/wiki/**Blog**:Recent_posts?oldid=3140" ...

CS 301: Using JFLAP
www.cs.colostate.edu/~massey/Teaching/.../**JFLAP**/gettingstarted.html
This course uses the **JFLAP** package. According to the **JFLAP** website, **JFLAP** is a package of graphical tools which can be used as an aid in learning the basic ...

[PDF] JFLAP Startup
www.inf.unibz.it/~calvanese/teaching/10-11-fl/.../**JFLAP**-manual.pdf
Download **JFLAP** and the files referenced in this book from www . j flap. org to get started. **JFLAP** is written in Java to allow it to run on a range of platforms.

AICTE- Sponsored SDP
"Workshop on Theory of Computation"
July 9, 2012 to July 13, 2012

Registration Form

Name: .....

Designation.....

Qualification:.....

Organization/Institute/University.....

Address for Correspondence .....

.....

.....

Phone(Mobile).....

Email.....

Gender & Age .....

Total Teaching / Research / Development Experience

.....

Accommodation Required: YES / NO

D. D. No:..... Dated: .....

Amount: ..... Name of Bank:.....

Date..... Signature of the Candidate

Certified that the candidate is employed in our institute and is sponsored to attend the course. He will be relieved for the said period if selected for the above course. The Candidate will abide by the rules and regulations governing the staff development program and will attend with full sincerity.

Date..... Signature of the Principal with seal

Scanned copy of above form filled with complete information duly signed by candidate and principal may be sent in advance to: toc@rgcer.edu.in or tocsdp@gmail.com

Patrons

Hon'ble Shri Dattaji Meghe
Chairman

Shri Sagarji Meghe
Secretary

Shri Sameerji Meghe
Treasurer

Dr. Avichal Kapur
Chief Executive Officer

Dr. M.M. Raghuwanshi
Principal

Organizing committee

Coordinator
Mr. M.M. Goswami

Co-Coordinator
Mr. Amit Khaparde
Mr. Alok Chauhan

ADDRESS FOR CORRESPONDENCE

SDP,
Rajiv Gandhi College of Engg. & Research,
Wanadongri, Hingna Road, Nagpur-441110
Maharashtra, India.

Ph.: 07104-649154,
Fax:07104-287950

Website:
www.rgcer.edu.in/toc.html

MEGHE GROUP

AICTE Sponsored
One Week
Staff Development Programme

Workshop
On
Theory of Computation

09th July to 13th July 2012

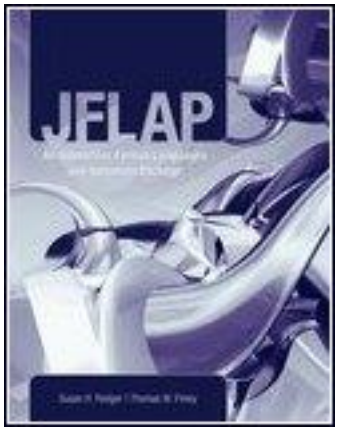
Organized by
Department
of
Information Technology
Nagar Yuwak Shikshan Sanstha's
Rajiv Gandhi College of Engineering & Research, Nagpur (M.S.)

COURSE CONTENTS

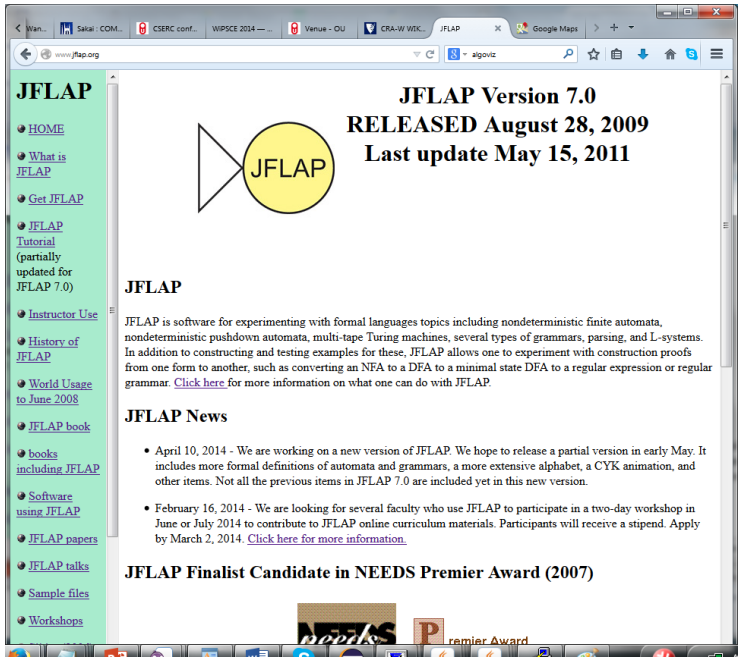
- Mathematical Foundation for Theory of computation
- Basic computational devices and their languages
- Finite Automata & Regular Languages
- Context Free Grammar & Context Free Languages, Pushdown Automata
- Turing Machines, Linear Bounded Automata & Context Sensitive Language.
- Undecidability,Reducability
- Practical applications of Theory of Computation
- Recent advancement in Theory of Computation

All the theory sessions will be supported by lab sessions with the hands on Practicals on tools such as LEX, YACC, JFLAP, Grammar Tool Box (GTB), etc.

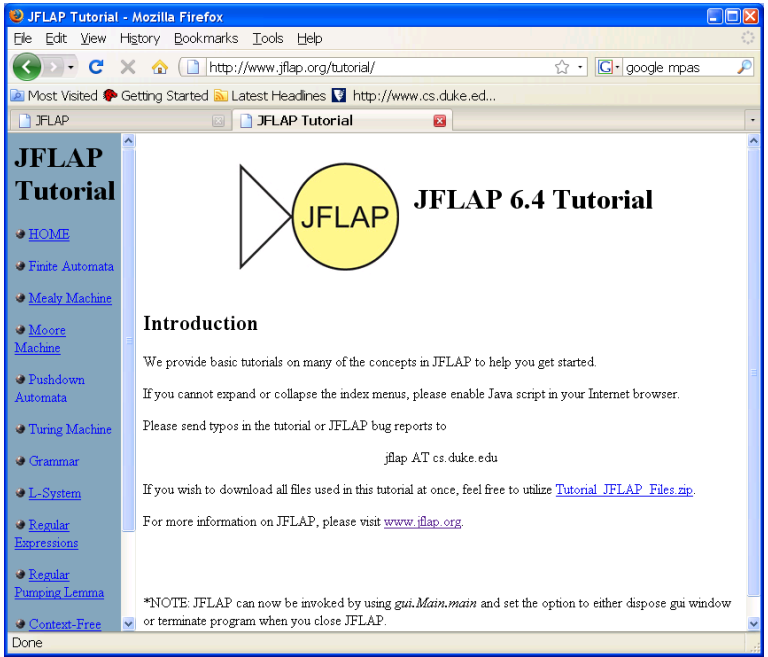
JFLAP is free

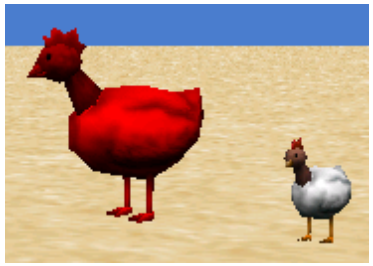


www.jflap.org

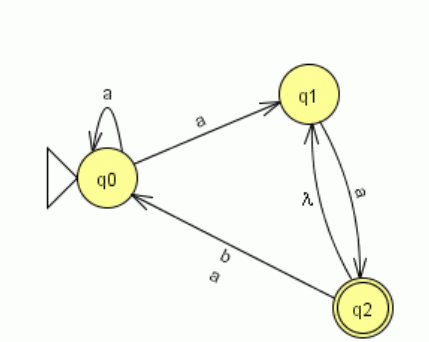


JFLAP tutorial





Outline



- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Additional Ways to Engage with CS
- Diversity Efforts

Middle School students sorting themselves with Bubblesort

Interaction in Class – Props
Passing "Parameters" in Class

- Pass by reference – throw frisbee

- Pass by value – throw copy of frisbee

- Pass by const reference – throw "protected" frisbee

CS UNPLUGGED

Computer Science without a computer

Search

People talking about #CSUnplugged

Tweets

Tracy Henderson
@henderson942

Saw this on Tuesday when I asked a student to teach #CSunplugged binary numbers. She did great and was challenged! twitter.com/justintarte/st...

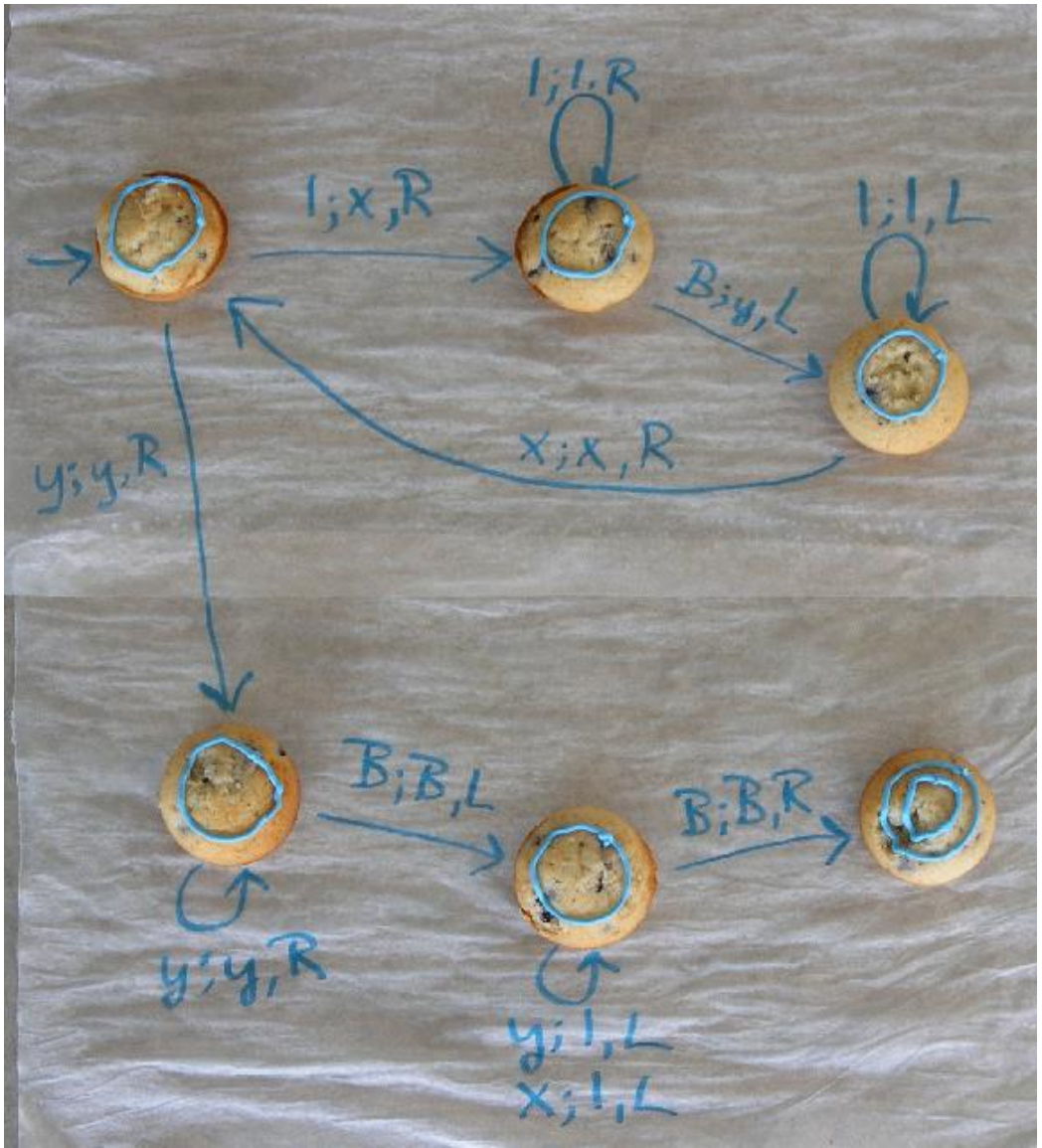
9h

What happens when your hobby and your career collide?

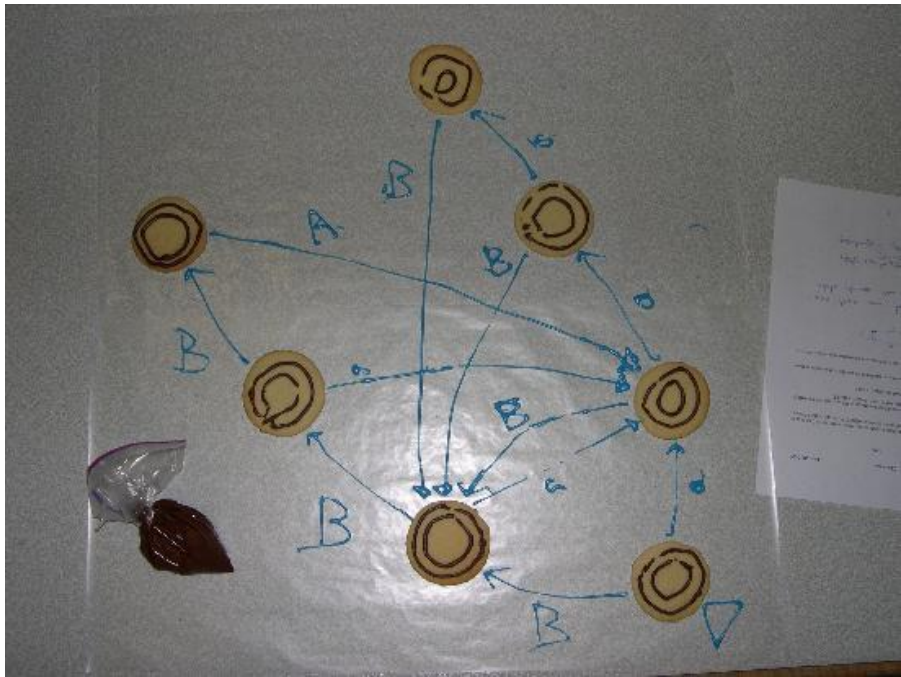
It is now time for engaging students with edible CS

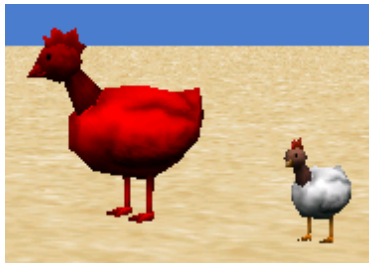
Automata Theory
Interaction in Class – Props
Edible Turing Machine

- TM for f(x)=2x where x is unary
- TM is not correct, can you fix it? Then eat it!
- States are blueberry muffins

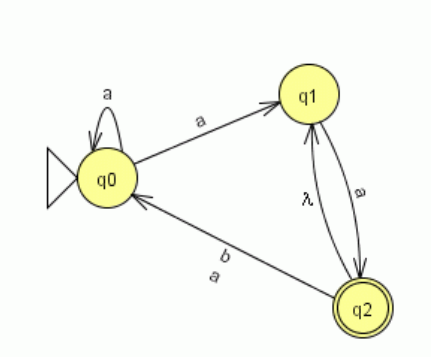


Students building DFA with cookies and icing





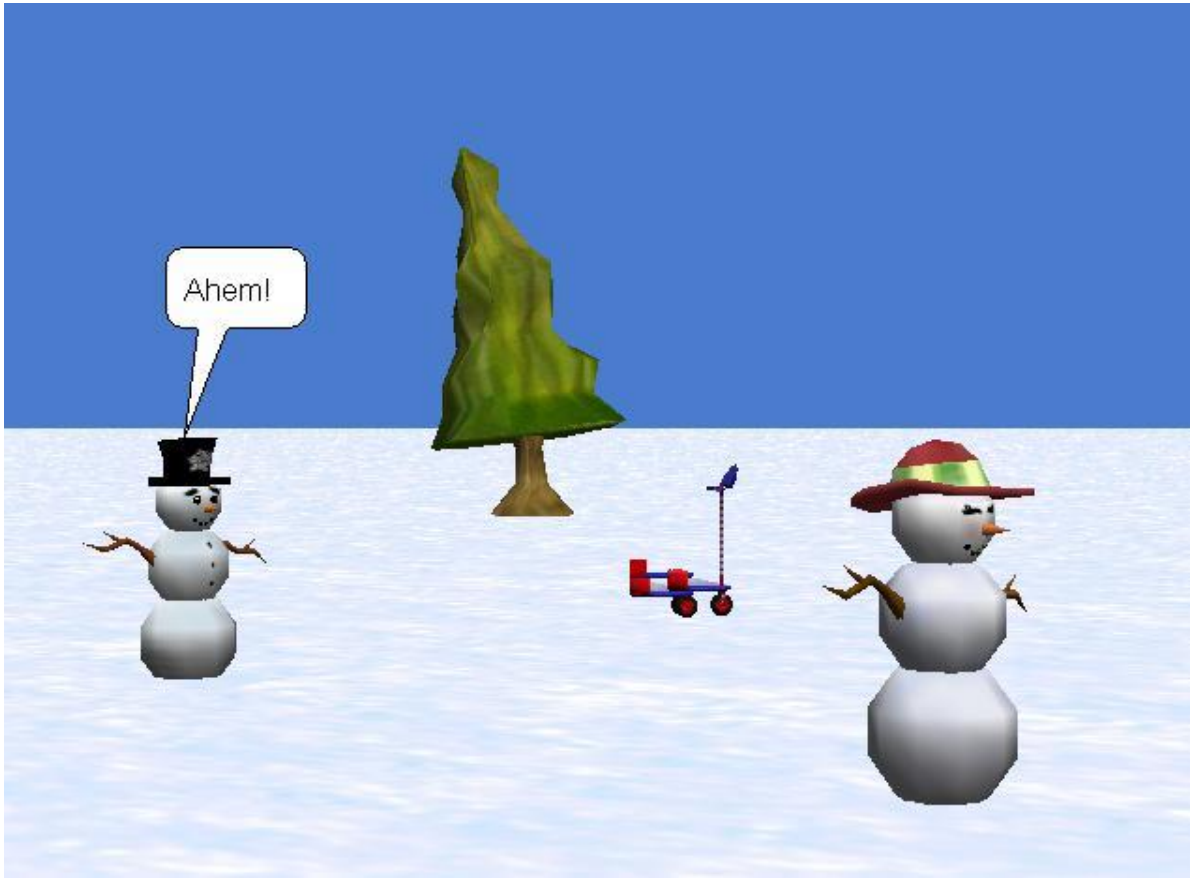
Outline



- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Additional Ways to Engage with CS
- Diversity Efforts

Success - Alice Excites 4th-6th Grade Girls

- Duke Femmes Event, April 07
- 60 girls – 4 groups of 15
- Taught them Alice for an hour
- Handout to take home
- Event again in 2008 ,almost every year since



Adventures in Alice Programming

www.cs.duke.edu/csed/alice/aliceInSchools

- 2-week Teacher workshops
  - Over 200 teachers, middle school, high school, some elementary
  - First week Teach Alice, Practice
  - Second week - Develop Lesson Plans
  - All disciplines: math, science, history, language arts, foreign language, art, music, business
  - Summers 2008-2015, funding for lodging
- Main Sites:
  - Duke University, Durham, NC
  - Charleston/Columbia, SC
  - San Jose, CA (started 2014)

Curriculum Materials
www.cs.duke.edu/csed/alice/aliceInSchools

- Over 90 tutorials available for free
- Beginner, advanced, challenges, projects
- Paper handouts and video
- Over 200 Teacher lesson plans
  - Organized by discipline and grade level

Tutorial for Adventure Game – Find objects in order

Harry Potter Challenge

- Mix of programming and math ch

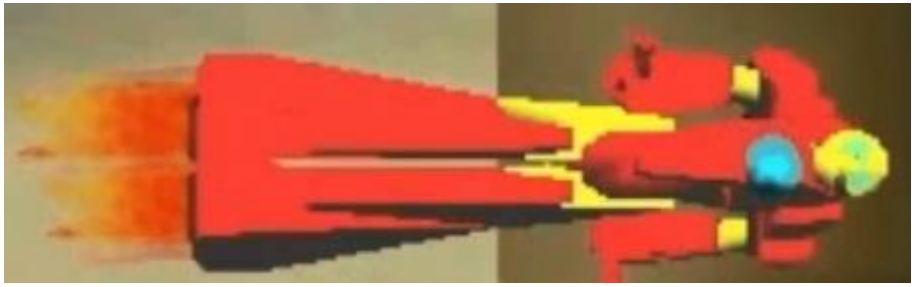
Harry Potter – Math/computing Level 1 Charms - before

Harry Potter – Math/Computing Level 1 Charms - after

Science Example
How a volcano is formed

What a 6th grader can do with Alice - teacher Chari Distler

No Superheros in Alice



How Visible are Notable Women in Computer Science?

- Pondered this question in early 2012
- Looked at Wikipedia
  - The internet encyclopedia
  - Who writes those pages?
  - Why did some notables have pages and others not?
- Turing Award Winners
  - Only two women at that time

Fran Allen

- School teacher – got a job at IBM
- Compilers and Optimization Technology
- IBM Fellow – First Women
- Turing Award (2006) – First Woman
- The Turing Award was announced on Feb. 21, 2007
- Her Wikipedia page was created on…
  - Feb. 6, 2007
- On Feb 21, 2007 the Turing Award was added to her Wikipedia page.

Here is that first page for Fran Allen

Frances E. Allen

From Wikipedia, the free encyclopedia

Fran Allen has made outstanding contributions to the field of programming languages for more than forty-five years, and her work has significantly influenced the wider computer science community.

Ms. Allen is a pioneer in the field of optimizing compilers. Her achievements include seminal work in compilers, code optimization, and parallelization. In the early 1980s, she formed the Parallel TRANslation (PTRAN) group to study the issues involved in compiling for parallel machines. The group was considered one of the top research groups in the world working with parallelization issues. Her work on these projects culminated in algorithms and technologies that form the basis for the theory of program optimization and are widely used in today's commercial compilers throughout the industry.

Ms. Allen's influence on the IBM community was recognized by her appointment as an IBM fellow, the first woman to receive this recognition. She was also president of the IBM Academy of Technology. The Academy plays an important role in the corporation by providing technical leadership, advancing the understanding of key technical areas and fostering communications among technical professionals.

In 1997, Ms. Allen was inducted into the WITI Hall of Fame. Ms. Allen retired from IBM in 2002.

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here

Three days later…

WIKIPEDIA
The Free Encyclopedia

Create account Log i

Article Talk | Read Edit View history | Search

Frances E. Allen

From Wikipedia, the free encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

Fran Allen is a pioneer in the field of optimizing compilers. Her achievements include seminal work in compilers, code optimization, and parallelization.

In the early 1980s, she formed the Parallel TRANslation (PTRAN) group to study the issues involved in compiling for parallel machines. The group was considered one of the top research groups in the world working with parallelization issues. Her work on these projects culminated in algorithms and technologies that form the basis for the theory of program optimization and are widely used in today's commercial compilers throughout the industry.

Ms. Allen's influence on the IBM community was recognized by her appointment as an IBM Fellow, the first woman to receive this recognition. She was also president of the IBM Academy of Technology. The Academy plays an important role in the corporation by providing technical leadership, advancing the understanding of key technical areas and fostering communications among technical professionals.

In 1997, Ms. Allen was inducted into the WITI Hall of Fame. Ms. Allen retired from IBM in 2002.

This article **has not been added to any categories**. Please help out by adding categories to it so that it can be listed with similar articles.

Turing Award Announced and added to her page

In 1997, Ms. Allen was inducted into the WITI Hall of Fame. Ms. Allen retired from IBM in 2002.

Early 2007, she became the first woman to win the the A.M. Turing Award.

V · T · E **A. M. Turing Award laureates** [hide]

Alan Perlis (1966) · Maurice Vincent Wilkes (1967) · Richard Hamming (1968) · Marvin Minsky (1969) · James H. Wilkinson (1970) · John McCarthy (1971) · Edsger W. Dijkstra (1972) · Charles Bachman (1973) · Donald Knuth (1974) · Allen Newell / Herbert A. Simon (1975) · Michael O. Rabin / Dana Scott (1976) · John Backus (1977) · Robert W. Floyd (1978) · Kenneth E. Iverson (1979) · Tony Hoare (1980) · Edgar F. Codd (1981) · Stephen Cook (1982) · Ken Thompson / Dennis Ritchie (1983) · Niklaus Wirth (1984) · Richard Karp (1985) · John Hopcroft / Robert Tarjan (1986) · John Cocke (1987) · Ivan Sutherland (1988) · William Kahan (1989) · Fernando J. Corbató (1990) · Robin Milner (1991) · Butler Lampson (1992) · Juris Hartmanis / Richard E. Stearns (1993) · Edward Feigenbaum / Raj Reddy (1994) · Manuel Blum (1995) · Amir Pnueli (1996) · Douglas Engelbart (1997) · Jim Gray (1998) · Fred Brooks (1999) · Andrew Yao (2000) · Ole-Johan Dahl / Kristen Nygaard (2001) · Ron Rivest / Adi Shamir / Leonard Adleman (2002) · Alan Kay (2003) · Vint Cerf / Bob Kahn (2004) · Peter Naur (2005) · **Frances E. Allen** (2006)

Categories: Turing Award laureates

In the next three days

- Over 30 edits, added awards, boards

Awards and honors

Allen is a member of the National Academy of Engineering, a fellow of the IEEE, the Association for Computing Machinery (ACM) and the American Academy of Arts and Sciences. She is currently on the Computer Science and Telecommunications Board, the Computer Research Associates (CRA) board and National Science Foundation's CISE Advisory Board.

In 1997, Allen was inducted into the WITI Hall of Fame.[3] She retired from IBM in 2002 and won the Augusta Ada Lovelace Award that year from the Association for Women in Computing. In 2007, she became the first woman to win the A.M. Turing Award.[4]

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
العربية

Frances E. Allen

From Wikipedia, the free encyclopedia

For the early American nun, see Frances Allen (nun).

Frances Elizabeth "Fran" Allen (born August 4, 1932) is an American computer scientist and pioneer in the field of optimizing compilers. Her achievements include seminal work in compilers, code optimization, and parallelization. She also had a role in intelligence work on programming languages and security codes for the National Security Agency.[2][3]

Allen was the first female IBM Fellow and in 2006 became the first woman to win the Turing Award.[4]

Contents [hide]

1 Career
2 Awards and honors
3 See also
4 References
5 External links

Career [edit]

Allen grew up on a farm in Peru, New York and graduated from The New York State College for Teachers (now State University of New York at Albany) with a B.Sc. degree in mathematics in 1954.[5] She earned an M.Sc. degree in mathematics at the University of Michigan in 1957 and began teaching school in Peru, New York.[6] Deeply in debt, she joined IBM on July 15, 1957 and planned to stay only until her school loans were paid, but ended up staying for her entire 45-year career.

Fran Allen's work has had an enormous impact on compiler research and

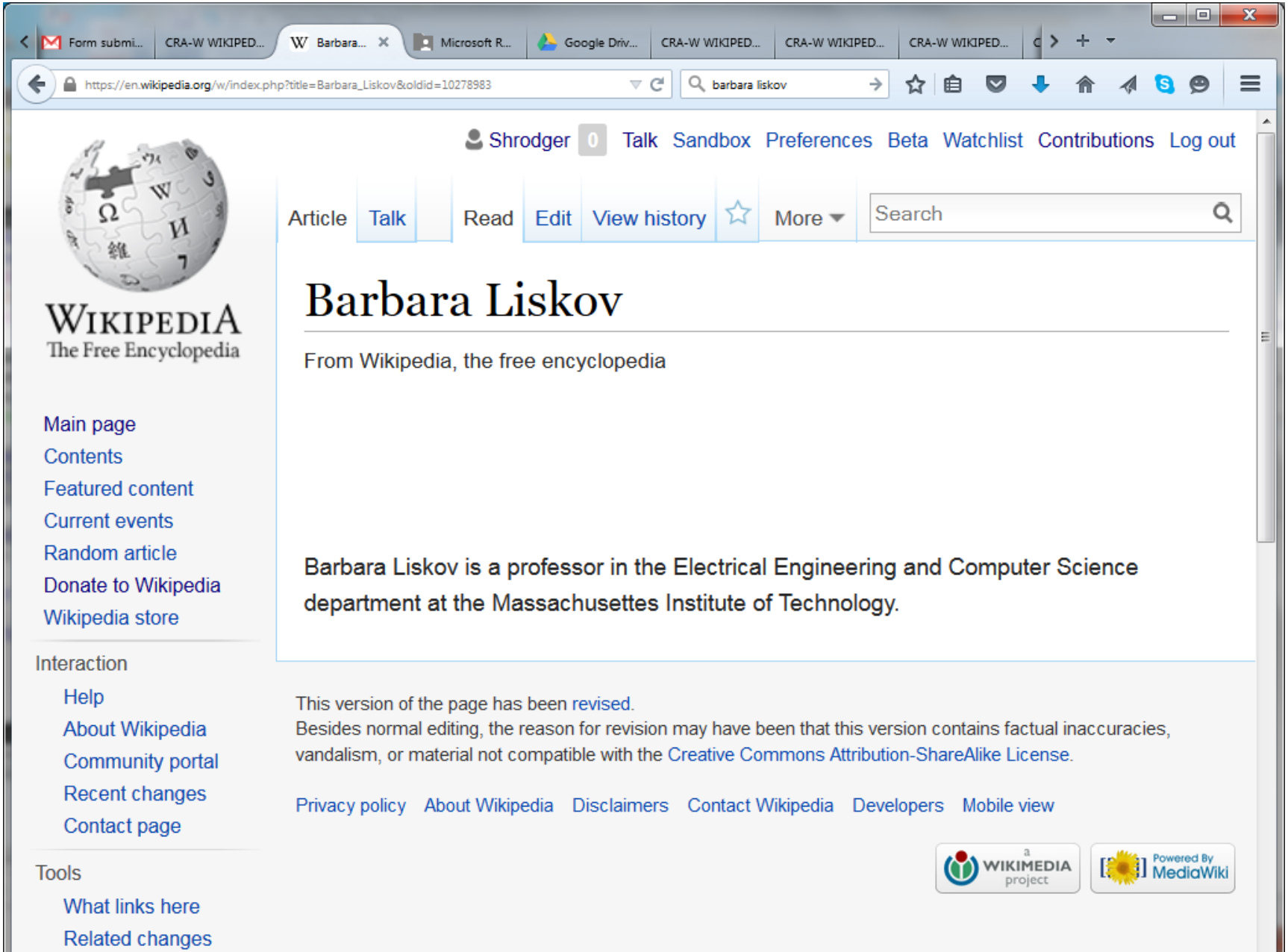
Frances Elizabeth "Fran" Allen

Born	August 4, 1932 (age 82) Peru, New York, United States[1]
Fields	computer science
Institutions	IBM
Alma mater	State University of New York at Albany, University of Michigan
Known for	high-performance computing, parallel computing, compiler organization, optimization
Notable awards	Turing Award (2006) Computer Pioneer Award (2004) Computer History Museum Fellow (2000)

Barbara Liskov

- Turing Award (2008)
- Currently Institute Professor at MIT
- One of the first women to get a PhD in CS in the U.S. (1968) – thesis on chess
- Research
  - Venus operating system
  - Design and Implementation of CLU
  - Argus, high-level language for distributed programs
  - Thor, object oriented database system
- Many awards – NAE, AAAS, Fellow ACM
- Had a Wikipedia page since before Turing Award

Wikipedia page created June 15, 2005



39 minutes later ….

about:sessionrestore | Pocket comes to F... | Lakewood Cake | Duke University Ali... | Barbara Liskov - W...
en.wikipedia.org/wiki/Barbara_Liskov
barbara liskov

Barbara Liskov

From Wikipedia, the free encyclopedia

Barbara Liskov (born November 7, 1939 as **Barbara Jane Huberman**) is an American computer scientist[2] who is an institute professor at the Massachusetts Institute of Technology and Ford Professor of Engineering in its School of Engineering's electrical engineering and computer science department.[3]

Contents [hide]

1 Life and career
2 Recognition and awards
3 See also
4 References
5 External links

Life and career [edit]

Liskov was born in 1939 California, the eldest of Jane (née Dickhoff) and Moses Huberman's four children.[4] She earned her BA in mathematics at the University of California, Berkeley in 1961. In 1968 she became one of the first women in the United States to be awarded a Ph.D. from a computer science department when she was awarded her degree from Stanford University.[5][6] The topic of her Ph.D. thesis was a computer

Barbara Liskov

Liskov in 2010.

Born	Barbara Jane Huberman November 7, 1939 (age 75) California
Nationality	American
Fields	Computer science

What about other Notable Women in Computer Science?

- ACM Fellows
  - Few women
    - 1994 first year over 130 Fellows
      - 9-12 were women? Less than 10%
  - About 20-50 Fellows per year
  - 2014 – 47 fellows, 6-8 women
  - Noticed few of Women had Wikipedia pages

Investigate New CRA-W Project

- Write Wikipedia pages for Notable women in Computing

- How hard is it to write a Wikipedia page?
  - Lots of rules you have to follow
- Another area with few women
    - 2013 study – 16% of Wikipedia writers are female

Some Rules in Writing Wikipedia Biography pages

- You cannot write your own page!
- Neutral point of view
- Person must be notable
- No original research
  - Must write only facts and reference them
  - Must be verifiable
  - Do not plagiarize – write in your own words
- Regard for subject's privacy
  - NOT A TABLOID!

Wrote a Wikipedia page

- We had no idea what we were doing….
- At a CRA-W Board meeting in April 2012

- Who to write?
  - Female Turing Award winners had pages
  - All two of them

Mary Jane Irwin

- Professor at Penn State University
- VLSI Architecture and Automated Design
- Board Level Designs
  - Arithmetic Cube, MGAP, and SPARTA
- Architecture, Logic & Circuit Design Tools
  - ARTIST, PERFLEX, LOGICIAN, DECOMPOSER
- Awards
  - National Academy of Engineering
  - ACM Fellow, IEEE Fellow
- No Wikipedia page!

In writing her Wikipedia page,

Here is what happens when you don’t know what you are doing…

Mary Jane Irwin - Wikipedia, the free encyclopedia - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Minecraft | LE... | Gift Card - My ... | Amazon.com ... | Oracle | People... | Welcome to G... | Mary Jane ... | Engl 10: Wester... | Erich Narten's ... | NCDMV, Onl

en.wikipedia.org/wiki/Mary_Jane_Irwin

janie irwin

Create accoun

WIKIPEDIA
The Free Encyclopedia

Article Talk | Read Edit View history | Search

Join Wikipedia Takes America photo hunts in 10+ cities in September.

Mary Jane Irwin

Main page

This article has multiple issues. Please help improve it or discuss these issues on the talk page.

- This biographical article **needs additional citations for verification**. *(April 2012)*
- **A major contributor to this article appears to have a close connection with its subject.** *(April 2012)*
- This article contains wording that **promotes the subject in a subjective manner without imparting real information**. *(April 2012)*
- The **neutrality of this article is disputed**. *(April 2012)*

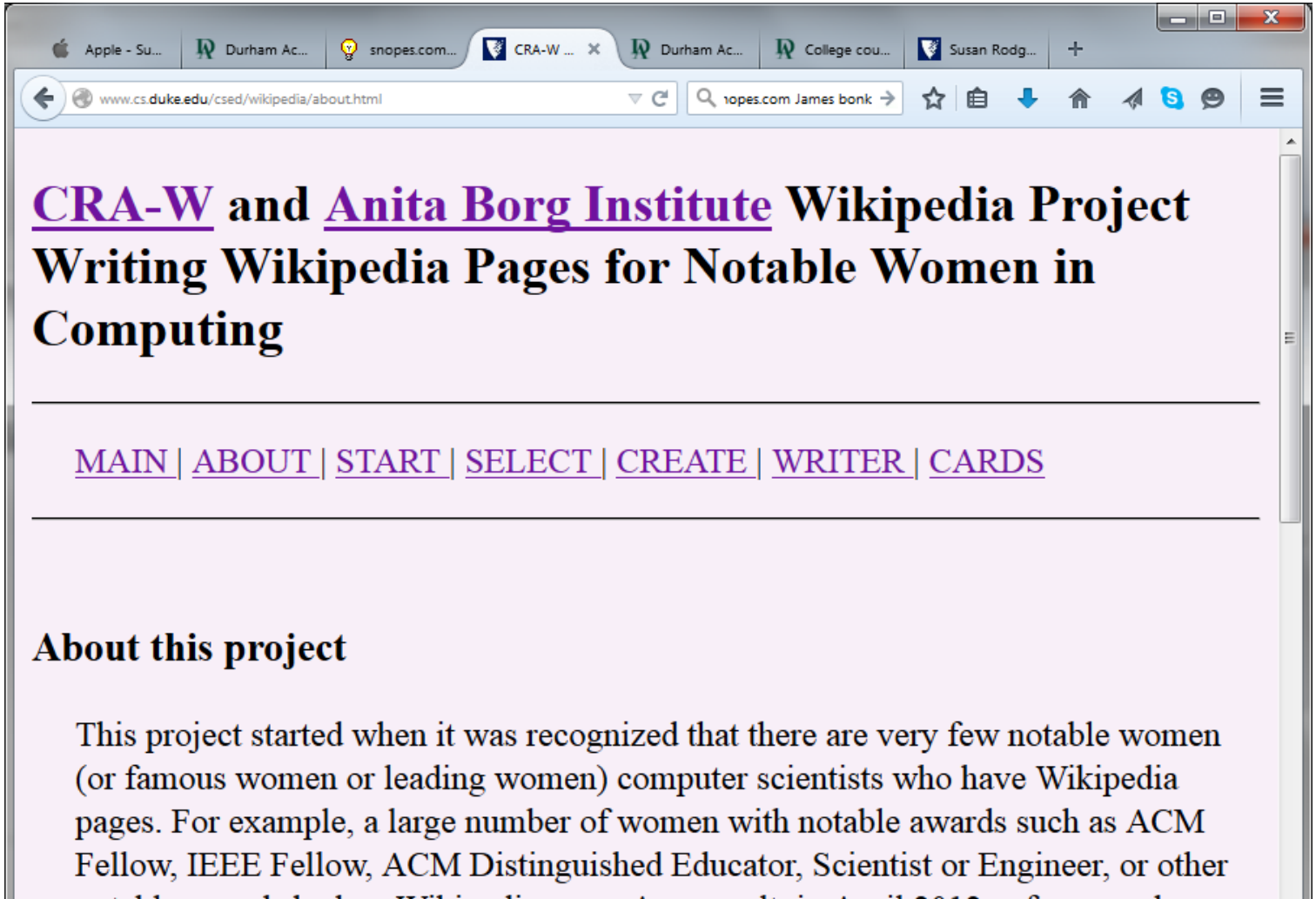
Toolbox

Print/export

international expert in computer architecture. Her research and teaching interests include computer architecture, embedded and mobile computing systems design, power and reliability aware design, and emerging technologies in computing systems. Dr. Irwin received her Ph.D. from the University of Illinois and an Honorary Doctorate from Chalmers University, Sweden.

Nationality	American
Fields	Computer Science, Com Architecture, Electronic Automation
Institutions	Pennsylvania State Univ
Alma	University of Illinois

Wrote a Guide on How to Write Wikipedia Biography www.cs.duke.edu/csed/wikipedia



Our Database of Notable Women in CS

- Over 300 women
- Why notable
- Status of their Wikipedia page
- Forms for adding women and updating status

	Title/Position	Web page	Prestigious Award or why notable	Wikipedia page?
	Professor of Human-Computer Interaction, CS	http://www.daimi.au.dk/~bodke	Member, CHI Academy	no page
	Founder	http://anitaborg.org/about/histor	WITI Hall of Fame, Fellow ACM, EFF Pioneer	has a page
at	Professor	http://polaris.gseis.ucla.edu/cb	ACM Fellow	has a page, needs work

3 female Turing Award Winners

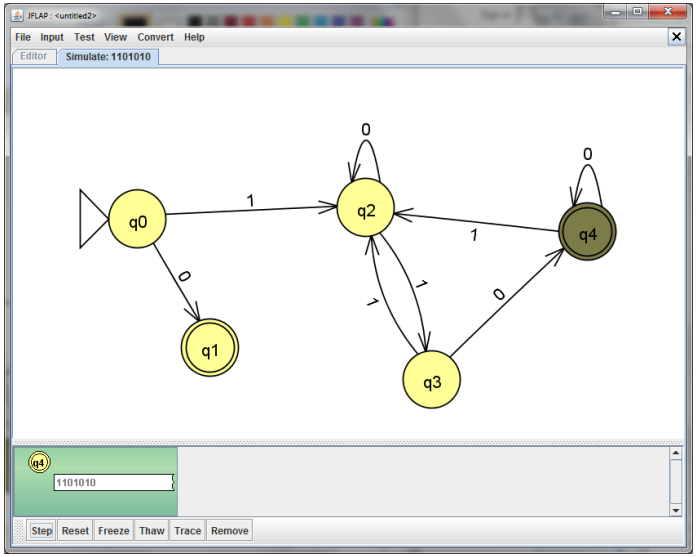
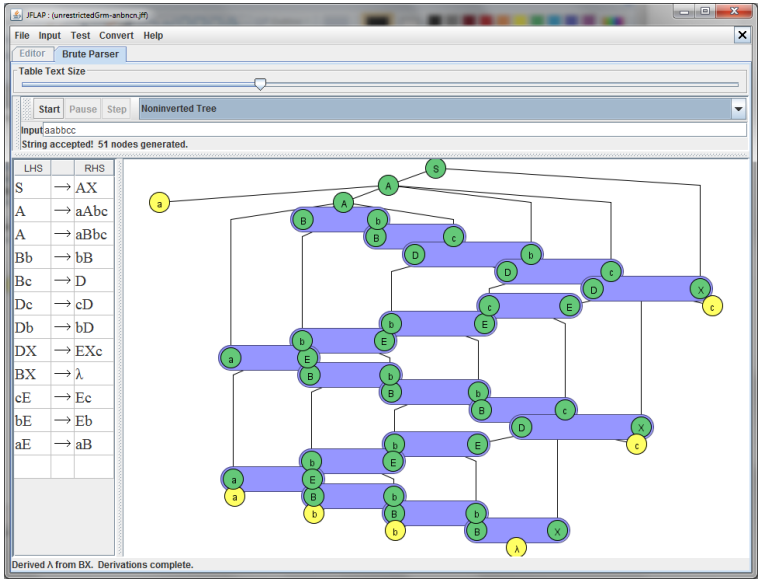
6 women Eniac Programmers

Anita Borg

Grace Hopper

ACM Fellows

Other

Thank You

- Questions?