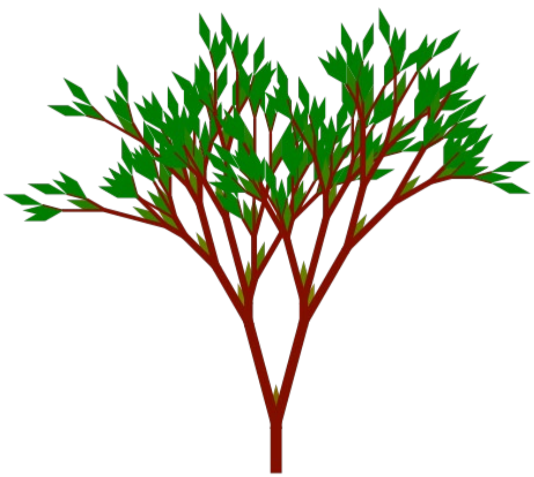
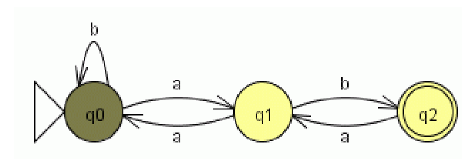
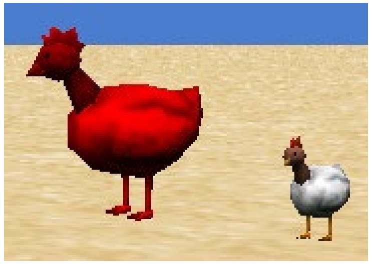
Through Visualization and Interaction, Computer Science Concepts Come Alive



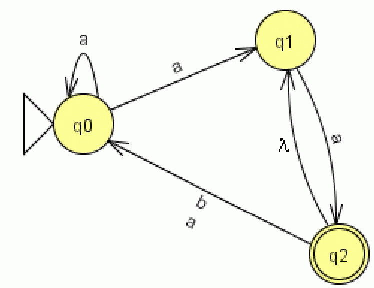
Susan H. Rodger
Duke University



March 29, 2022
Purdue University

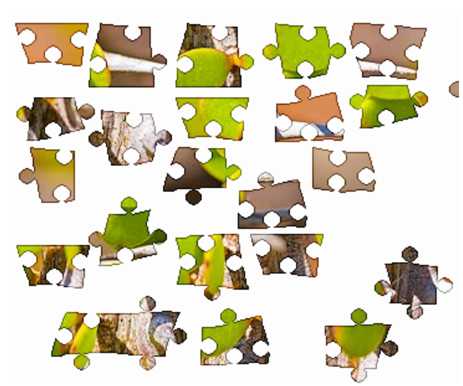


Outline



- My Path
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Solving Problems with Seven Steps
- Diversity Efforts

A long time ago, back in 1979….



NC State

B.S. Computer Science and Mathematics

- My first semester, my first course in programming - PL/I

```
Hello2: proc options(main);
     put list ('Hello, world!');
end Hello2;
```

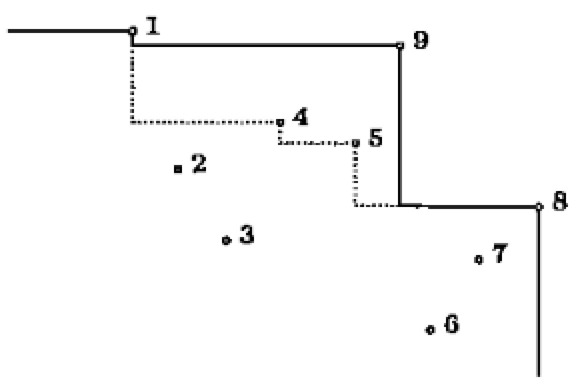
Decisions? Industry? Grad School?

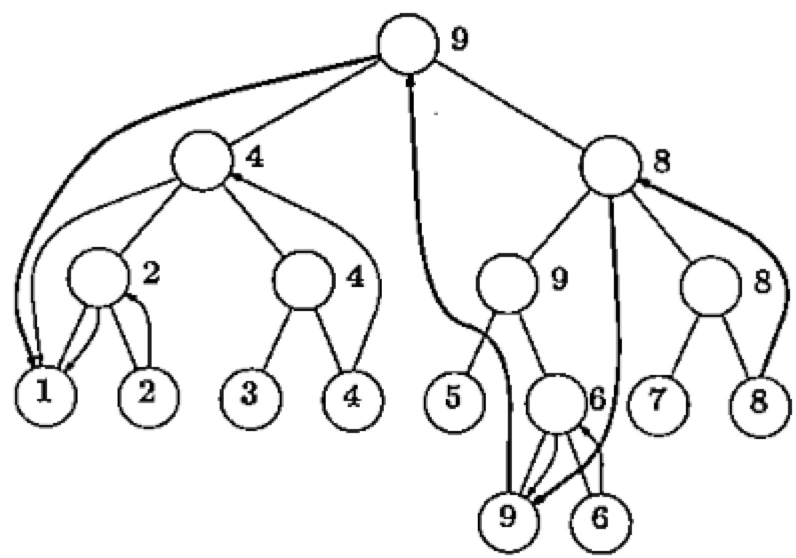
- Systems Programmer
  - NCSU,
    University Systems Control Center
- Undergraduate Research
  - Cleanup data from buoys in the water
- Wasn't thinking about grad school
- Be sure to encourage students to think about graduate school!

- Started in 1983
- Teaching Assistant for intro programming in Fortran
- Punch cards…
- In trouble with email…

Finished Graduate School!

- PhD Purdue University 1989
  - Computational Geometry
  - Parallel Scheduling Algorithms
- New Data Structure
  - Dynamic contour search tree





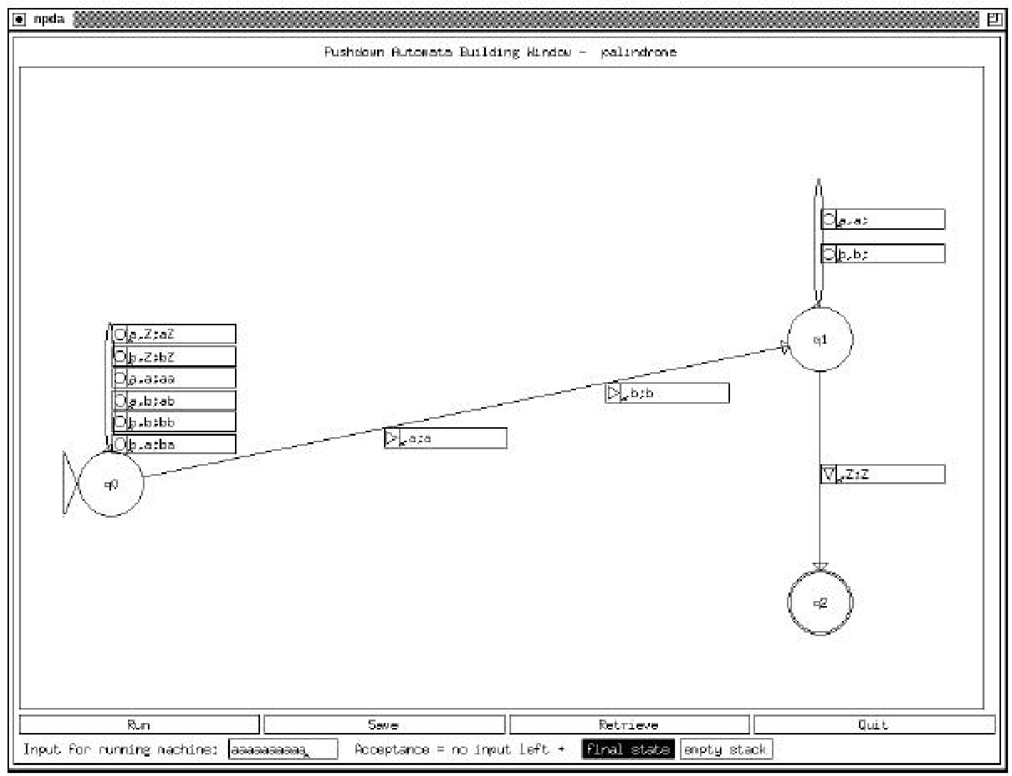
Rensselaer

Assistant Professor

- Continued research in algorithms
- CAREER CHANGE….
- Got more interested in education

Started developing education tools Changed area to Visualization Tools and CS Education

- Tool – NPDA - to experiment with pushdown automata



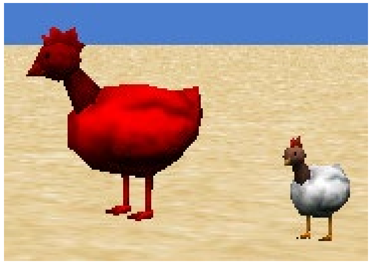
1994 – Moved to Duke University Professor of the Practice

- Position focuses on Education in the Discipline

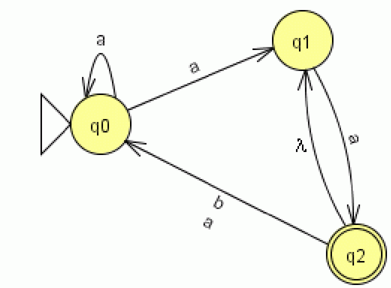
About Me - Hobby – Baking Shape cakes

The Wiggles magazine Issue No. 42

How do you make those cakes?



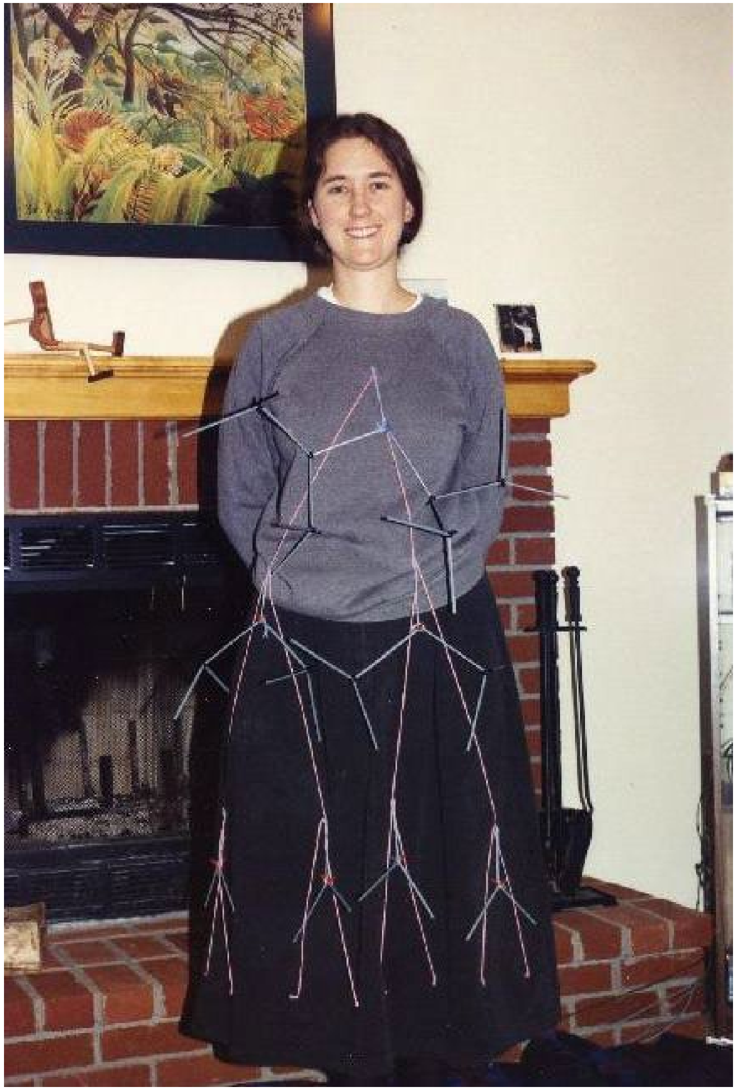
Outline



- My Path
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Solving Problems with Seven Steps
- Diversity Efforts

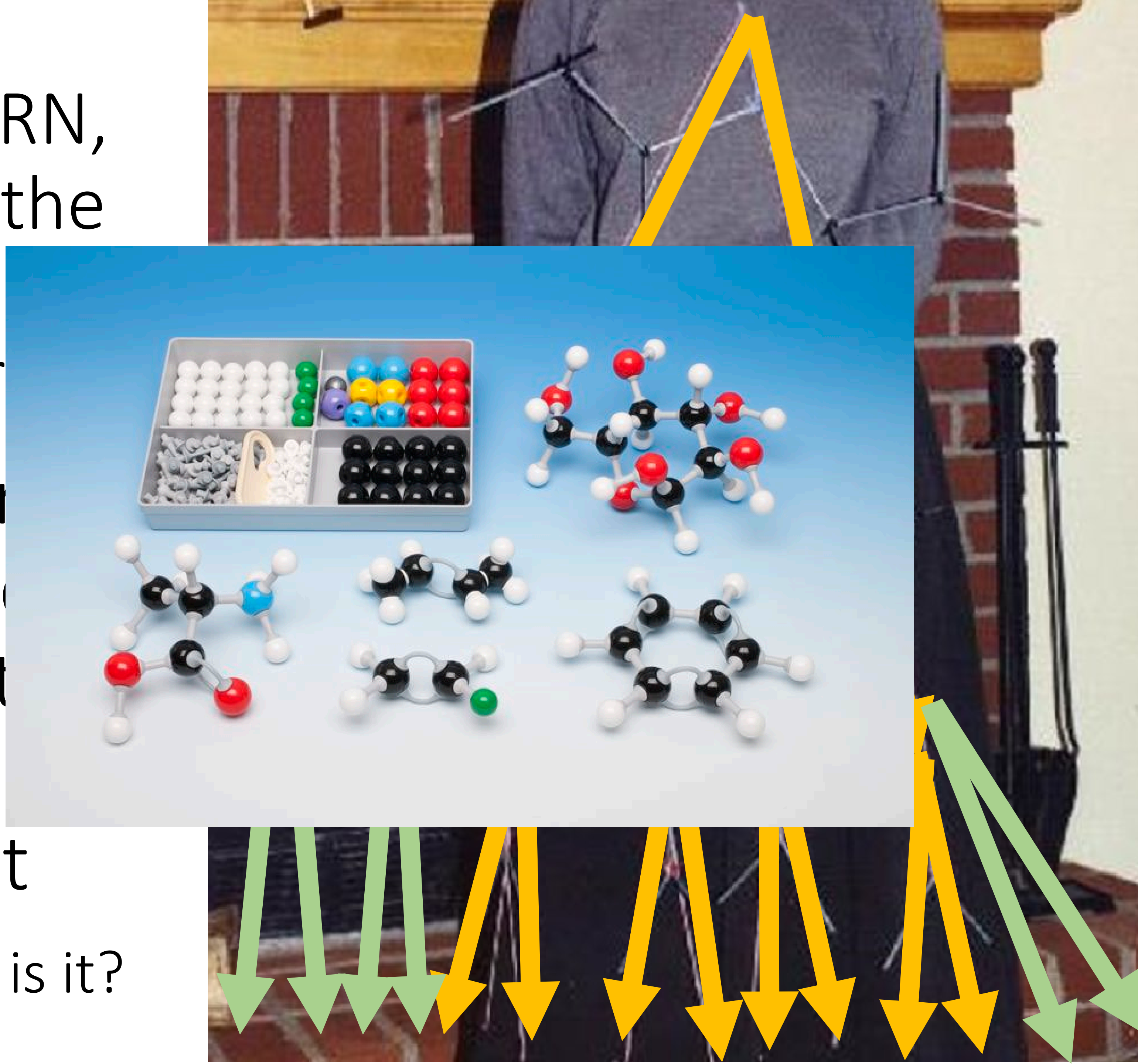
CS Concepts Coming Alive

- What data structure is this?



YARN,
in the
shap
binar

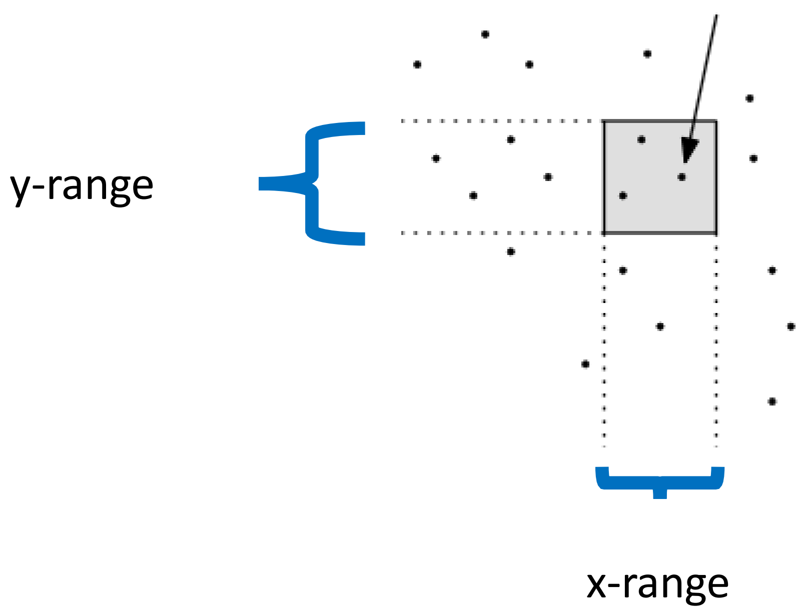
Subtr
mac
wit
mole
kit



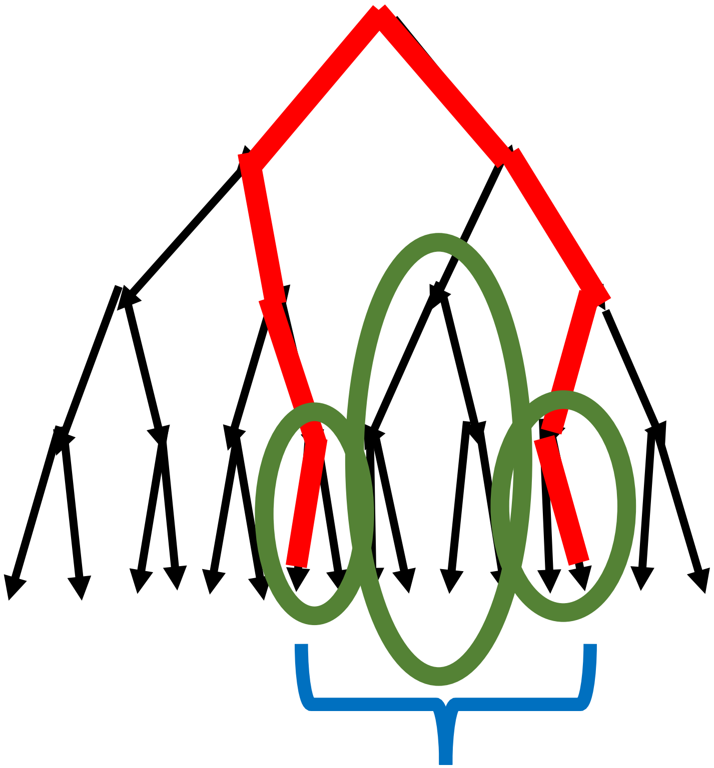
What is it?

2D-range tree

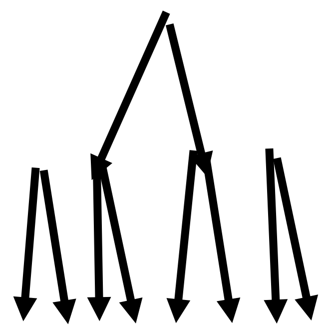
- Search in x-y plane
- Main tree organized by x-values
- Subtree organized by y values



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

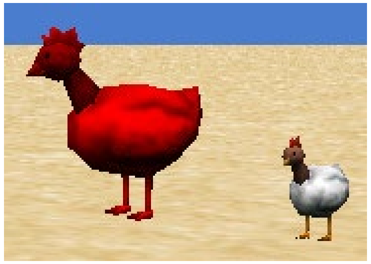


In the x-range

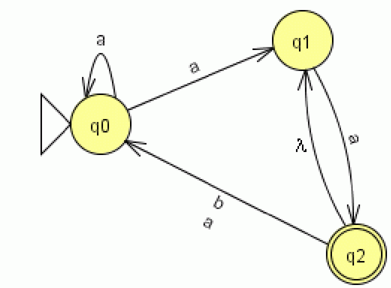


Each subtree organized by y-value

Search each subtree by y-value



Outline



- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Solving Problems with Seven Steps
- Diversity Efforts Sprinkled in…

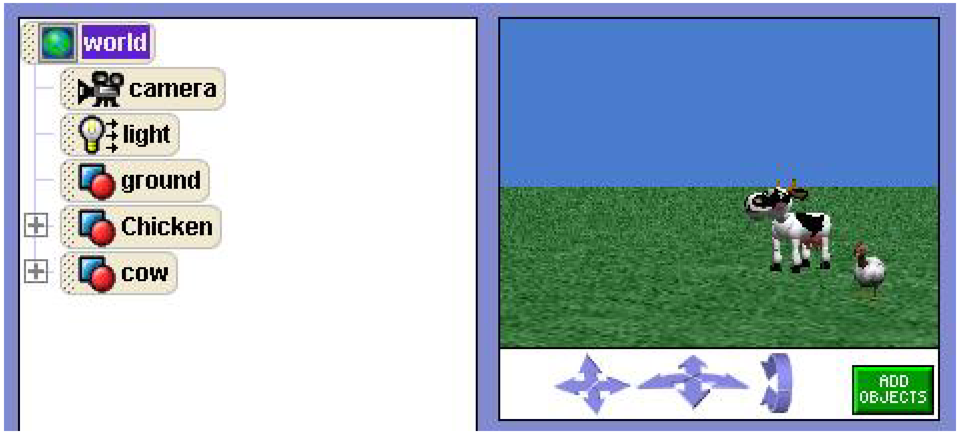
Alice Programming Language

- 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

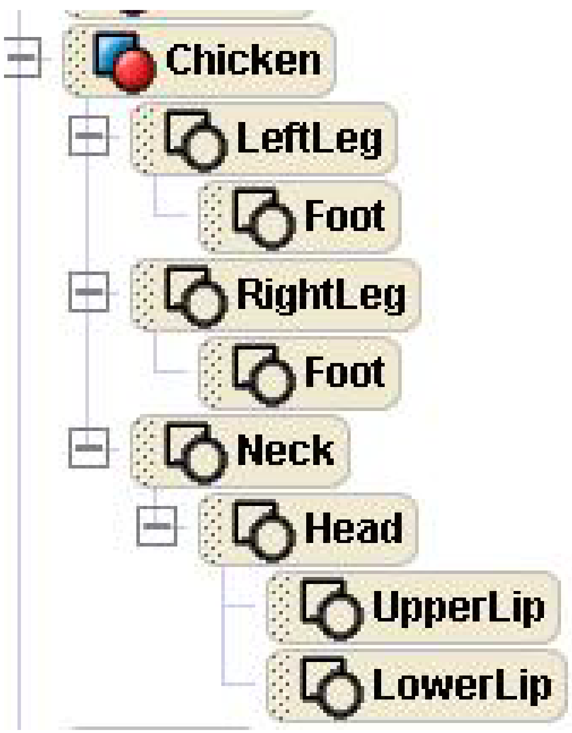
More on … Alice Programming Language

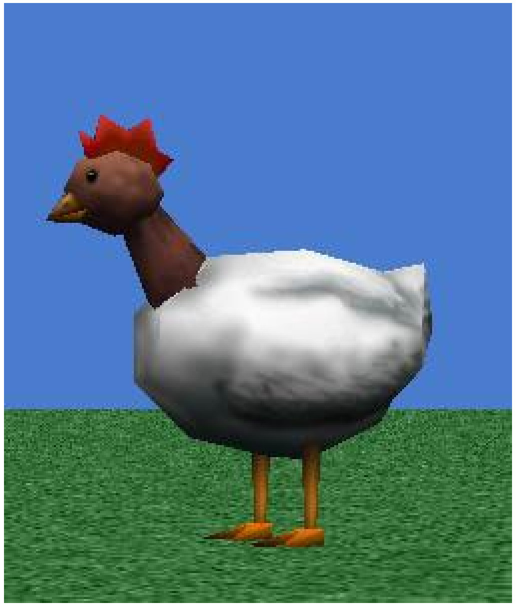
- Has libraries of 3D objects

- Keeps Track of objects you select



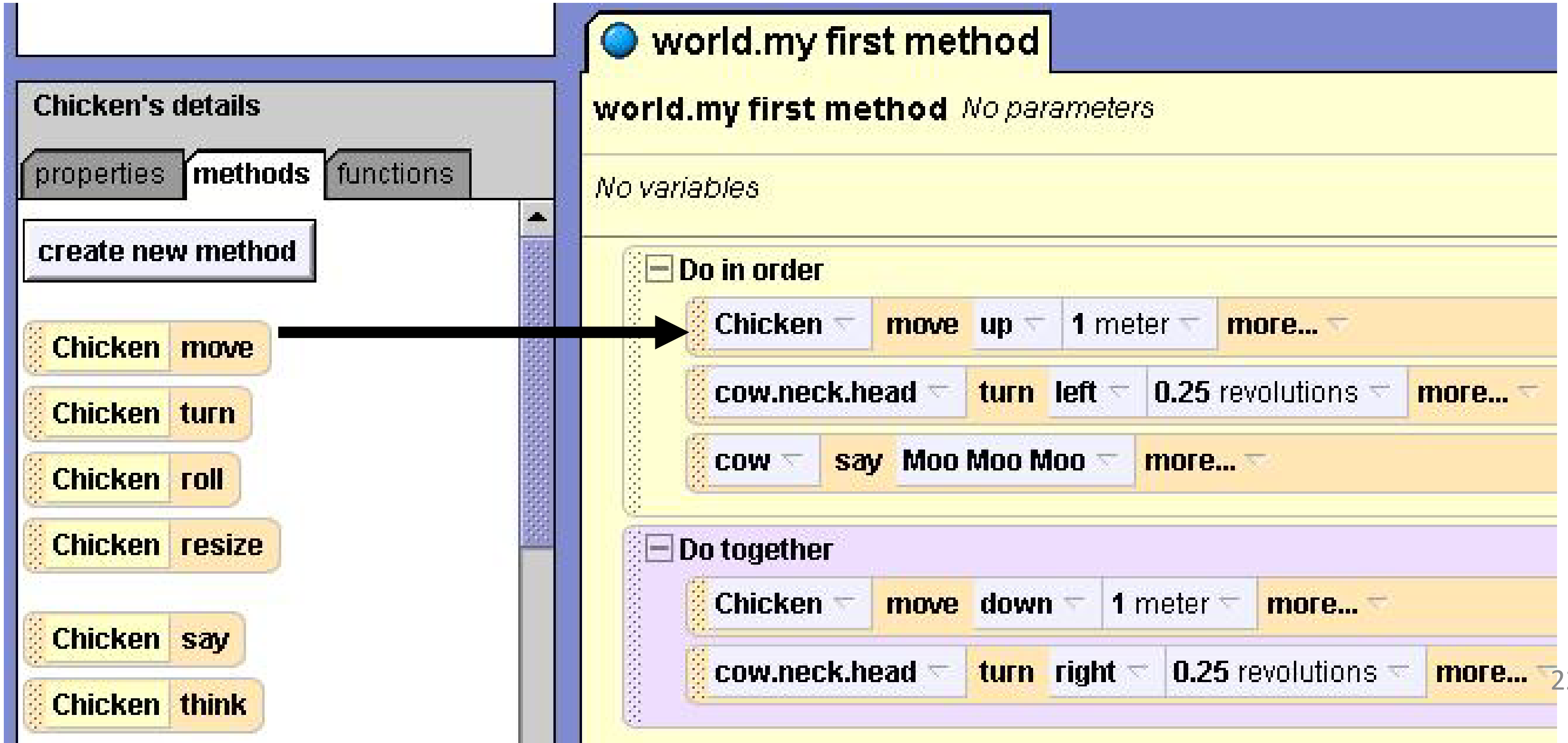
Objects Have Multiple Parts that are moveable





Alice Code is Easy to Learn

Select Code, Drag-and-Drop code in program



Play Alice Animation

- Chicken rises, cow turns head and talks

Computer Science Concepts come alive with Alice - Examples

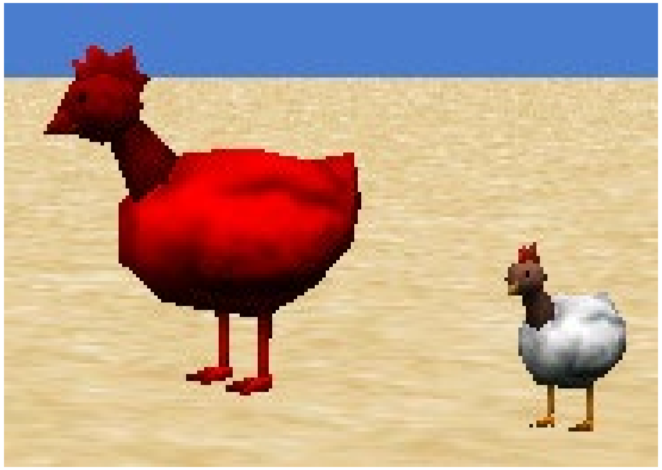
- Objects - visible
- Variables - see how they are changing
- Inheritance - visual
- Lists/Arrays - visual

Objects are visible

Variables – Timer and Score

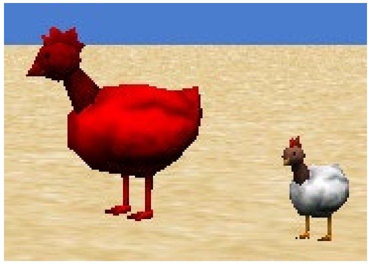
Example - Inheritance

- 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

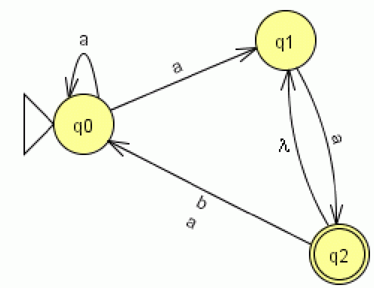


Example list

Example – Arrays
Shuffle, then Selection Sort



Outline



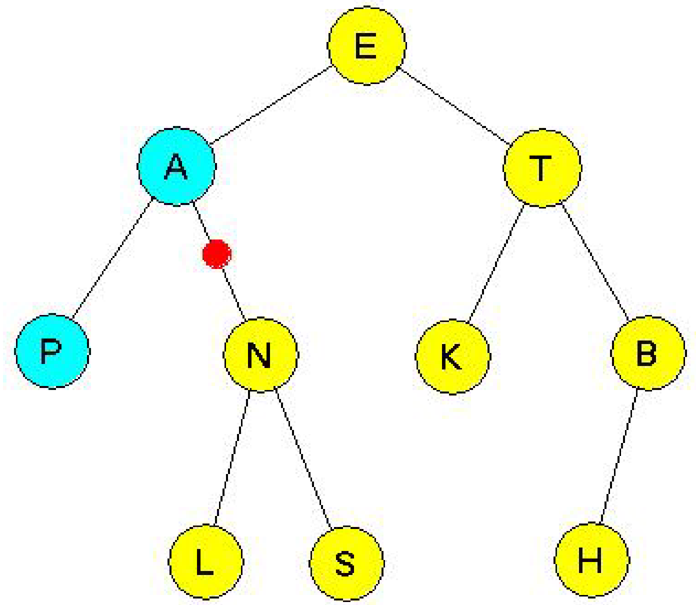
- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Additional Ways to Engage with CS
- Diversity Efforts

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!

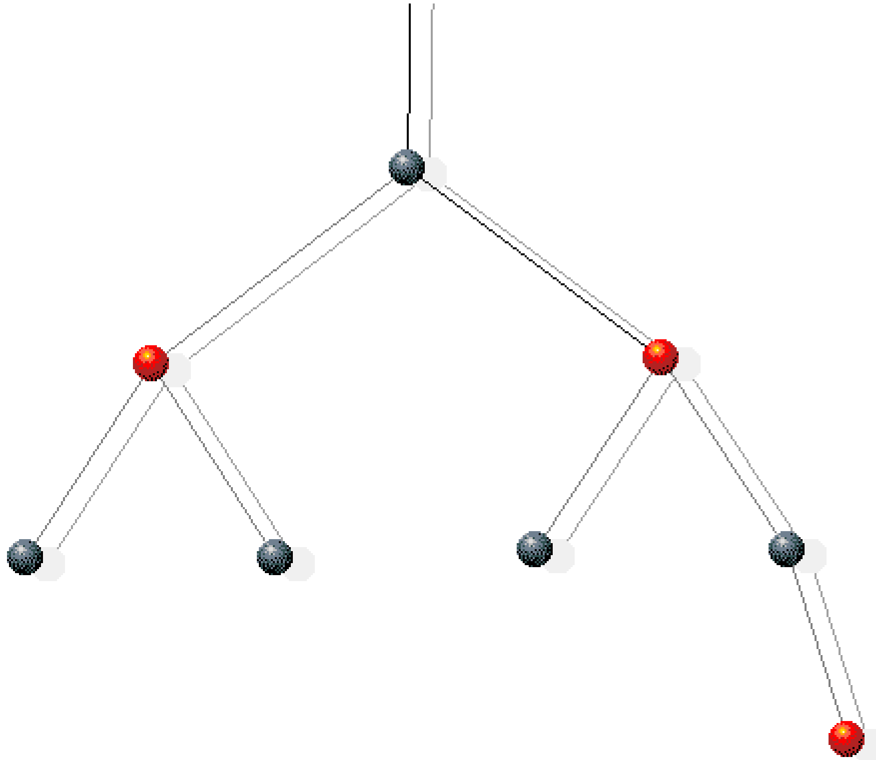
Use of Algorithm Animation in CS 1/2

- 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]

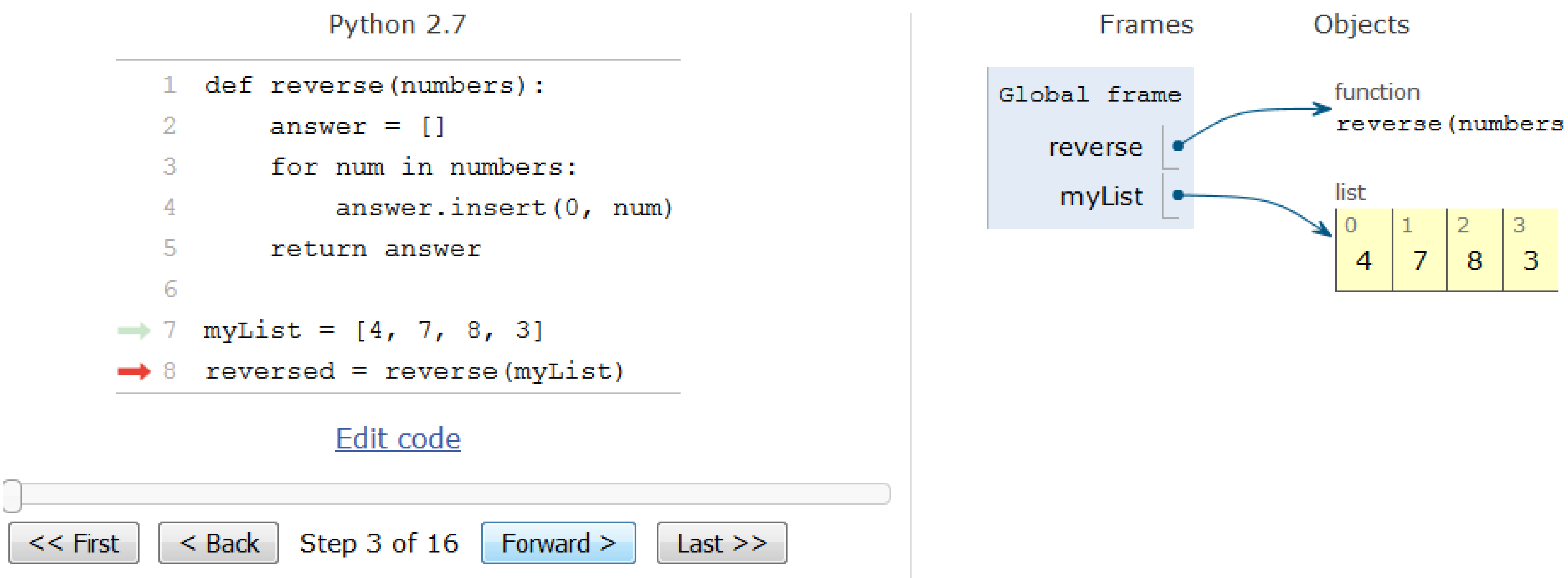


Student must have graduated. Link no longer works!

X

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

Python Tutor
Compute reverse of a list



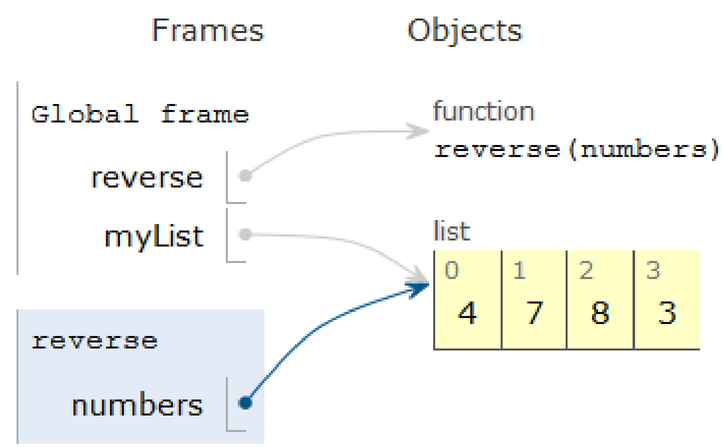
Python Tutor
Compute reverse of a list

Python 2.7

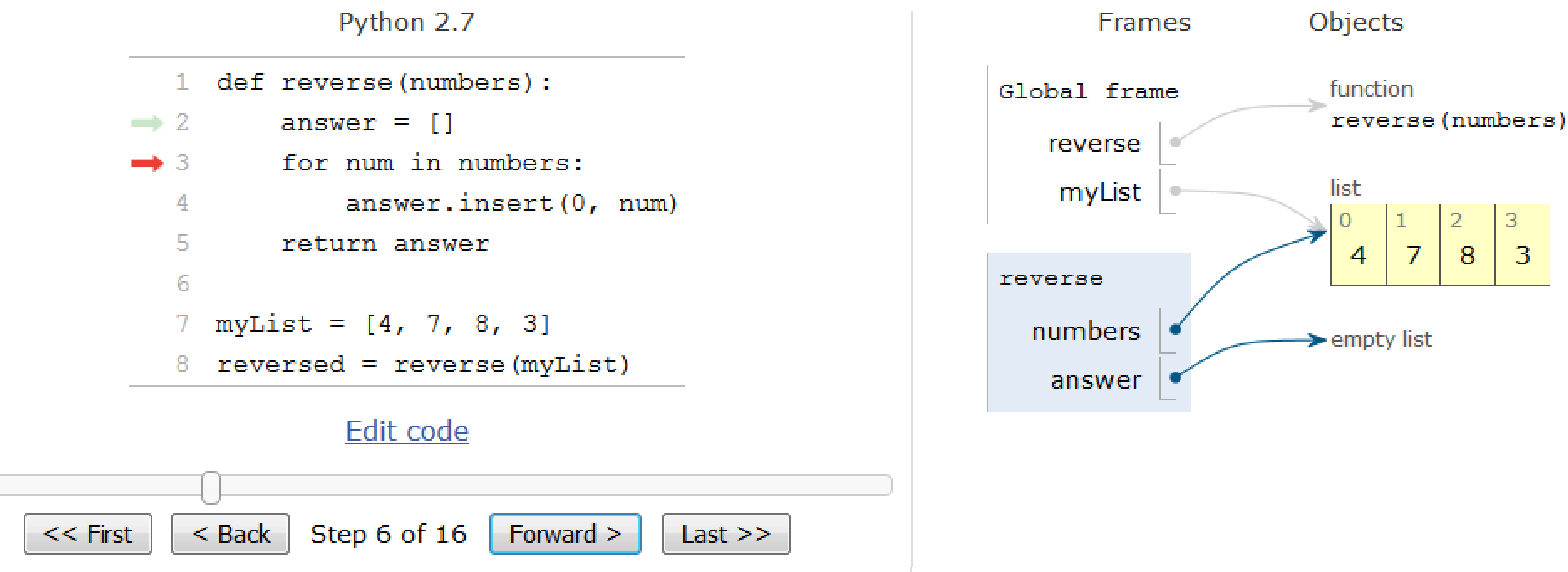
```
def reverse(numbers):
    answer = []
    for num in numbers:
        answer.insert(0, num)
    return answer

myList = [4, 7, 8, 3]
reversed = reverse(myList)
```

Edit code

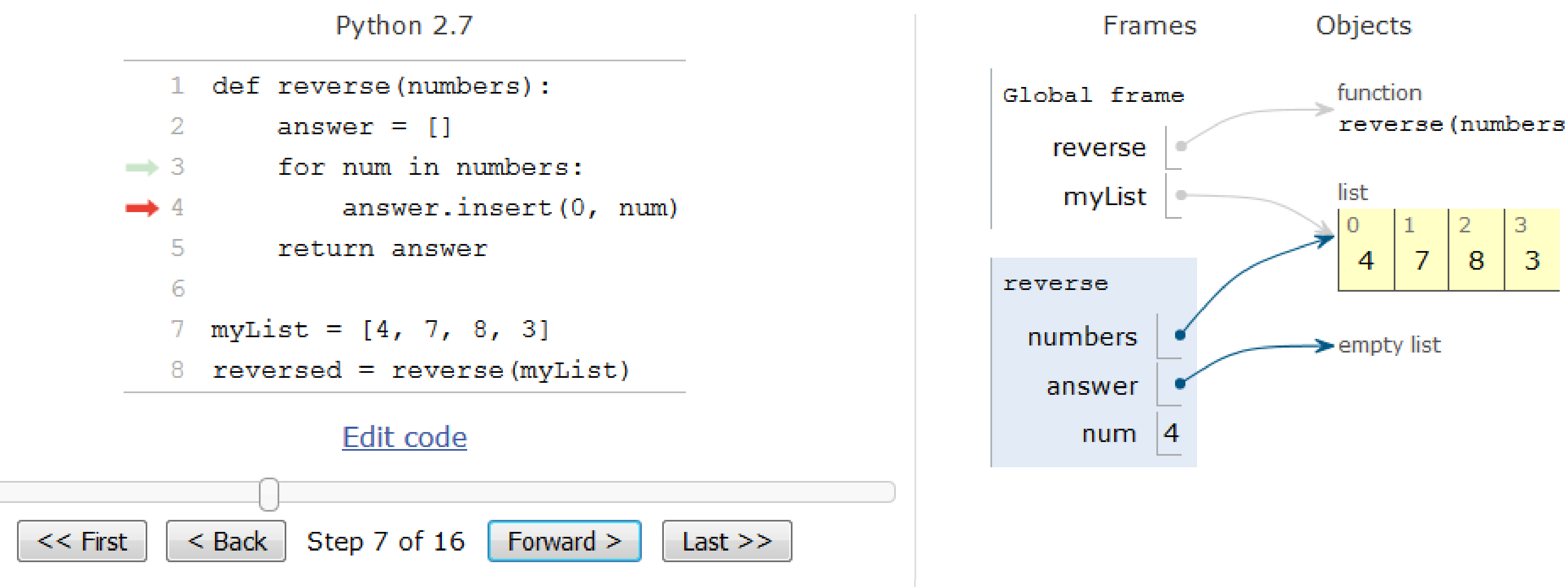
<< First | < Back | Step 4 of 16 | Forward > | Last >>



Python Tutor
Compute reverse of a list



Python Tutor
Compute reverse of a list



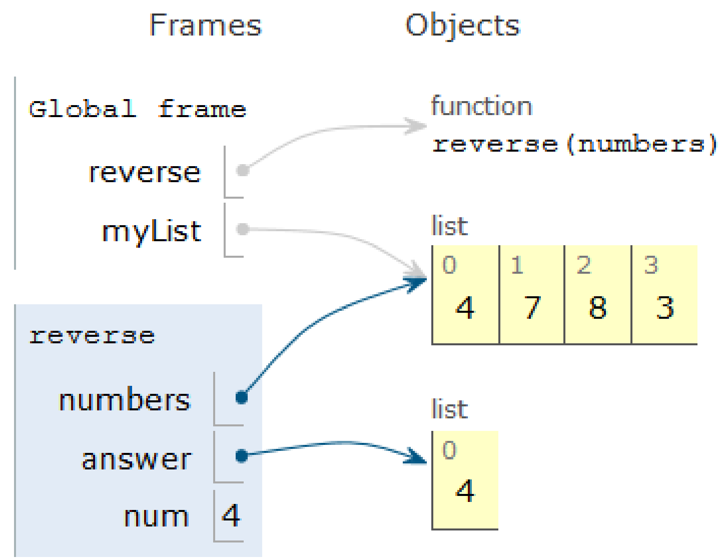
Python Tutor
Compute reverse of a list

Python 2.7

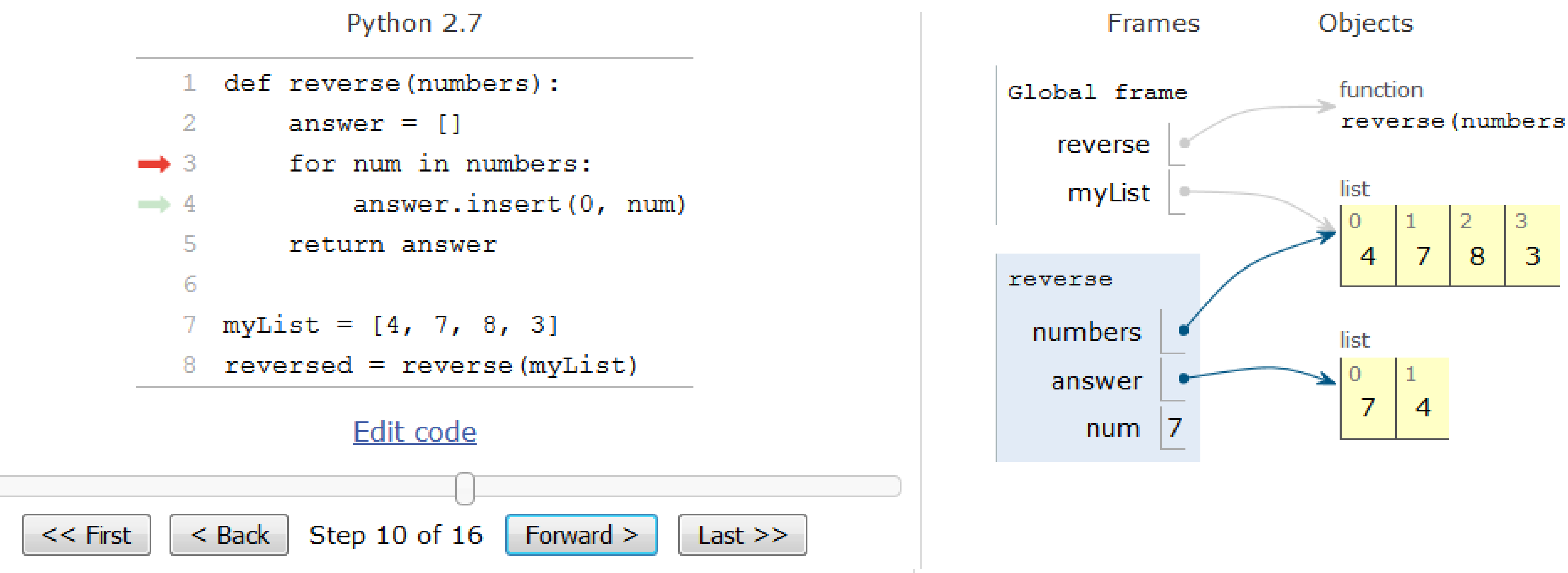
```
def reverse(numbers):
    answer = []
    for num in numbers:
        answer.insert(0, num)
    return answer

myList = [4, 7, 8, 3]
reversed = reverse(myList)
```

Edit code

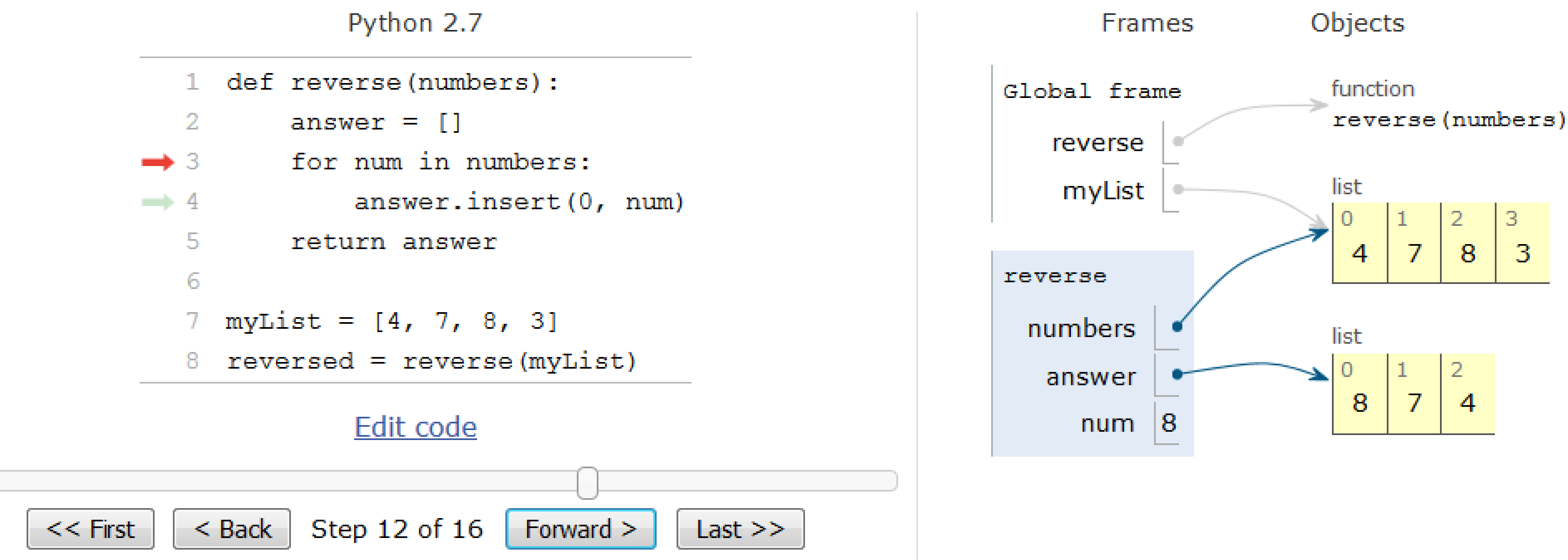
<< First | < Back | Step 8 of 16 | Forward > | Last >>



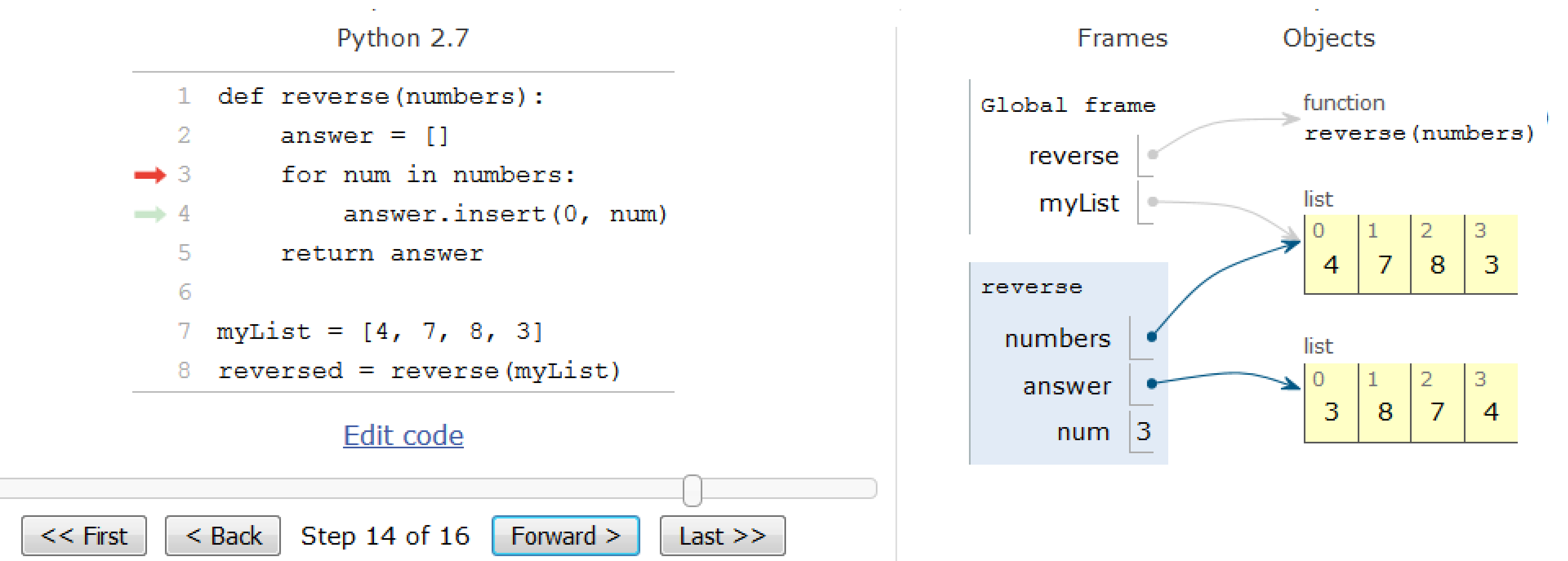
Python Tutor
Compute reverse of a list



Python Tutor
Compute reverse of a list



Python Tutor
Compute reverse of a list



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
- ZyBooks – 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.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
L	u	t	h	e	r		C	o	l	l	e	g	e
-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

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

```
school = "Luther College"
m = school[2]
print(m)

lastchar = school[-1]
print(lastchar)

```

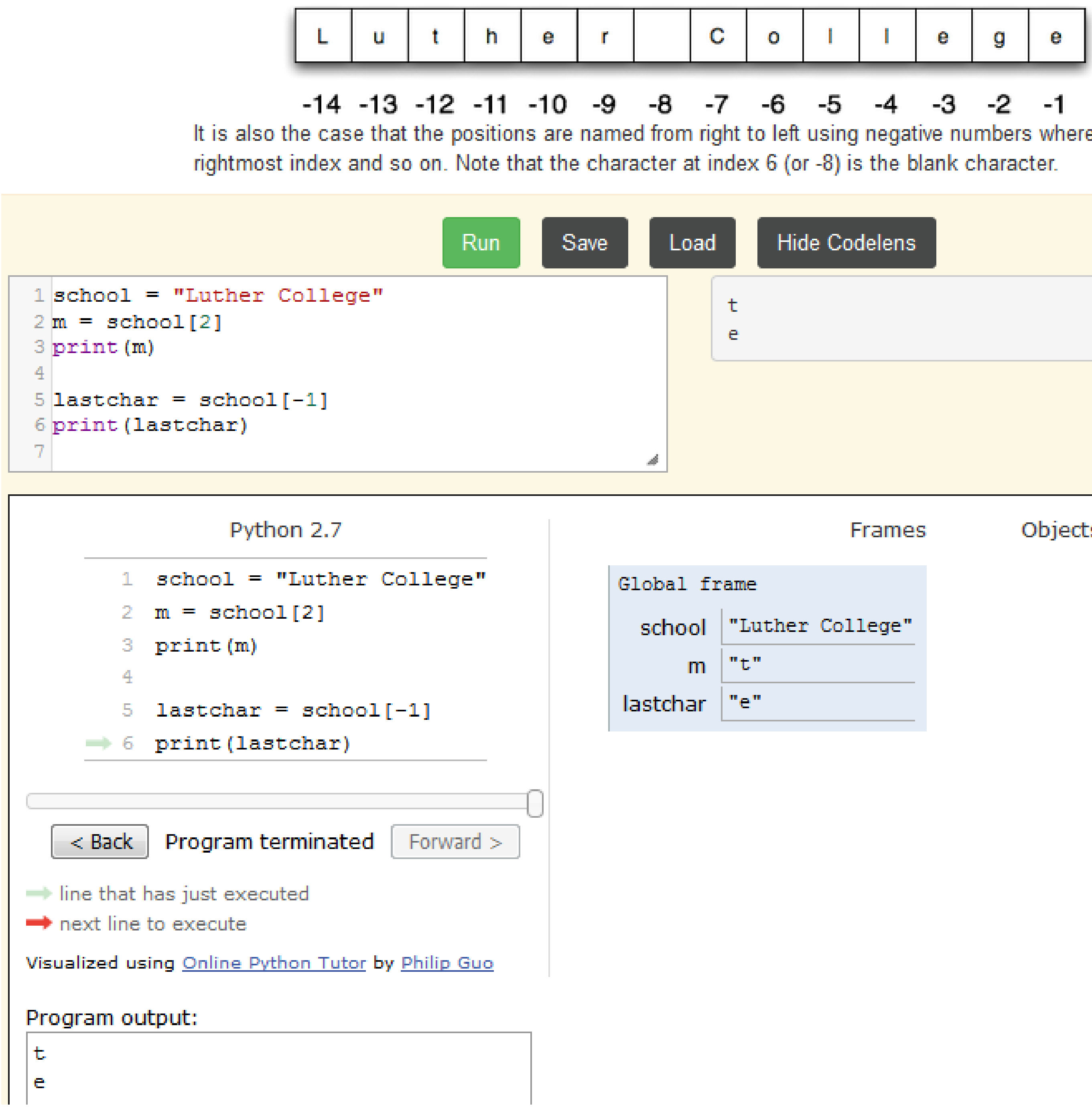
Run and edit code in the book

Run Save Load Show CodeLens

```python
school = "Luther College"
m = school[2]
print(m)

lastchar = school[-1]
print(lastchar)
```

```
t
e
```

Integrates in Python Tutor



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.

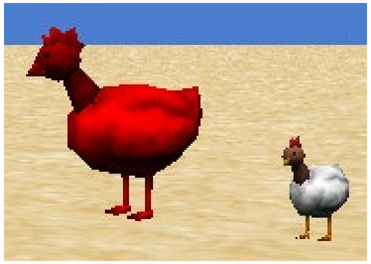
strings-4-2: What is printed by the following statements?

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

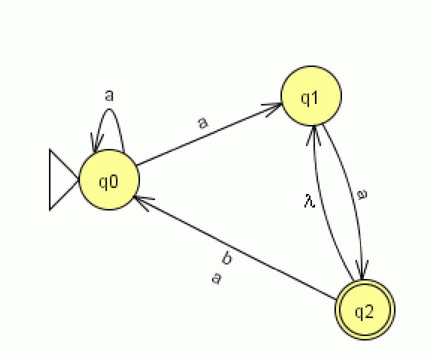
- tr
- ps
- nn
- Error, you cannot use the [ ] operator with the + operator.

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
  - Solving Problems with Seven Steps
  - Additional Ways to Engage with CS
- Diversity Efforts

How does a compiler work? 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. (modelled using a pushdown automaton)

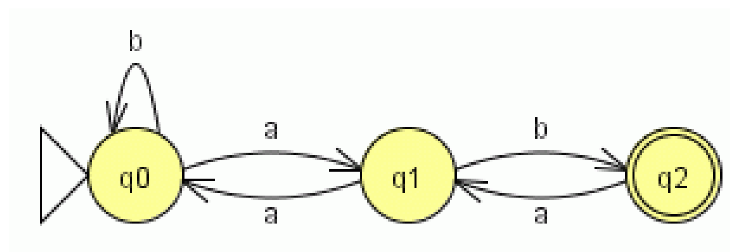
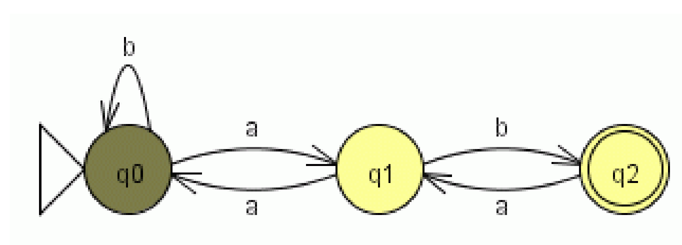
- This area is known as Formal languages and Automata theory

Formal Languages and Automata Theory

- Traditionally taught
  - Pencil and paper exercises
  - No immediate Feedback!

- More mathematical than programming
- Less hands-on than most CS courses

Why Develop Tools for Automata?

<table>
<tr><td>Textual</td><td>$(\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$
$\delta = \{(q_0, b, q_0), (q_0, a, q_1), (q_1, a, q_0), (q_1, b, q_2), (q_2, a, q_1)\}$</td></tr>
<tr><td>Tabular</td><td>$$\begin{array}{c||c|c} & \text{a} & \text{b} \\ \hline\hline q_0 & q_1 & q_0 \\ \hline q_1 & & q_2 \\ \hline q_2 & & \end{array}$$</td></tr>
<tr><td>Visual</td><td>
</td></tr>
<tr><td>Interactive</td><td>
</td></tr>
</table>

Overview of JFLAP

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

Thanks to Students - Worked on JFLAP and Automata Theory Tools

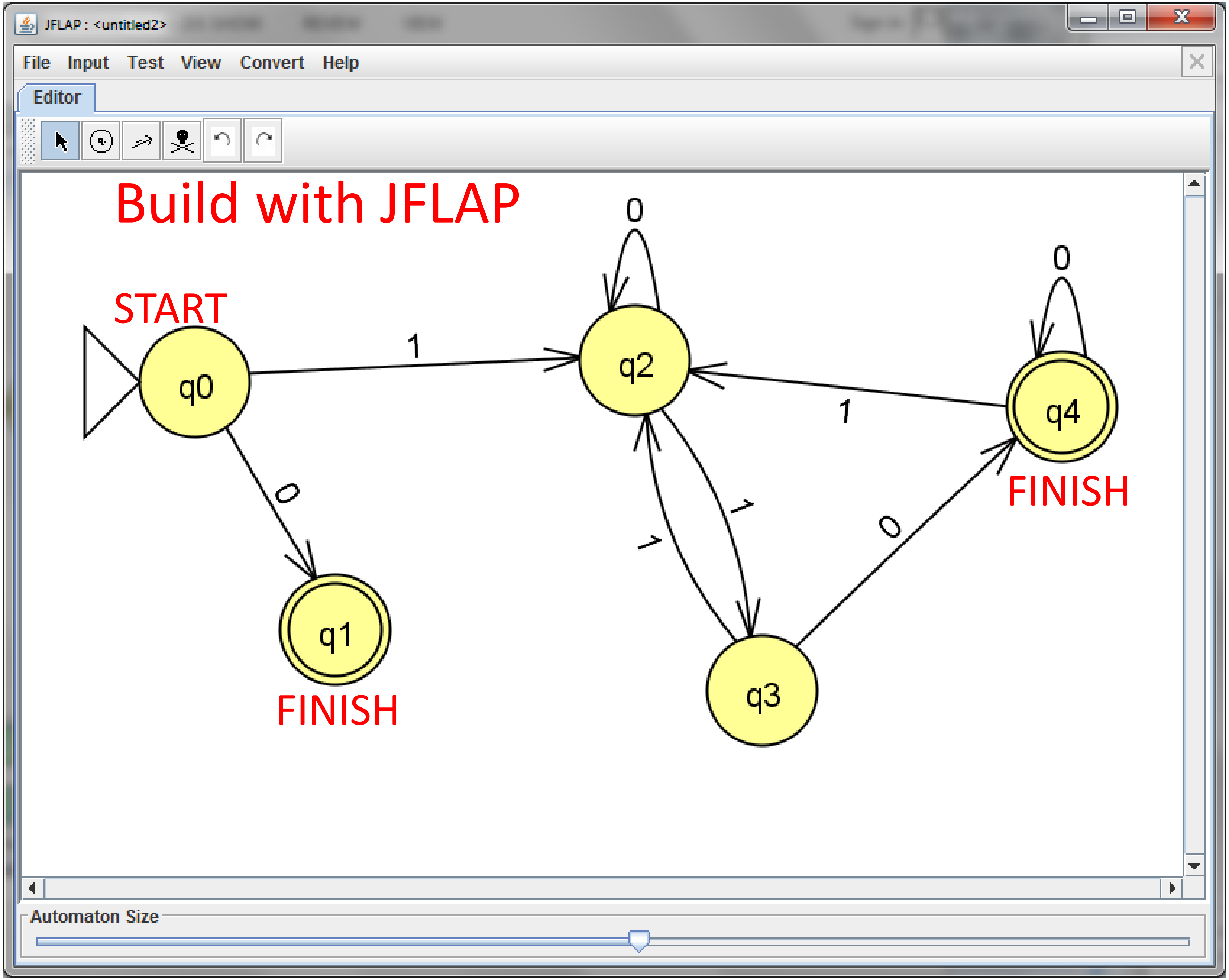
Over 30 years!

- NPDA - 1990, C++, Dan Caugherty
- FLAP - 1991, C++, Mark LoSacco, Greg Badros
- JFLAP - 1996-1999, Java version
  Eric Gramond, Ted Hung, Magda and Octavian Procopiuc
- Pâté, JeLLRap, Lsys
  Anna Bilska, Jason Salemme, Lenore Ramm, Alex Karweit, Robyn Geer
- JFLAP 4.0 – 2003, Thomas Finley, Ryan Cavalcante
- JFLAP 6.0 – 2005-2008 Stephen Reading, Bart Bressler, Jinghui Lim, Chris Morgan, Jason Lee
- 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
- Yu and Pester (2016), Yeh and Fang (2017), Patel (2018)

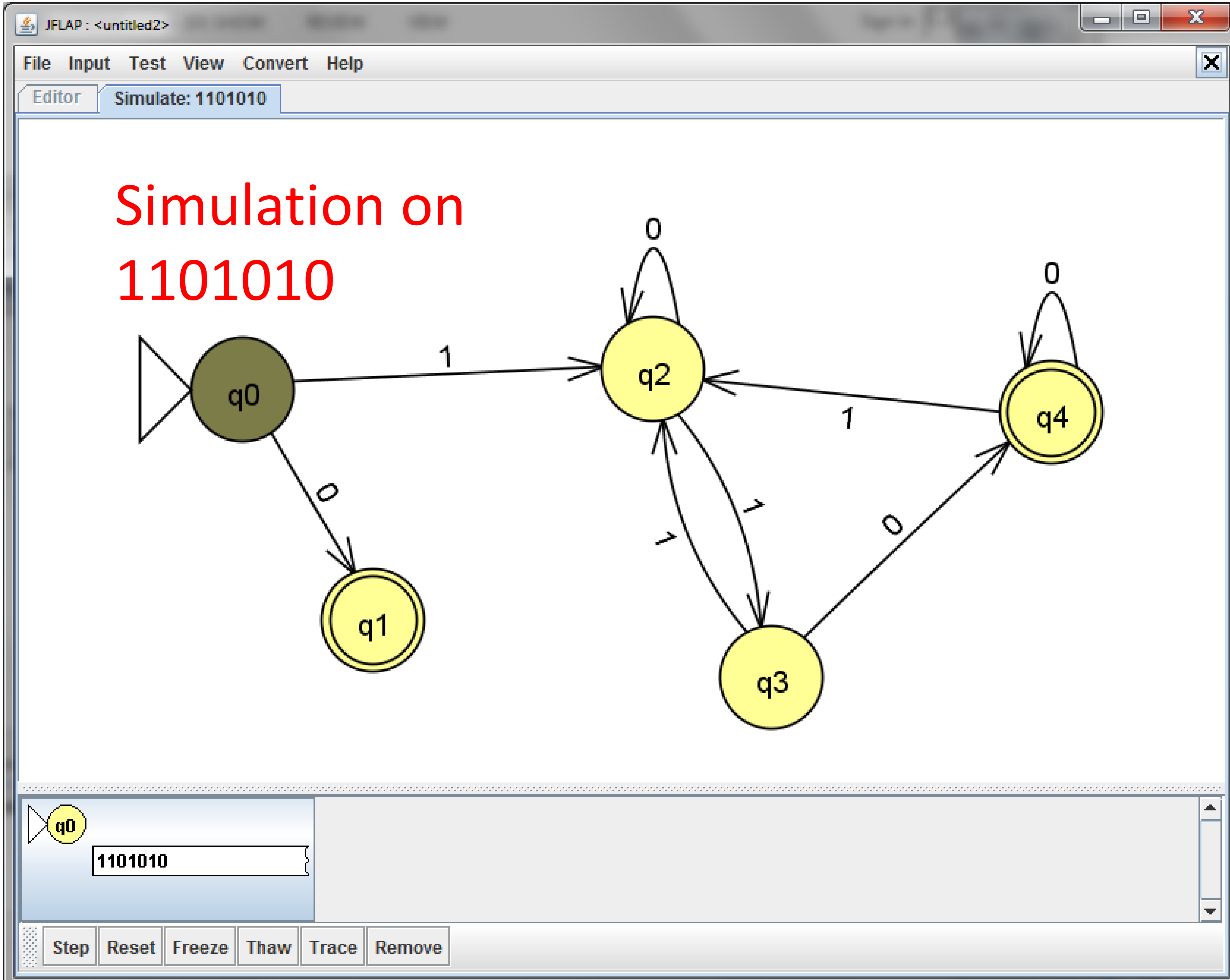
DFA Example

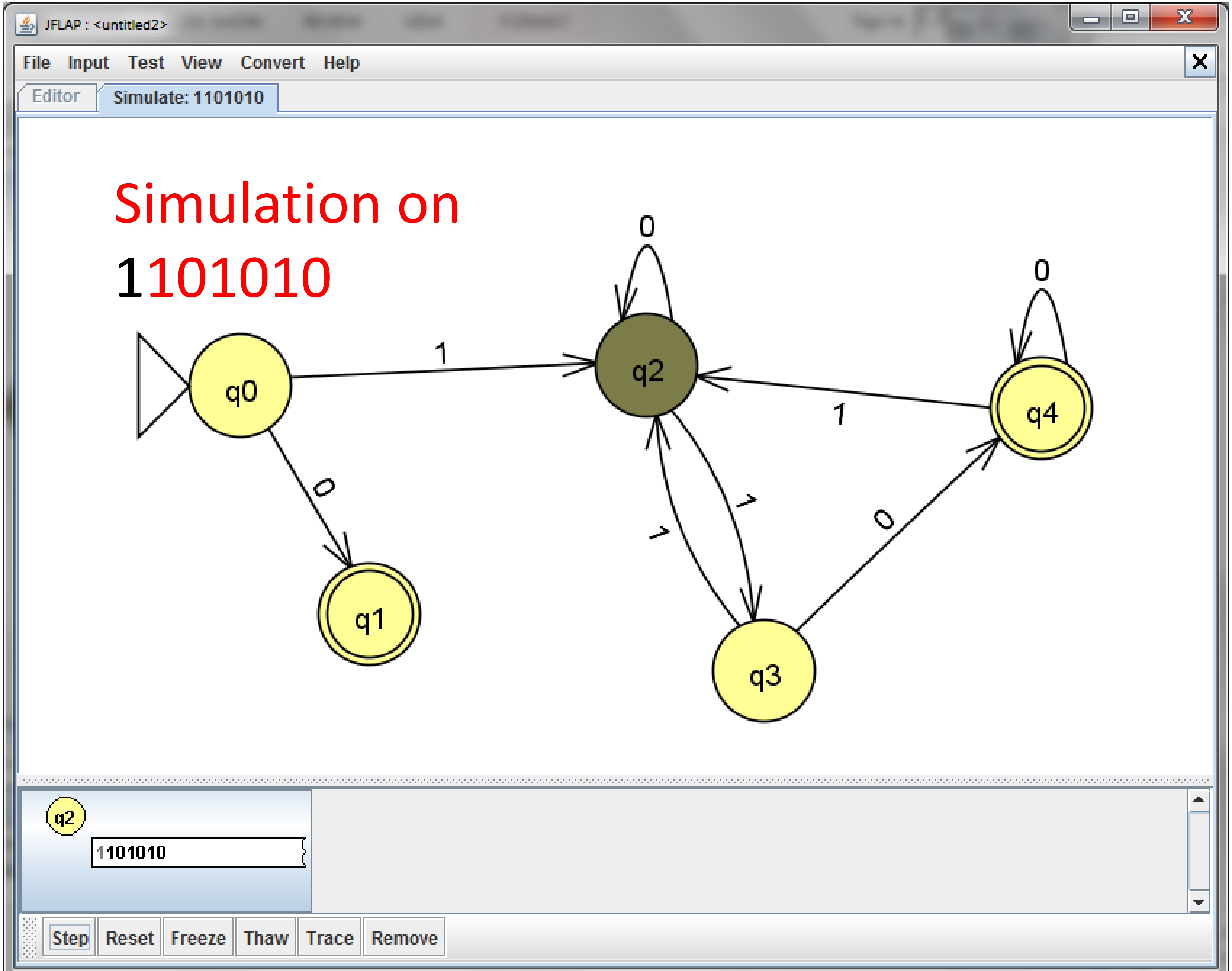
- Build a deterministic finite automaton(DFA) to recognize even binary numbers with an even number of 1s.
- Only use symbols 0 and 1
- Binary numbers: 0, 1, 10, 11, 100, 101, 110, 111, …
- When is a binary number an even number?
  - Ends in 0
- Which strings should be accepted?
- 11010, 10010, 1111,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

11010,	10010,	1111,	10100
No, odd no. of 1's	Yes	No, ends In 1	Yes

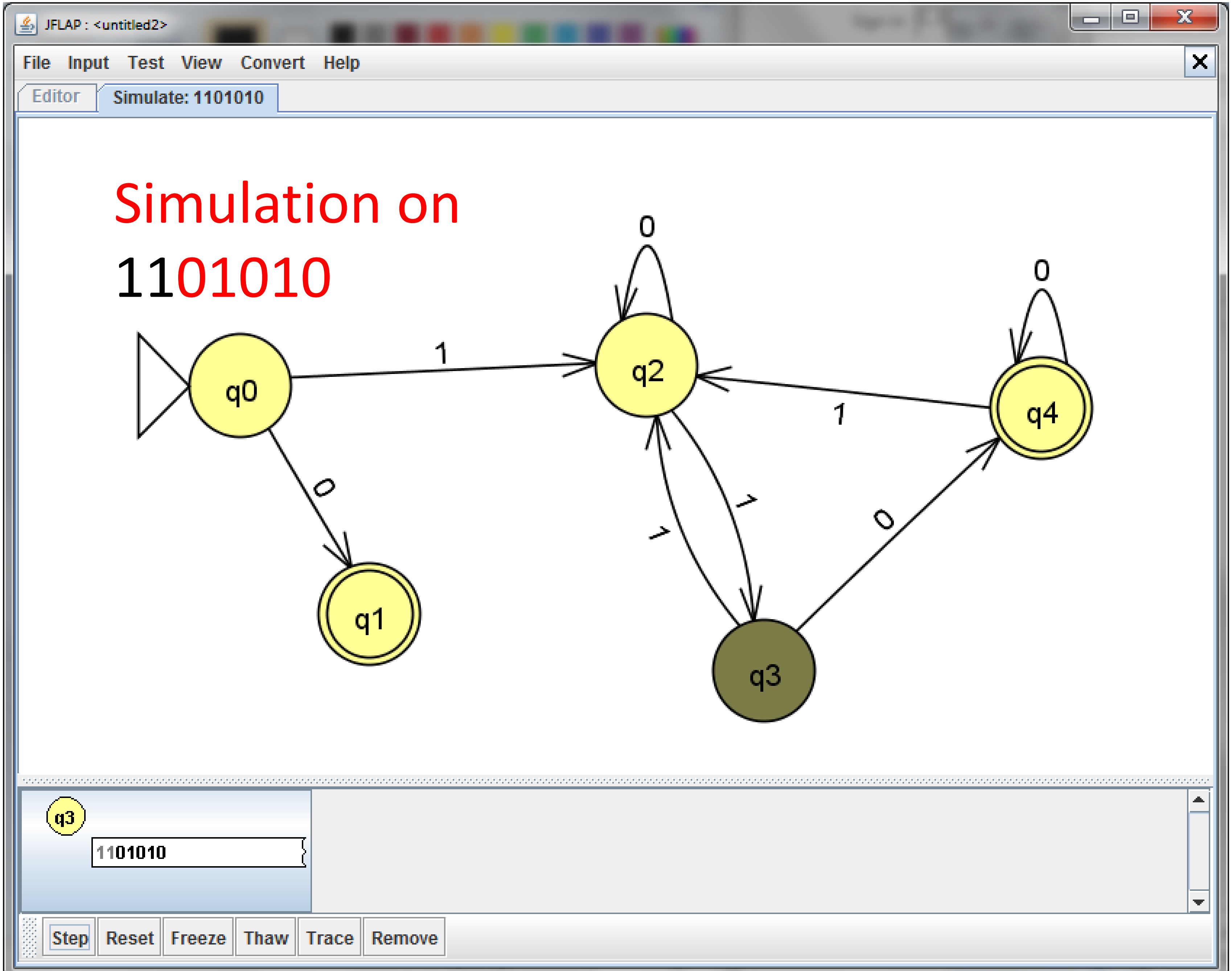

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Build with JFLAP
START
q0
1
q2
0
0
q4
1
0
q1
FINISH
1
1
0
FINISH
q3
Automaton Size

Simulation on 1101010





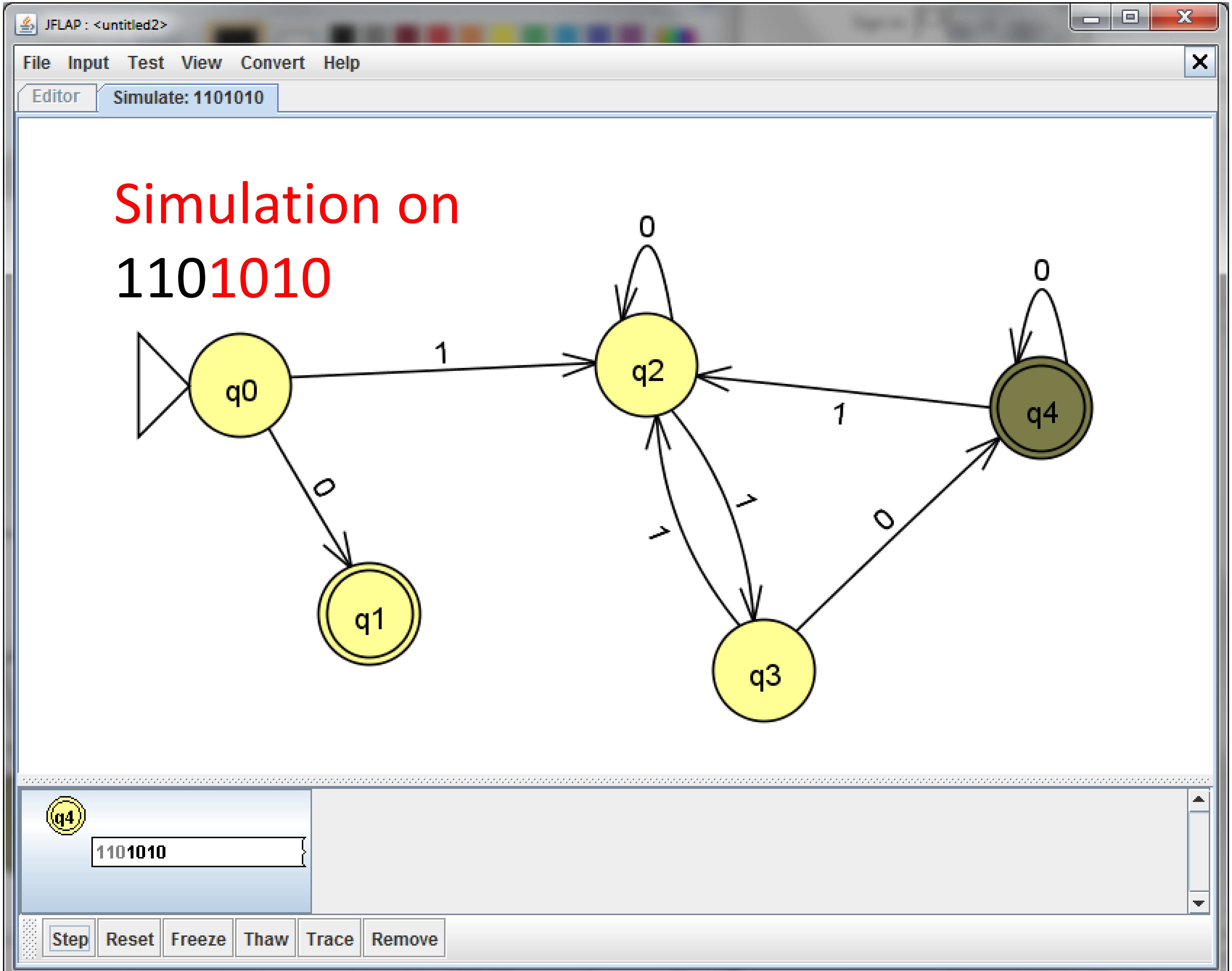
JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on
1101010
q0
q1
q2
q3
q4
0
1
0
1
1
1
0
0
q2
1101010
Step Reset Freeze Thaw Trace Remove



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

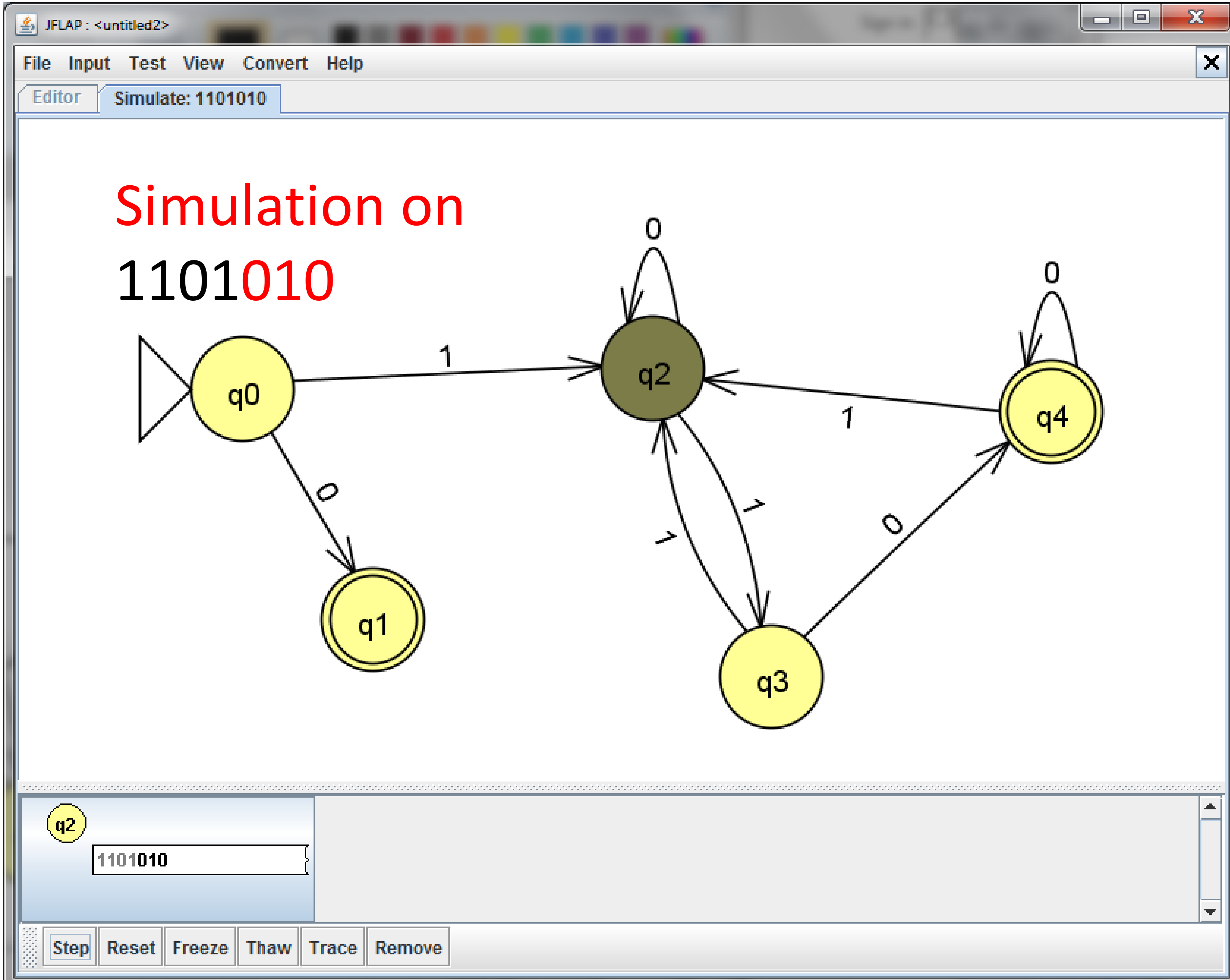
Simulation on

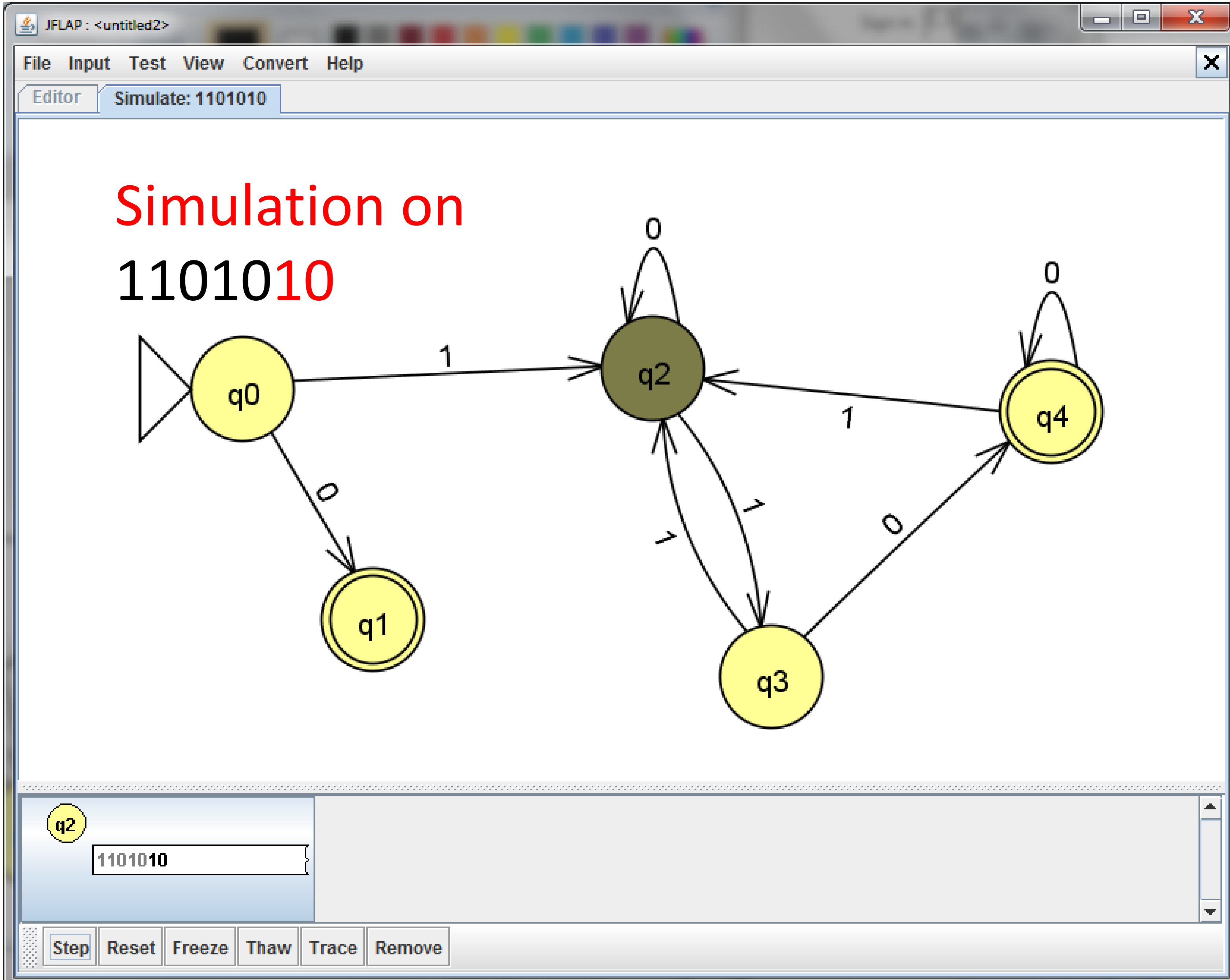
1101010

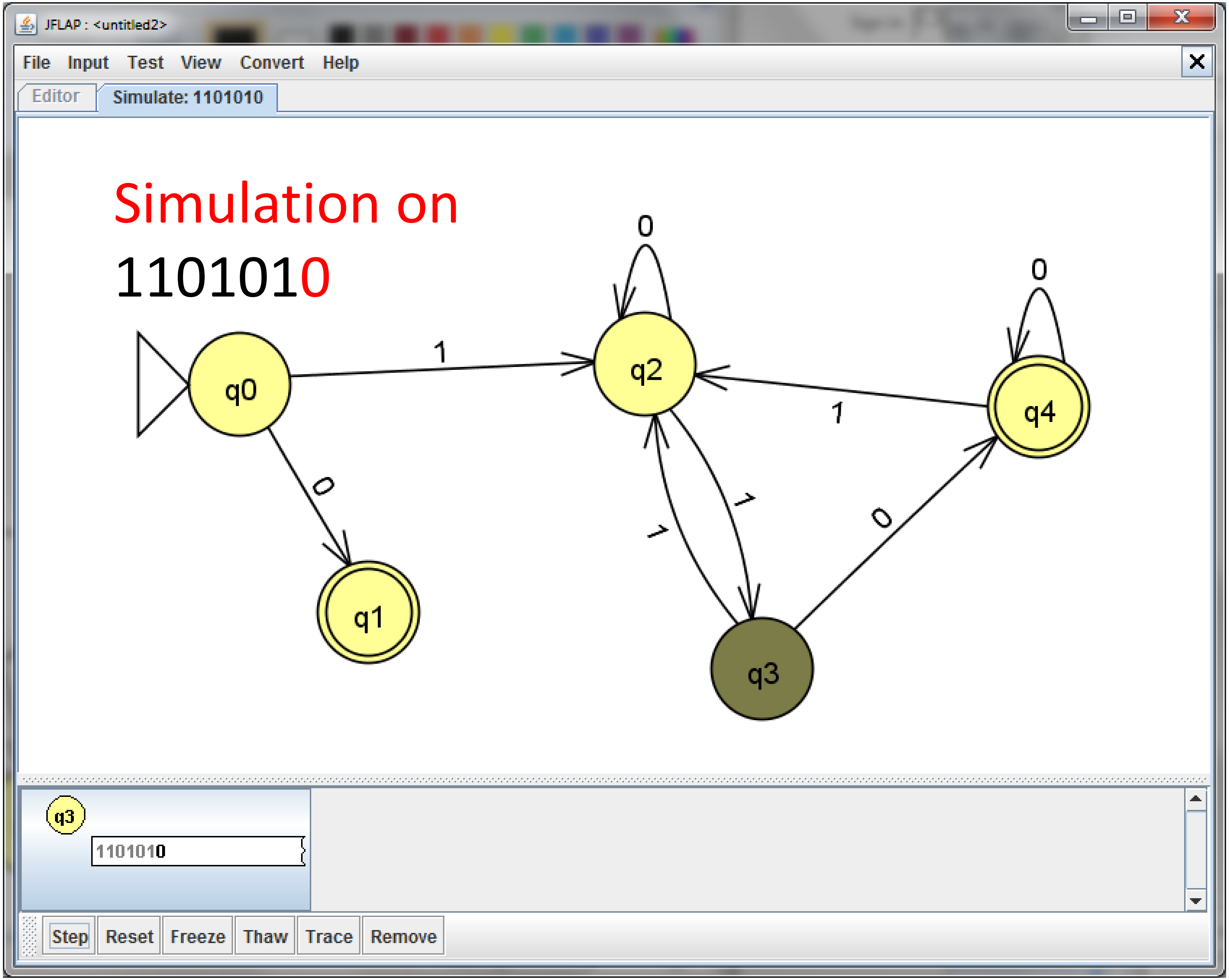


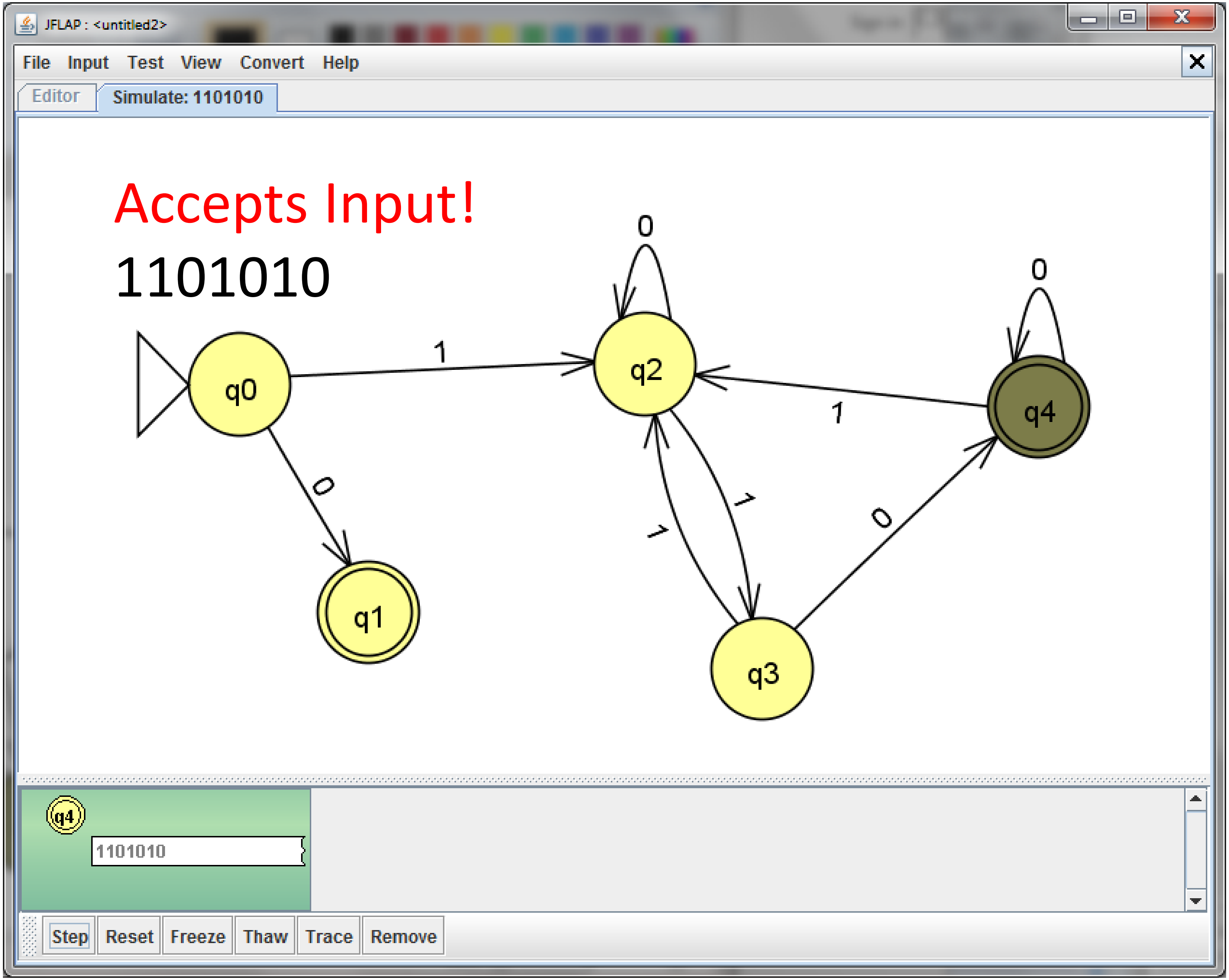
Simulation on

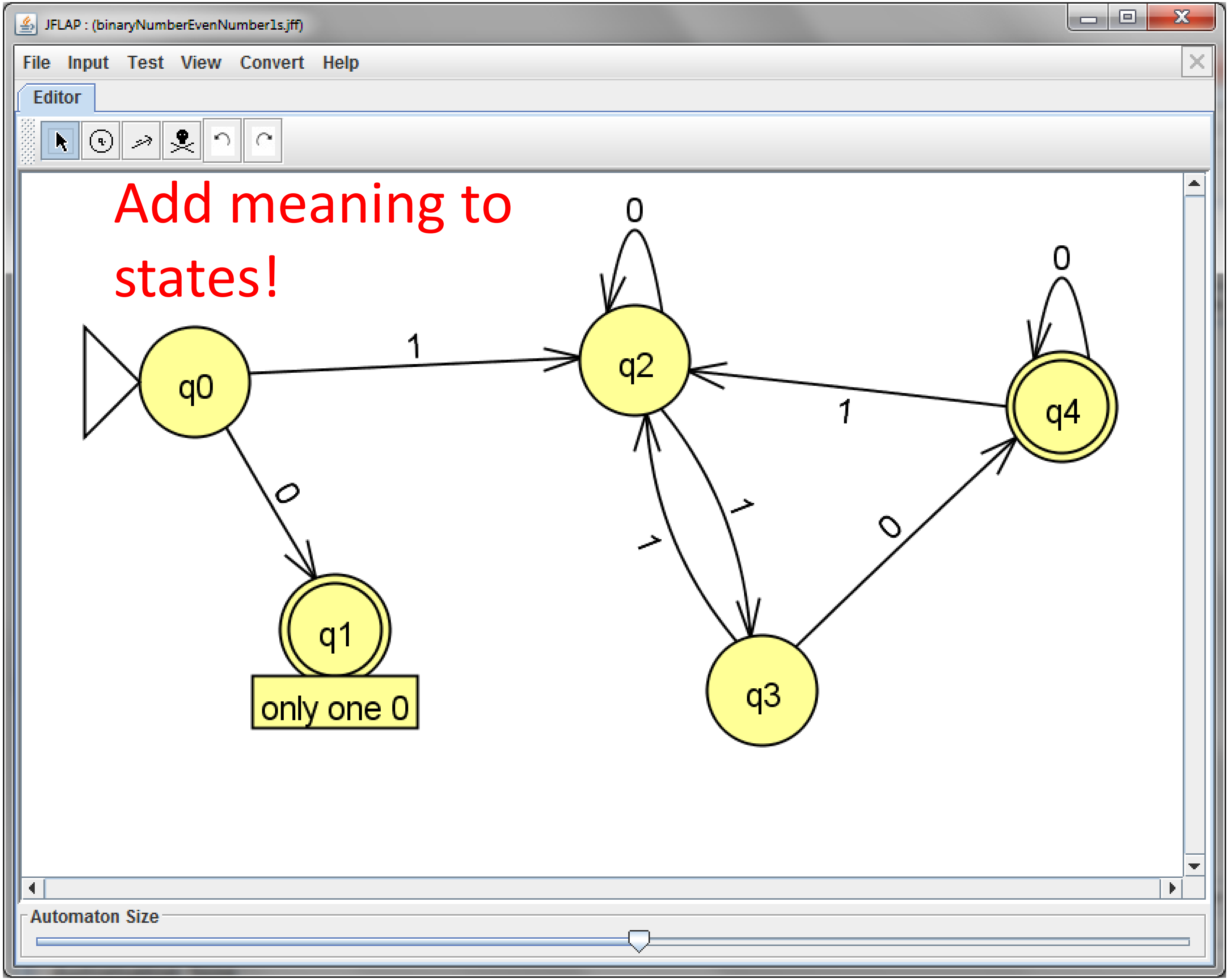
1101010



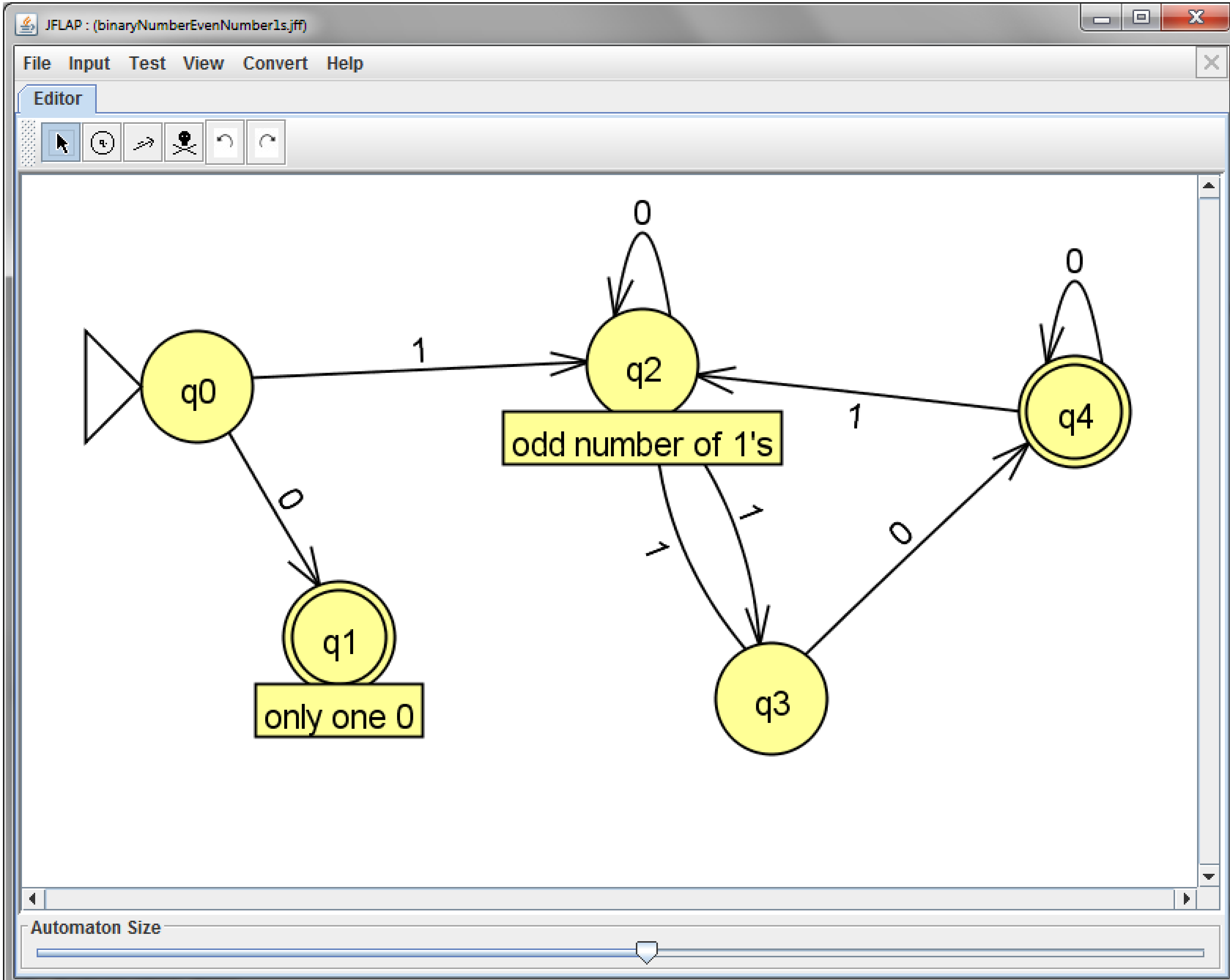

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Simulation on
1101010
q0
q1
q2
q3
q4
1
0
0
0
1
1
1
0
q2
1101010
Step Reset Freeze Thaw Trace Remove


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
q3
1101010
Step Reset Freeze Thaw Trace Remove

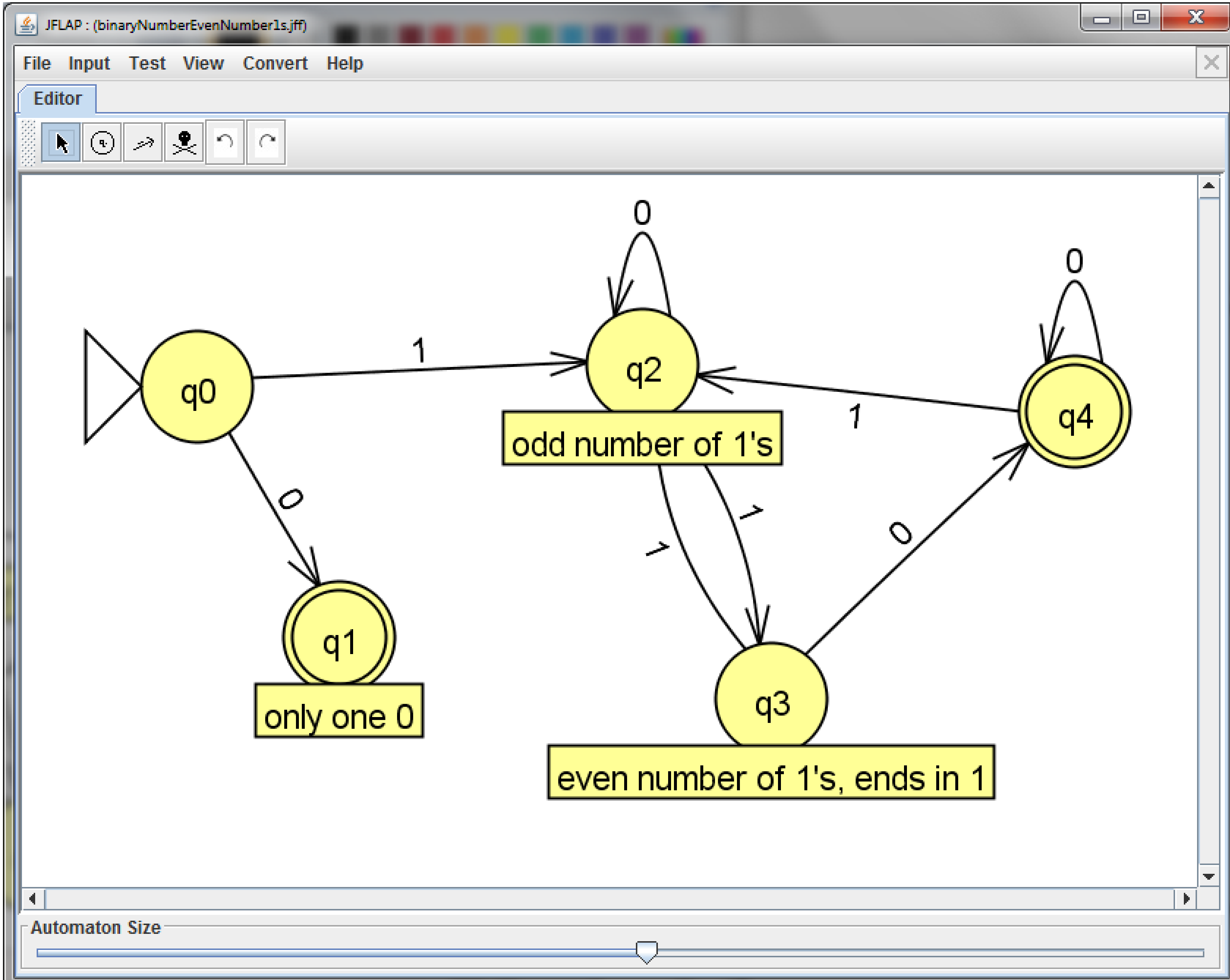

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
Simulate: 1101010
Accepts Input!
1101010
q0
q1
q2
q3
q4
0
1
0
1
1
0
1
0
q4
1101010
Step Reset Freeze Thaw Trace Remove

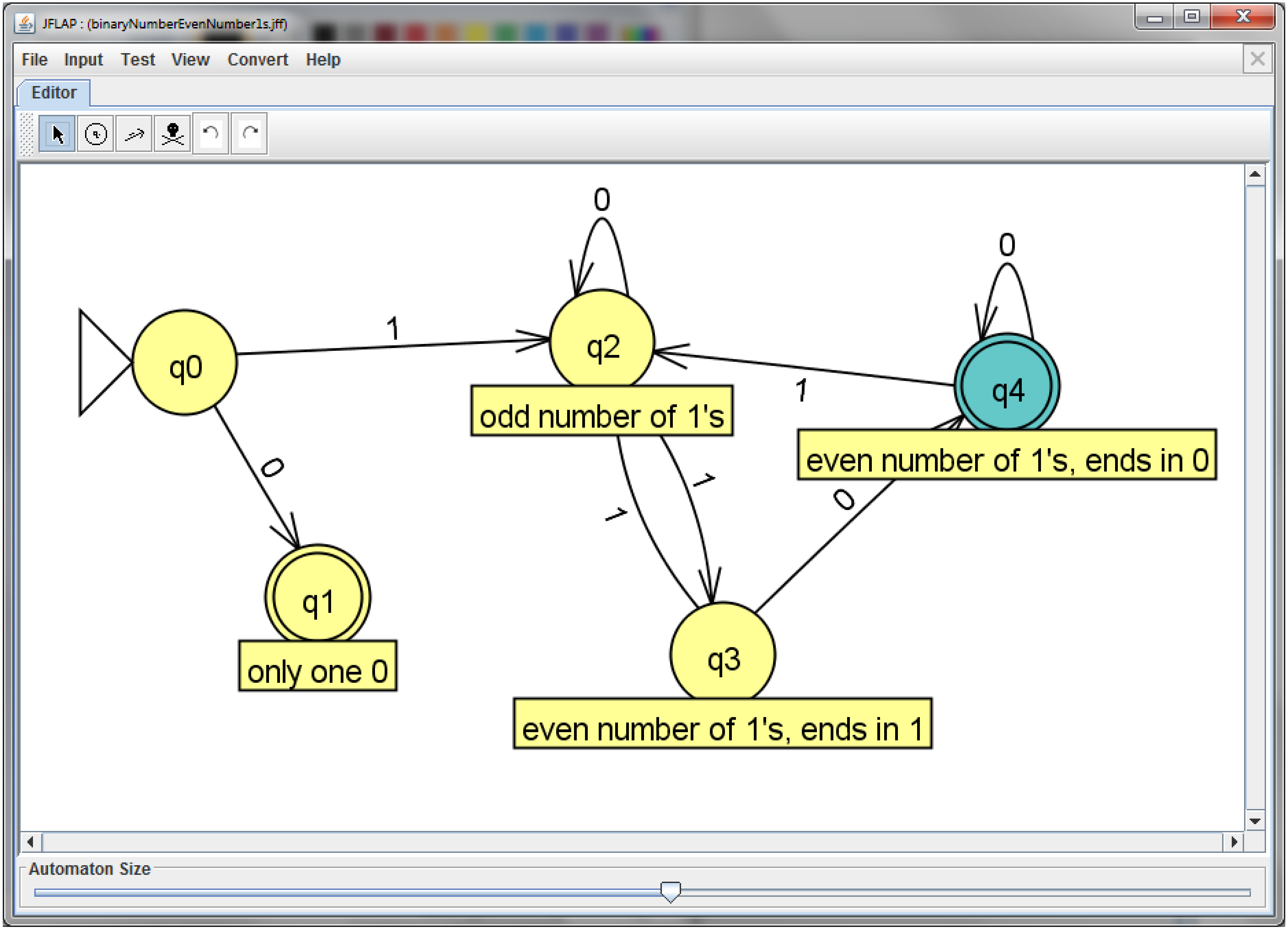


JFLAP : (binaryNumberEvenNumber1s.jff)
File Input Test View Convert Help
Editor
Add meaning to states!
q0
q1
q2
q3
q4
1
0
0
0
1
1
1
0
only one 0
Automaton Size

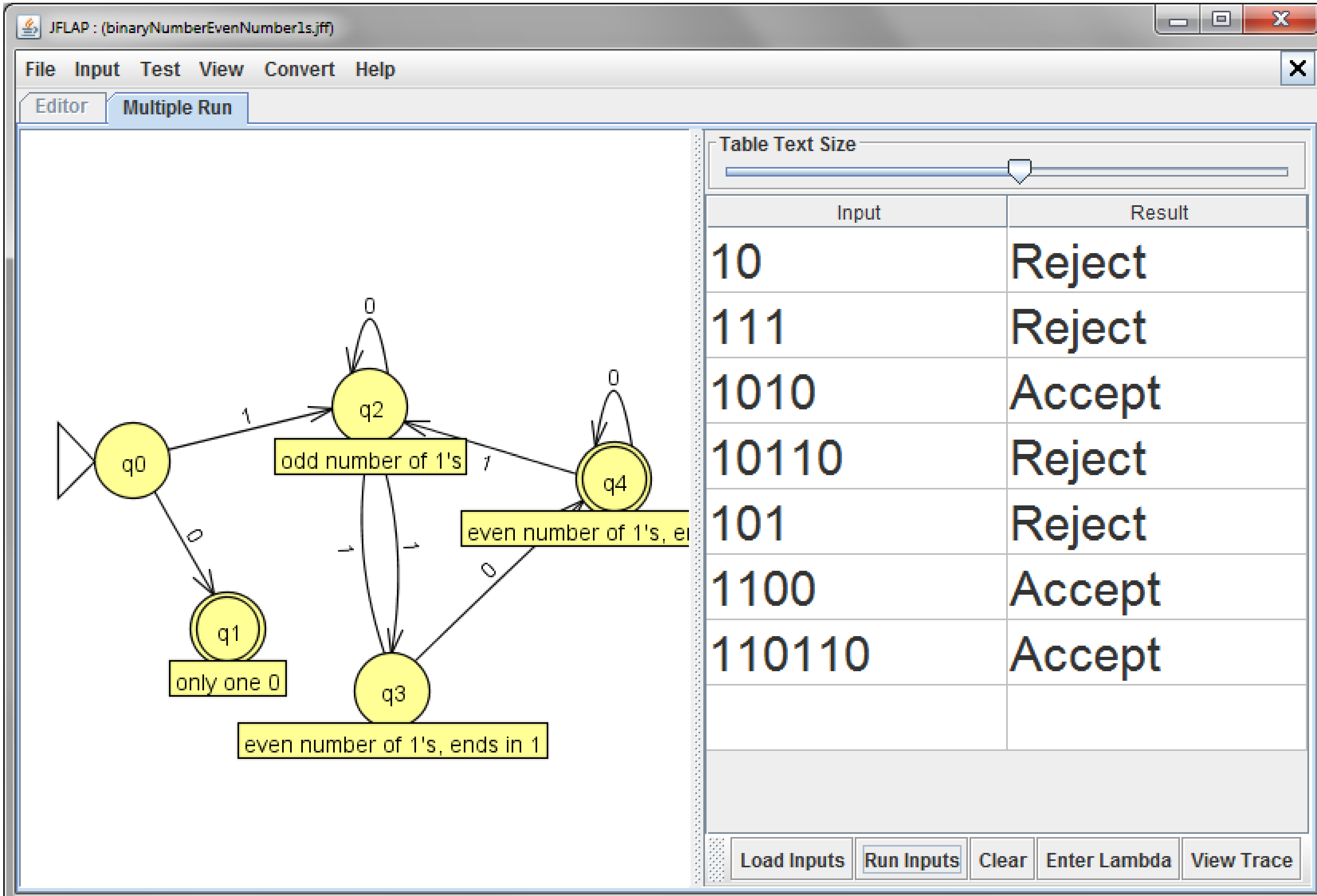


JFLAP : (binaryNumberEvenNumber1s.jff)
File Input Test View Convert Help
Editor
q0
q1
only one 0
q2
odd number of 1's
q3
q4
0
1
Automaton Size


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


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

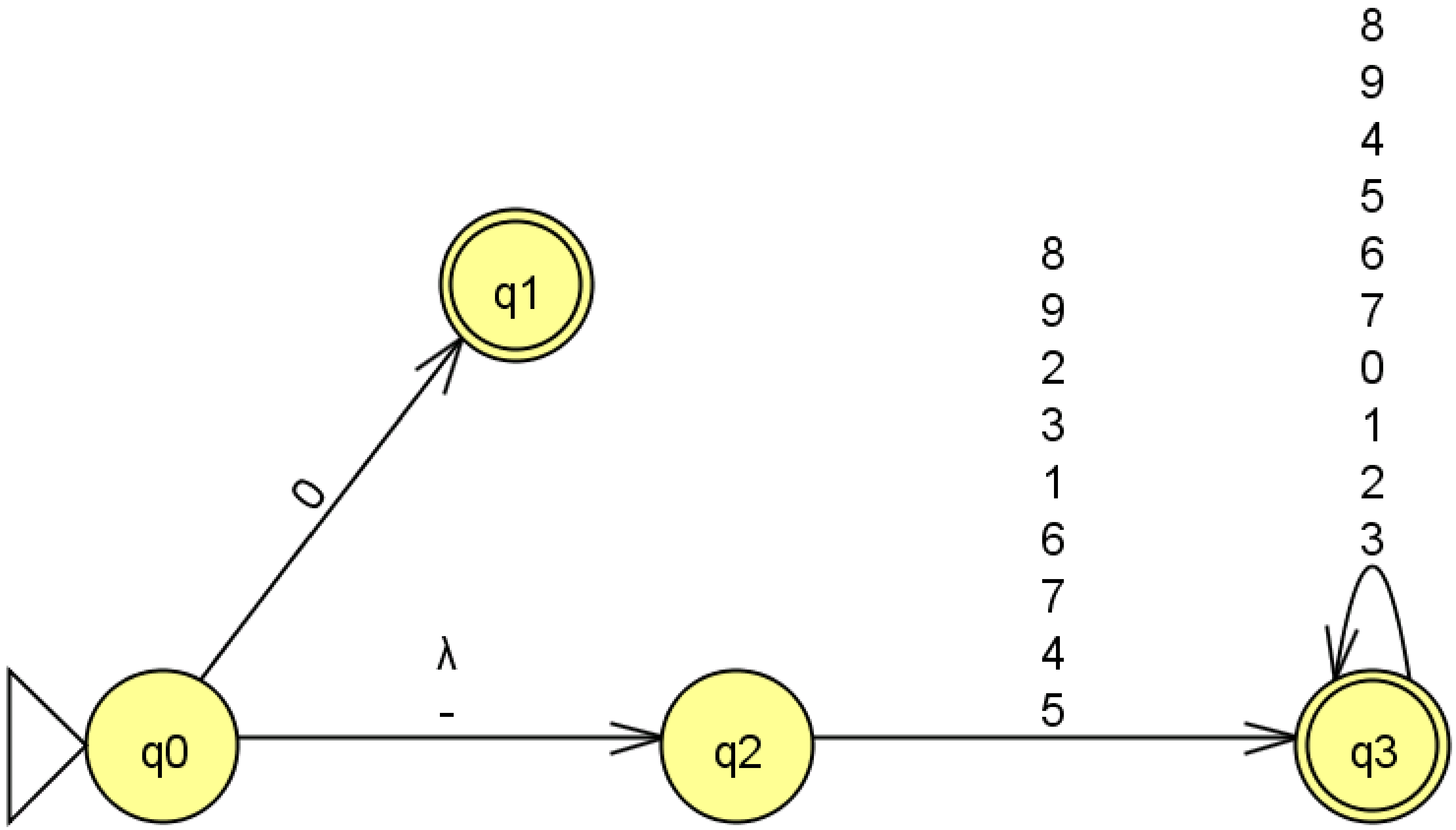
Test Multiple Inputs



Example: Build an NFA for valid integers

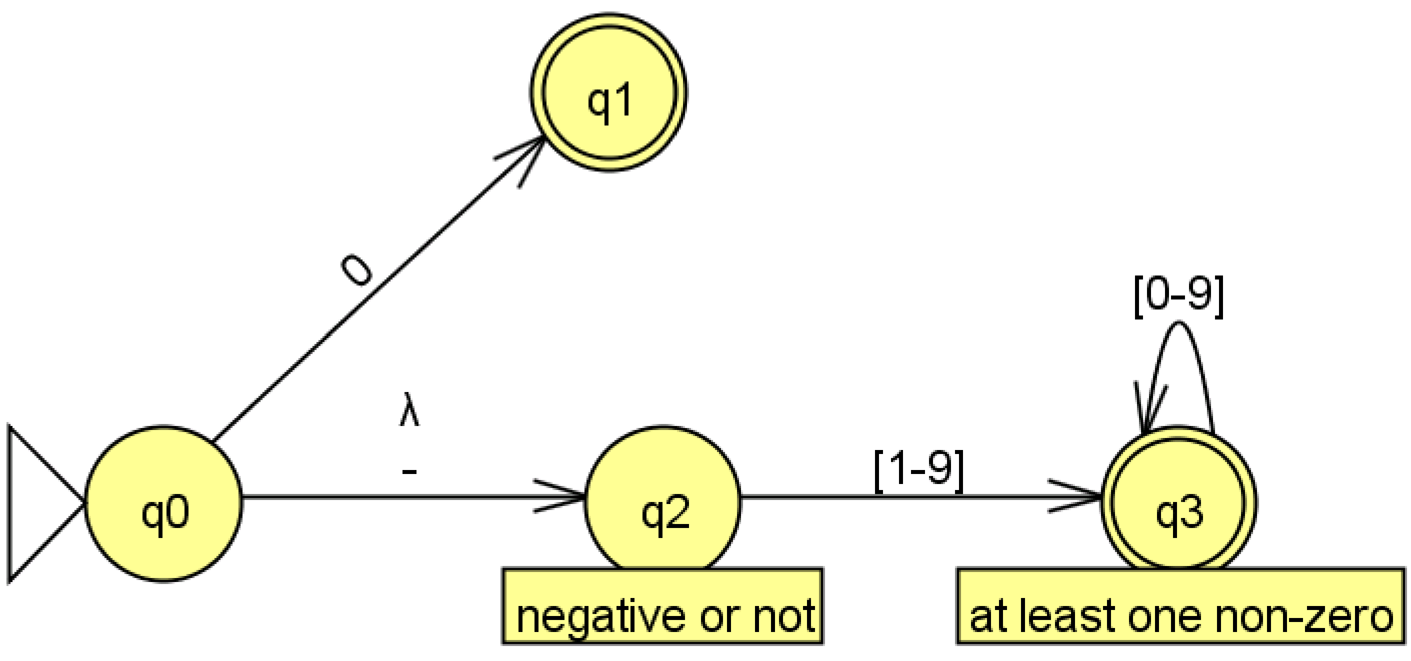
- Example:
  - Valid integers {-3, 8, 0, 456, 13, 500, …}
  - Not valid: {006, 3-6, 4.5, …}

Example: NFA for all valid integers



NFA annotated and shortcut

- Shortcut: [1-9] on labels



Another Example: Grammar

- Grammar – set of replacement rules to define a language
- Grammar for $a^n b^n c^n$
- Why look at such a grammar?
- 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

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

$\rightarrow aAbcX$ rule: A -> aAbc

Example Derivation for aabbcc

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

$\rightarrow aAbcX$ rule: A -> aAbc

$\rightarrow aaBbcbcX$ rule: A -> aBbc

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

Example Derivation for aabbcc

$S \to AX$	rule: S -> AX
$\to aAbcX$	rule: A -> aAbc
$\to aaBbcbcX$	rule: A -> aBbc
$\to aabBcbcX$	rule: Bb -> bB

Example Derivation for aabbcc

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

$\rightarrow aAbcX$ rule: A -> aAbc

$\rightarrow aaBbcbcX$ rule: A -> aBbc

$\rightarrow aabBcbcX$ rule: Bb -> bB

$\rightarrow aabDbcX$ rule: Bc -> D

Note: the D absorbed the c!

Example Derivation for aabbcc

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

rule: S -> AX
rule: A -> aAbc
rule: A -> aBbc
rule: Bb -> bB
rule: Bc -> D
rule: Db -> bD

Example Derivation for aabbcc

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

rule: S -> AX
rule: A -> aAbc
rule: A -> aBbc
rule: Bb -> bB
rule: Bc -> D
rule: Db -> bD
rule: Dc -> cD

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
→ aabbcEXc	rule: DX -> EXc

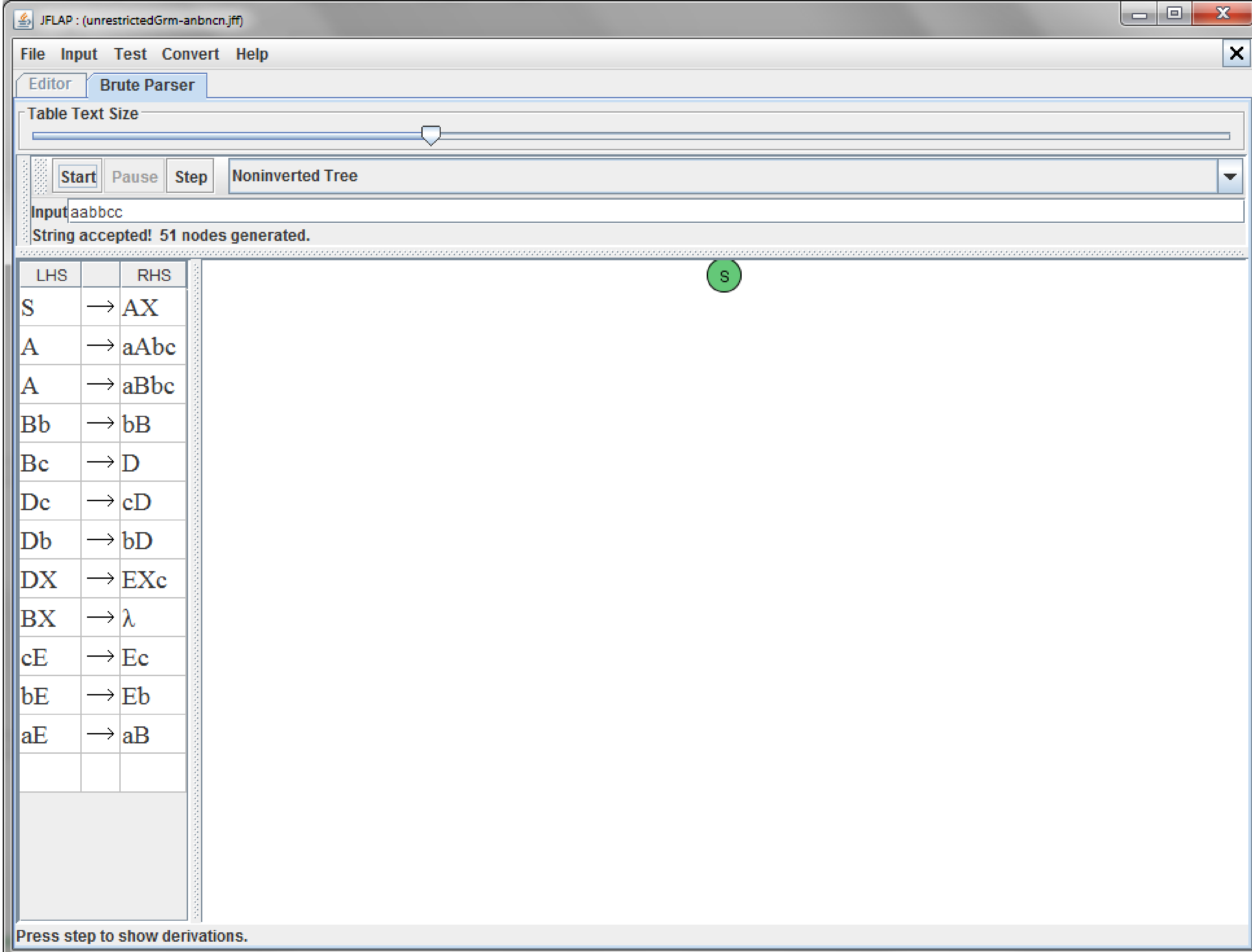
Note the c spit out on right end!

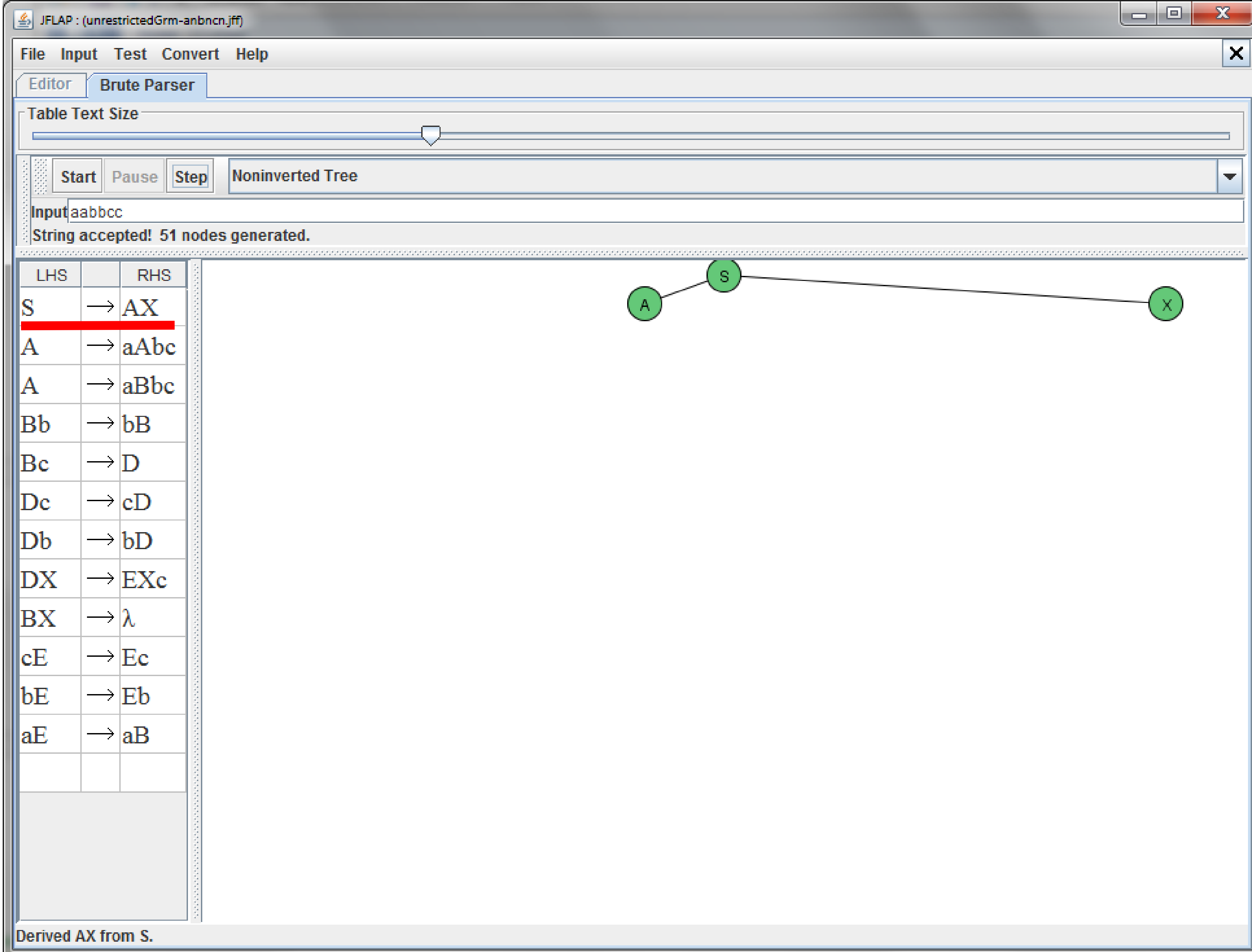
Eventually … → aabbcc

We could have done this derivation of aabbcc with JFLAP.

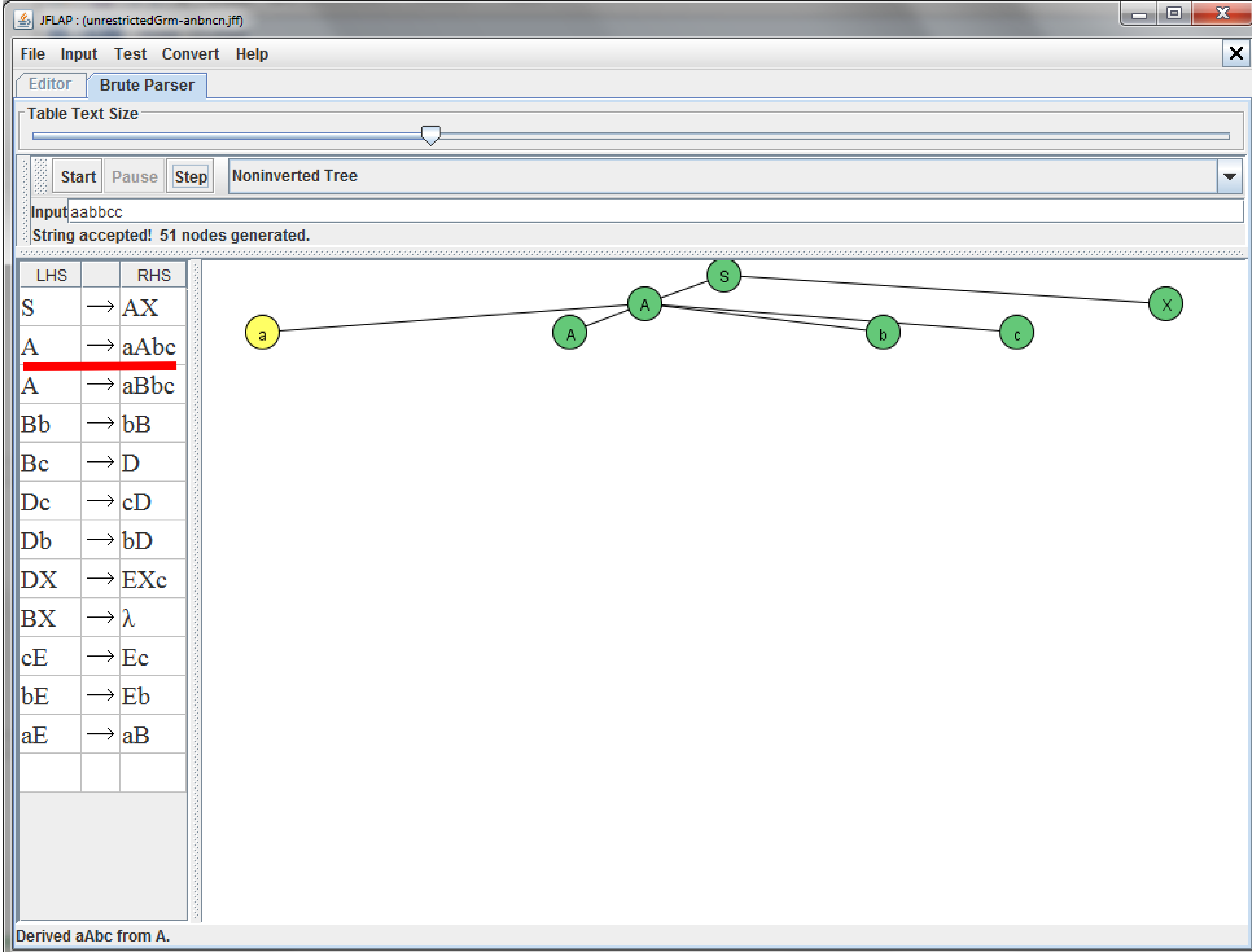
Now let's see how JFLAP visualizes this derivation with a "~~parse tree~~"

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
S
A
X
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
S
A
X
a
A
b
c
Derived aAbc from A.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor Brute Parser

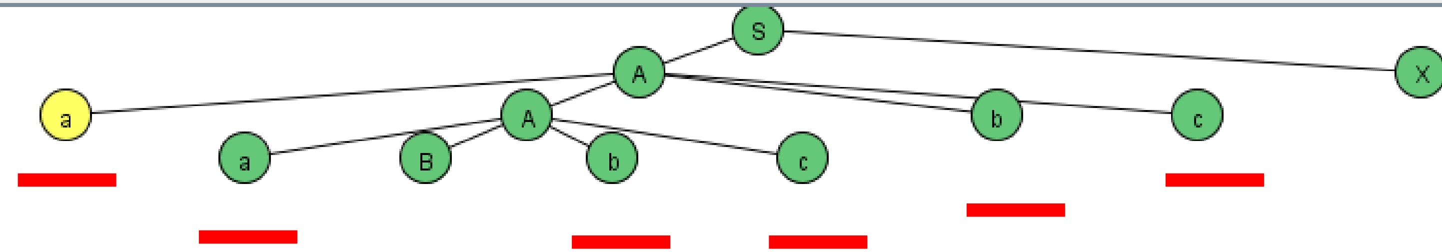
Table Text Size

Start Pause Step Noninverted Tree

Input aabbcc

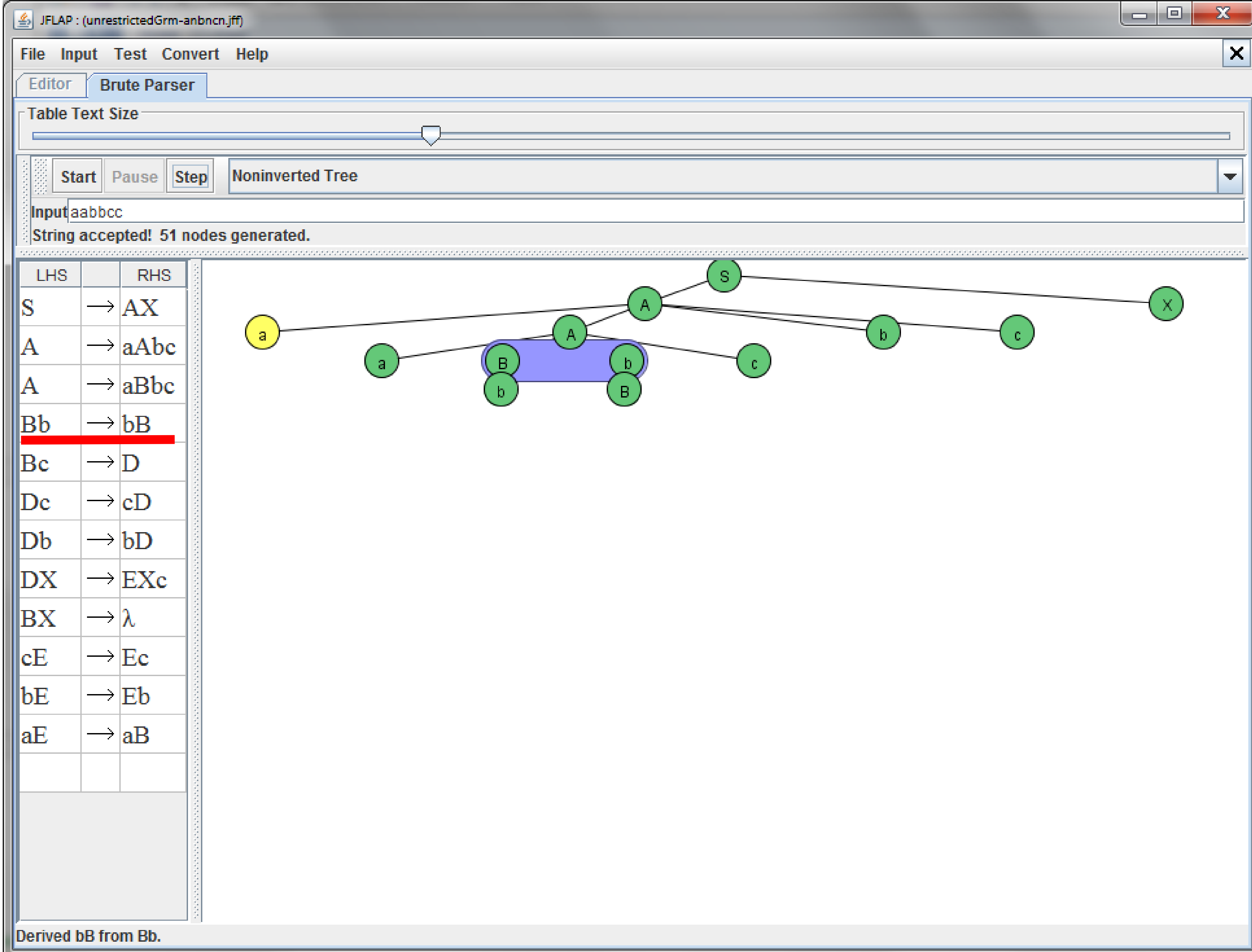
String accepted! 51 nodes generated.

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

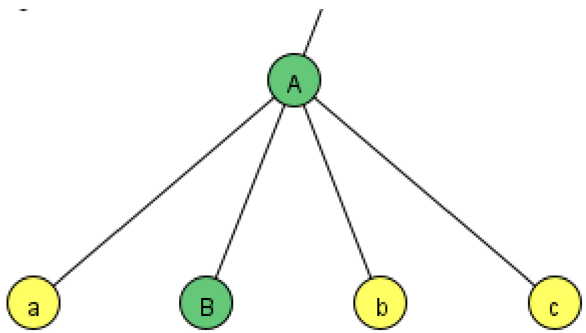
Derived aBbc from A.

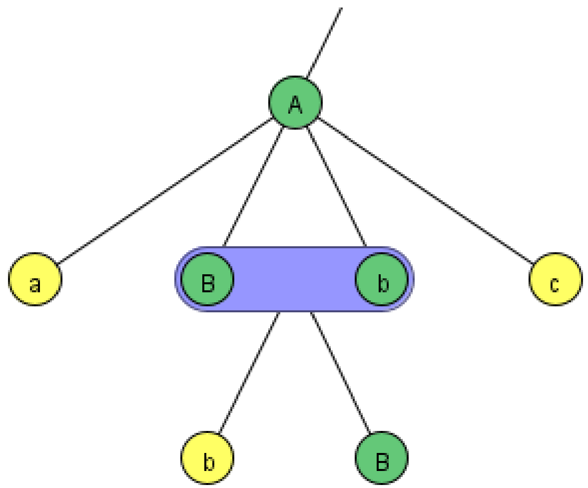


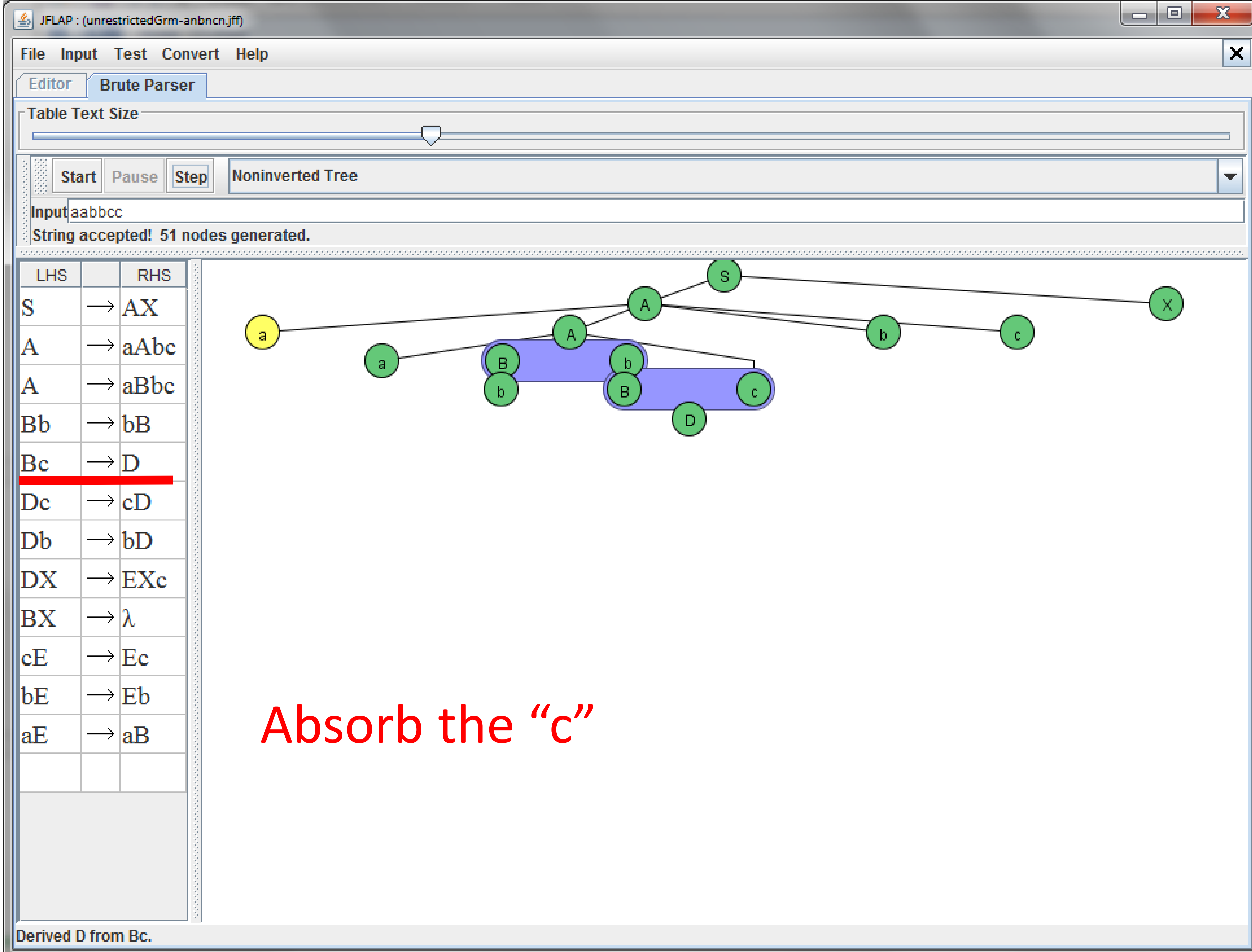
JFLAP : (unrestrictedGrm-anbncn.jff)
File Input Test Convert Help
Editor
Brute Parser
Table Text Size
Start
Pause
Step
Noninverted Tree
Input aabbcc
String accepted! 51 nodes generated.
LHS
RHS
S → AX
A → aAbc
A → aBbc
Bb → bB
Bc → D
Dc → cD
Db → bD
DX → EXc
BX → λ
cE → Ec
bE → Eb
aE → aB
S
A
X
a
a
A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

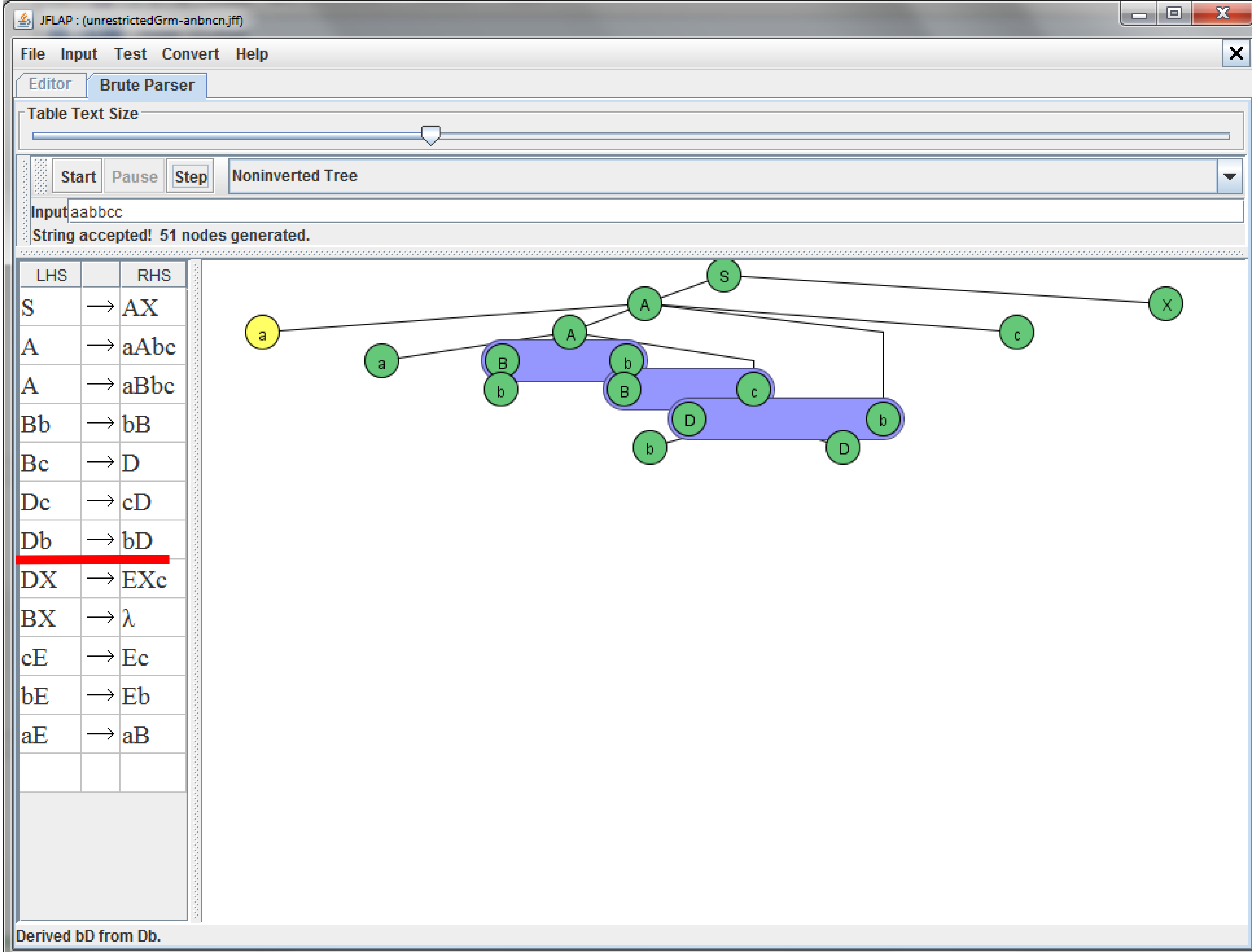
What’s happening?

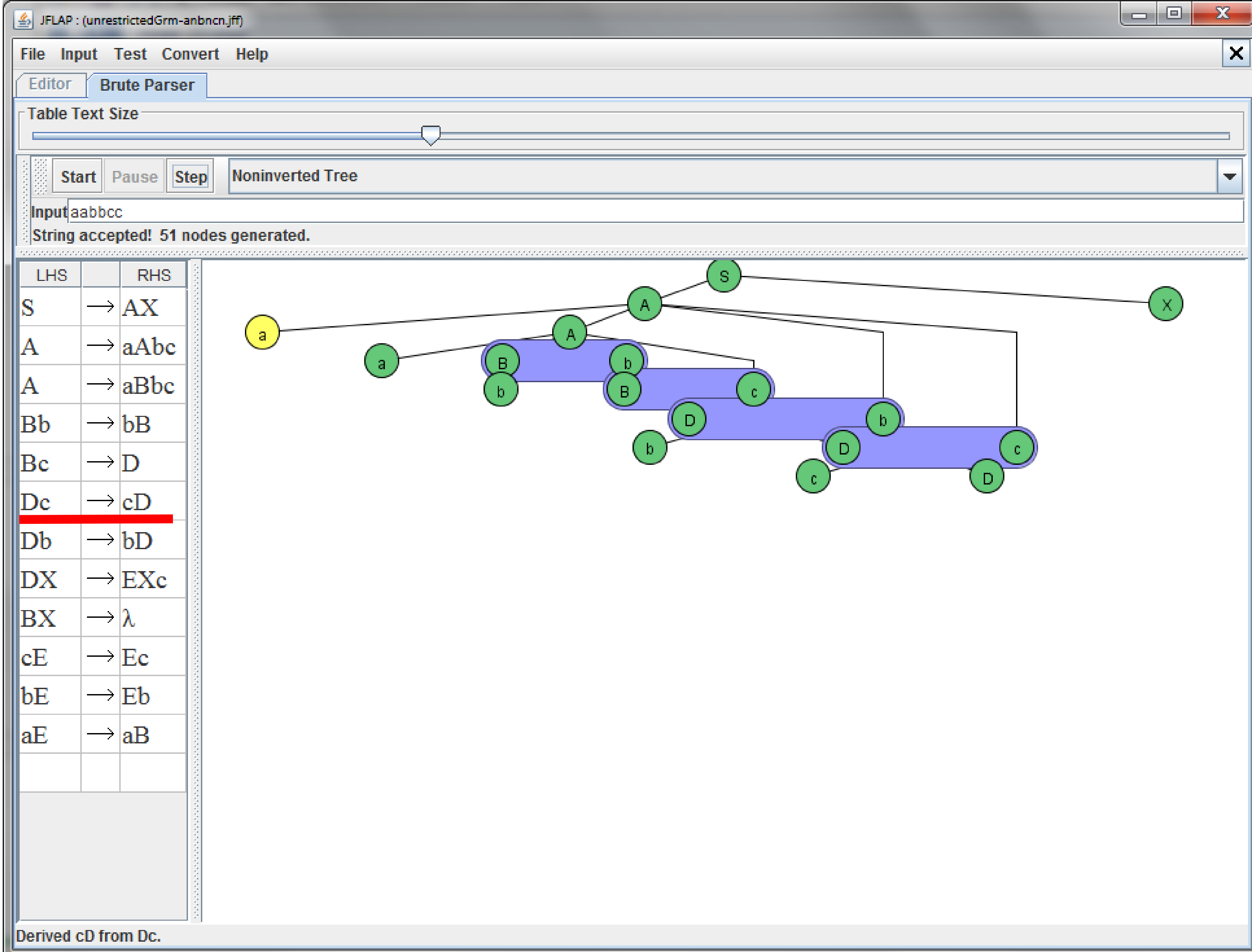
Bb $\longrightarrow$ bB






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
S A X a a A b c B b b B c D
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
S
A
X
a
A
c
a
B
b
b
B
c
D
b
b
D
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
S
A
X
a
a
A
B
b
b
B
c
b
D
b
D
c
c
D
Derived cD from Dc.

JFLAP : (unrestrictedGrm-anbncn.jff)

File Input Test Convert Help

Editor | Brute Parser

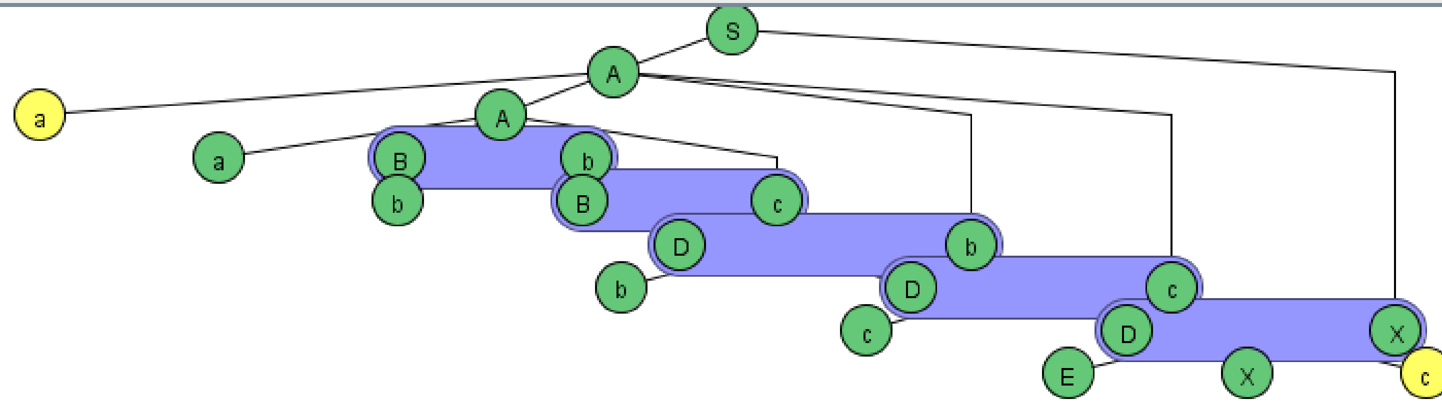
Table Text Size

Start Pause Step | Noninverted Tree

Input aabbcc

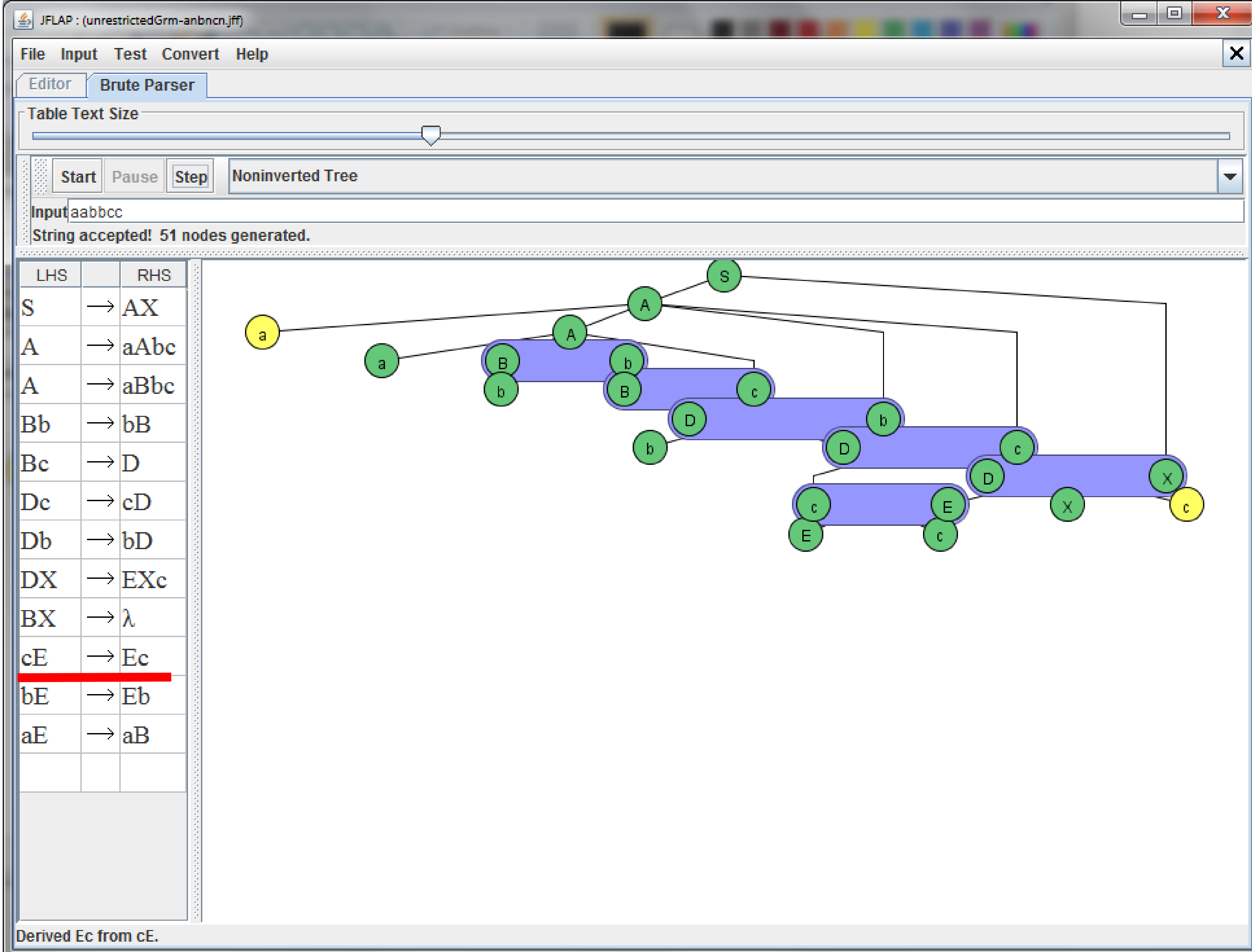
String accepted! 51 nodes generated.

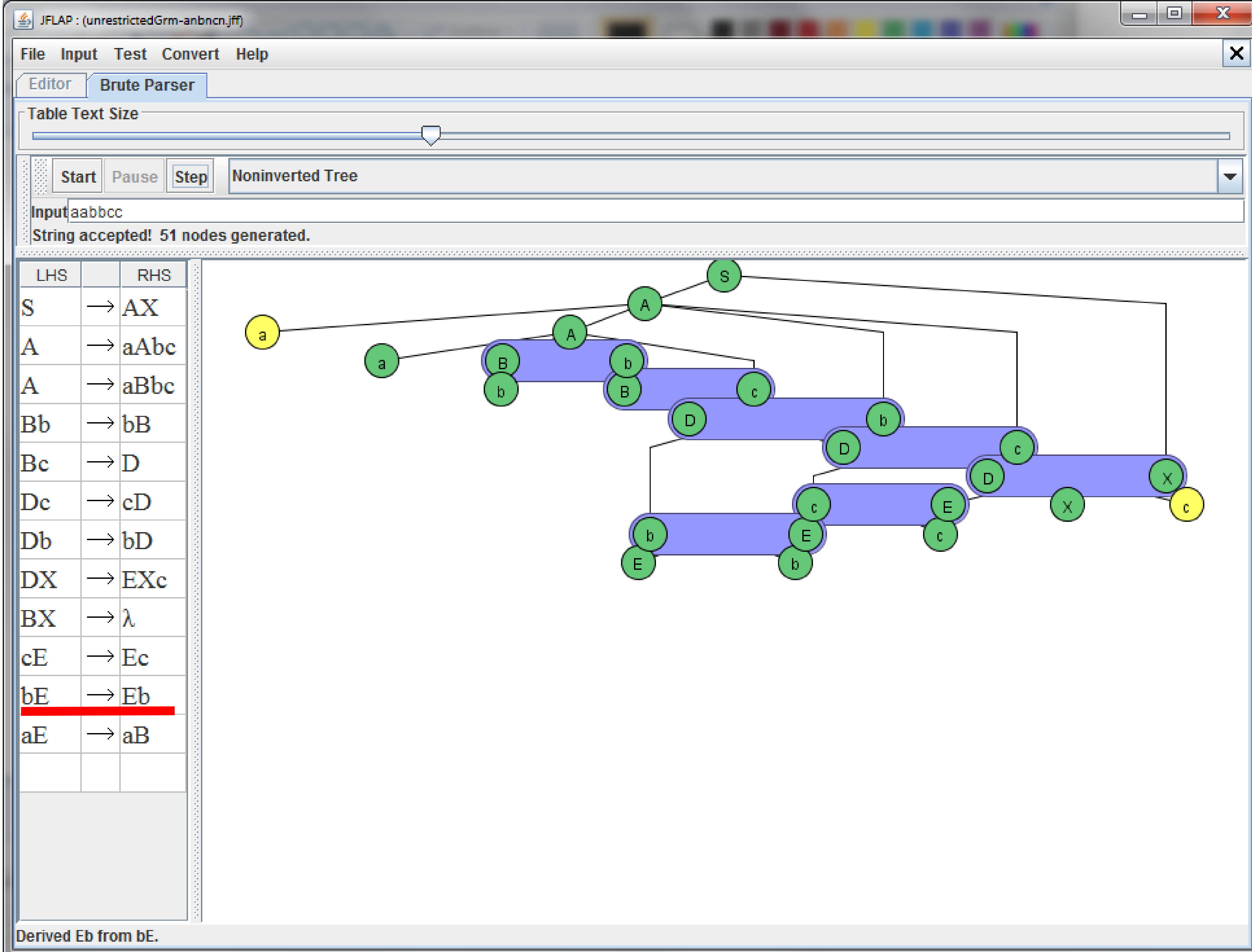
LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB

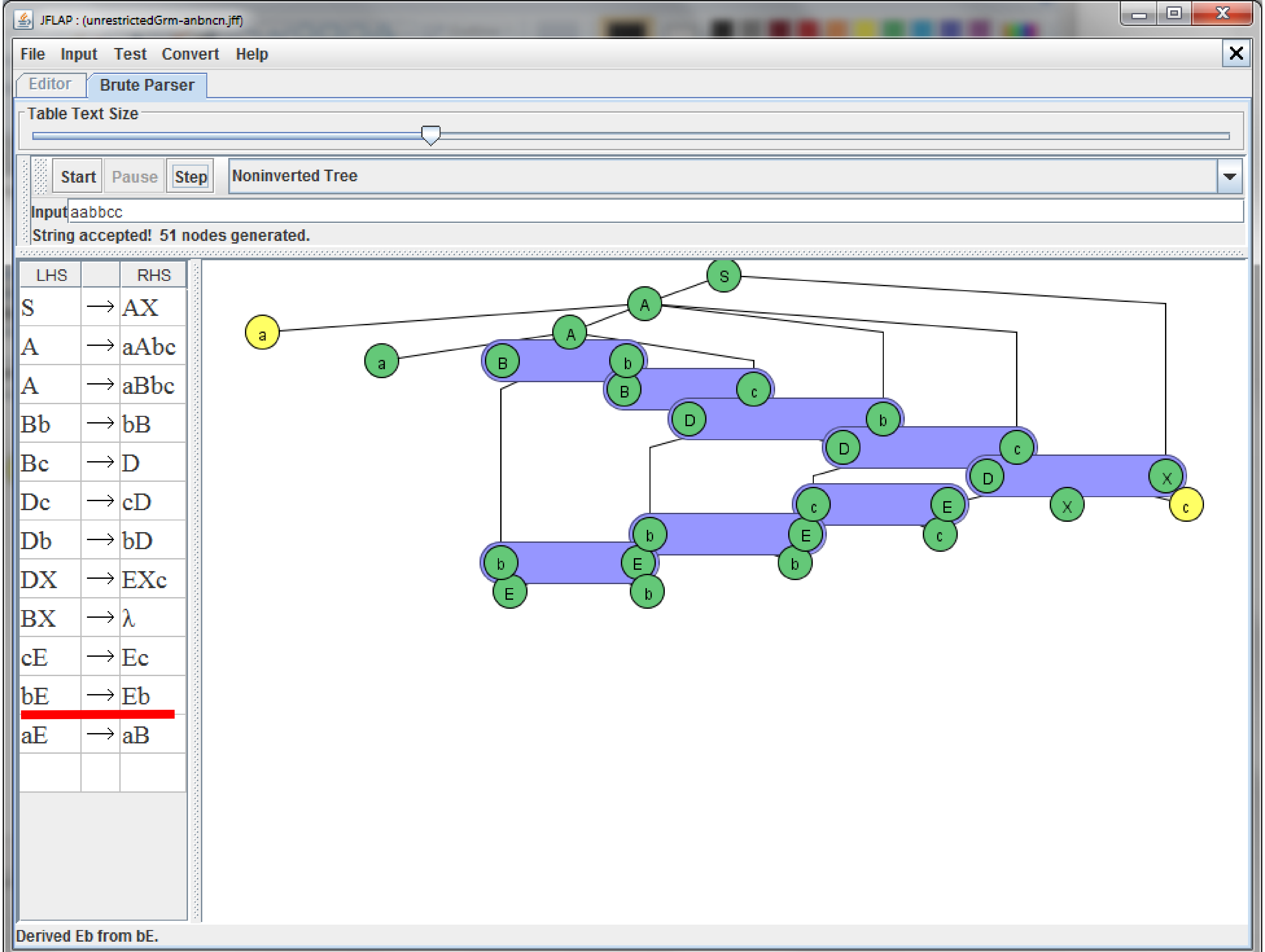


Spit out the "c" at the right end

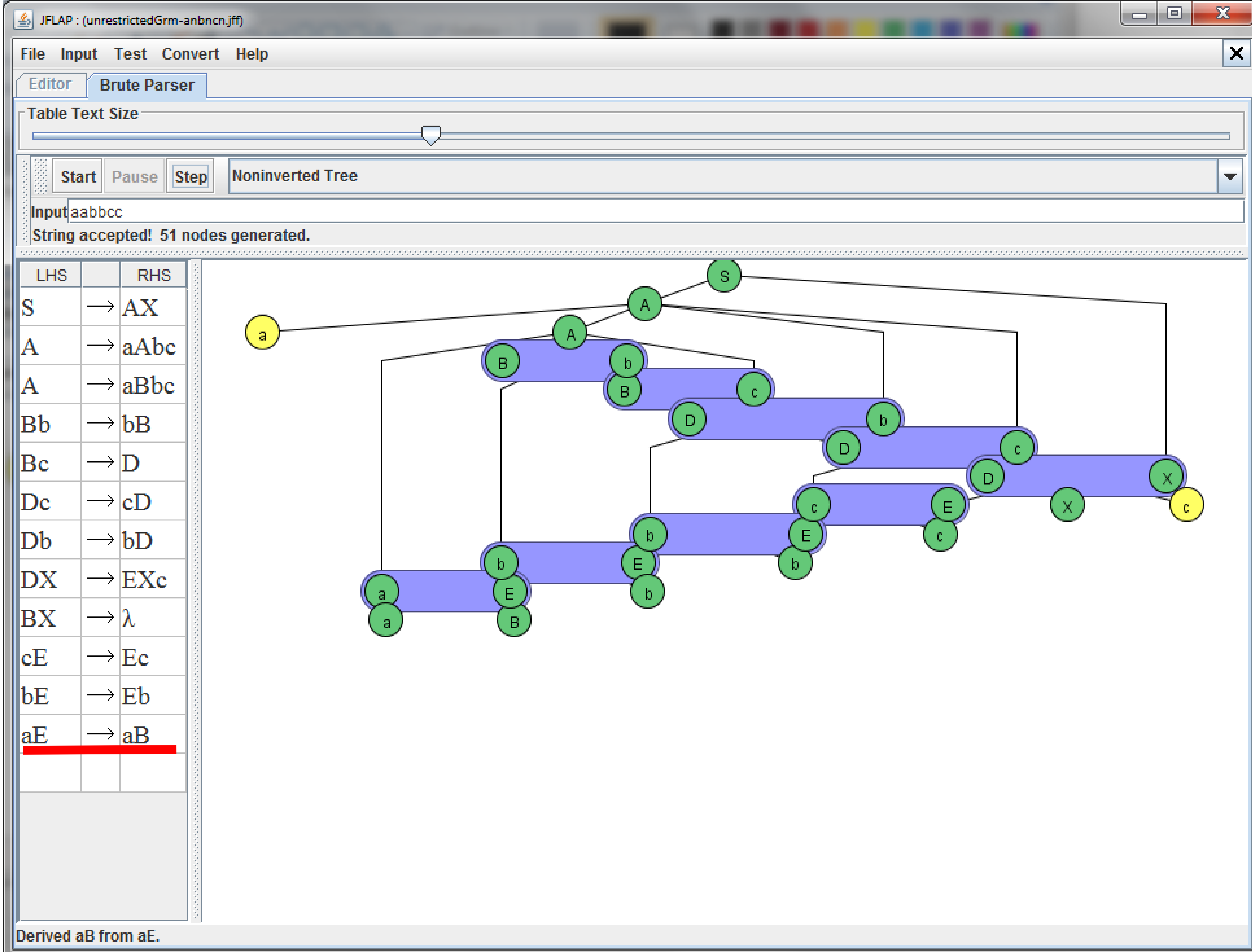
Derived EXc from DX.

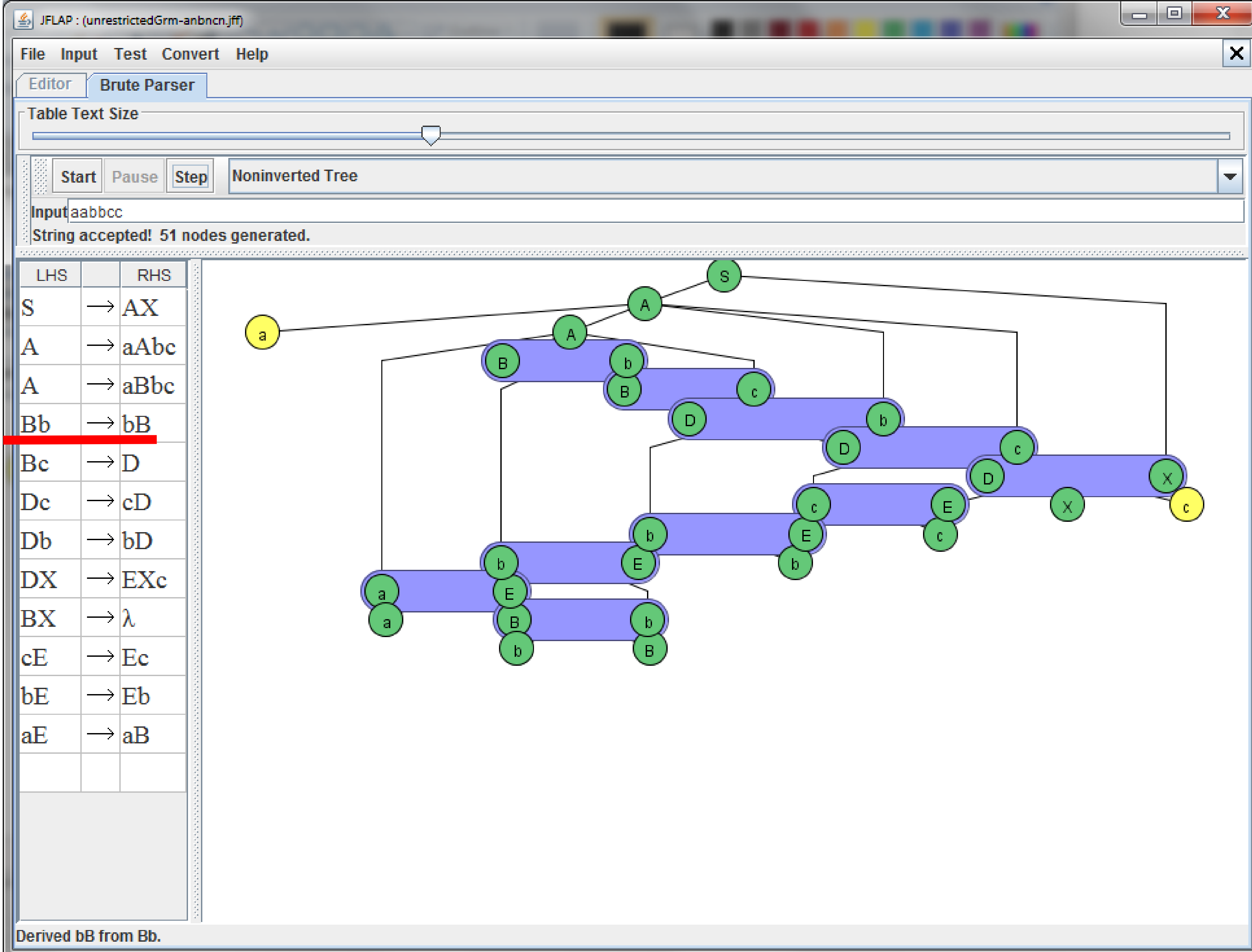

JFLAP : (unrestrictedGrm-anbncn.jff)
File Input Test Convert Help
Editor
Brute Parser
Table Text Size
Start Pause Step
Noninverted Tree
Input aabbcc
String accepted! 51 nodes generated.
LHS RHS
S → AX
A → aAbc
A → aBbc
Bb → bB
Bc → D
Dc → cD
Db → bD
DX → EXc
BX → λ
cE → Ec
bE → Eb
aE → aB
S
A
a
A
a
B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

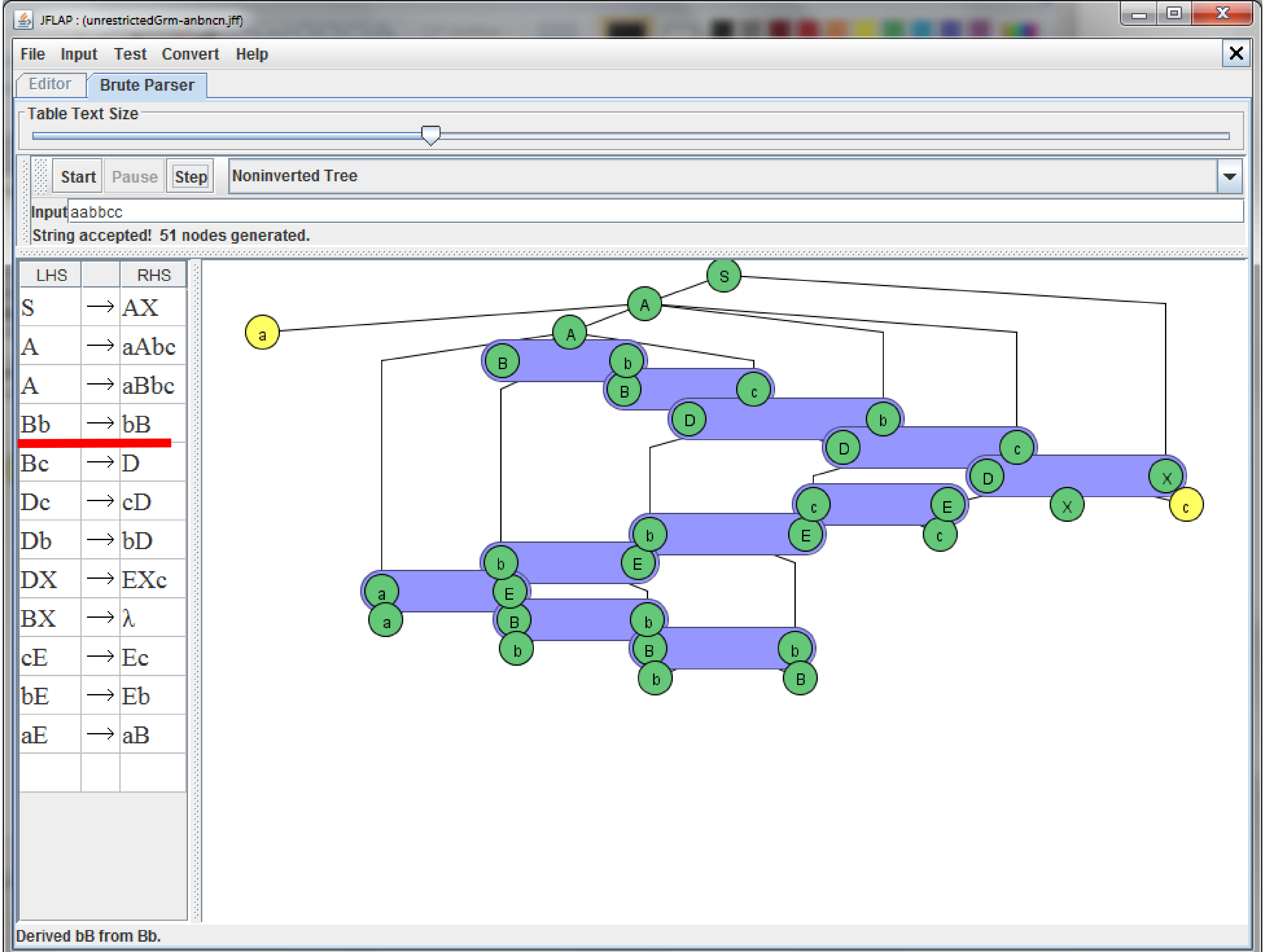


JFLAP : (unrestrictedGrm-anbncn.jff)
File Input Test Convert Help
Editor
Brute Parser
Table Text Size
Start Pause Step
Noninverted Tree
Input aabbcc
String accepted! 51 nodes generated.
LHS RHS
S → AX
A → aAbc
A → aBbc
Bb → bB
Bc → D
Dc → cD
Db → bD
DX → EXc
BX → λ
cE → Ec
bE → Eb
aE → aB
S
A
A
a
a
B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

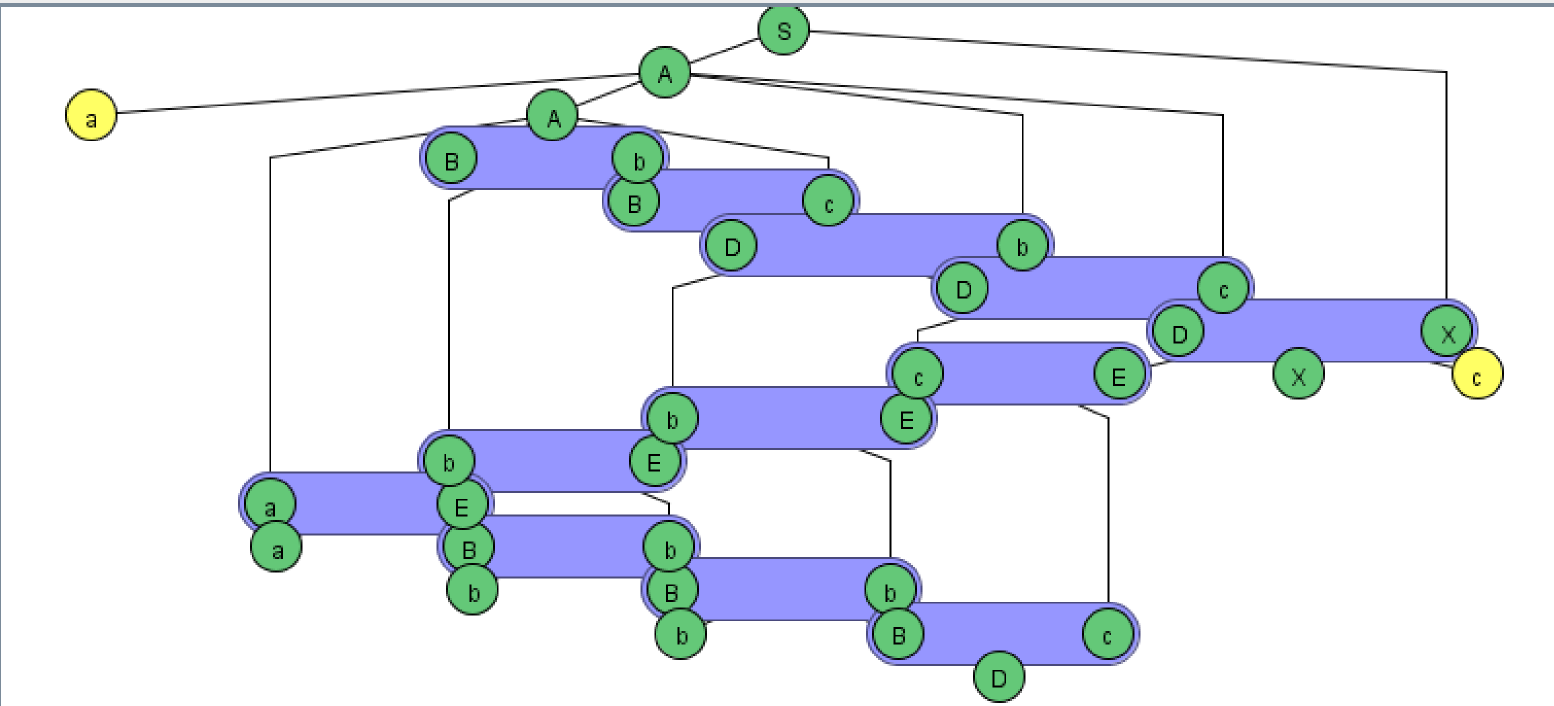
Table Text Size

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

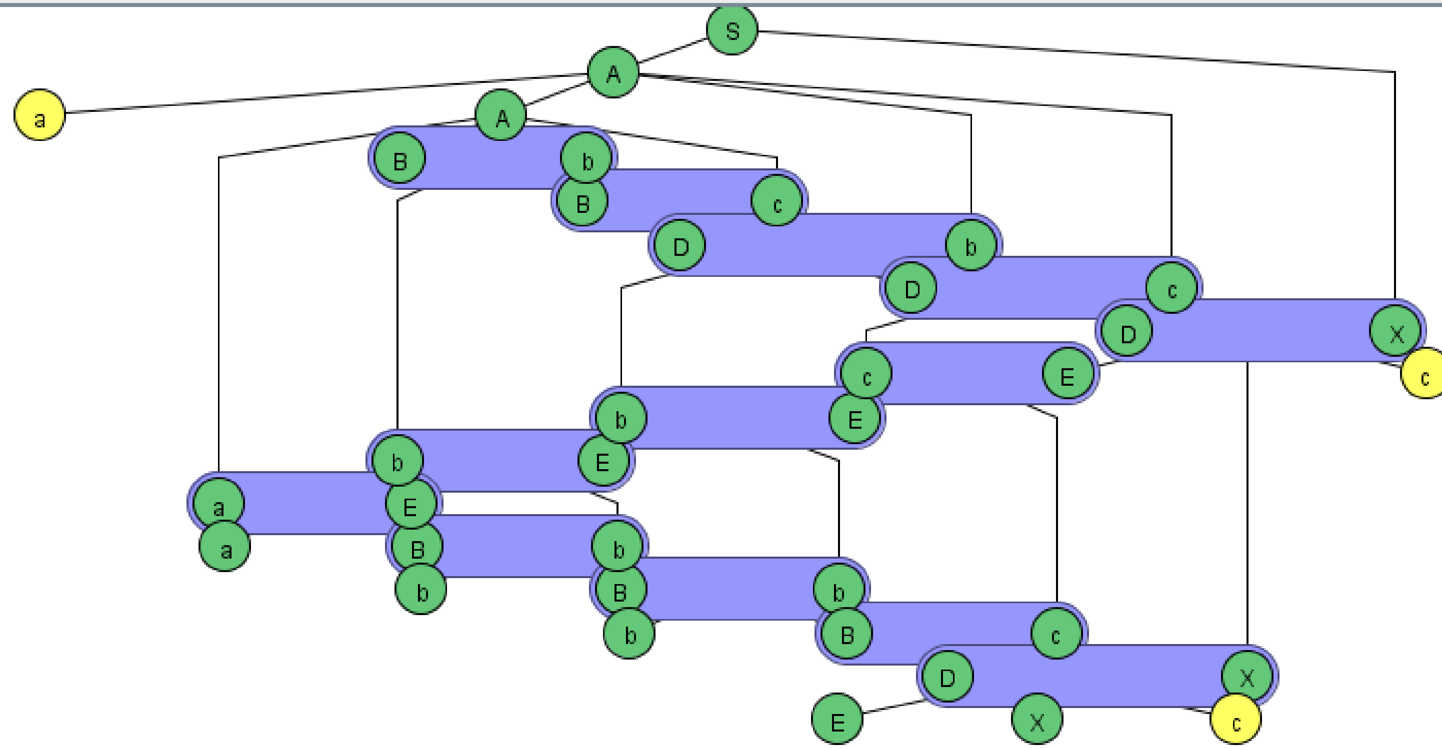
Table Text Size

Start Pause Step | Noninverted Tree

Input aabbcc

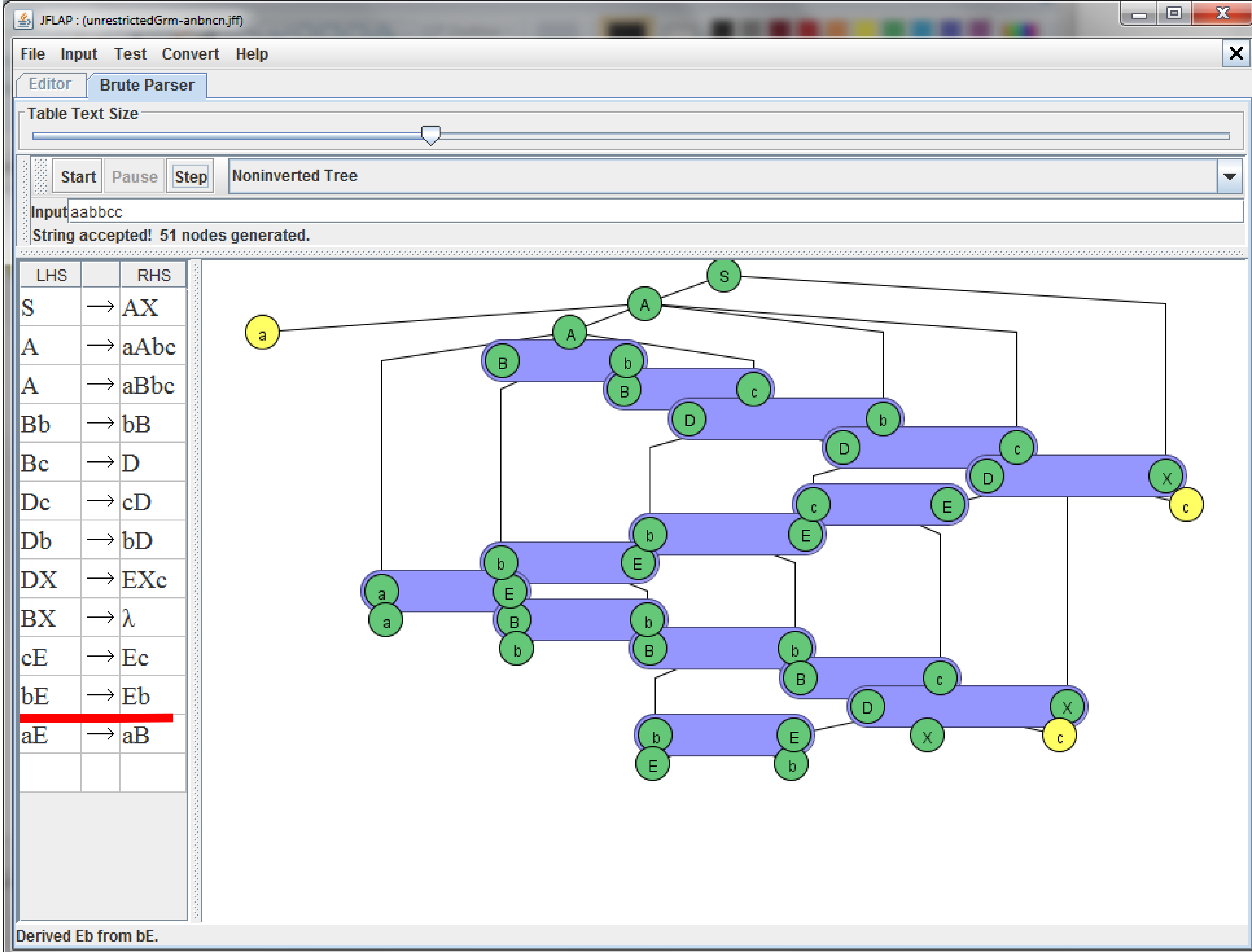
String accepted! 51 nodes generated.

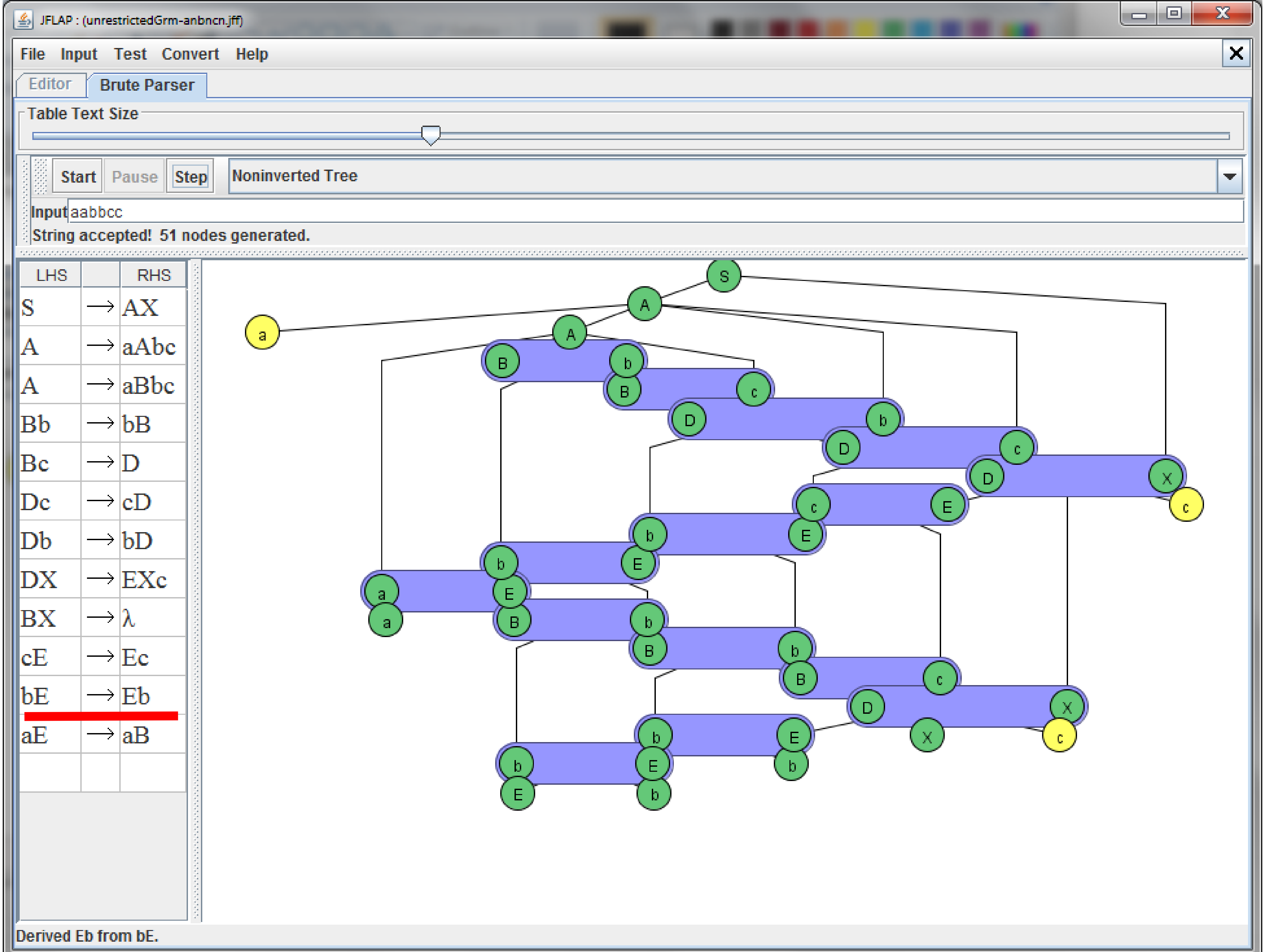
LHS		RHS
S	→	AX
A	→	aAbc
A	→	aBbc
Bb	→	bB
Bc	→	D
Dc	→	cD
Db	→	bD
DX	→	EXc
BX	→	λ
cE	→	Ec
bE	→	Eb
aE	→	aB

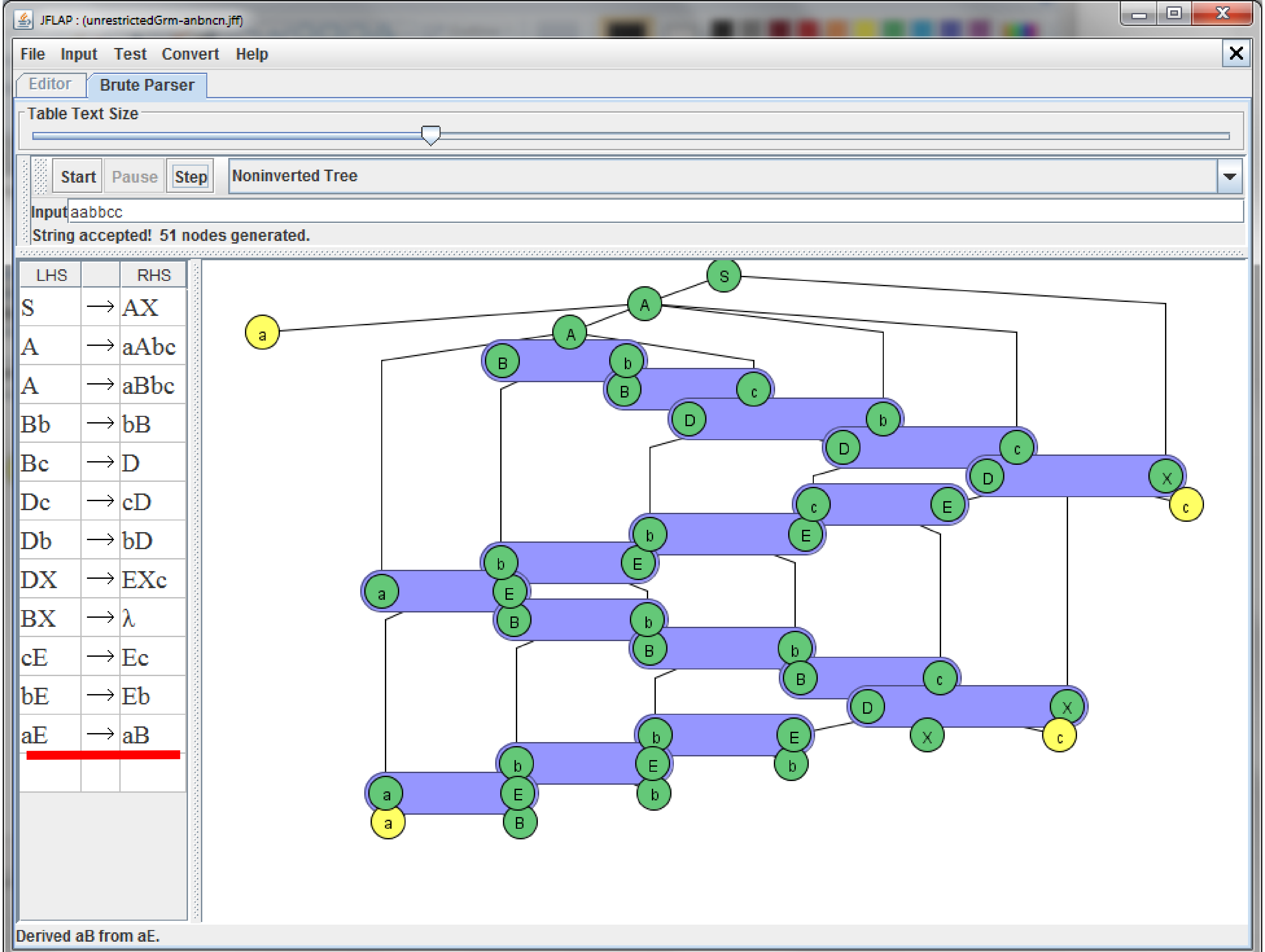


Spit the "c" out at right end

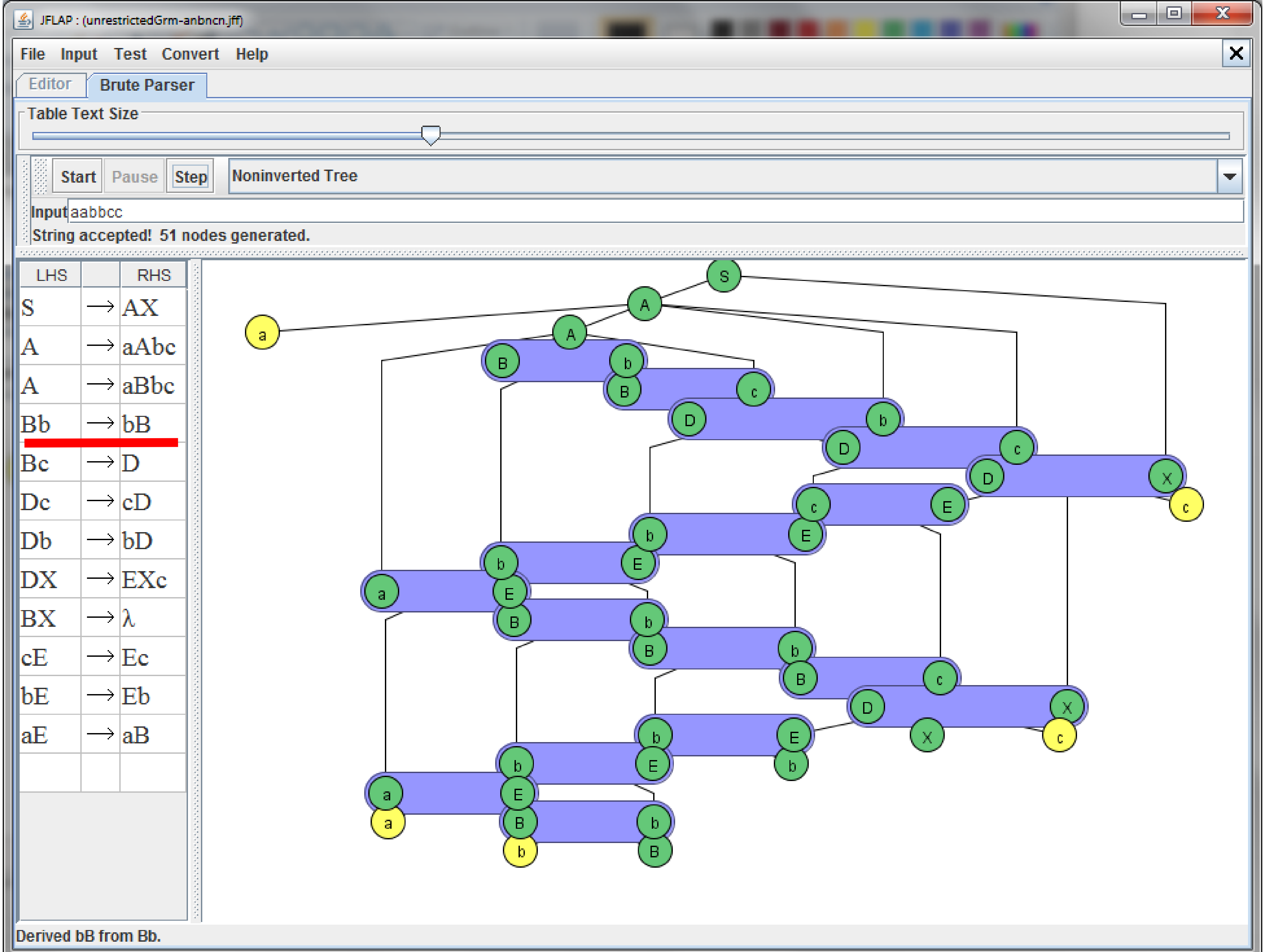
Derived EXc from DX.

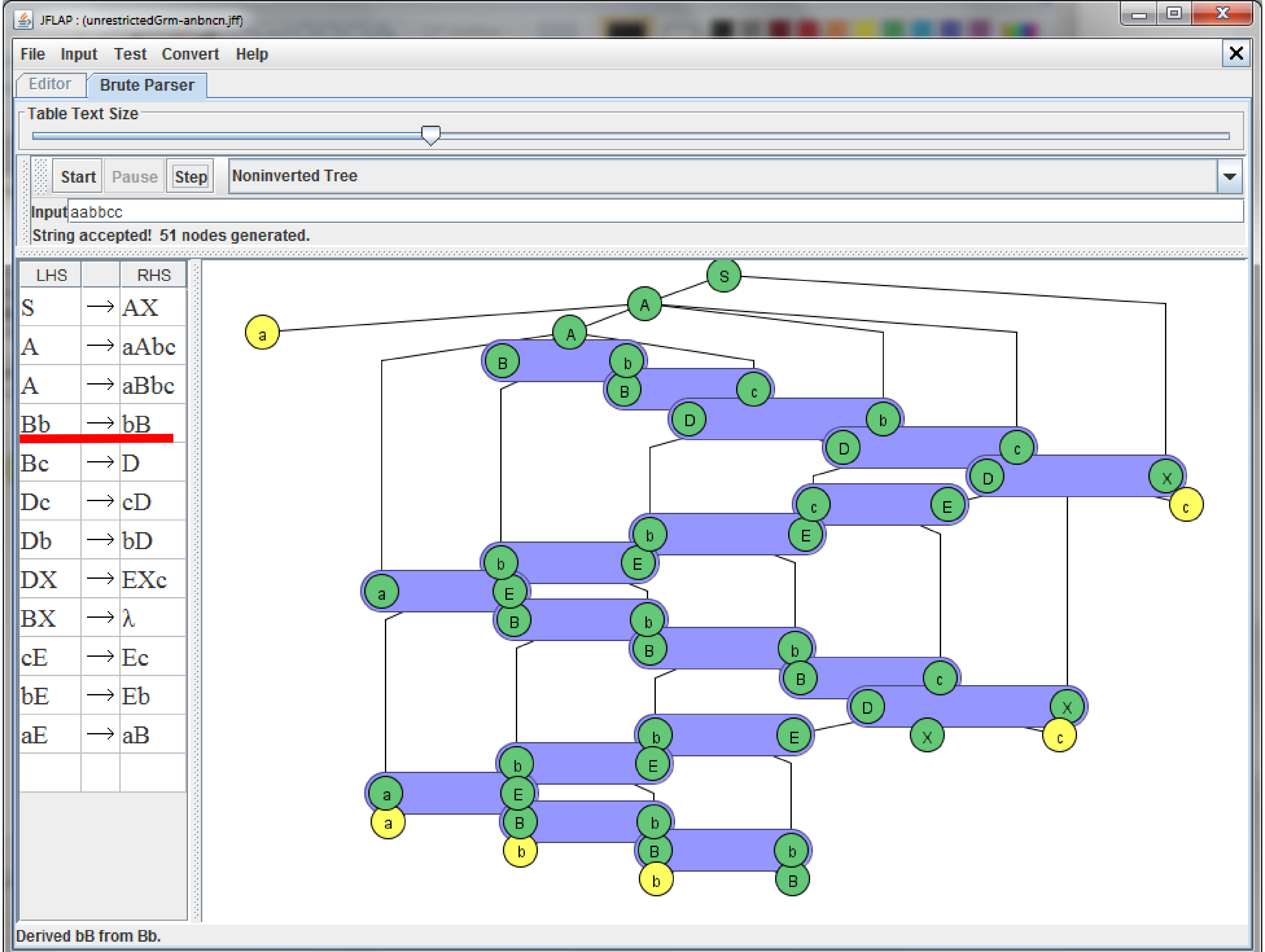

JFLAP : (unrestrictedGrm-anbncn.jff)
File Input Test Convert Help
Editor
Brute Parser
Table Text Size
Start
Pause
Step
Noninverted Tree
Input aabbcc
String accepted! 51 nodes generated.
LHS RHS
S → AX
A → aAbc
A → aBbc
Bb → bB
Bc → D
Dc → cD
Db → bD
DX → EXc
BX → λ
cE → Ec
bE → Eb
aE → aB
Derived Eb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

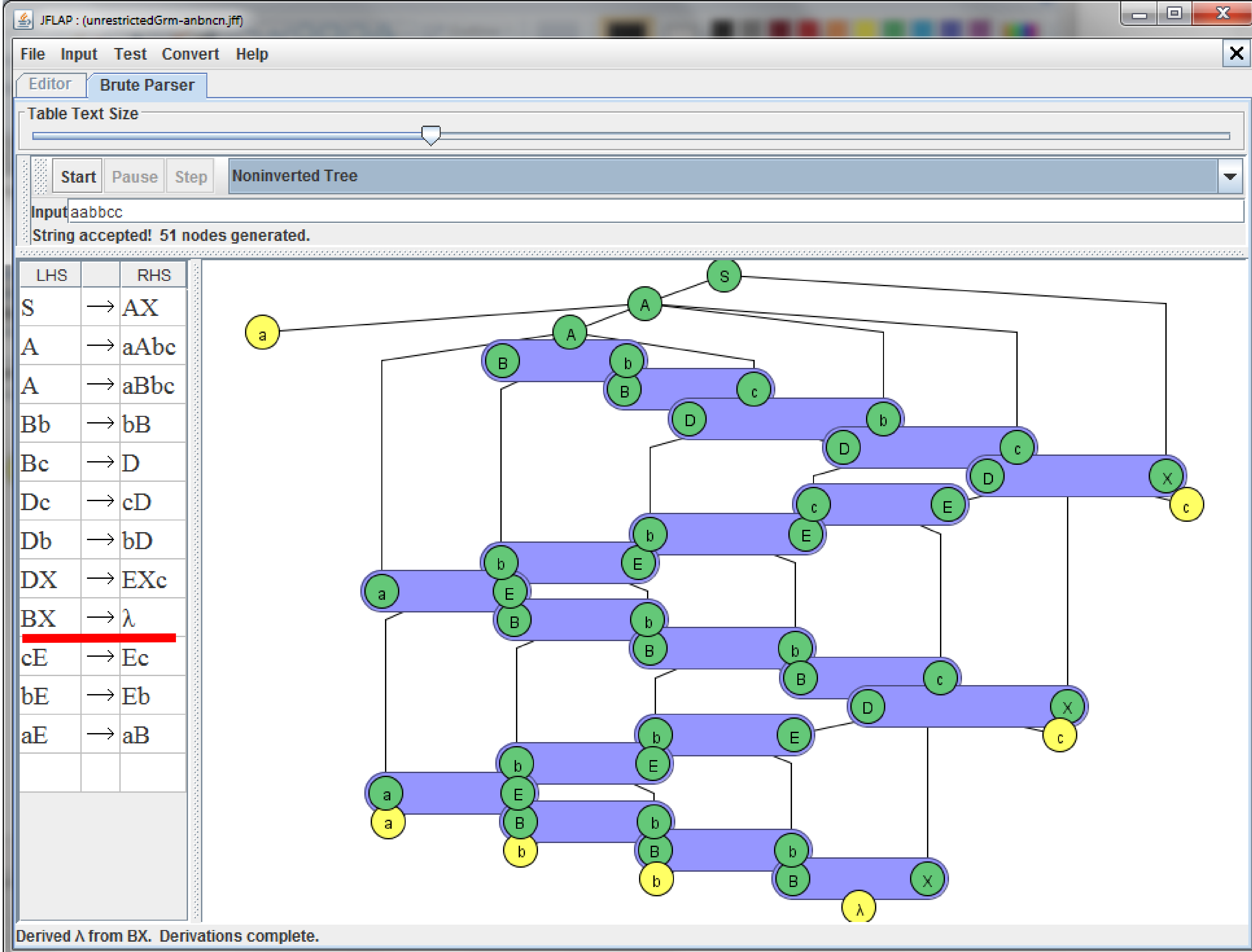

JFLAP : (unrestrictedGrm-anbncn.jff)
File Input Test Convert Help
Editor
Brute Parser
Table Text Size
Start
Pause
Step
Noninverted Tree
Input aabbcc
String accepted! 51 nodes generated.
LHS RHS
S → AX
A → aAbc
A → aBbc
Bb → bB
Bc → D
Dc → cD
Db → bD
DX → EXc
BX → λ
cE → Ec
bE → Eb
aE → aB
Derived Eb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

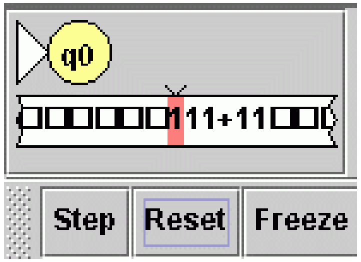

JFLAP : (unrestrictedGrm-anbncn.jff)
File Input Test Convert Help
Editor
Brute Parser
Table Text Size
Start
Pause
Step
Noninverted Tree
Input aabbcc
String accepted! 51 nodes generated.
LHS RHS
S → AX
A → aAbc
A → aBbc
Bb → bB
Bc → D
Dc → cD
Db → bD
DX → EXc
BX → λ
cE → Ec
bE → Eb
aE → aB
Derived bB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

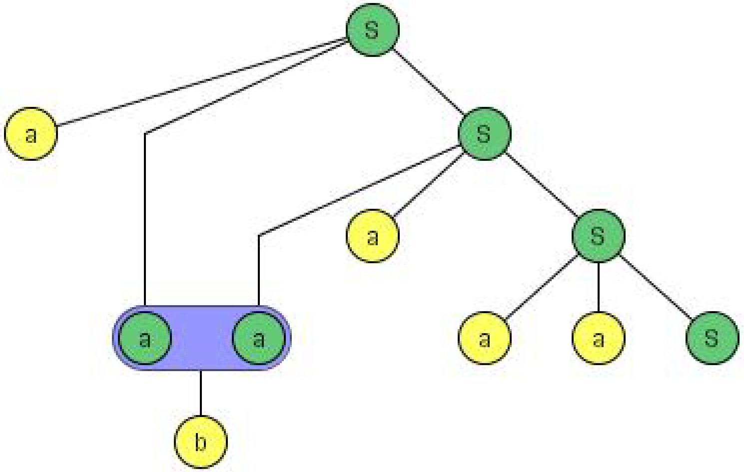
What else can JFLAP do?

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



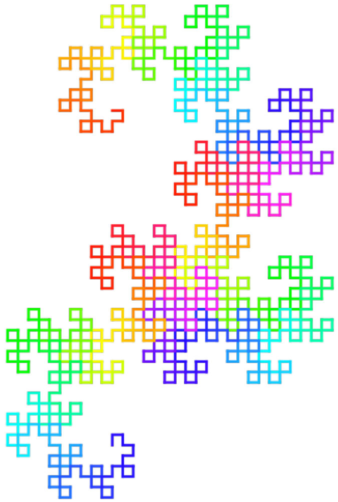
- Parsing of grammars
  - regular, context-free grammars
  - Unrestricted grammar

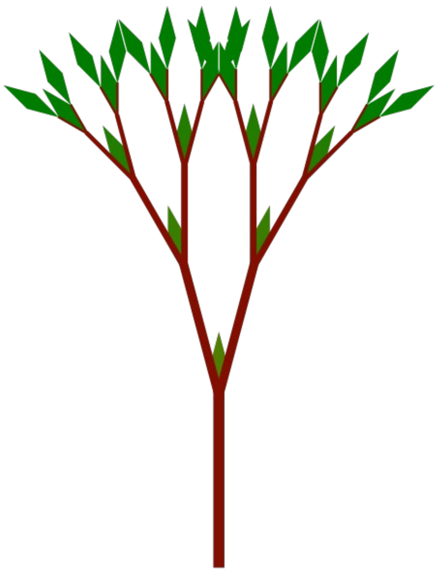
- Conversions for proofs
  - NFA to DFA to minimal DFA
  - NFA ←→ regular expression
  - NFA ←→ regular grammar
  - CFG ←→ NPDA

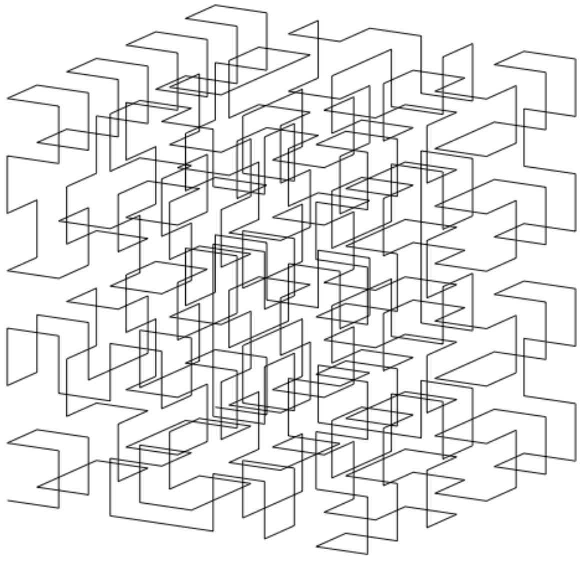


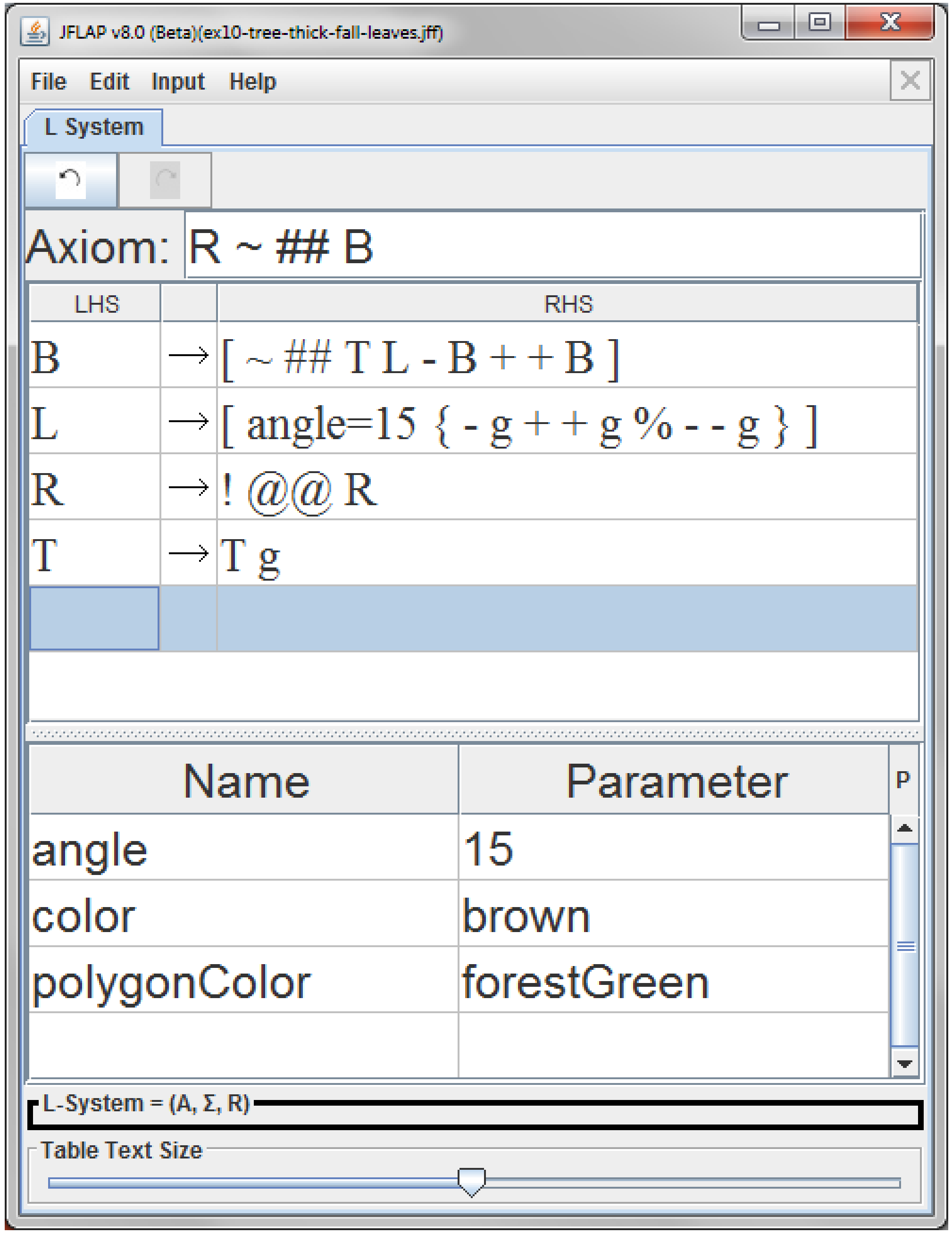
JFLAP - L-Systems

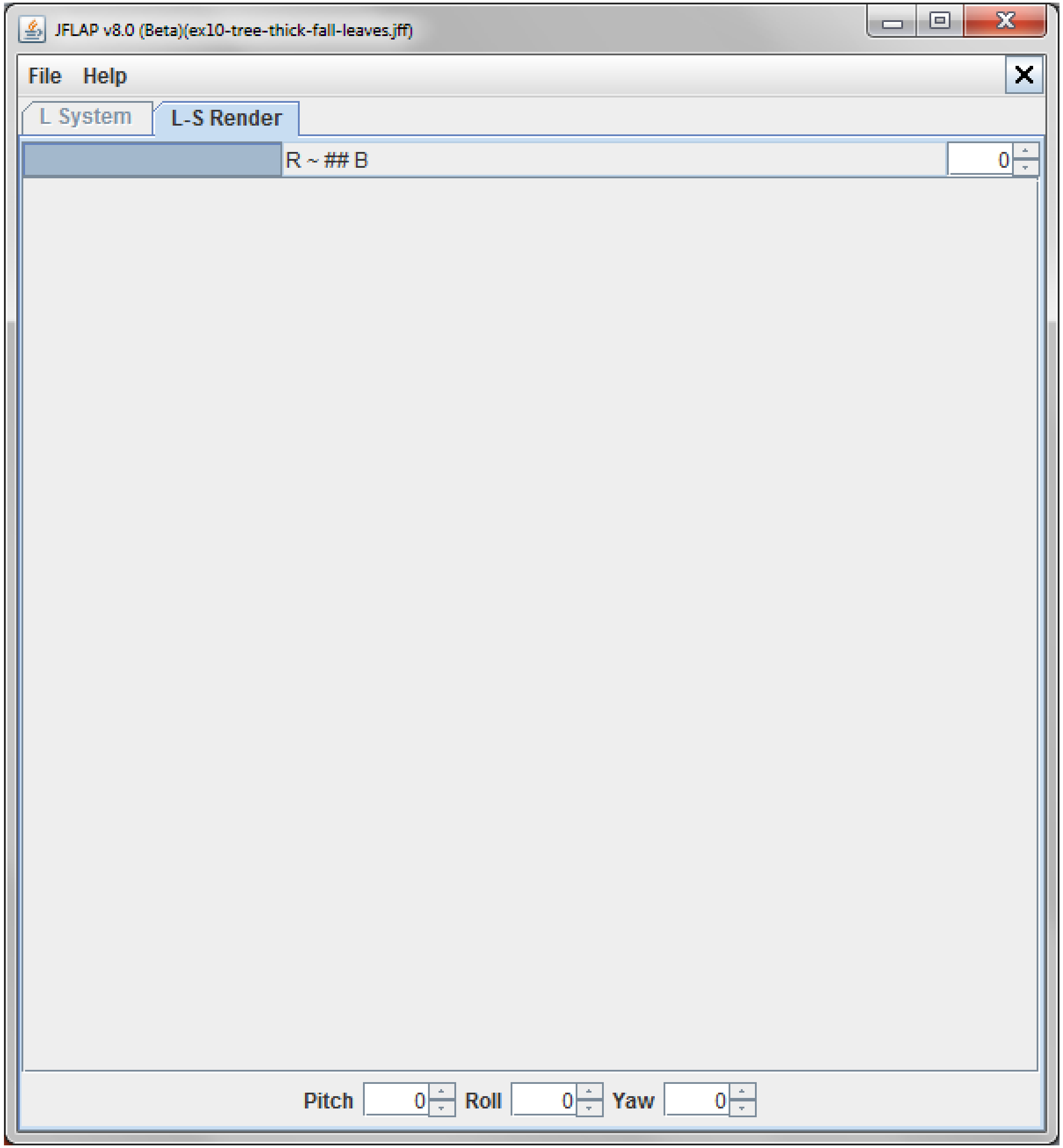
- L-Systems may be used to model biological systems and create fractals.
- Similar to Chomsky grammars, except *all* variables are replaced in each derivation step, not just one!
- Commonly, strings from successive derivations are interpreted as strings of render commands and are displayed graphically.

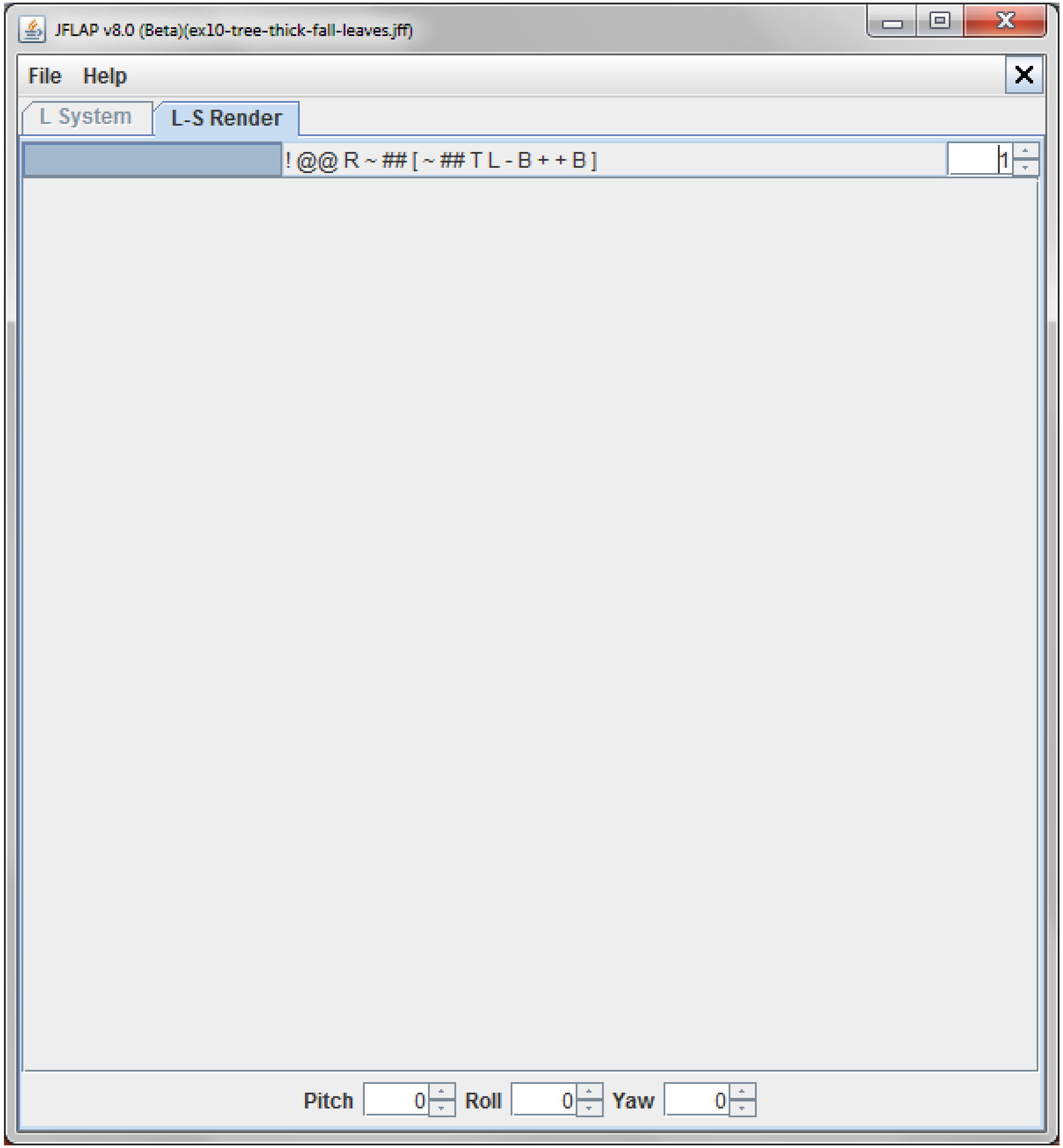


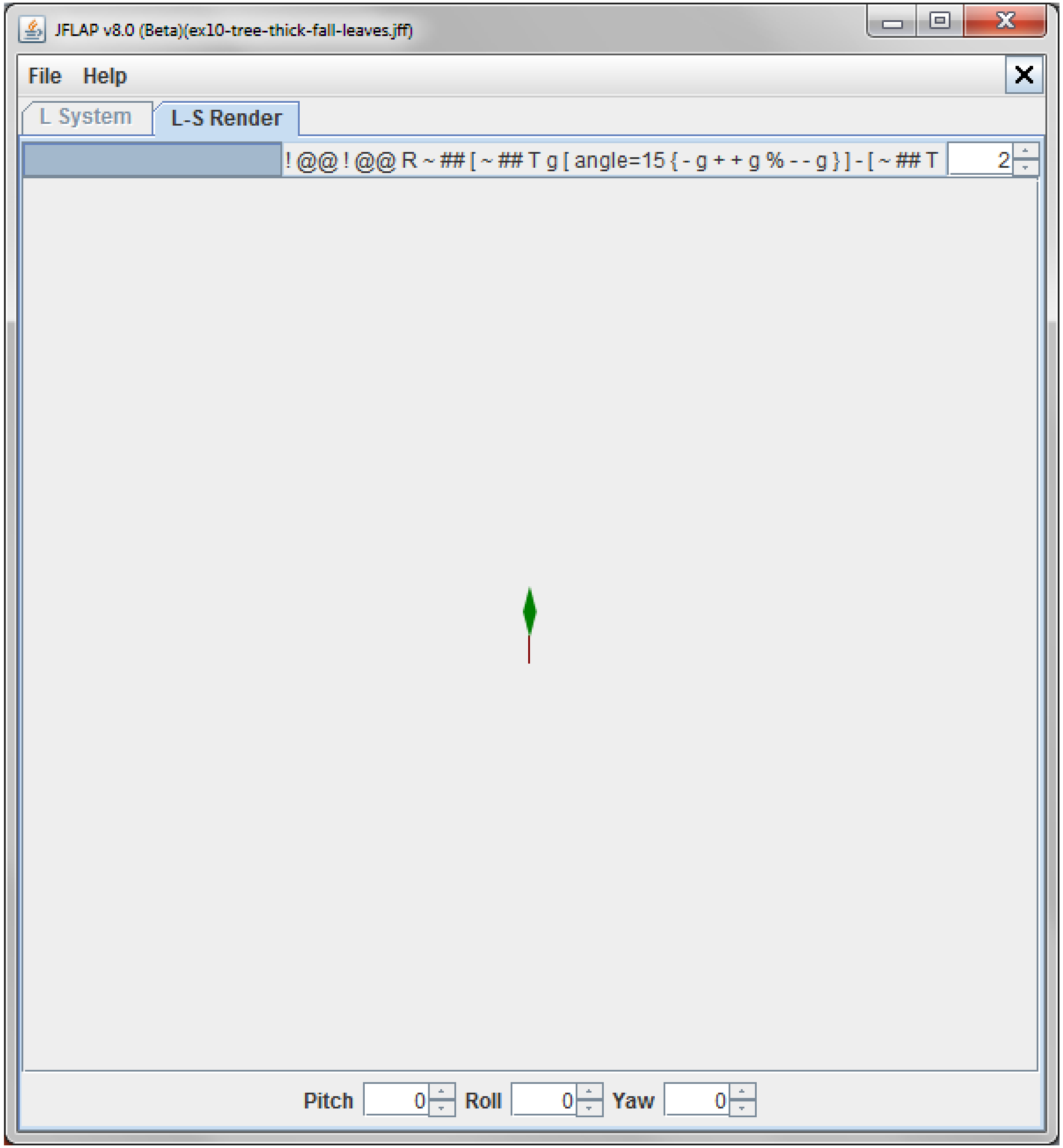




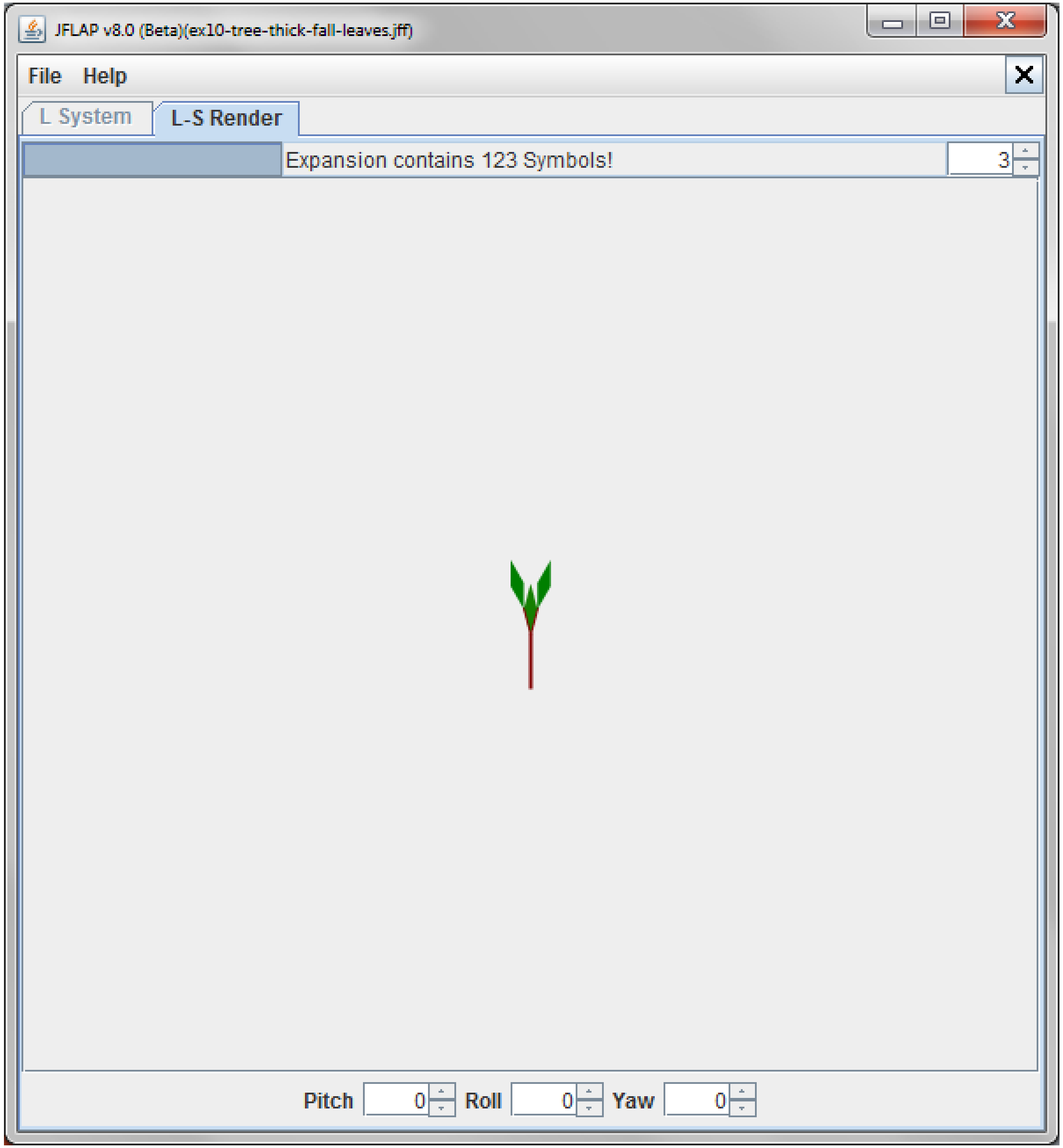

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File Edit Input Help
L System
Axiom: R ~ ## B
LHS
RHS
B → [~ ## T L - B + + B]
L → [angle=15 { - g + + g % - - g }]
R → ! @@ R
T → T g
Name
Parameter
P
angle
15
color
brown
polygonColor
forestGreen
L-System = (A, Σ, R)
Table Text Size

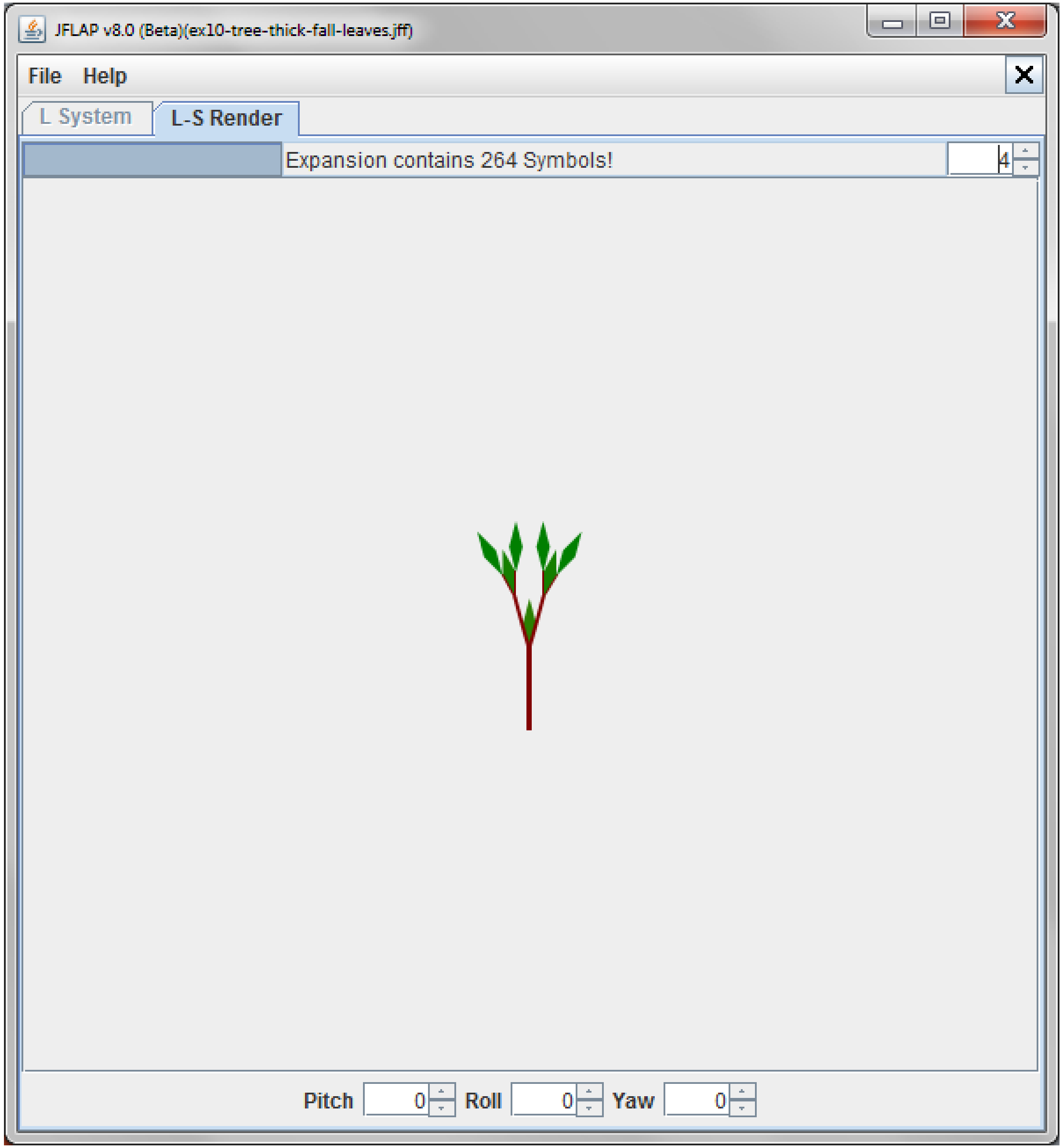

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
R ~ ## B
0
Pitch
0
Roll
0
Yaw
0

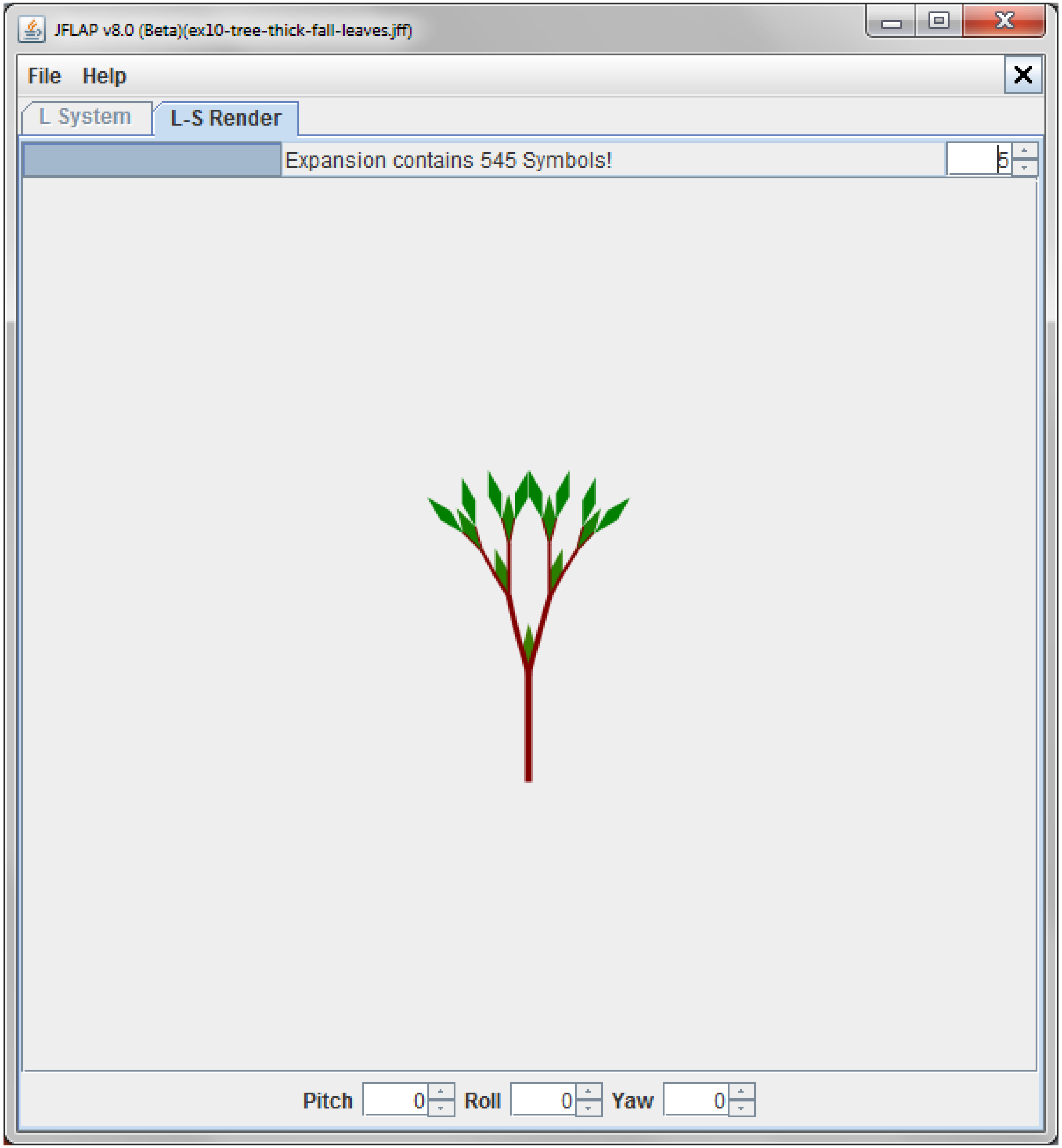

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
! @@ R ~ ## [~ ## T L - B + + B]
1
Pitch
0
Roll
0
Yaw
0

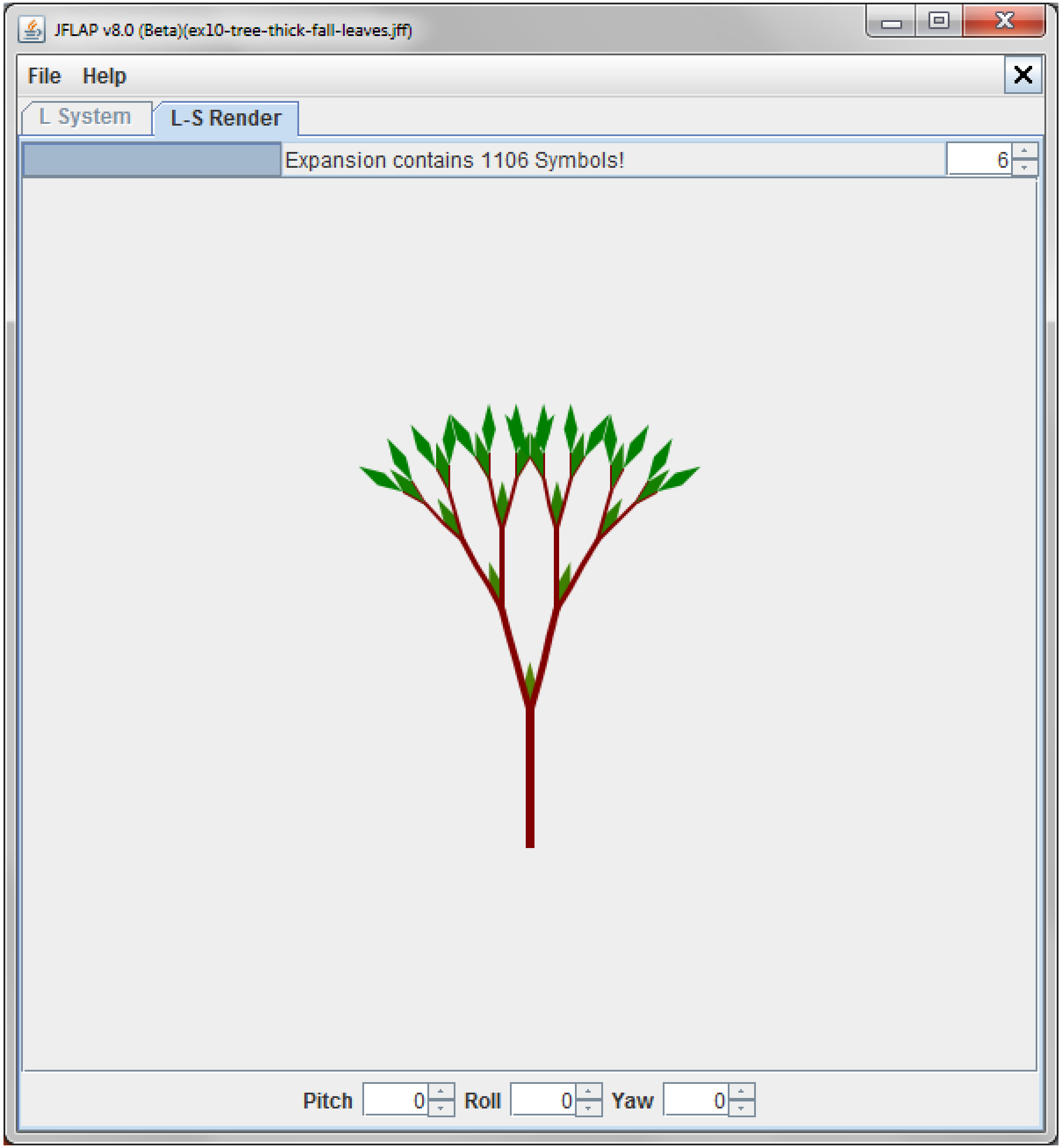


JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
! @@ ! @@ R ~ ## [~ ## T g [angle=15 { - g + + g % - - g }] - [~ ## T
2
Pitch
0
Roll
0
Yaw
0

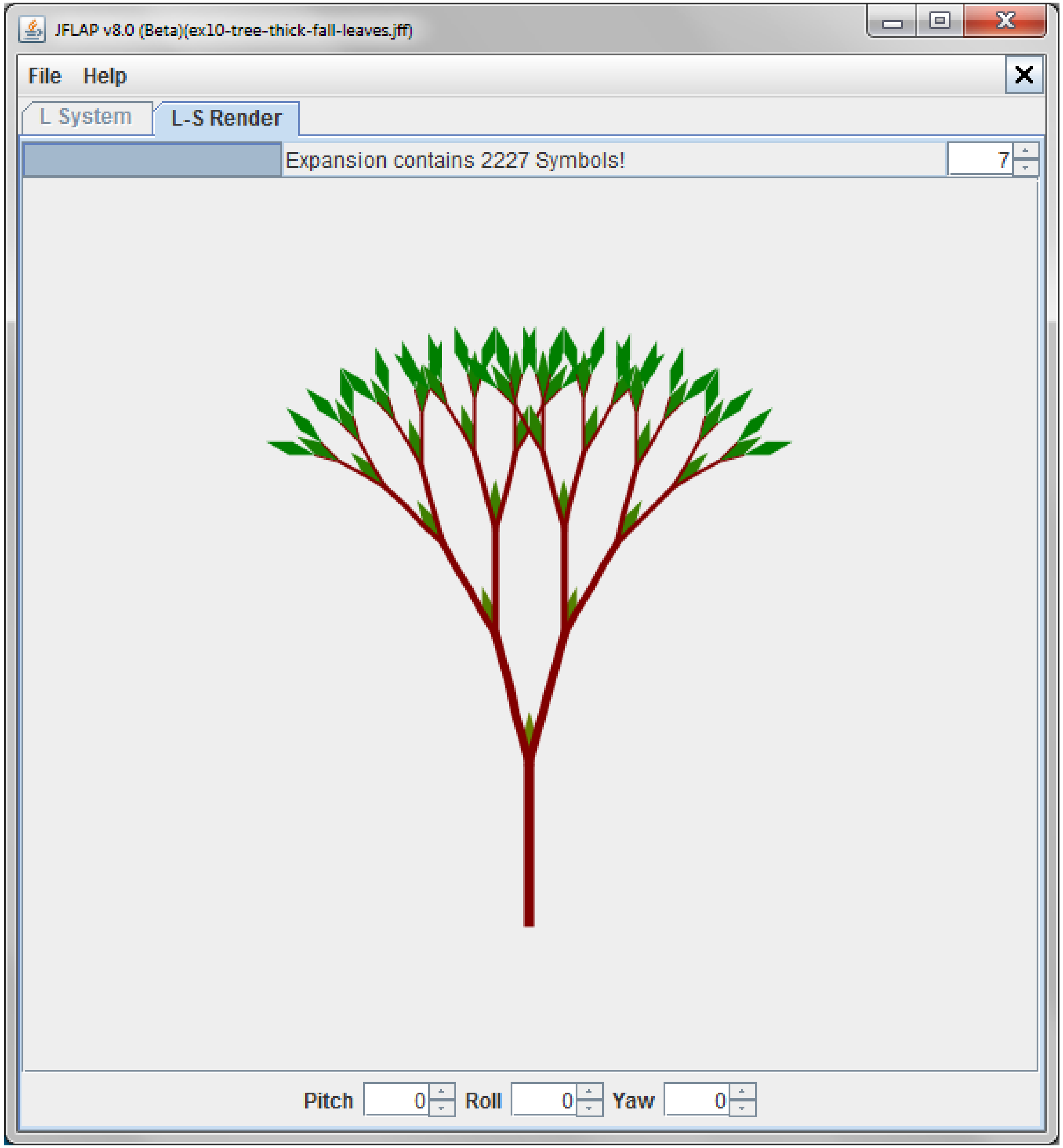

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 123 Symbols!
3
Pitch
0
Roll
0
Yaw
0


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

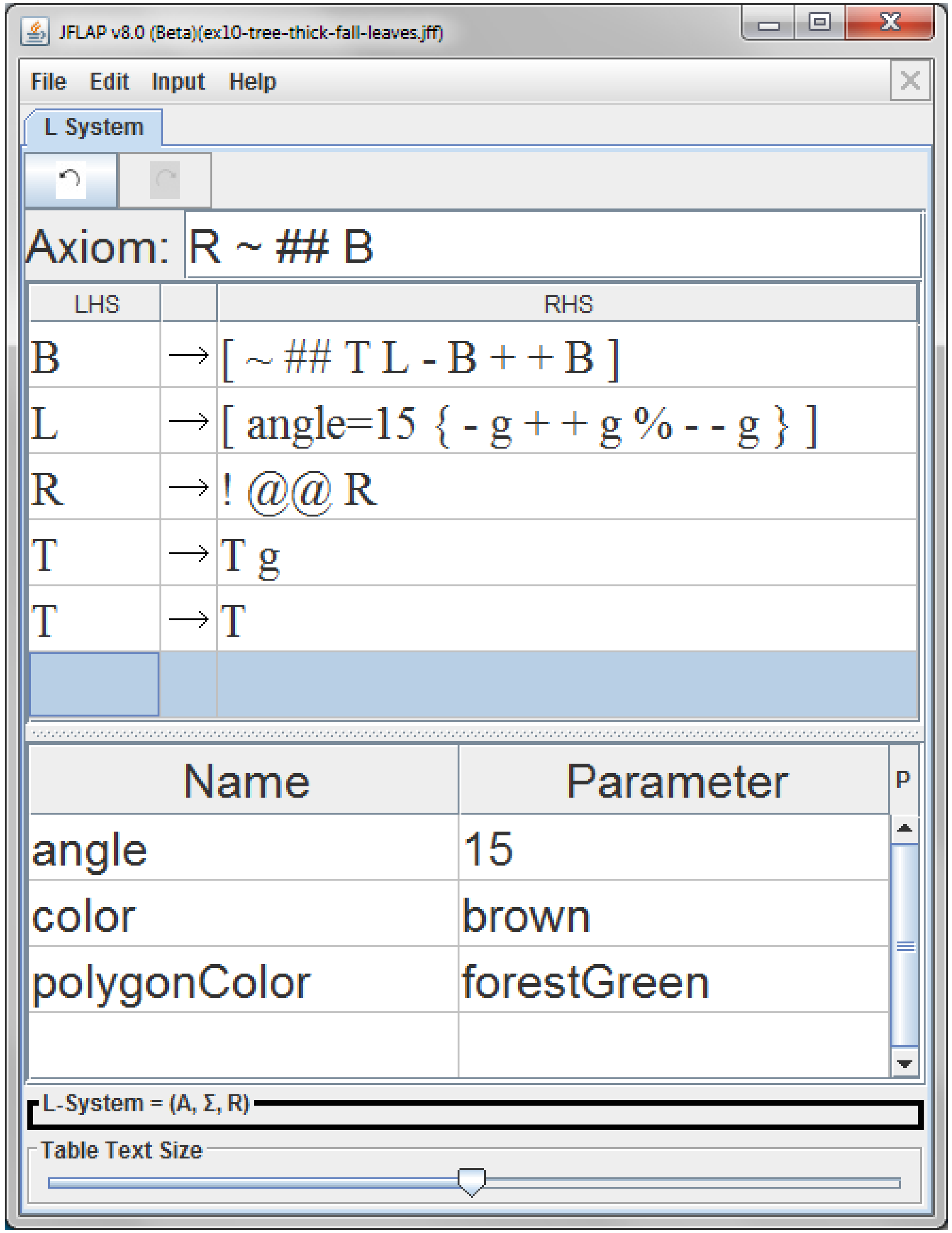


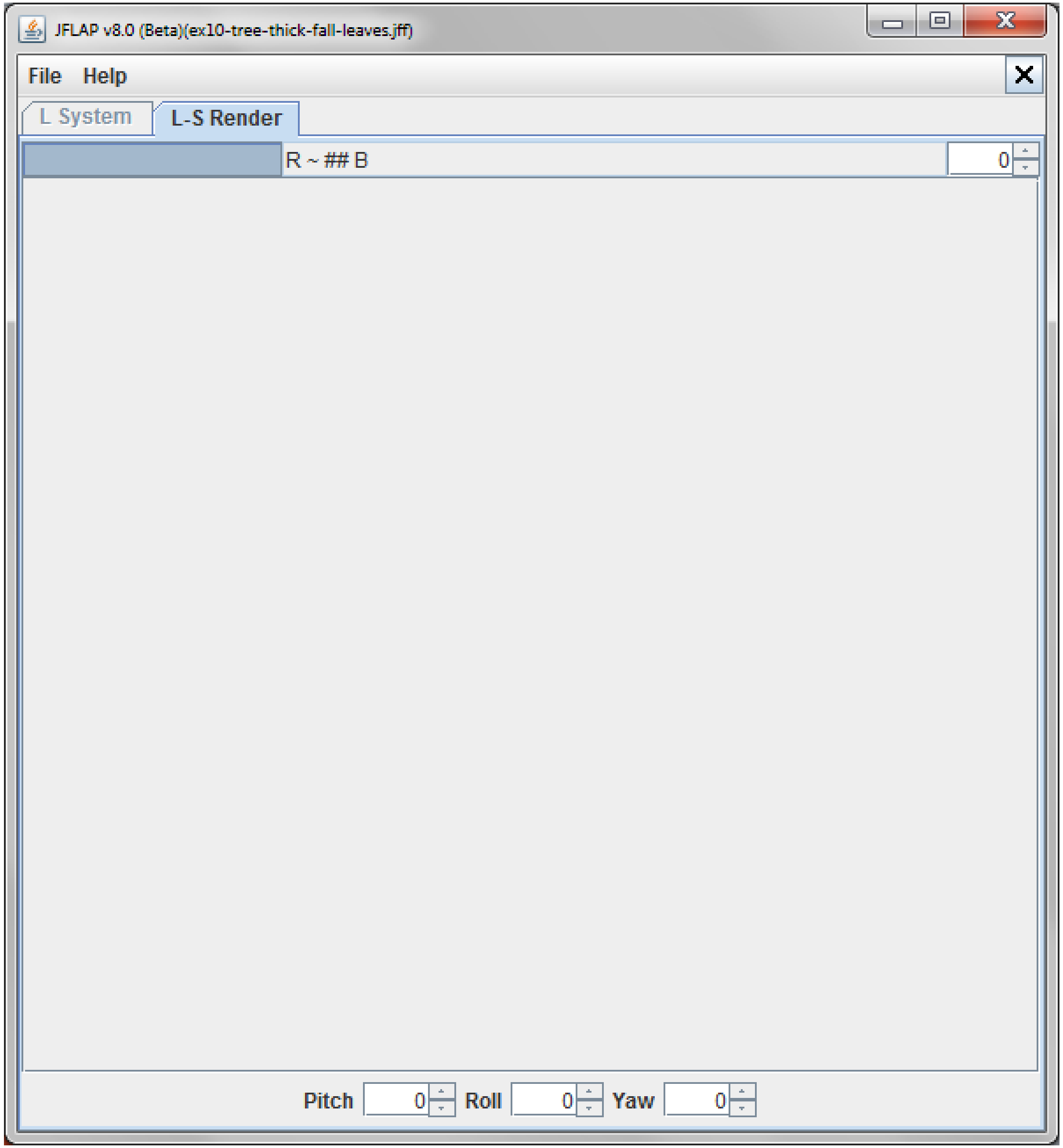
JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 1106 Symbols!
6
Pitch
0
Roll
0
Yaw
0

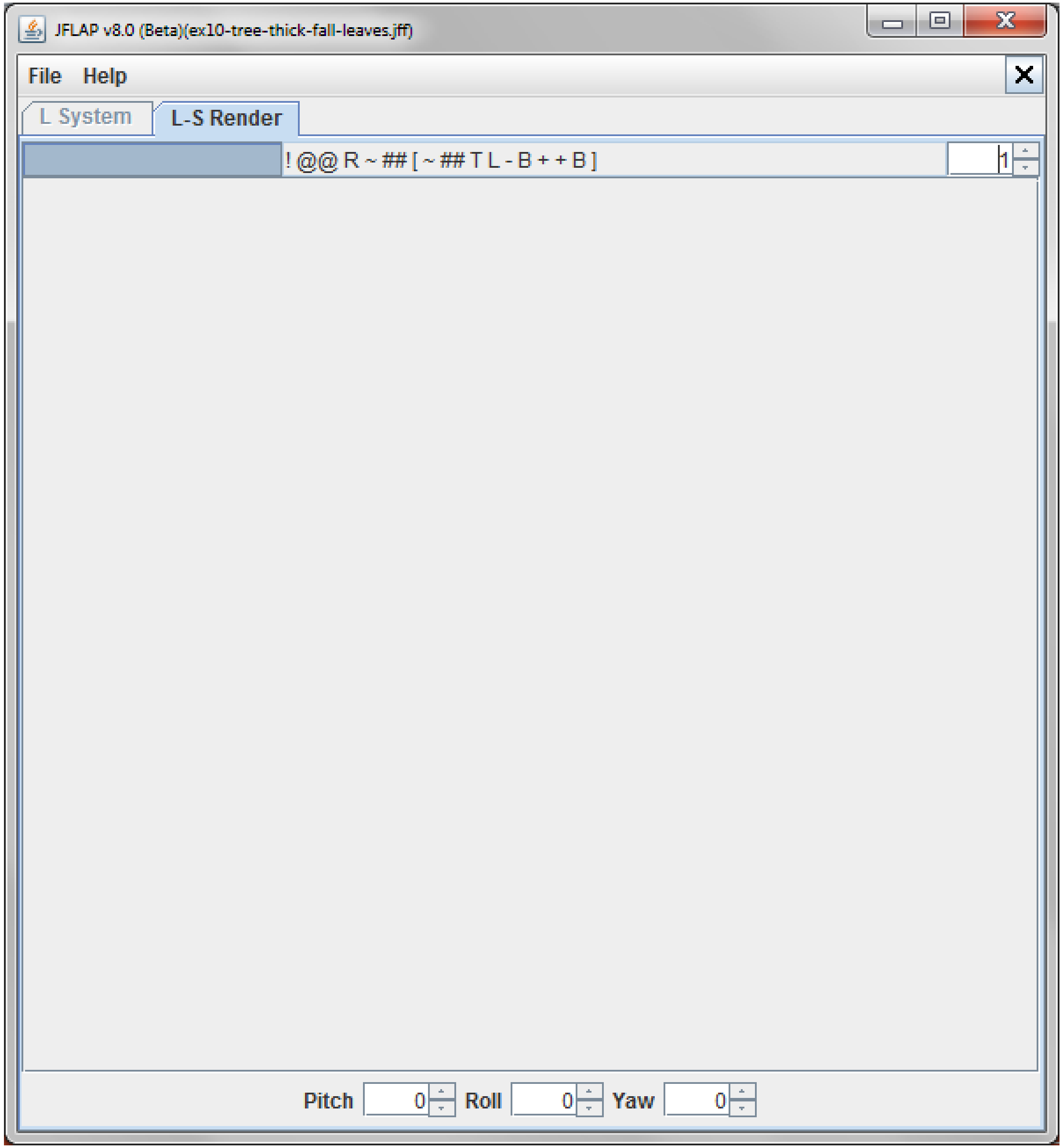


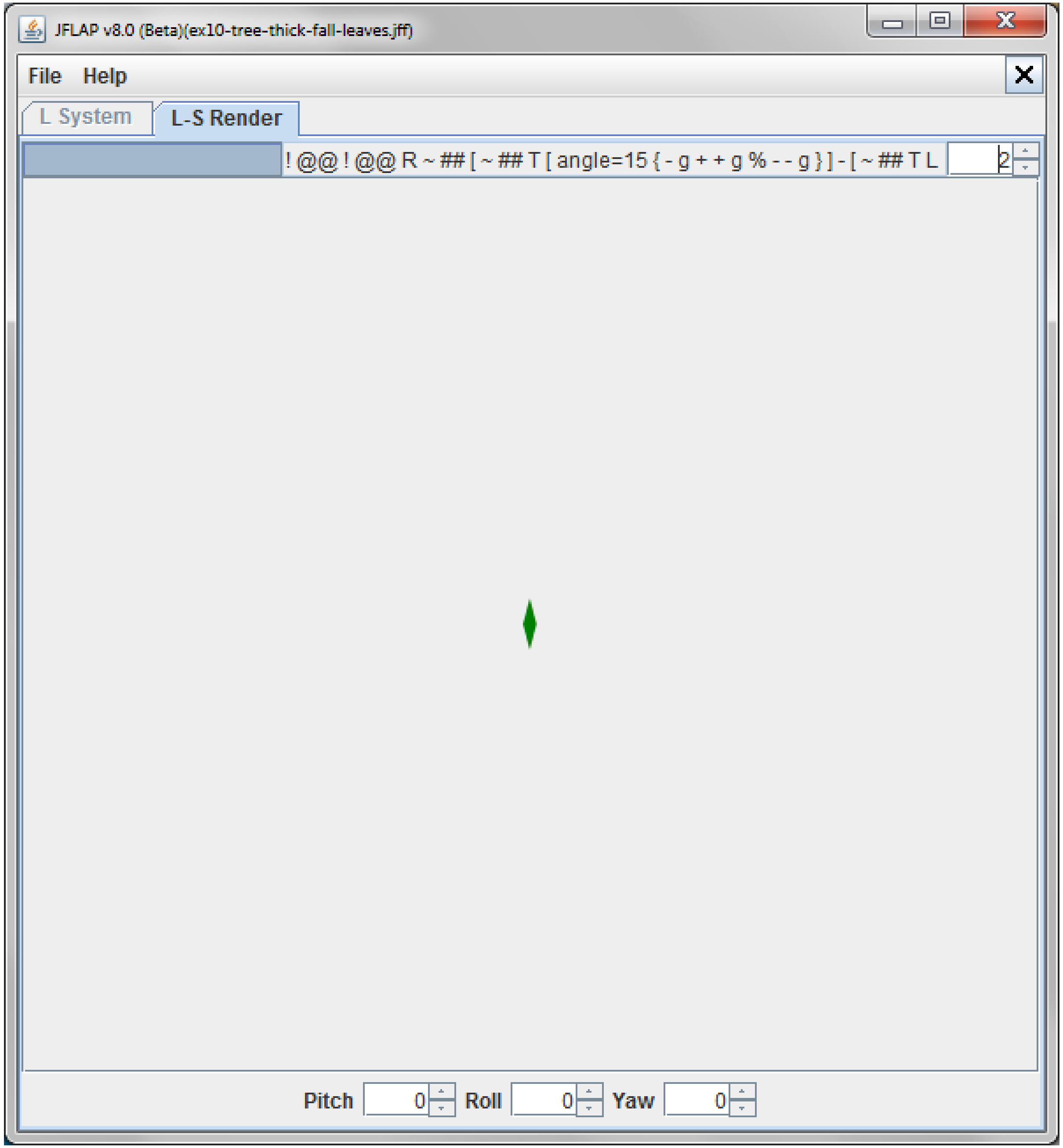
JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 2227 Symbols!
7
Pitch
0
Roll
0
Yaw
0

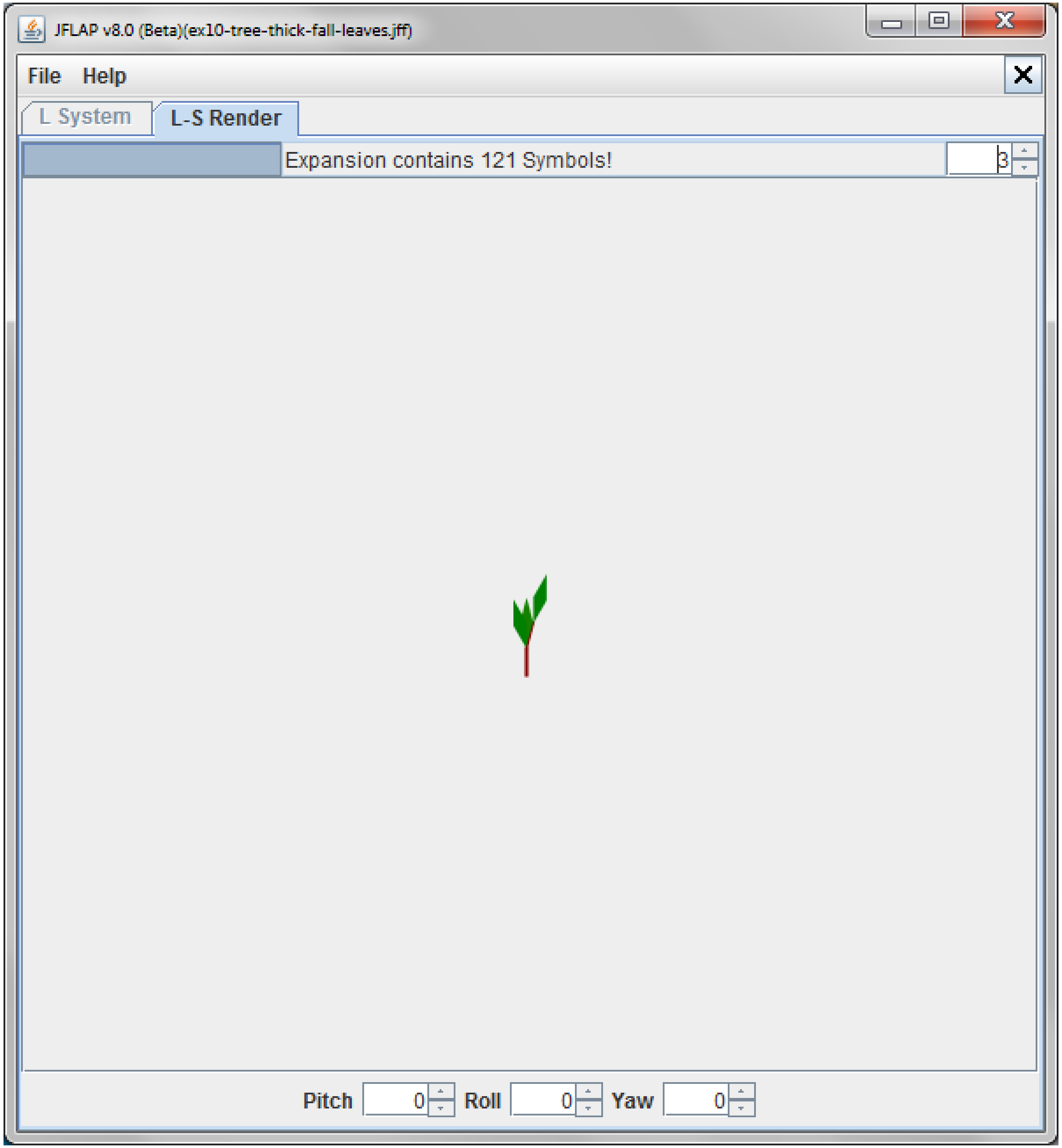
Add second T rule

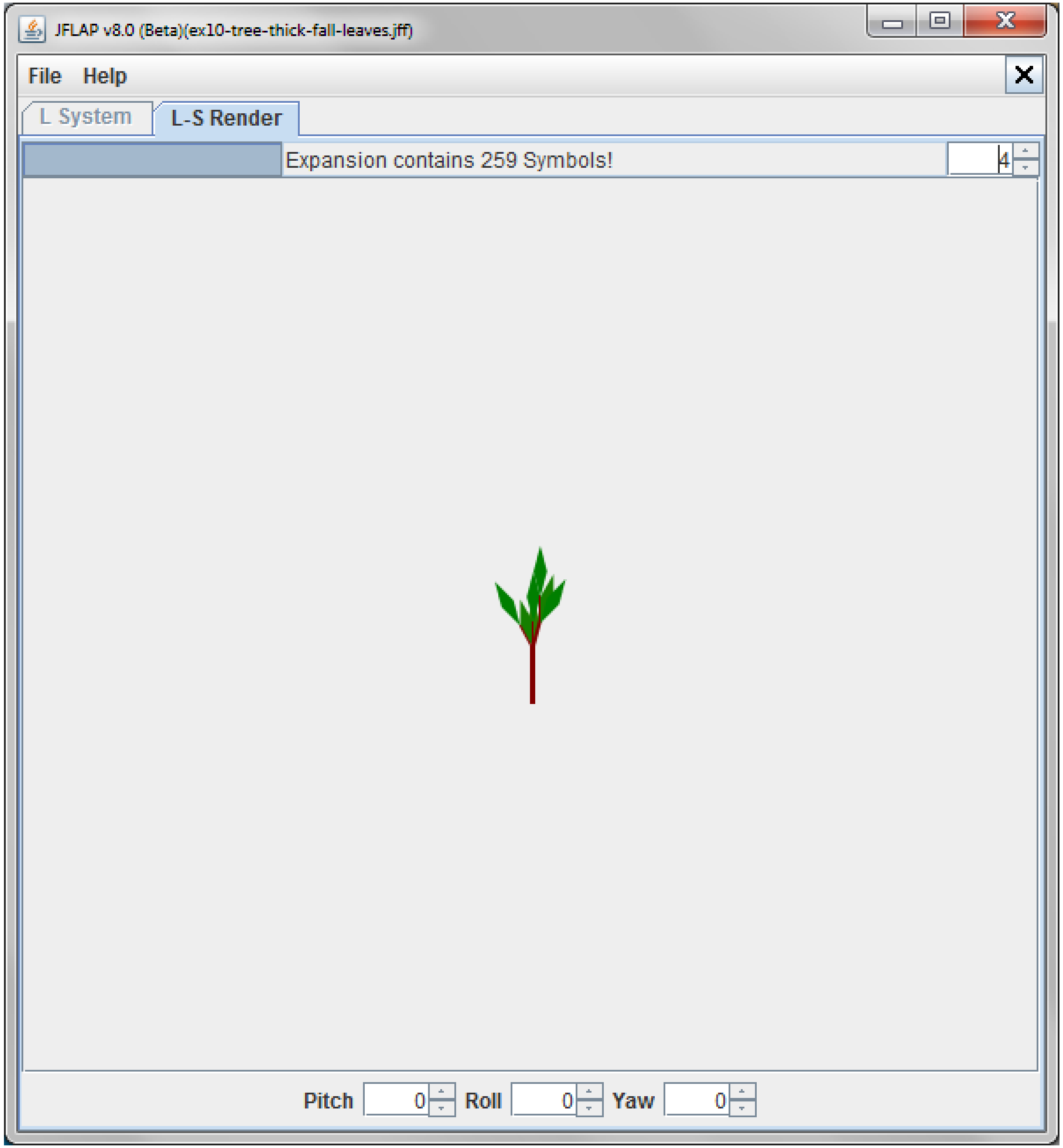


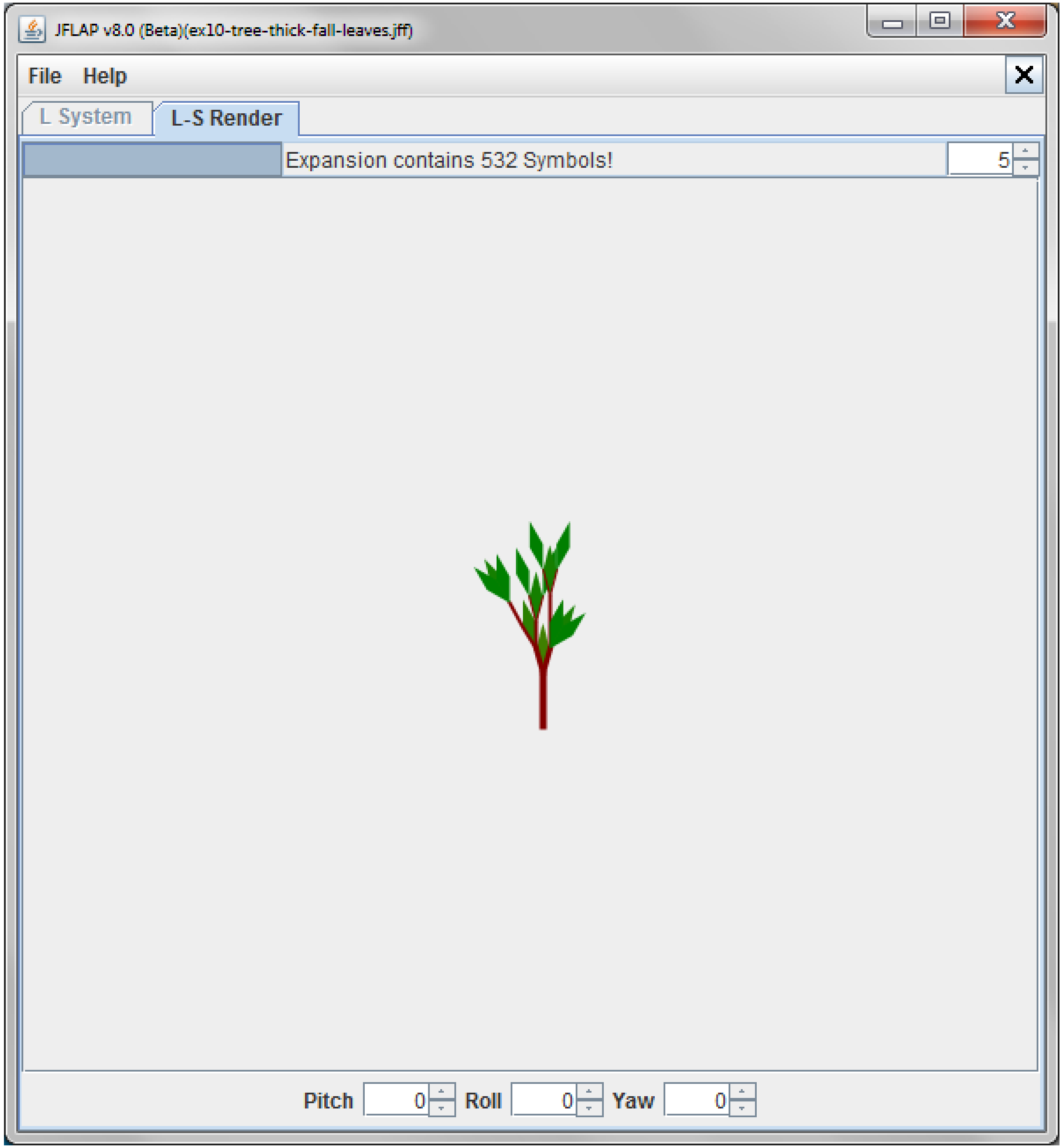

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
R ~ ## B
0
Pitch
0
Roll
0
Yaw
0

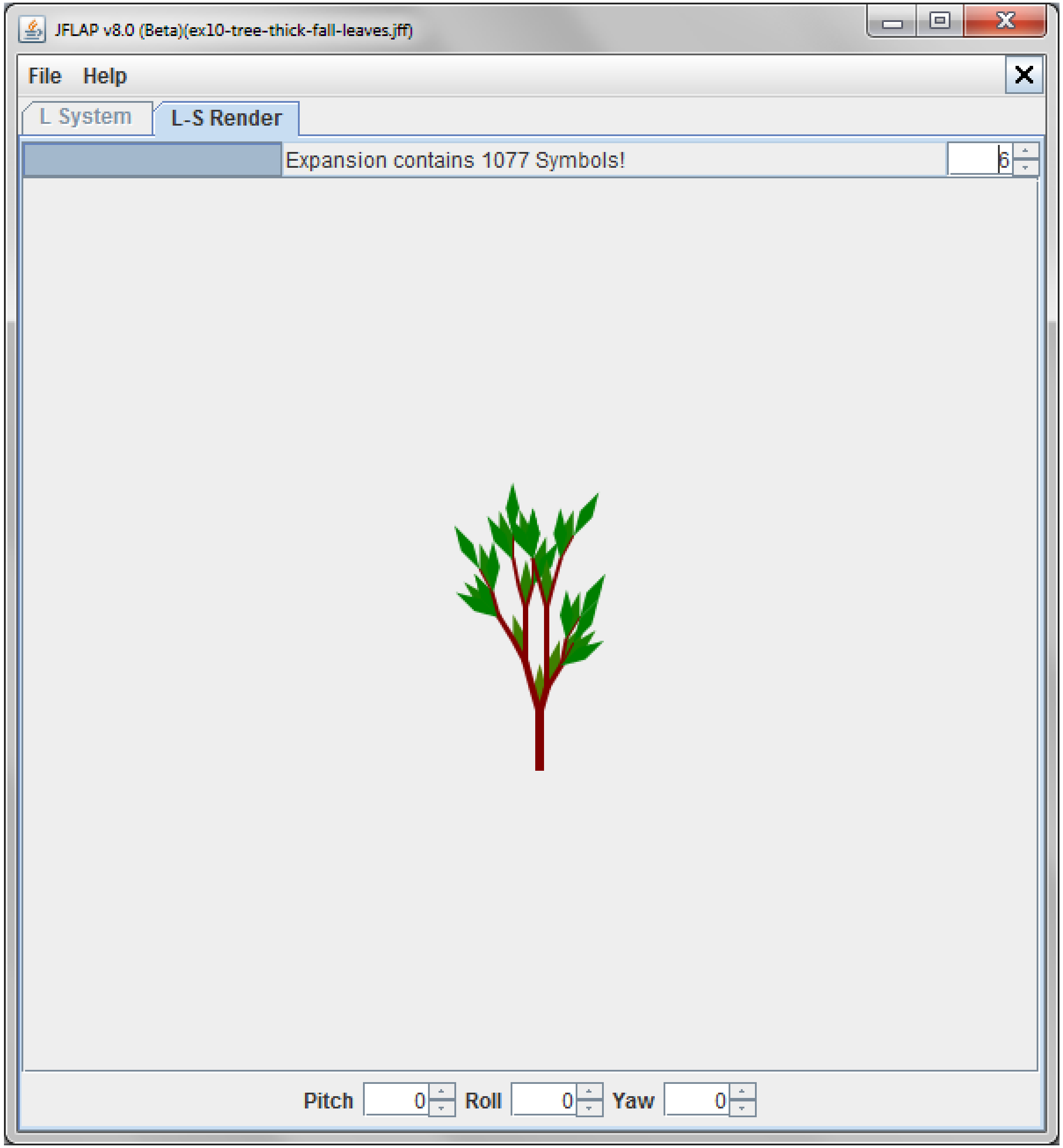

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File Help
L System
L-S Render
! @@ R ~ ## [~ ## T L - B + + B]
1
Pitch 0 Roll 0 Yaw 0


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

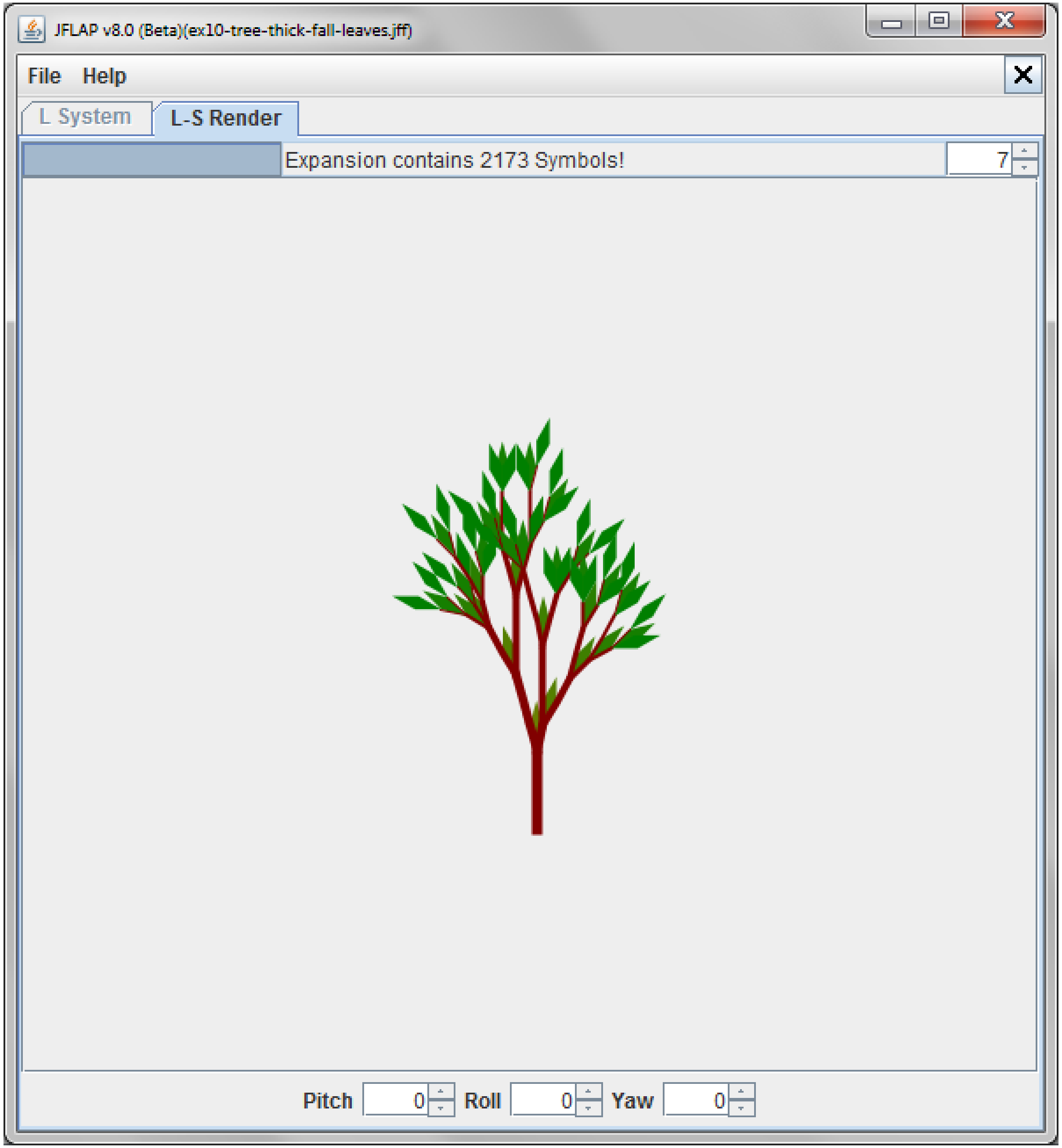

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 121 Symbols!
3
Pitch
0
Roll
0
Yaw
0

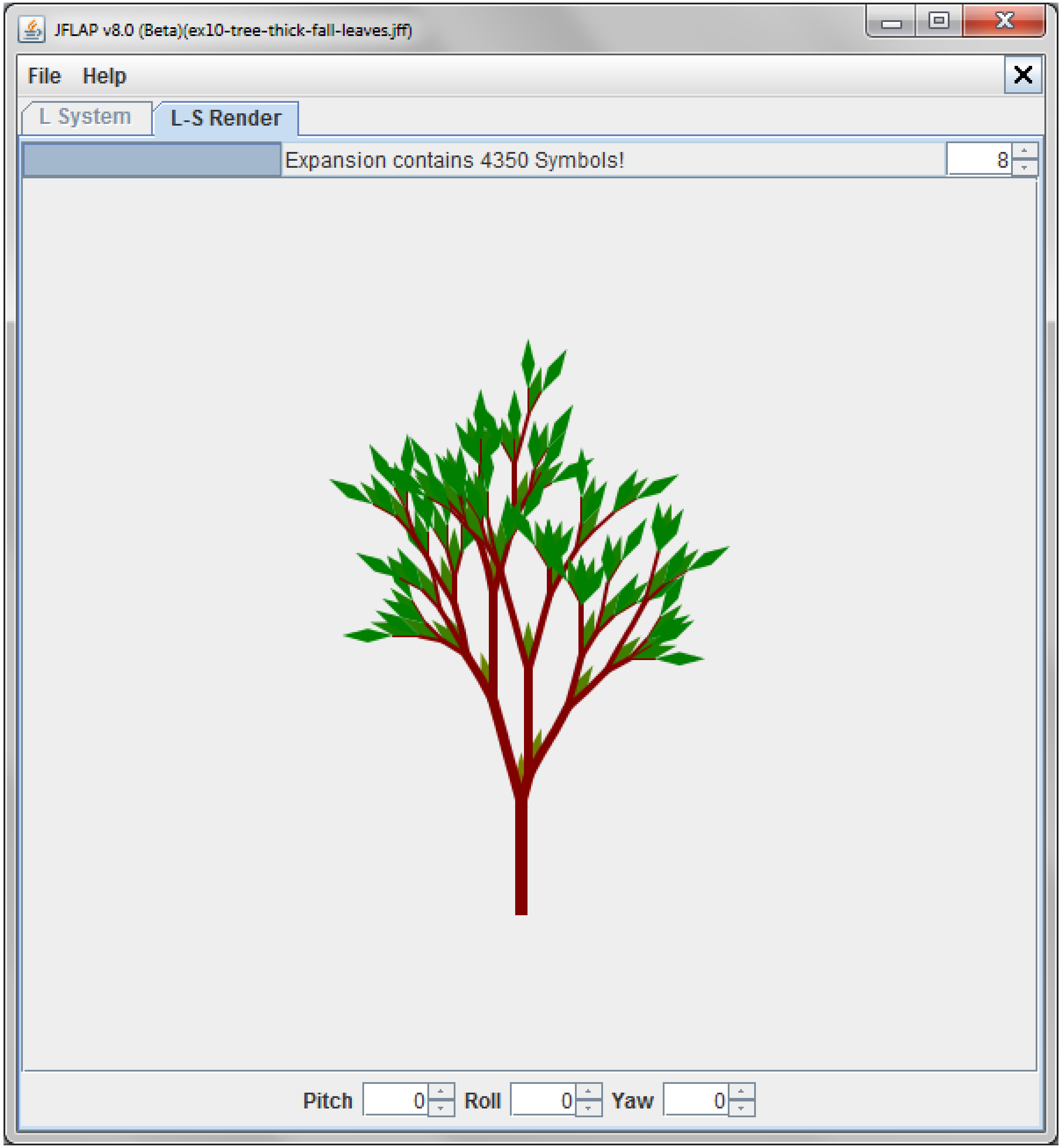

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 259 Symbols!
4
Pitch
0
Roll
0
Yaw
0

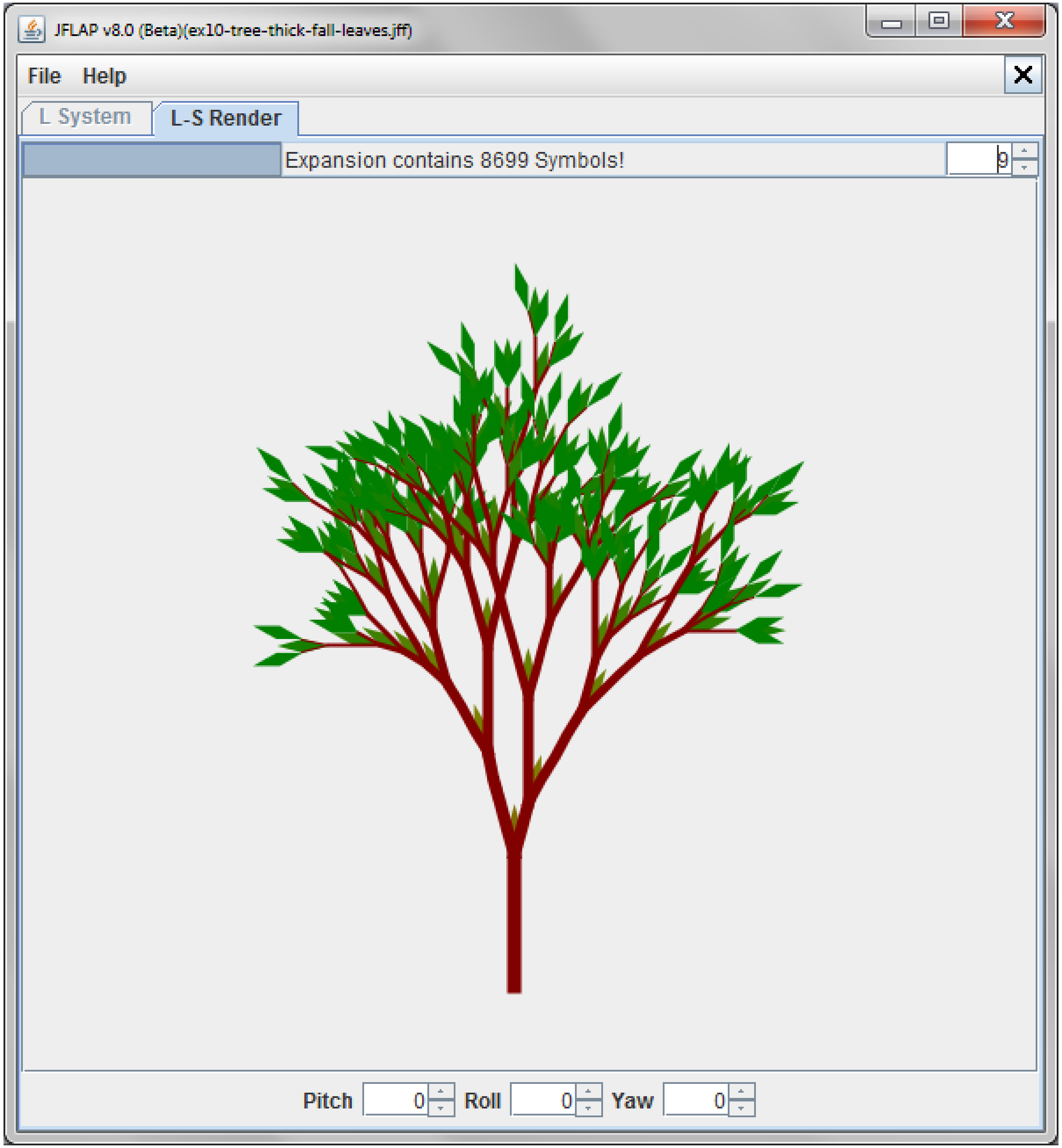

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File Help
L System
L-S Render
Expansion contains 532 Symbols!
5
Pitch 0 Roll 0 Yaw 0

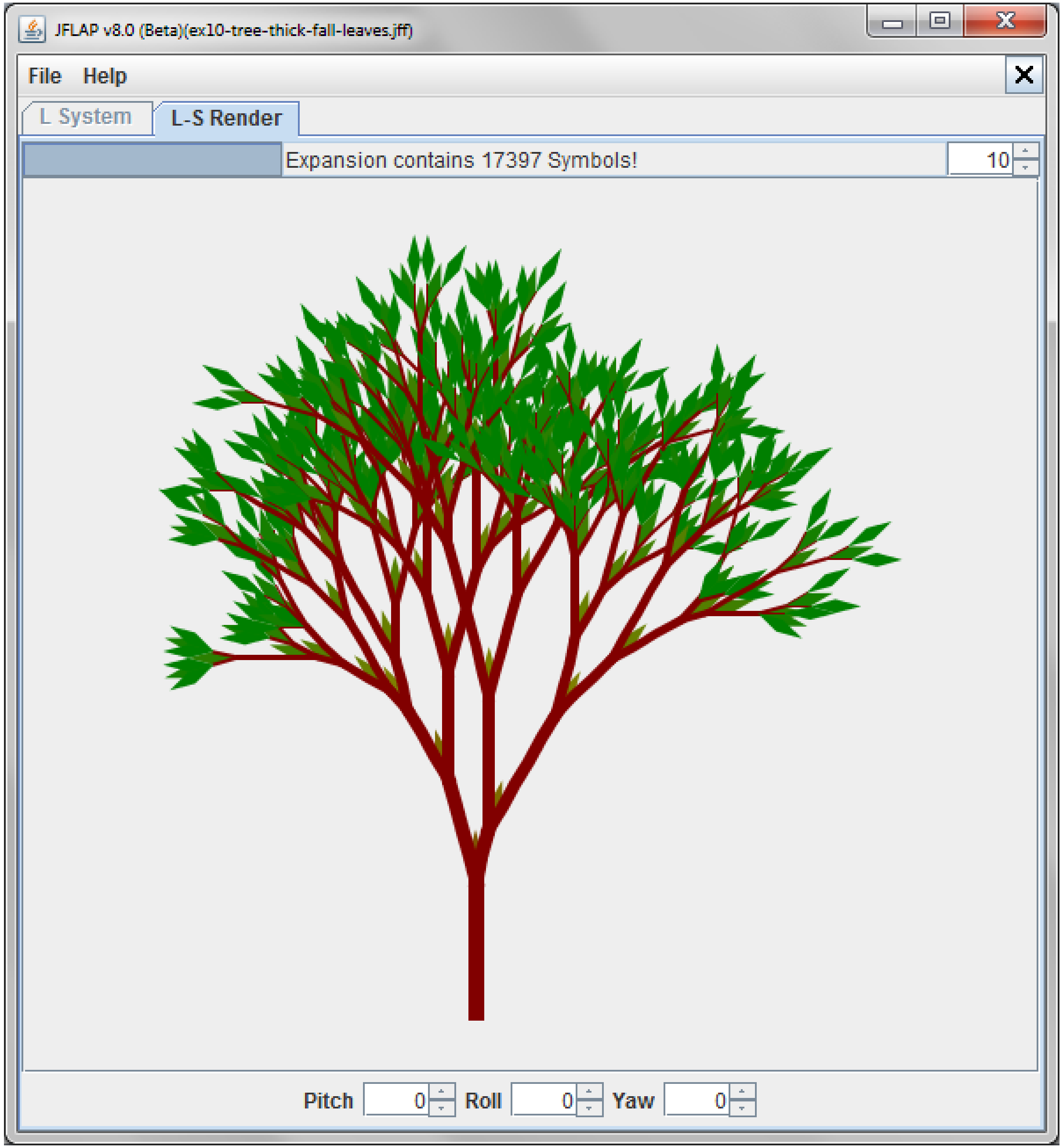


JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 1077 Symbols!
6
Pitch
0
Roll
0
Yaw
0


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

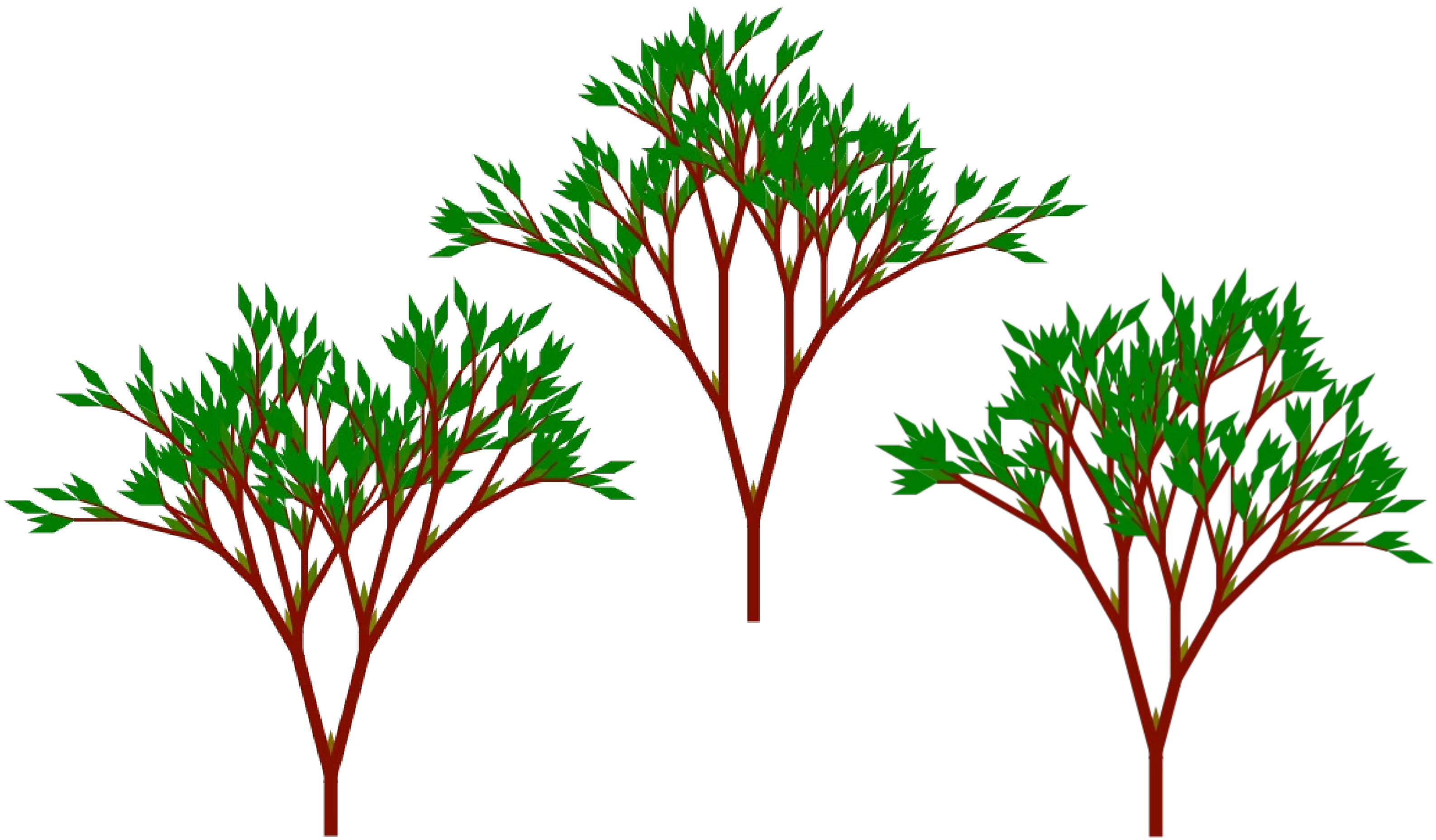

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 4350 Symbols!
8
Pitch
0
Roll
0
Yaw
0


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

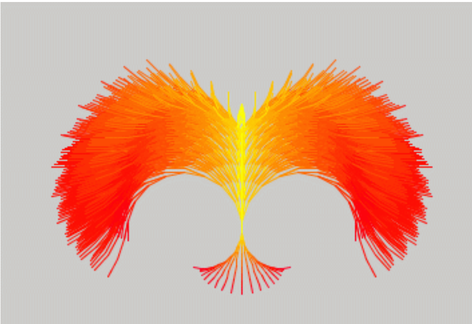

JFLAP v8.0 (Beta)(ex10-tree-thick-fall-leaves.jff)
File
Help
L System
L-S Render
Expansion contains 17397 Symbols!
10
Pitch
0
Roll
0
Yaw
0

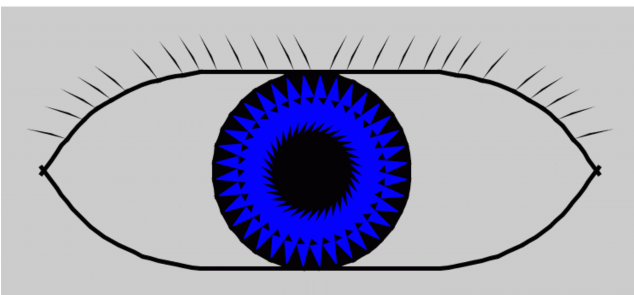
L-Systems

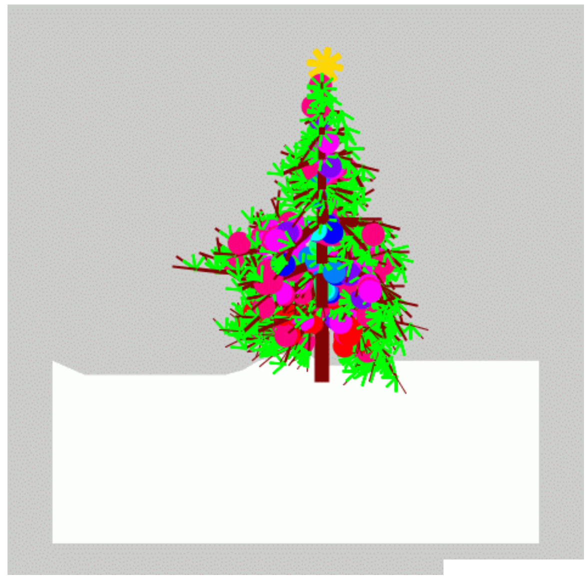
The same stochastic L-system, rendered
3 different times all at the 9th derivation.

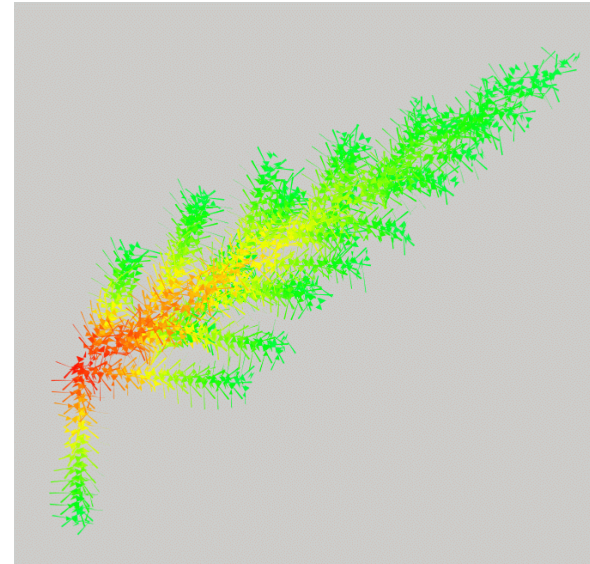


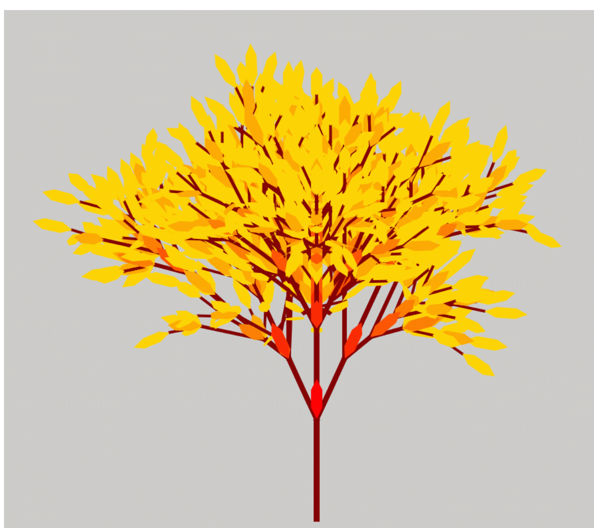
Students like L-systems

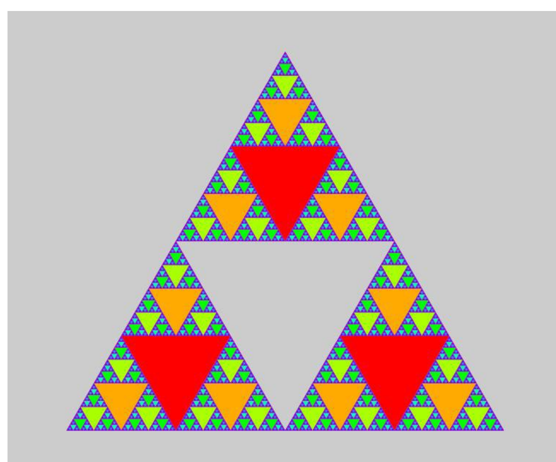


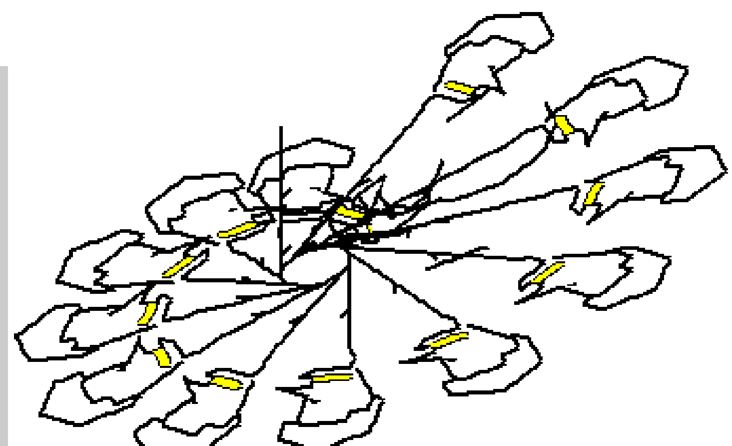












Two-year JFLAP Study
2005-2007

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

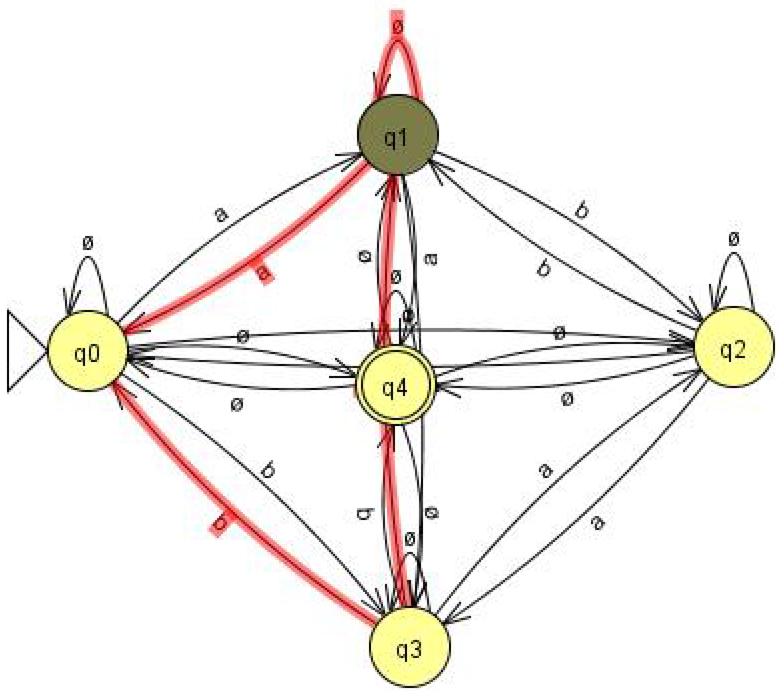
Conclusions From Study

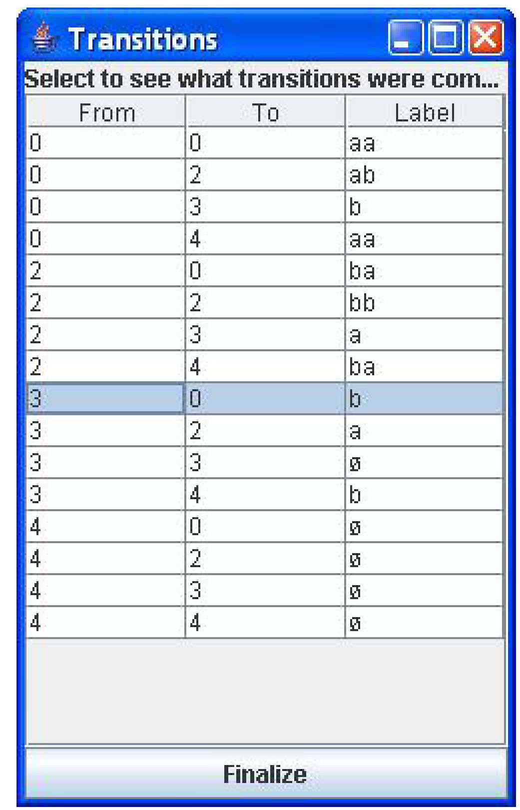
- Results of Study showed
  - All the faculty used JFLAP in their courses, mostly for homework, some in lecture
  - Students had a high opinion of JFLAP
  - Majority of students felt access to JFLAP
    - Made learning course concepts easier
    - Made them feel more engaged
    - Made the course more 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

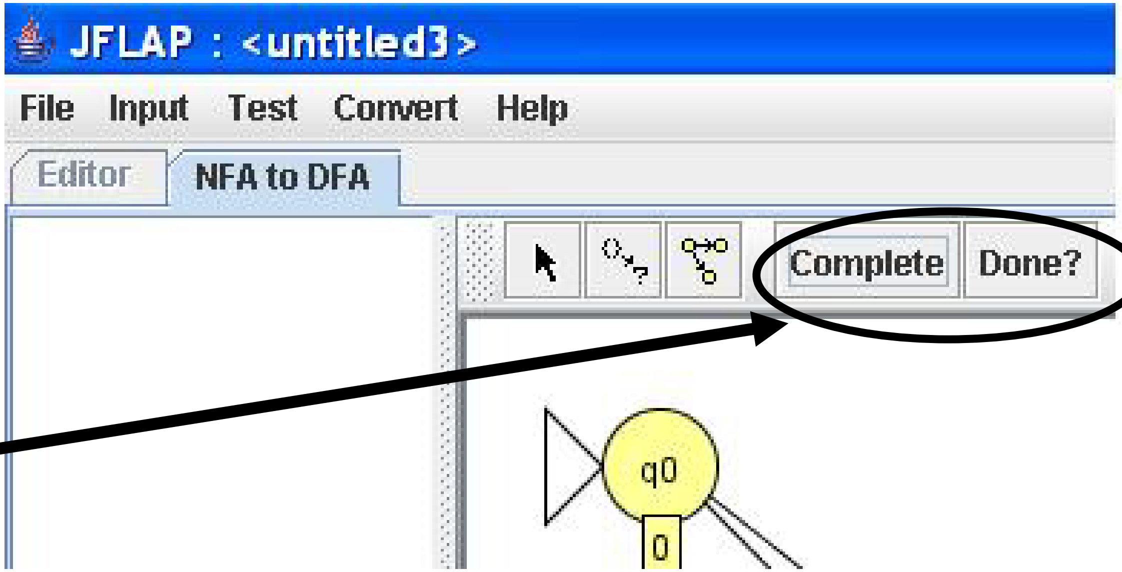




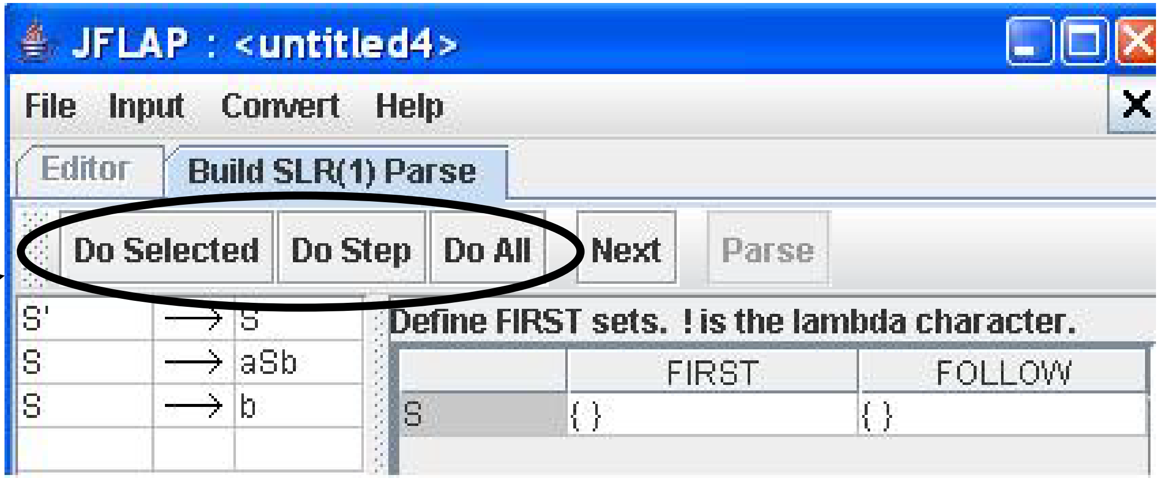
From	To	Label
0	0	aa
0	2	ab
0	3	b
0	4	aa
2	0	ba
2	2	bb
2	3	a
2	4	ba
3	0	b
3	2	a
3	3	ø
3	4	b
4	0	ø
4	2	ø
4	3	ø
4	4	ø

Allow user to proceed on if they got it

- Complete the rest for them

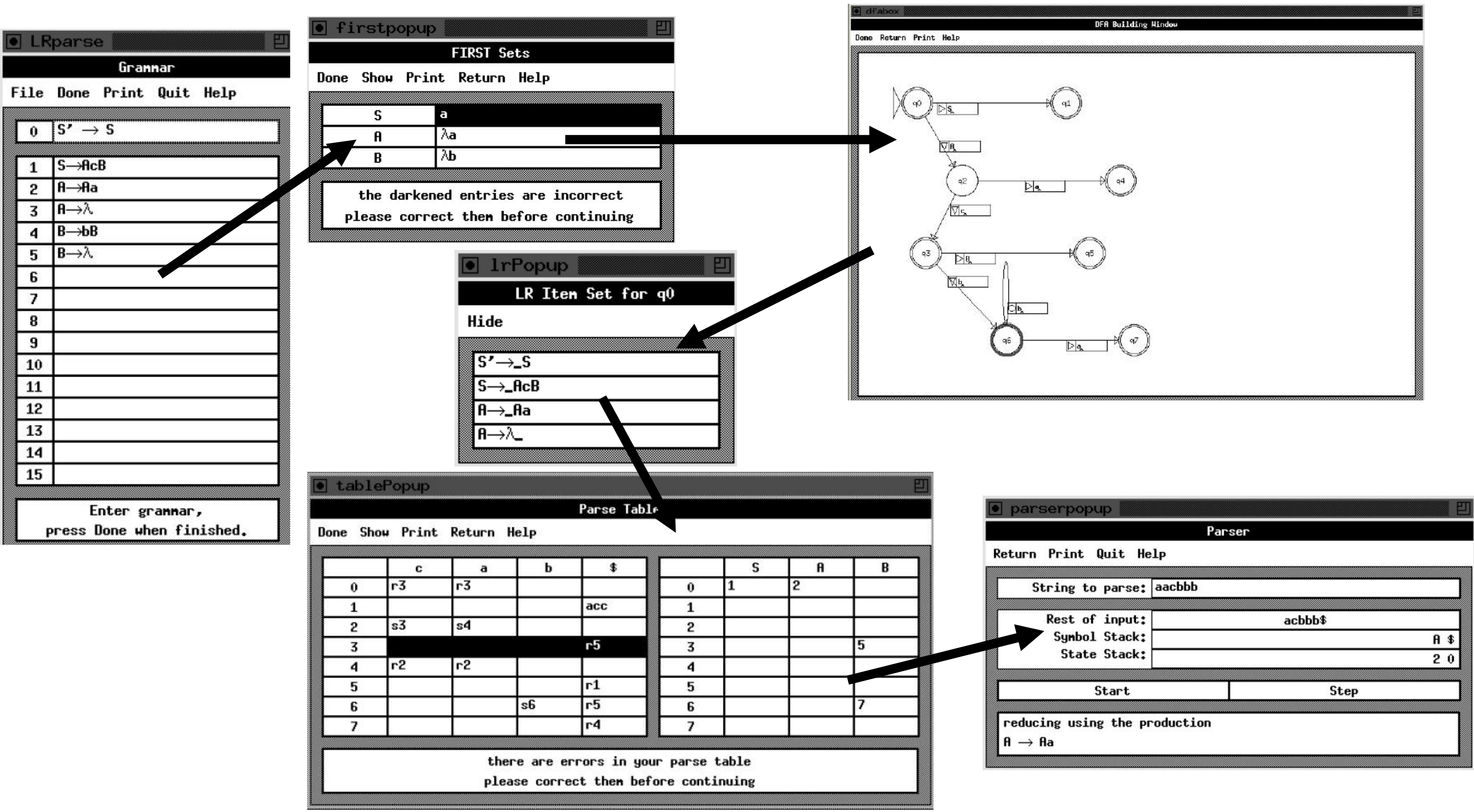


- Complete parts for them



Avoid Too Many Pop up windows

- OLD JFLAP LR PARSE TOOL

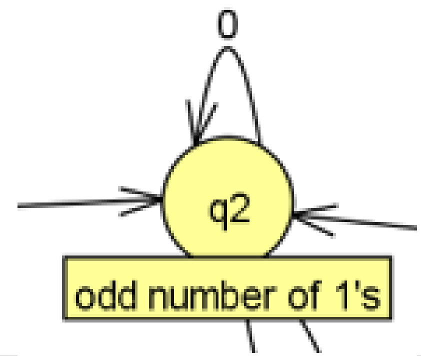


Add Pause/Checkpoint questions

- 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

Multiple Run

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

- General definitions
  - FA – recognize one or more symbols
  - NPDA – pop or push 0 or more symbols
- Batch processing

Naming your software

What is a “good” name for your tool?

Jawaa

- Algorithm Animation tool

BITS Goa Campus Internet Portal | Tech or treat! 40 geeky Hallowee... | Jawaa Commands

www.cs.duke.edu/csed/jawaa2/commands.html

Rectangle

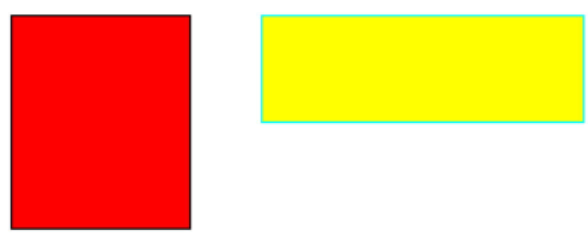
Parameters:

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

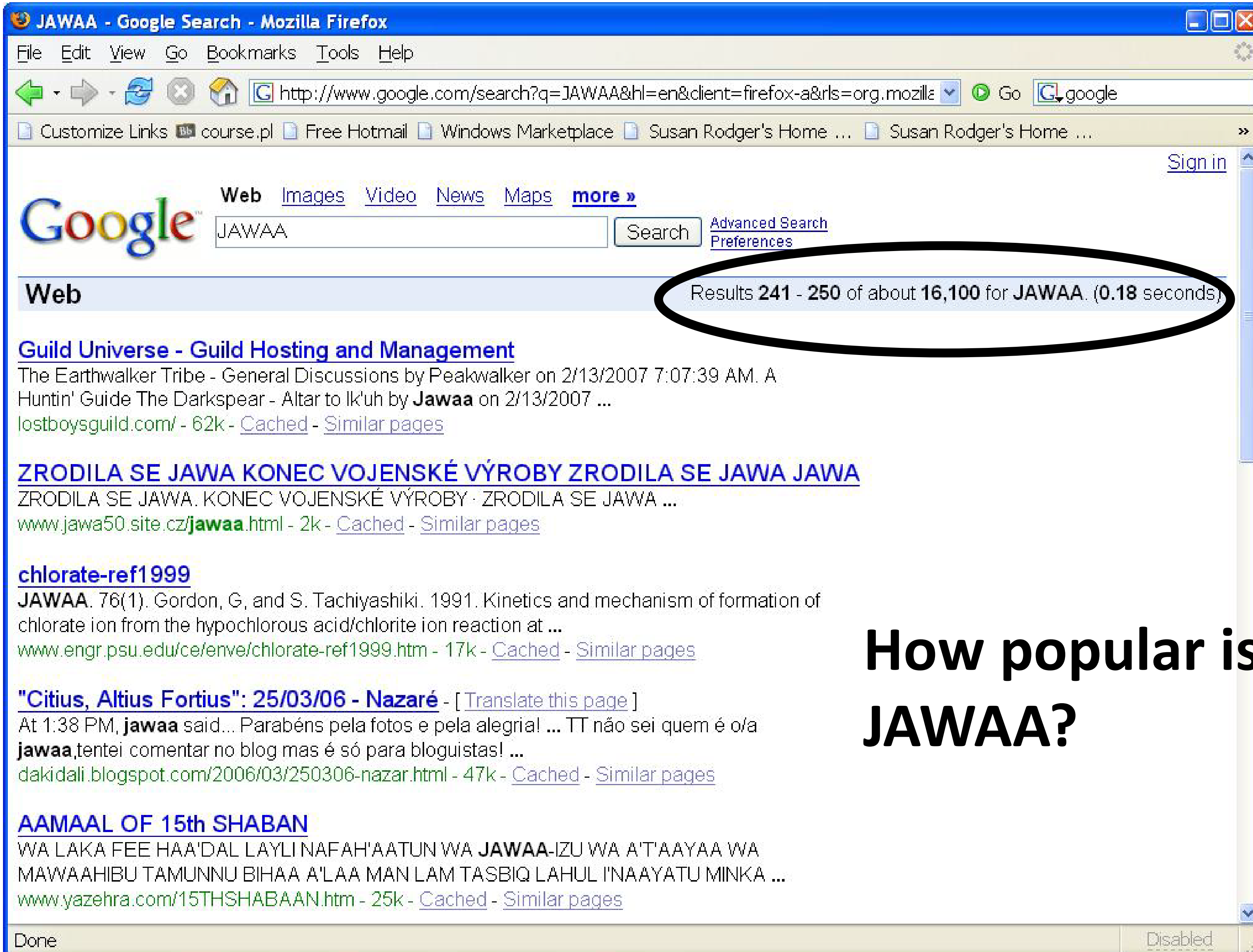
Example:

```
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



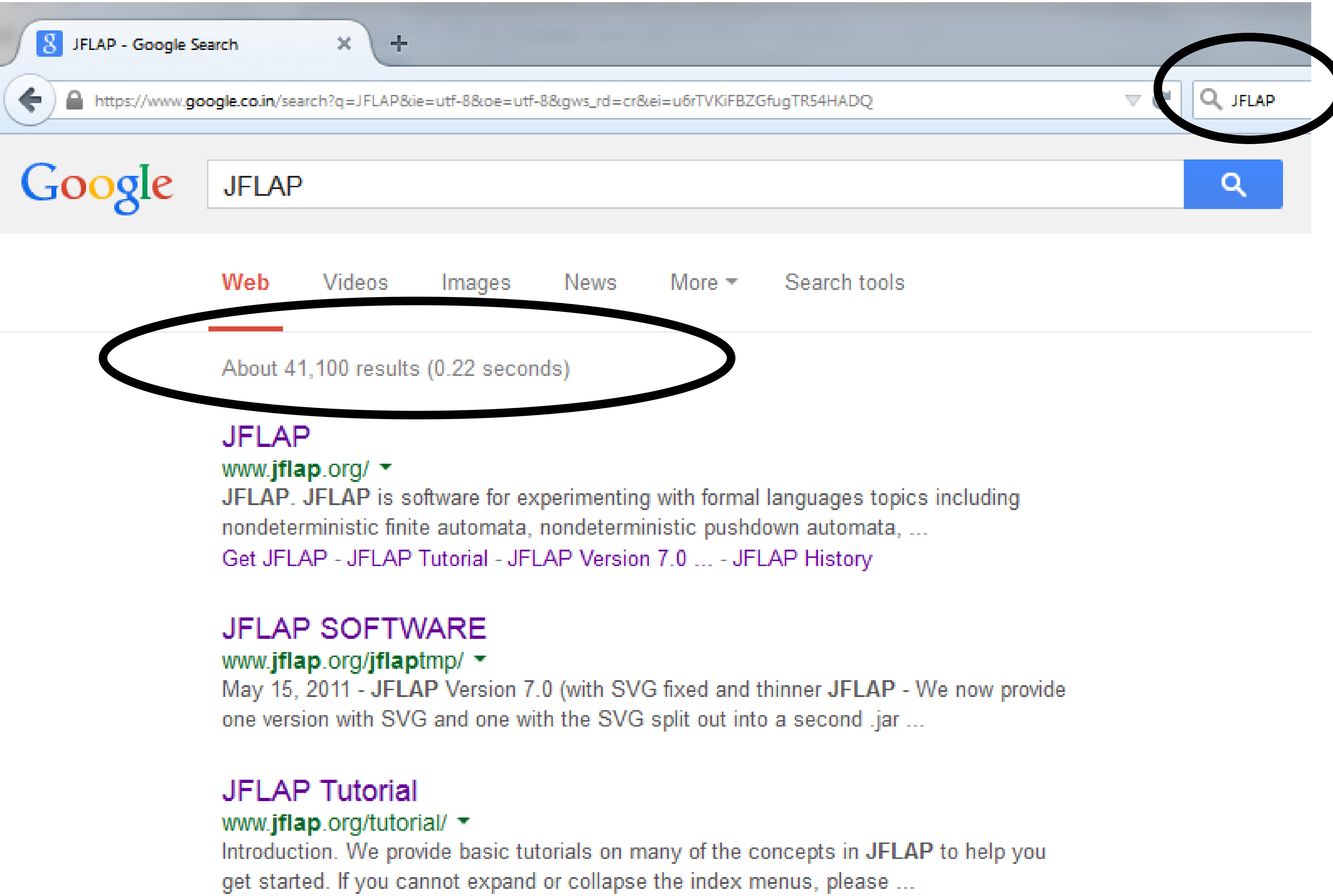
How popular is JAWAA?

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

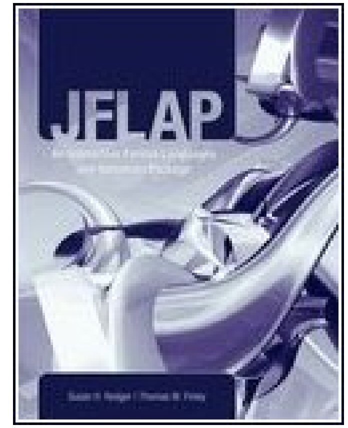
Newest 'jflap' Questions - Stack Overflow
stackoverflow.com/questions/tagged/**jflap**
We can use small letters for terminals and caps for Non-terminals in **JFLAP** while entering grammar. But this restricts to only 26 options. Can we have more ...

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 ...

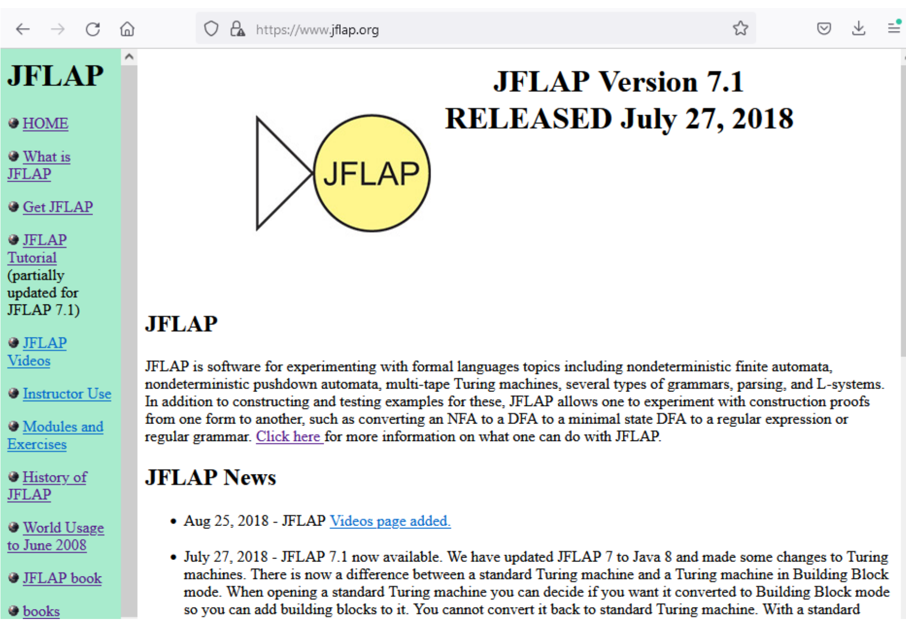
[PDF] **JFLAP Startup**
www.inf.unibz.it/~calvanese/teaching/10-11-fl/.../**JFLAP**-manual.pdf
Download **JFLAP** and the files referenced in this book from www . j flap. org to get started. **JFLAP** is written in Java to allow it to run on a range of platforms.

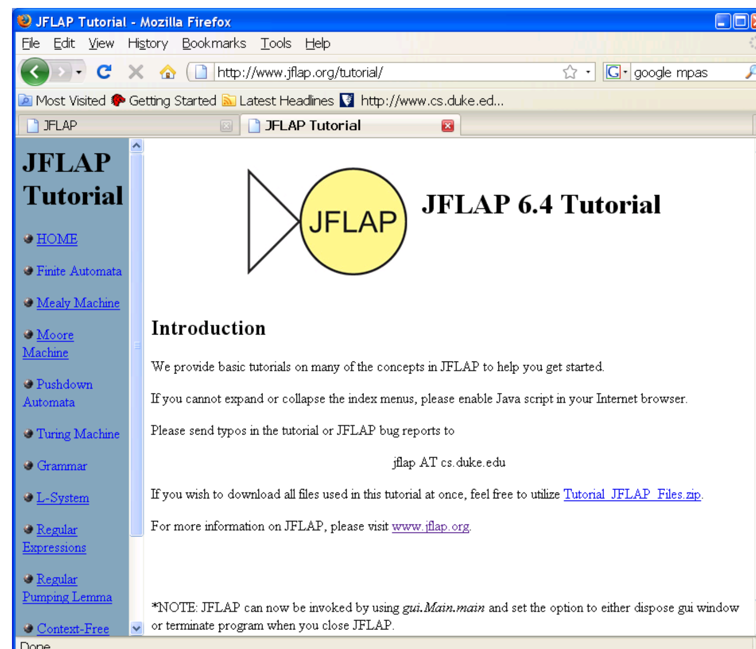
JFLAP is free

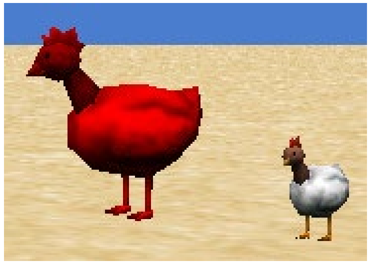


www.jflap.org

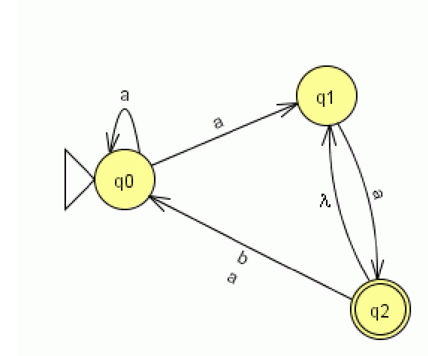
JFLAP tutorial







Outline



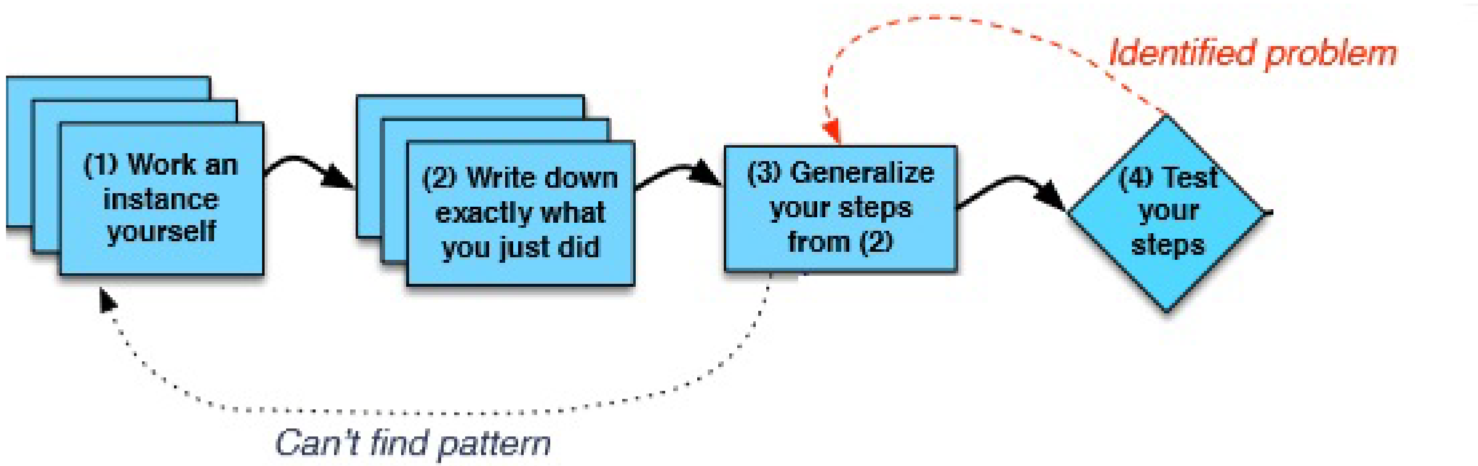
- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Solving Problems with Seven Steps
- Diversity Efforts

Stuck on solving a problem? Don’t know where to start?

- Use the 7 step process!

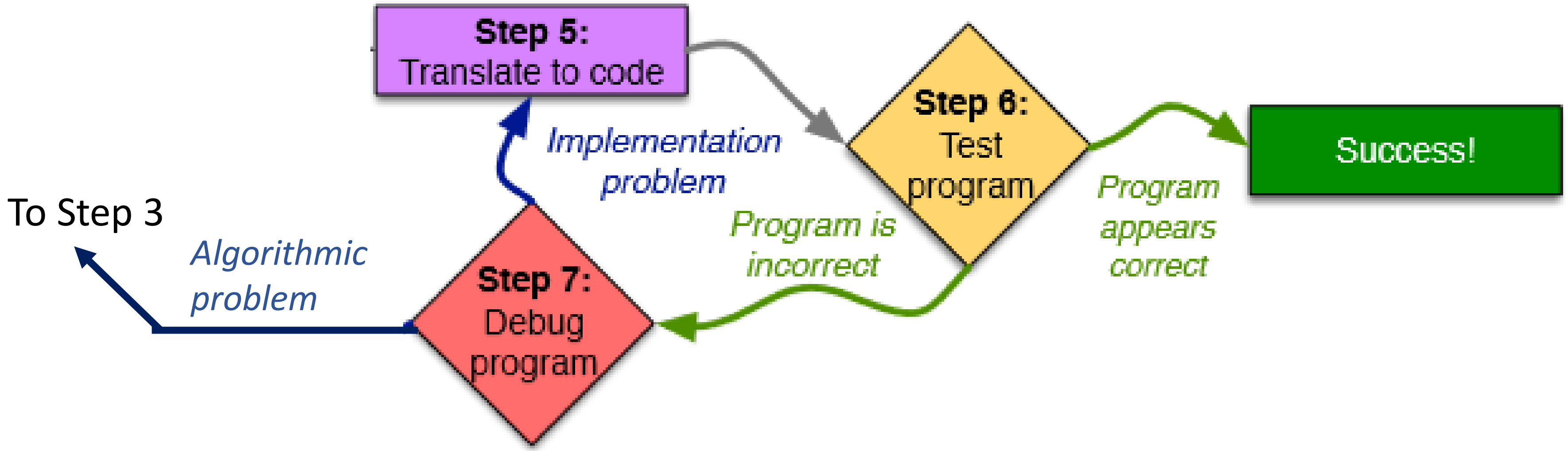
- CompEd 2019, Translation from Problem to Code in Seven Steps, Hilton, Lipp and Rodger

Problem Solving to Code – Steps 1-4



1. Work small examples by hand
2. Write down what you did in words (algorithm)
3. Find Patterns (generalize algorithm)
4. Work another example by hand (algorithm work? If not, go back to 3, or 1)

Problem Solving to Code – Steps 5-7



5. Translate to code
6. Test several cases
7. Debug failed test cases

Problem - TxMsg

Problem Statement

Strange abbreviations are often used to write text messages on uncomfortable mobile devices. One particular strategy for encoding texts composed of alphabetic characters and spaces is the following:

- Spaces are maintained, and each word is encoded individually. A word is a consecutive string of alphabetic characters.
- If the word is composed only of vowels, it is written exactly as in the original message.
- If the word has at least one consonant, write only the consonants that do not have another consonant immediately before them. Do not write any vowels.
- The letters considered vowels in these rules are 'a', 'e', 'i', 'o' and 'u'. All other letters are considered consonants.

For instance, "ps i love u" would be abbreviated as "p i lv u" while "please please me" would be abbreviated as "ps ps m". You will be given the original message in the string parameter `original`. Return a string with the message abbreviated using the described strategy.

Specification

```
filename: TxMsg.py

def getMessage(original):
    """
    return String that is 'textized' version
    of String parameter original
    """

    # you write code here
```

Examples

Examples

1. "text message"

   Returns "tx msg"

5. "aeiou bcdfghjklmnpqrstvwxyz"

   Returns: "aeiou b"

Focus on transforming one word Write helper function *transform*

- How?
- Use seven steps
- Work an example by hand

Transform word - Step 1: work small example by hand

- Word is “please”
- Letter is ‘p’, YES
- answer is “p”
- Letter is ‘l’, NO
- Letter is ‘e’, NO
- Letter is ‘a’, NO
- Letter is ‘s’, YES
- answer is “ps”
- Letter is ‘e’, NO

Step 2: Describe what you did

- Word is “please”, create an empty answer
- Letter is ‘p’, consonant, no letter before, YES
- Add ‘p’ to answer
- Letter is ‘l’, consonant, letter before “p”, NO
- Letter is ‘e’, vowel, letter before ‘l’, NO
- Letter is ‘a’, vowel, letter before ‘e’, NO
- Letter is ‘s’, consonant, letter before ‘a’, YES
- Add ‘s’ to answer
- Letter is ‘e’, vowel, letter before ‘s’, NO
- Answer is “ps”

Step 3: Find Pattern and generalize

Need to initialize letter before, pick “a”

answer is empty

for each letter in word

- If it is a **consonant**, and the **letter before** is a vowel, then add the letter to the answer
- This letter is now the letter before

return answer

Step 4 – Work another example

Use vowel not part of word

- Word is message
- Letter is ‘m’, before is ‘a’, add ‘m’ to answer
- Letter is ‘e’, before is ‘m’, NO
- Letter is ‘s’, before is ‘e’, add ‘s’ to answer
- Letter is ‘s’, before is ‘s’, NO
- Letter is ‘a’, before is ‘s’, NO
- Letter is ‘g’, before is ‘a’, add ‘g’ to answer
- Letter is ‘e’, before is ‘g’, NO
- Answer is “msg” WORKS!!

Step 5: Translate to Code

Letter before is “a” # start with a vowel

answer is empty

for each letter in word

Step 5: Translate to Code

Letter before is “a”   # start with a vowel

before = ‘a’

answer is empty

answer = [ ]   # or this could be an empty string

for each letter in word

for ch in word:

Step 5: Translate to Code (code)

```
    #If it is a consonant, and the letter before is
    a #vowel, then add the letter to the answer


    #This letter is now the letter before

# return answer
```

Step 5: Translate to Code (code)

```
#If it is a consonant, and the letter before is
a #vowel, then add the letter to the answer
if !(isVowel(ch)) and isVowel(before):
    answer += ch
#This letter is now the letter before
before = ch
```

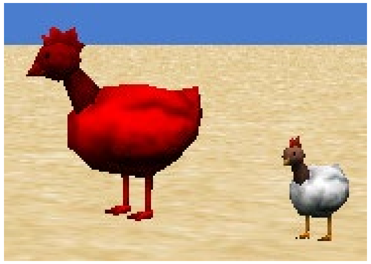
```
# return answer
return answer
```

Student Anecdotes

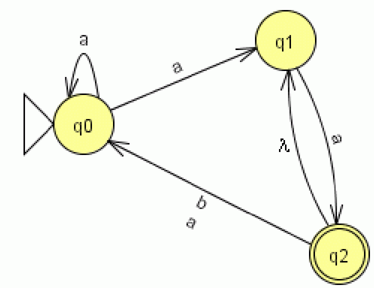
- From CompSci 101
  - “I just want to tell you that I tried the seven step method, and **I worked on all of my code** for one or two hours **before I even looked at the computer**. AND IT WORKED! I got all my code right on the first try! For the first time ever, I don’t have to go to the help lab …”

Student Anecdotes

- From Coursera course
  - “I have been programming for a couple of years. Learned from so many resources but **none said how to write the algorithm**, they just say you should write your algorithm first. The steps illustrated here are beautiful and definitely help to understand how to decompose a problem.“



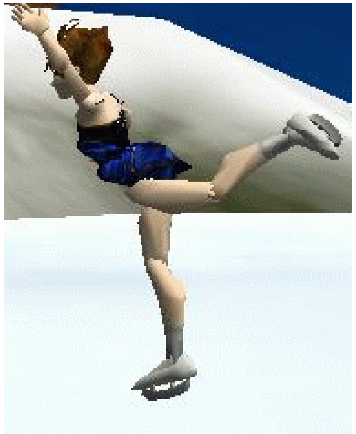
Outline



- Introduction
- CS Concepts Come Alive
  - Alice Programming Language
  - Algorithm Visualization
  - Automata Theory with JFLAP
  - Solving Problems with Seven Steps
- Diversity Efforts

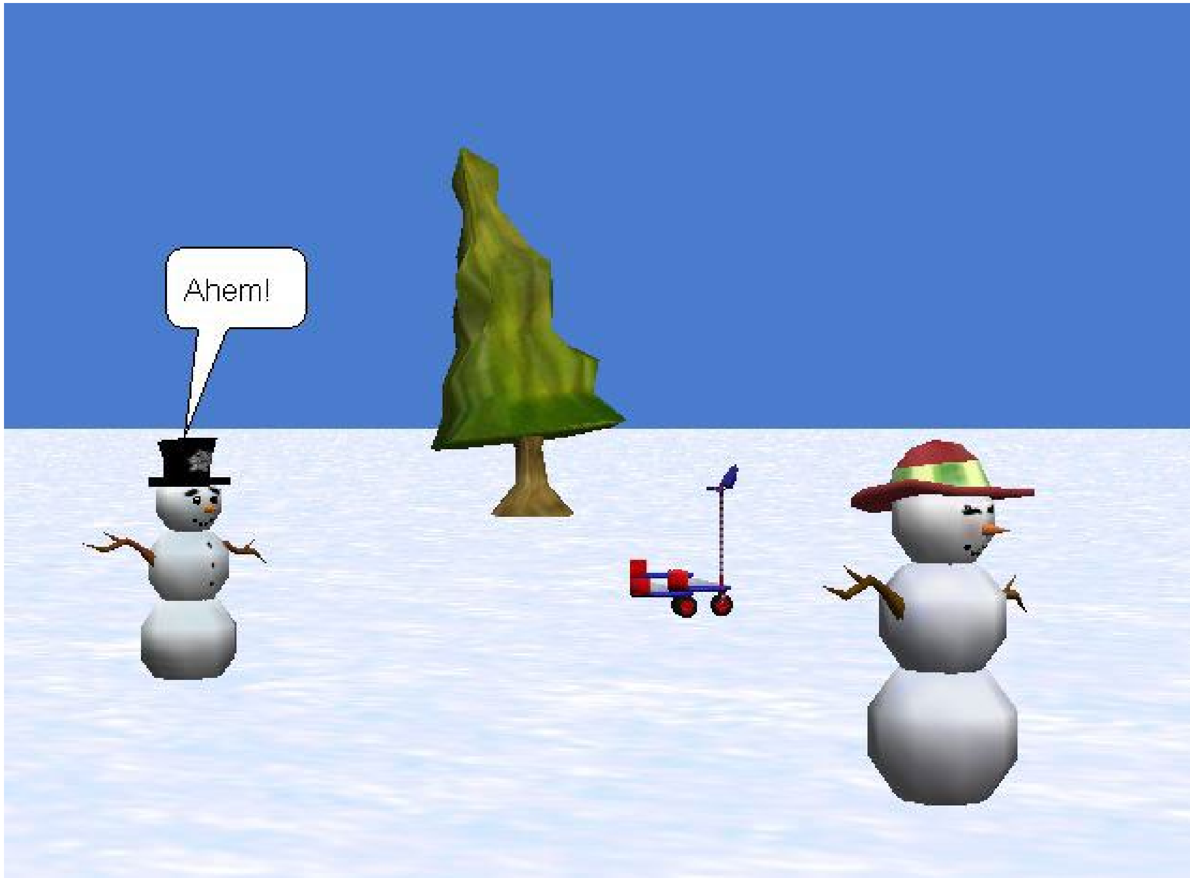
Success - Alice attracts diverse group

- 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



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 500 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 2008-2017

- Main Sites:
  - Duke University, Durham, NC
  - Charleston/Columbia, SC
  - San Jose, CA
  - Lincoln, Nebraska
- THANKS IBM and NSF

CRA-WP
Computing Research Association
Widening Participation

CRA-WP Board

- Organize Career Mentoring Workshops for Women and underrepresented groups
  - Early Career Workshop
    - Asst Prof, PhD students, PostDocs, Industry
  - Mid-Career Workshop
    - Assoc Prof, Industry Equiv
- Grad Cohort for Women
  - For Graduate students in first 3 years

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

Create account Log in

Wikipedia
The Free Encyclopedia

Article Talk Read Edit View history Search

- 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

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.

Three days later…

Create account Log i

Article Talk Read Edit View history Search

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

Frances E. Allen

From Wikipedia, the free encyclopedia

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]

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]

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)

Contents [hide]

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

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

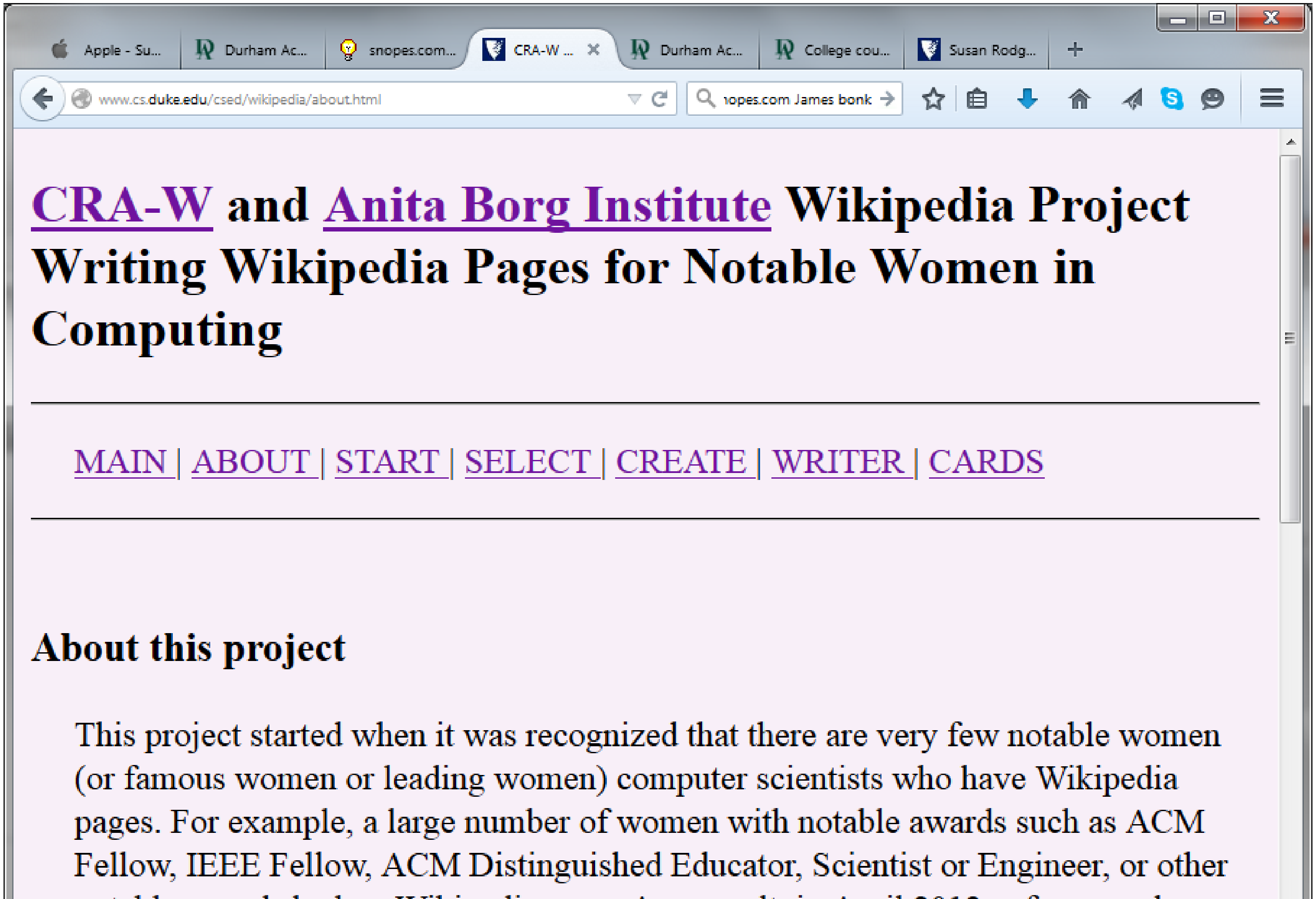
Write Wikipedia pages for Notable women in Computing

- 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
- Be careful!
  - 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 Guide on How to Write Wikipedia Biography www.cs.duke.edu/csed/wikipedia



CRA-W and Anita Borg Institute Wikipedia Project Writing Wikipedia Pages for Notable Women in Computing

MAIN | ABOUT | START | SELECT | CREATE | WRITER | CARDS

About this project

This project started when it was recognized that there are very few notable women (or famous women or leading women) computer scientists who have Wikipedia pages. For example, a large number of women with notable awards such as ACM Fellow, IEEE Fellow, ACM Distinguished Educator, Scientist or Engineer, or other

Our Database of Notable Women in CS

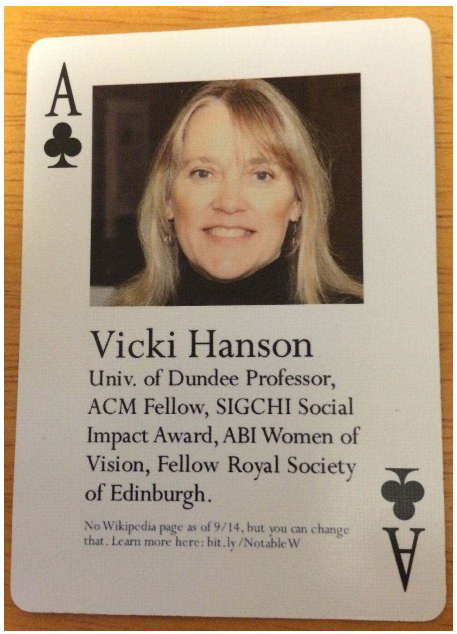
- Over 300 women
- Why notable
- Status of their Wikipedia page
- Forms for adding women and updating status

	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

To Share These Achievements….

- August 2014, with Katy Dickinson and Jessica Dickinson Goodman….
- Created Notable Women in Computing cards

Vicki Hanson
Had no Wikipedia page, now does

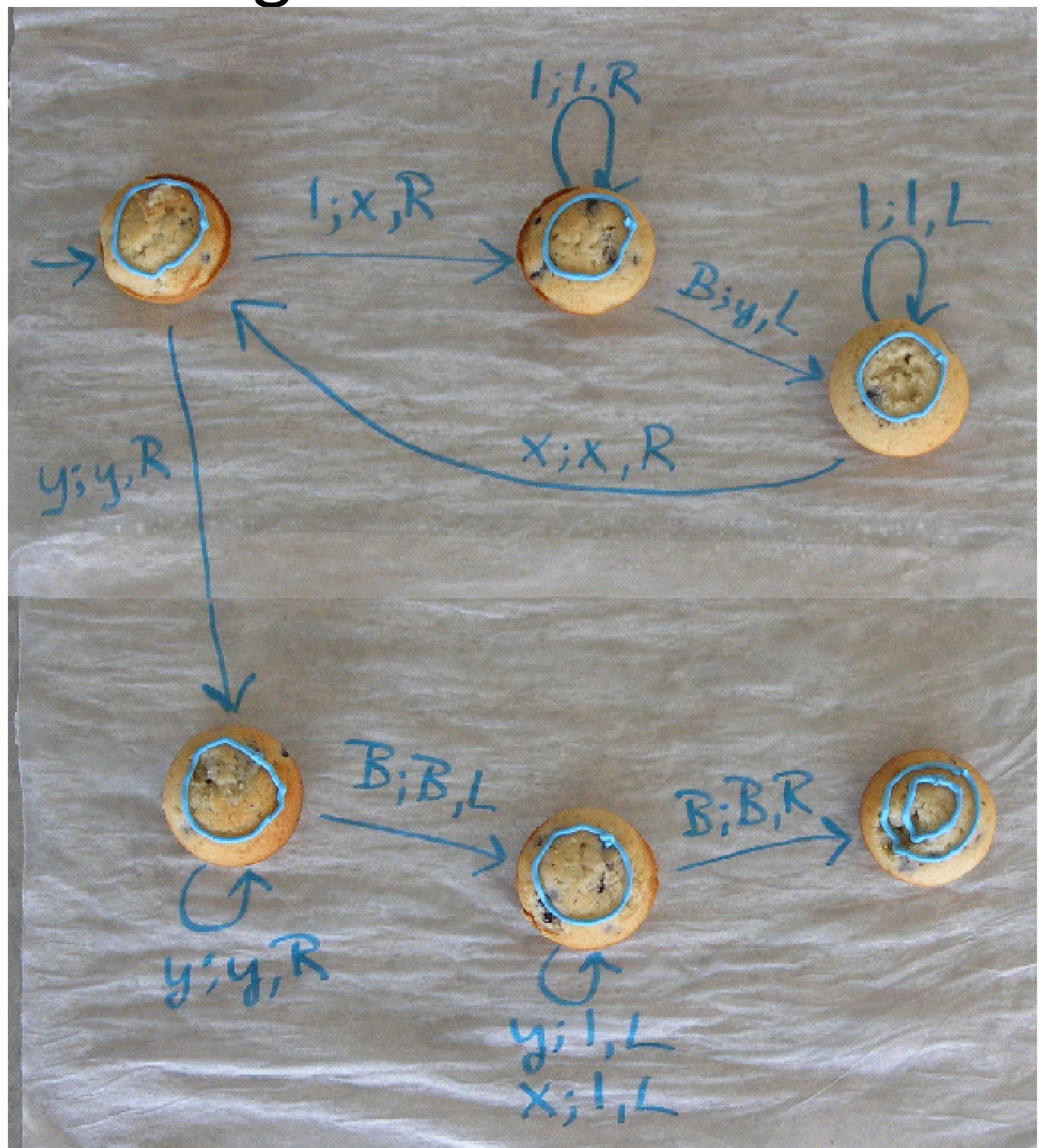


What happens when your hobby and your career collide?

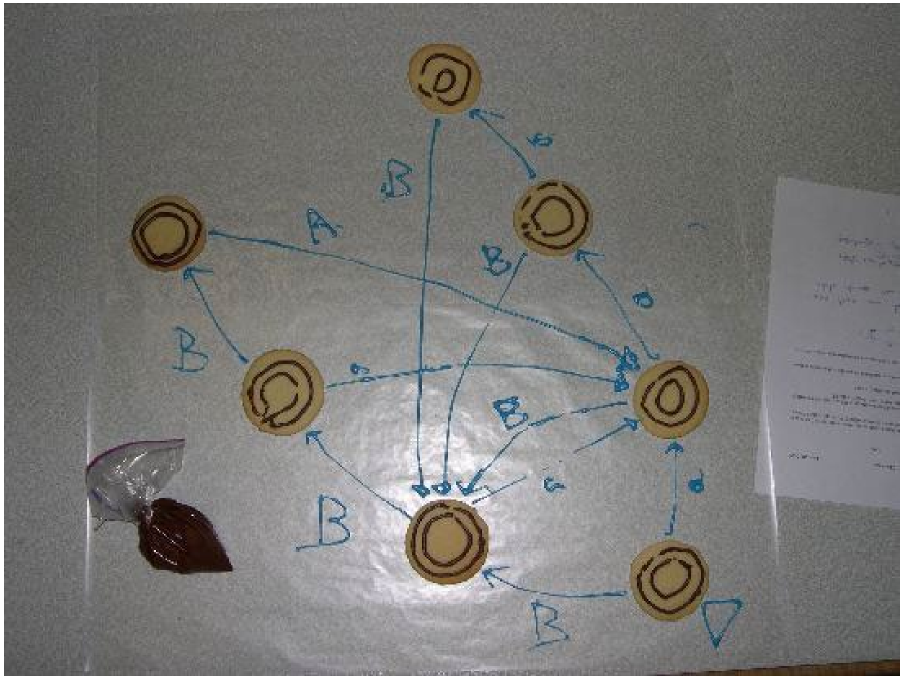
It is now time for engaging students with edible CS

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



CS 1 Sorting Cookies

Cookies for CS 1 - Python

CS 1 had around 300 students

Thank You

- Questions?