Computer Science Concepts Come Alive

Susan H. Rodger
Duke University
Feb. 6, 2015
ACM India
Goa, India

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

Thanks for inviting me!

• Tough choice

or

?

This is my classroom now

• Classroom rule:
  NO SITTING IN THE LAST FOUR ROWS

• Come forward

• Yes that is YOU who is sitting in the last four rows.

About me

Rode motorbikes in the past

Not sure I could do it here!!!
Graduate School

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

Became a professor - 1989

- New Assistant Professor at Rensselaer Polytechnic Institute

  - New Course: Combined automata theory with CS1 and CS 2 (data structures)
  - Student wanted feedback on all the answers in the book!

Back in 1990

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

- Tool – NPDA - to experiment with pushdown automata

CS Concepts Coming Alive- Back in 1989

- What data structure is this?
YARN, in the shape of a binary tree
Subtrees made with molecule kit
What is it?

2D-range tree
• Search in x-y plane
• Main tree organized by x-values
• Subtree organized by y values

1994 – Moved to Duke University
Professor of the Practice
• Position focuses on Education in the Discipline
  • Assistant Prof of the Practice
  • Associate Prof of the Practice
  • Professor of the Practice

Outline
• CS Concepts Come Alive with Software
  • Automata Theory with JFLAP
  • Learning Programming with Alice
  • Algorithm Animation
• Challenges in Designing Educational Software
• CS Concepts Come Alive in other ways
  • Manipulatives
  • Group Activities
  • Edible CS
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

Outline

- CS Concepts Come Alive with Software
  - Formal Languages and Automata Theory with JFLAP
  - Learning Programming with Alice
  - Algorithm Animation
- Challenges in Designing Educational Software
- CS Concepts Come Alive in other ways
  - Manipulatives
  - Group Activities
  - Edible CS

ITiCSE Working Group Report 2002 (Naps et al.)
Formal Languages and Automata Theory

- Traditionally taught
  - Pencil and paper exercises
  - No immediate feedback

- More mathematical than most CS courses
- Less hands-on than most CS courses
- Appears to be no programming? Unlike most other CS courses

Students Ready to learn Automata Theory!

Things start well enough ...

But soon, instead of pictures, there are WORDS.
Big words! The type with more than one syllable!

VIOLENCE AMONG STUDENTS AS NERVES FRAY!

We only wanted to learn automata theory! Isn’t there a better way?

Try JFLAP ...
Students Learning Automata with JFLAP

Why Develop Tools for Automata?

<table>
<thead>
<tr>
<th>Textual</th>
<th>( ({q_0, q_1, q_2}, {a, b}, \delta, q_0, {q_2}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta = { (q_0, b, q_1), (q_0, a, q_1), (q_1, a, q_2), (q_1, b, q_2), (q_2, a, q_1) } )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tabular</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>( q_0 )</td>
</tr>
<tr>
<td>( q_1 )</td>
</tr>
</tbody>
</table>

Visual

Interactive

Overview of JFLAP

- **Java Formal Languages and Automata Package**
- Instructional tool to learn concepts of Formal Languages and Automata Theory
- Topics:
  - Regular Languages
  - Context-Free Languages
  - Recursively Enumerable Languages
  - Lsystems
- With JFLAP your creations come to life!

Thanks to Students - Worked on JFLAP and Automata Theory Tools

- NPDA - 1990, C++, Dan Caugherty
- 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

Over 20 years!
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    Yes    No, ends    Yes
no. of 1’s    In 1
Simulation on 1101010

Accepts Input! 1101010

Add meaning to states!

only one 0

odd number of 1's
Example: proof
NFA to DFA

1. Start with NFA

2. Construct new DFA
   On q0 with an a, go to q0, q1 and q2

3. Final DFA
**DFA to Min DFA**

- Start with DFA

**DFA to Min DFA (2)**

- Start tree of distinguishable states
- Complete tree!

**DFA to Min DFA (3)**

- Determine states in min DFA
- Add arcs to complete it

**Another Example: Grammar**

- Grammar – set of replacement rules to define a language
- Examples:
  - Grammar for English
    - defines English sentences
  - Grammar for Python programming language
    - defines syntactically correct programs
  - Grammar for a formal language (simpler)
Grammar for $a^n b^n c^n$

- 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
  - aaaaabbbbcbbbbccaaaaabbbbbccccc...

Example Derivation for aabbcc

S $\rightarrow$ AX  
$\rightarrow$ aAbcX  
$\rightarrow$ aAbcX  
$\rightarrow$ aBbcX

Example Derivation for aabbcc

S $\rightarrow$ AX  
$\rightarrow$ aAbcX  
$\rightarrow$ aAbcX  
$\rightarrow$ aAbcX  
$\rightarrow$ aBbcX

NOTE: We have generated the correct symbols, aabcbbc, but they are in the wrong order!
Example Derivation for aabbcc

S → AX
→ aAbcX
→ aaBbcX
→ aabBbcX

rule: S → AX
rule: A → aAbc
rule: A → aBbc
rule: Bb → bB

Example Derivation for aabbcc

S → AX
→ aAbcX
→ aaBbcX
→ aabBbcX
→ aabDbcX
→ aabDcX
→ aabbDcX

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

Note: the D absorbed the c!
Example Derivation for aabbcc

S \rightarrow AX
S \rightarrow aAbcX
S \rightarrow aabBcX
S \rightarrow aabBcX
S \rightarrow aabbcX
S \rightarrow aabbcc

Example Derivation for aabbcc

S \rightarrow AX
A \rightarrow aAbc
A \rightarrow aBbc
Bb \rightarrow bB
Bc \rightarrow D
Db \rightarrow bD
Dx \rightarrow Exc

Eventually ... \rightarrow 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

Note the c spit out on right end!
Note all letters there, but wrong order: aabcbc

Absorb the “c”
Spit out the "c" at the right end
Absorb second “c”

Spit the “c” out at right end
Parsing in JFLAP

- Brute Force Parsing
  - Reg. Grammars, CFG, unrestricted grammars
- LL(1) Parsing
- SLR(1) Parsing
  - Application with
    - DFA
    - Pushdown Automata
  - Can parse grammars with conflicts!
- CYK Parsing

Example Parsing with SLR

- Ambiguous Grammar
- Will have conflicts in the parse table, but can still parse strings

1. Define FIRST and Follow sets
2. Build DFA
3. Define parse table

orange is conflict
Parse of aaba with reduce conflicts

- Parse entry highlighted
- Stack
- Rule used
- Parse tree

Recall the conflicts

- When click on orange entry, can choose a different entry to resolve conflict
- For both, let’s choose the shift instead of the reduce

Parse of aaba complete

Parse of aaba with shift conflicts

- Note tree is a different shape
Comparison Reduce vs Shift Conflicts

With Reduce Entrees

With Shift Entrees

Compare SLR(1) with NPDA

• Convert the CFG to an NPDA

\[ S \rightarrow SS \\
S \rightarrow a \\
S \rightarrow b \]

\[ \lambda, b ; S \\
\lambda, a ; S \\
\lambda, SS ; S \\
b, \lambda, b \\
a, \lambda, a; \]

\[ q_0 \] \[ q_1 \] \[ q_2 \]

\[ \lambda, S ; \lambda \\
\lambda, Z ; \lambda \]

Trace same string: aaba

• Note the nondeterminism
• Discuss how lookaheads in SLR(1) make it deterministic

Finish the trace: aaba

• 5 paths accepted
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 $\leftrightarrow$ regular expression
  - NFA $\leftrightarrow$ regular grammar
  - CFG $\leftrightarrow$ 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.
Add second T rule
L-Systems

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

Using JFLAP during Lecture

- Use JFLAP to build examples of automata or grammars
- Use JFLAP to demo proofs
- Load a JFLAP example and students work in pairs to determine what it does, or fix it if it is not correct.

JFLAP’s use Outside of Class

- Homework problems
  - Turn in JFLAP files
  - OR turn in on paper, check answers in JFLAP
- Recreate examples from class
- Work additional problems
  - Receive immediate feedback
- Talk more on using JFLAP with teaching in my talk tomorrow

Two-year JFLAP Study 2005-2007

Fourteen Faculty Adopter Participants

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

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

JFLAP 8.0 Beta– Where we are now!

- Formal definition
- Variables and terminals with multiple symbols
- Students: Genkins, McMahon, Godbey, Lin

JFLAP 8.0 Beta (cont)

- Language generator

JFLAP 8.0 Beta (cont)

- CYK animation
JFLAP is free

www.jflap.org

Outline

• CS Concepts Come Alive with Software
  – Automata Theory with JFLAP
  – Learning Programming with Alice
  – Algorithm Animation
• Challenges in Designing Educational Software
• CS Concepts Come Alive in other ways
  – Manipulatives
  – Group Activities
  – Edible CS

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

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

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

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

Game: Eragon

4 tasks to win the game

Example list
Example – Arrays
Shuffle, then Selection Sort

Outline

• CS Concepts Come Alive with Software
  • Automata Theory with JFLAP
  • Learning Programming with Alice
    • Algorithm Animation
• Challenges in Designing Educational Software
• CS Concepts Come Alive in other ways
  • Manipulatives
  • Group Activities
  • Edible CS

Algorithm Animation
Software/Aps/Videos

• AlgoViz.org – collection of algorithm visualizations
• 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
• 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

![Red Black Tree Animation](http://aleph0.clarku.edu/~achou/cs102/Examples/bst_animation/RedBlackTree-Example.html)

Student must have graduated. Link no longer works!

Another red-black tree animation

![Another Red-Black Tree Animation](http://www.ece.uc.edu/~franco/C321/html/RedBlack/redblack.html)

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)
Outline

• CS Concepts Come Alive with Software
  • Automata Theory with JFLAP
  • Learning Programming with Alice
  • Algorithm Animation

• Challenges in Designing Educational Software
• CS Concepts Come Alive in other ways
  • Manipulatives
  • Group Activities
  • Edible CS

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

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 usable?

- 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

Naming your software

What is a “good” name for your tool?

Jawaa

- Algorithm Animation tool
JAWAA name is not unique

How popular is JAWAA?

JFLAP name is unique

Much more than Google Analytics
Forums, Blogs, Course websites

FLAP

- Formal Languages and Automata Package
- 1996 – converted to Java
- FLAP -> JFLAP

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/Recent_posts

Watchlist Random page Recent changes Create blog post Recent posts Blog posts
Retrieved from “http://jflap.wikia.com/wiki/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 ...

JFLAP file.pdf
Download JFLAP and the files referenced in this book from www.jflap.org to get started. JFLAP is written in Java to allow it to run on a range of platforms.
Outline

• CS Concepts Come Alive with Software
  – Automata Theory with JFLAP
  – Learning Programming with Alice
  – Algorithm Animation

• Challenges in Designing Educational Software

• CS Concepts Come Alive in other ways
  – Manipulatives
  – Group Activities
  – Edible CS

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
Interaction in Class – Props
Linked List and Memory Heaps

ITICSE 98 – Astrachan – “Concrete Teaching: Hooks and Props as Instructional Technology”

Interaction in Class – Props
Memory Heap

Cards
- Insertion Sort
- Card Class – shuffling, dealing hands
- Poker hands – Full house, Flush, etc.
Royal Flush

Notable Women in Computing Cards
bit.ly/NotableW

- Based on Wikipedia project – wrote guide on how to write a Wikipedia page on a Notable women in Computing
- Picked 54 Women - deck of cards
- Page on using cards to teach CS
- Poster of the women

Outline

- CS Concepts Come Alive with Software
  - Automata Theory with JFLAP
  - Learning Programming with Alice
  - Algorithm Animation
- Challenges in Designing Educational Software
- CS Concepts Come Alive in other ways
  - Manipulatives
  - Group Activities
  - Edible CS
CS Unplugged

• Great activities for exploring computer science concepts – without a computer

Outline

• CS Concepts Come Alive with Software
  • Automata Theory with JFLAP
  • Learning Programming with Alice
  • Algorithm Animation
• Challenges in Designing Educational Software
• CS Concepts Come Alive in other ways
  • Manipulatives
  • Group Activities
  • Edible CS

Middle School students sorting themselves with Bubblesort

About Me - Hobby – Baking Shape cakes

The Wiggles magazine Issue No. 42
How do you make those cakes?

What happens when your hobby and your career collide?

It is now time for engaging students with edible CS

Automation 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 2 – Data Structures
Red-Black Tree (cookies)

CS 1
Sorting Cookies

Discrete math
A graph and its Dual Graph

Discrete Math
Stacking Pancakes Problem
Conclusions

- We have shown several ways for CS concepts to come alive
- Questions?