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....

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!

Finished Graduate School!

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

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

How do you make those cakes?

The Wiggles
magazine Issue No. 42
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 shape of a binary tree built with a molecule kit.

What is it?

2D-range tree

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

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

Alice Code is Easy to Learn
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
Outline

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

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

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
• Lots of animations and systems on the web!
• Lots of videos of algorithm animations on the web!
Lots of other software/programs for algorithm animation

- **Red Black Tree** – animation on web page

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

Student must have graduated. Link no longer works!
Python Tutor
Compute reverse of a list

Electronic Textbooks (ebooks) engage students
• OpenDSA (Shaffer, Virginia 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)

Run and edit code in the book
Integrates in Python Tutor

Questions for feedback

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

<table>
<thead>
<tr>
<th>Tabular</th>
<th><img src="image" alt="Tabular Example" /></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Visual</th>
<th><img src="image" alt="Visual Example" /></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Interactive</th>
<th><img src="image" alt="Interactive Example" /></th>
</tr>
</thead>
</table>

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!

Why Develop Tools for Automata?

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

Thanks to Students - Worked on JFLAP and Automata Theory Tools

Over 30 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

<table>
<thead>
<tr>
<th>No, odd</th>
<th>Yes</th>
<th>No, ends</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of 1’s</td>
<td></td>
<td>in 1</td>
<td></td>
</tr>
</tbody>
</table>

Build with JFLAP

Simulation on 1101010

Simulation on 1101010
Simulation on 1101010

Accepts Input! 1101010

Add meaning to states!

odd number of 1's
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 \(a^n b^n c^n\)

- 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:
aabc
aabbbccc
aaaabbbbcbbbbccccc
aaaaabbbbbbbbbccccc
...
Example Derivation for aabbcc

S → AX  
→ aAbcX  
→ aAbBcbcX

rule: S -> AX  
rule: A -> aAbc  
rule: A -> aBbc

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

S → AX
→ aAbcX
→ aaBbcbcX
→ aabBcbcX
→ aabDbcX

Note: the D absorbed the c!

Example Derivation for `aabbcc`

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

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”
Note all letters there, but wrong order: aabcbbc

What’s happening?
Bb → bB

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

Spit the “c” out at right end
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.
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

- 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

Allow user to proceed on if they got it

- Complete the rest for them
- Complete parts for them

Avoid Too Many Pop up windows

- OLD JFLAP LR PARSE TOOL
Add Pause/Checkpoint questions

• Allow for pause to think about what comes next
• Undo/go back
• Pop up a quiz question to see if the user understands what he/she just did
  • JHAVE tool does this
  • Can integrate into ebooks

What can make the tool more useable?

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

Naming your software

What is a “good” name for your tool?

Algorithm Animation tool
JAWAA name is not unique

How popular is JAWAA?

FLAP

• Formal Languages and Automata Package

• 1996 – converted to Java

• FLAP -> JFLAP

JFLAP name is unique

Much more than Google Analytics

Forums, Blogs, Course websites
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

Examples

1. "text message"
   Returns "tx msg"

5. "aeiou bcdfghjklmnoprstvwxyz"
   Returns: "aeiou b"

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.

Examples

1. "text message"
   Returns "tx msg"

5. "aeiou bcdfghjklmnoprstvwxyz"
   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

- 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!!

Use vowel not part of word
Step 5: Translate to Code

# Letter before is “a”       # start with a vowel

# answer is empty

# for each letter in word

```python
before = 'a'
answer = []          # or this could be an empty string

for ch in word:
    # If it is a consonant, and the letter before is a vowel, then add the letter to the answer
    if not isVowel(ch) and isVowel(before):
        answer += ch

    # This letter is now the letter before
    before = ch

# return answer
return answer
```

Step 5: Translate to Code (code)

```python
if not isVowel(ch) and isVowel(before):
    answer += ch

before = ch
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

- 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

- 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 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 notable 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.

Three days later...

Turing Award Announced and added to her page

In 1997, Ms. Allen was inducted into the WiTII 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.

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. 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.

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
- Respect for subject’s privacy
  - NOT A TABLOID!

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

Our Database of Notable Women in CS

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

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?