Introduction to Logical Reasoning with RESOLVE

Joseph E. Hollingsworth, IUS
Murali Sitaraman, Clemson

This research is funded in part by NSF grants CCF-1161916 and DUE-1022941.
Where Can These Ideas Be Taught?

- High school computing courses (in progress)
- Beginning programming courses (e.g., Ohio State intro sequence)
- Discrete structures and data structures courses (e.g., IUS)
- Programming language courses (e.g., Denison)
- Software development and software engineering courses (e.g., Clemson)
This Workshop

- High school computing courses (in progress)
- Beginning programming courses (e.g., Ohio State intro sequence)
- Software development and software engineering courses (e.g., Clemson)
Resources

- Google: Clemson RESOLVE
- [https://www.cs.clemson.edu/resolve/](https://www.cs.clemson.edu/resolve/)
  - Go to Tab Teaching
  - Go to Tab Research for publications
  - Go to Tab Web IDE for tools
Workshop Organization: Session I

- Collaborative, active learning pedagogy
- “Hands on” beginner reasoning
- Reasoning with objects
  - Mathematical modeling made simple
- Design by contract
  - Introductory reasoning activities
- Discussion
Workshop Organization: Session II

- Detailed specification example
- “Hands on” reasoning with objects
- Software engineering reasoning exercises and projects
- Discussion
Beginner Reasoning: Single Assignment Statement

- What will be the output for input 5?
- What will be the output for input -5?

Read(I);

I = I + 1; -- This is the assignment statement

Output(I);
Beginner Reasoning: Single Assignment Statement

- What will be the output for input 5?
- What will be the output for input -5?
- We use := for assignment to avoid confusion between equals in math and assignment statements

Read(I);
I := I + 1;  -- This is the assignment statement
Output(I);
Beginner Reasoning: Single Assignment Statement

- What will be the output for input 5?
- What will be the output for input -5?
- What will be the output for input max_int? [Not for beginner!]

Read(I);
I := I + 1;     -- This is the assignment statement
Output(I);
Beginner Logical Reasoning: Single Assignment Statement

What is the relationship between input and output?

Read(I);

**Remember**;  -- Remember the value of I as #I here

I := I + 1;

**Confirm** ???  -- How is I related to #I?

Output(I);
Beginner Logical Reasoning: Single Assignment Statement

What is the relationship between input and output?

Read(I);

**Remember;**  -- Remember the value of I as #I here

I := I + 1;

**Confirm** I = #I + 1;  -- Logical assertion, not assignment!

Output(I);
Beginner Logical Reasoning: Single Assignment Statement

Other acceptable answers

Read(I);

**Remember**; -- Remember the value of I as #I here

I := I + 1;

**Confirm** #I + 1 = I;

Output(I);
Beginner Logical Reasoning:
Single Assignment Statement

Other acceptable answers

Read(I);

Remember;  -- Remember the value of I as #I here
I := I + 1;

Confirm  #I = I - 1;

Output(I);
“Hands On” Beginner Logical Reasoning

- Google Clemson RESOLVE
- Bookmark: https://www.cs.clemson.edu/resolve/
- Go to Tab: Web IDE
- Click on Link: Begin to Reason
“Hands On” Basic Logical Reasoning

- Google Clemson RESOLVE
- Bookmark: https://www.cs.clemson.edu/resolve/
- Go to Tab: Web IDE
- Click on Link: Reason with Components
Basic Activities and Reasoning

- Click on Item Components to bring up Finder
- Select Programs in the Finder
- Select the first activity on reasoning
- Proceed with the activities in the order they appear to learn to reason about various constructs
Creating New Activities

- Sign in with an e-mail using the “sign in” link near top right
- Once you’re signed in, click on Item Components to bring up Finder
- Right Click on Programs in the Finder to create a new program
- Right Click on Concepts in the Finder to create a new (reusable) concept
Contract specifications – comparing informal specs with formal specs
- Mathematical modeling – abstraction
- Generating test data from specs
- Reasoning assistant tool
CPSC 3720 – Software Engineering at Clemson

- Usual Topics
  - Requirements analysis
  - Design and specification
  - Component-based implementation
  - Quality assurance

- Formal Reasoning
All Levels

- Collaborative Approach
  - Pairs or small groups
  - In class or homework
Collaborative Method

- Pairs or small groups
- With or without tools
- Each team presents their findings
- Collaboration both within teams and among teams
Selective Adaptation

- Pick and choose appropriate reasoning concepts and/or tools
- Faculty expertise
- Student background
Objectives

- Read formal specifications
  - Create test points from the specs
- Use component specifications to build larger systems
  - Work in teams
- Carry out formal verification of components
  - Use automated rules
Methods

- Collaborative learning
  - Teams of 2 to 4 members
  - Read specs
  - Implement specs
  - Verify implementations
  - Build larger systems.