Analyzing and Improving Visualizations of Formal Models

Avinash Palliyil[1]
Mentors: Yiliang Liang[2], Eunsuk Kang[2], Joshua Sunshine[2]
[1] Georgia Institute of Technology [2] Carnegie Mellon University

Background

Formal methods are powerful tools in validating system behavior, but their use of complex notations make understanding and debugging difficult.
Visualizations may help in these tasks.

Research Questions

- **How do practitioners use these visualizations and how are they helpful?**
- **What are the limitations of current tools and what do they desire?**

Interview Study

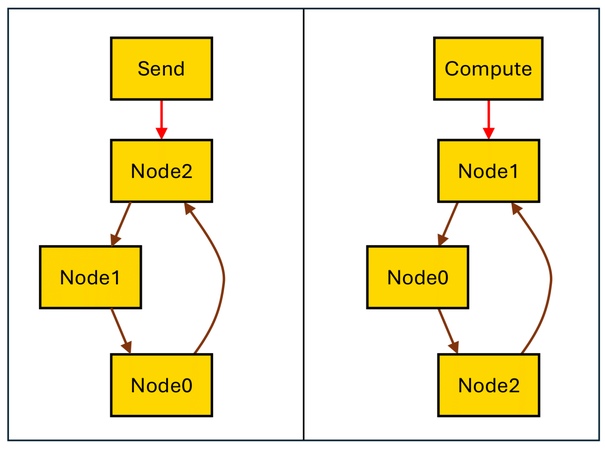
- Pre-screened interview study
- Semi-structured interviews with 15 users of formal modeling tools, including: Alloy (8), TLA+(3), BMethod (2), Spin (2), TSL Synthesis (2), P (1), Racq (1), Lean (1)

How are visualizations helpful?

- **Simplify understanding of models**
  - *"When the interactions between state machines are really complicated, … I go look for the visualization … it helps me keep track of what are the state machines that are running"* (P4)
- **Quickly validate and debug models**
  - *"Visualizations act really well as some sort of sanity check for our specifications … [and are] extremely helpful in finding sub-graphs that are closed loops … this tells you that maybe you need to go back and look at your specifications"* (P3)
- **Abstract away technical details to present to non-expert stakeholders**
  - *"There is a large audience of people who cannot understand Alloy text, … but these visualizations can make perfect sense to them."* (P2)

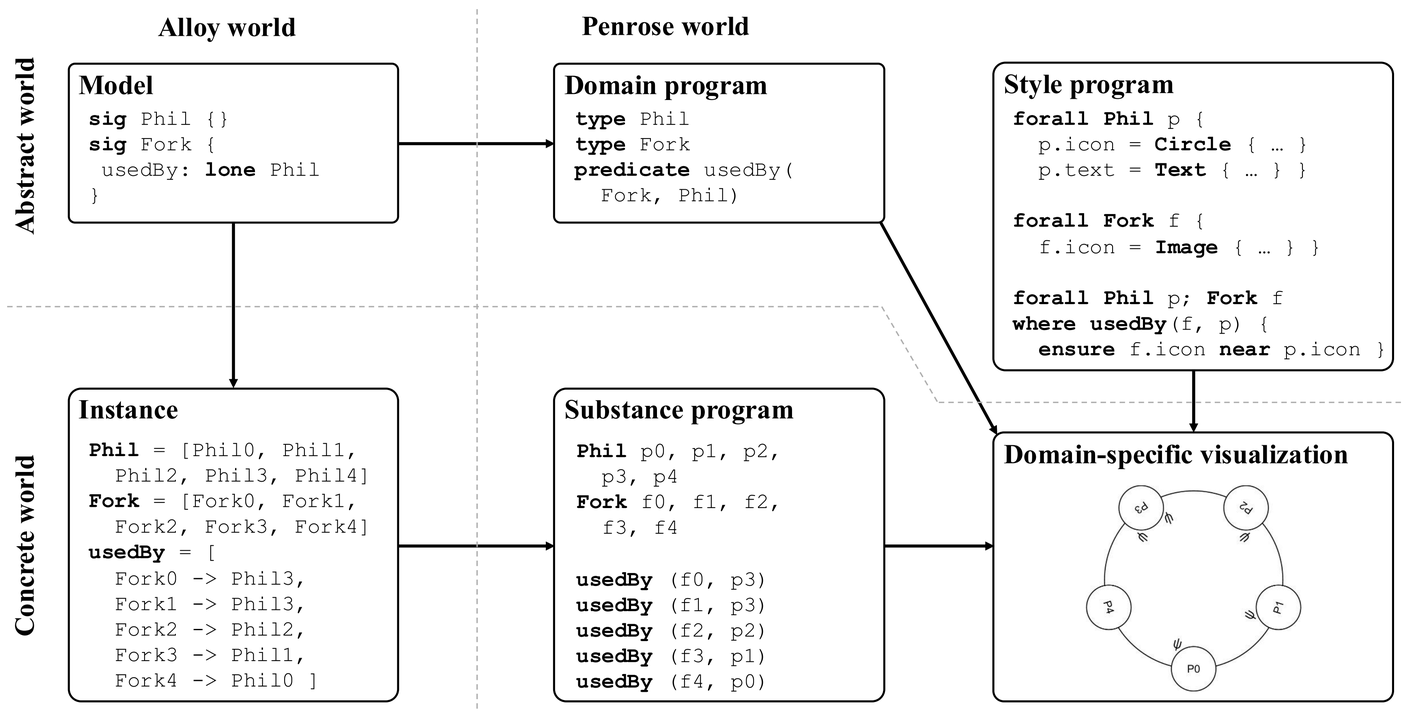
What are the limitations of current tools?

- **Lack of domain specificity make visualizations confusing.**
  - *"We understand how a train moves and how it visually looks, and none of this [built-in visualization] matches that. For exploratory tasks, it is challenging to use the visualizer."* (P7)
- **Lack of positional consistency of elements across states complicates bug finding.**
  - *"These graphs look isomorphic, the whole thing looks like identical structure. … But the node I was looking at before has moved down to the bottom…That's weird."* (P2)



An example of an Alloy visualization modeling the leader-election protocol that lacks both domain specificity and positional consistency across states.

Penlloy Visualization Tool for Domain-Specific Visualizations



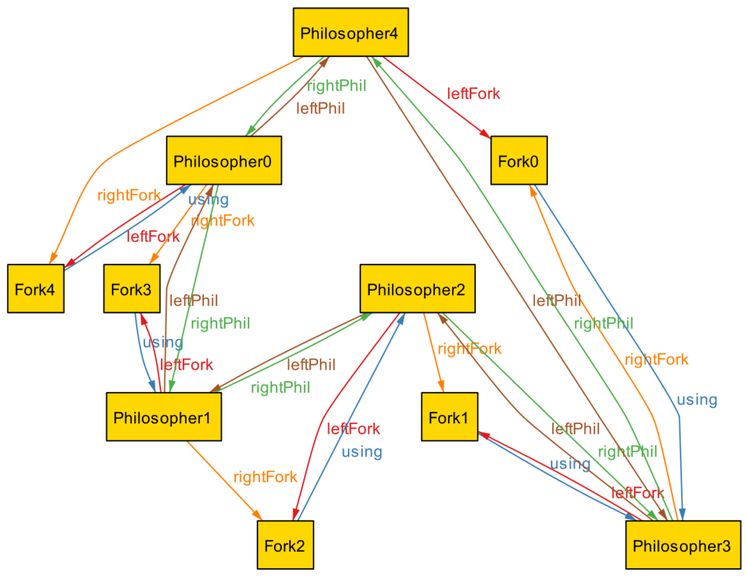
How Penlloy works, visualized with the Dining Philosopher's Problem

We explore domain specificity in visualizations with **Penlloy**, an **Alloy** visualizer using the **Penrose** engine.

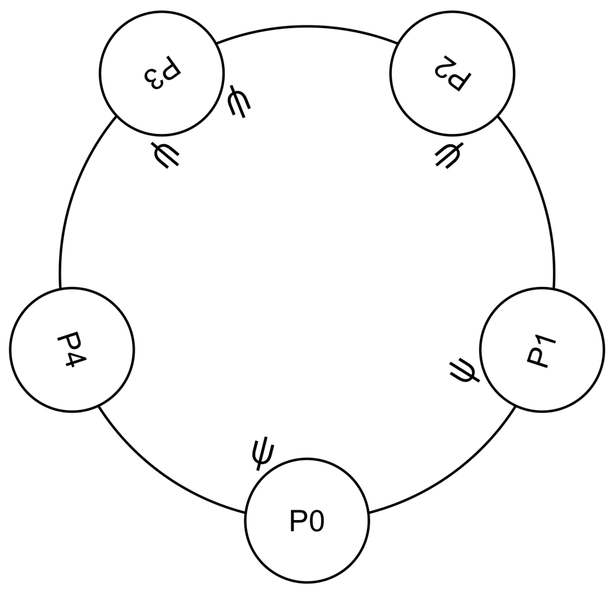
Why Alloy?
- Built-in visualizer to work off
- Widely used across many SE applications

Why Penrose?
- Versatility in creating complex visualizations
- Strength in reasoning about geometric relations between shapes

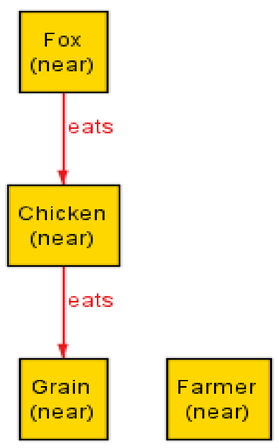
Alloy vs Penlloy Visualization Examples



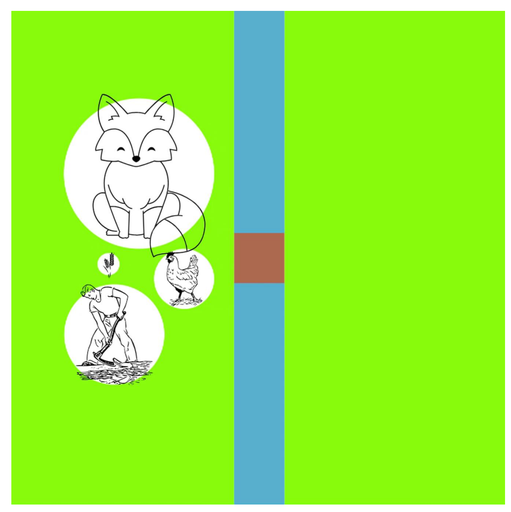
Default Alloy output – Dining Philosophers Problem



Penlloy-styled output - Dining Philosophers Problem



Default Alloy output – River Crossing Puzzle



Penlloy-styled output – River Crossing Puzzle

Current and Future Work

- **Addressing visual consistency in multi-state visualizations**
  - Encoding multiple notions of visual consistency as constraints and objectives within Penlloy.
- **Empirical evaluation of visual consistency**
  - Conducting a human-subject experiment to evaluate the effects of visual consistency in formal methods visualizations.
  - Experiment includes debugging tasks with consistent and inconsistent visualizations.
- **Automatic inference of domain-specific visualizations**
  - We want to generate constraints automatically from user provided diagrams.