Using Alloy in a Language Lab Approach to Introductory Discrete Mathematics

Charles Wallace

Michigan Technological University

In collaboration with Laura Brown, Adam Feltz

Supported by the National Science Foundation under grant DUE-1504860

Future of Alloy Workshop – Cambridge MA, 30 April 2018

Our “standard” Discrete Structures course

CS 2311 - Discrete Structures
Presents fundamental concepts in discrete structures that are used in computer science. Topics include sets, trees, graphs, functions, relations, recurrences, proof techniques, logic, combinatorics, and probability.
Credits: 3.0
Lec-Rec-Lab: (0-3-0)

Too many topics – not enough time

A break in the continuum: “all theory, no practice”

Lost in translation

- Behind every successful mongoose is a wombat.

$$\boxed{(\forall m)(\exists w) mongoose(m) \wedge successful(m) \Rightarrow wombat(w) \wedge behind(w,m)}$$

$$\boxed{(\forall mongoose) successful \Rightarrow behind(wombat, mongoose)}$$

$$\boxed{(\forall m)(\exists w) mongoose(m) \wedge successful(m) \wedge wombat(w) \wedge behind(w,m)}$$

$$\boxed{(\forall m)(\forall w) mongoose(m) \wedge successful(m) \Rightarrow wombat(w) \wedge behind(w,m)}$$

Lost in translation

- Without early, repeated feedback, these syntactic and semantic errors can persist.
- Traditional “pencil-and-paper” exercises typically delay feedback until it’s too late.
- Alloy provides an opportunity for just-in-time feedback on student input – just like in their programming courses.

Scaffolding Alloy

- Powerful tool – use with caution!
- Ease students into Alloy gradually:
  - Observing: Students run a complete program and interpret Analyzer output.
  - Tweaking: Students play with highly constrained parameters of a complete program.
  - Building: Students complete portions of an incomplete program.

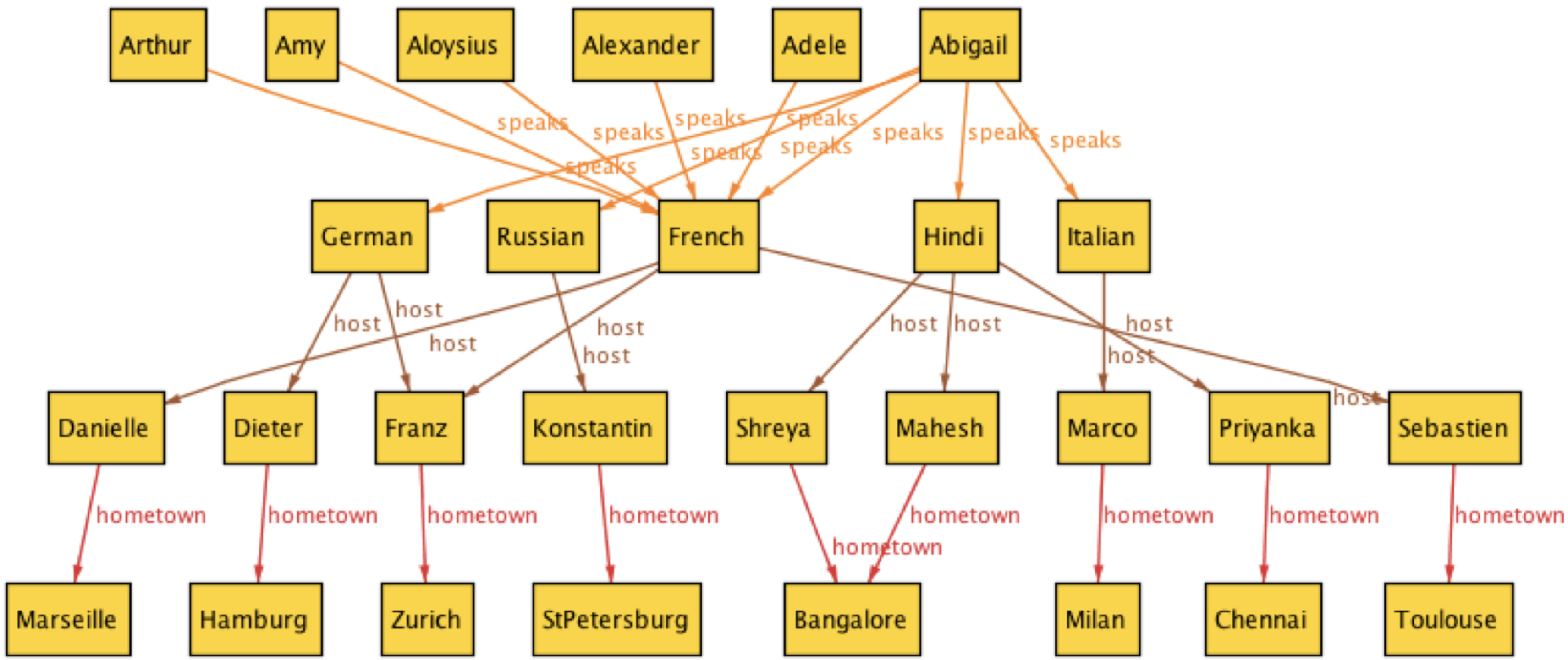
Declarative paradigm rocks their world

- Strategies for debugging, testing?
- Googling Stack Overflow = FAIL

More scaffolding

- Worked examples
- Lab assignments, with human support
- Alloy cheat sheet
- Guided inquiry process
- Start with single scenario, then generalize

Example: Study abroad



Test predicates, functions in single scenario; then generalize by pulling out constraints

Example: Hipster café lab

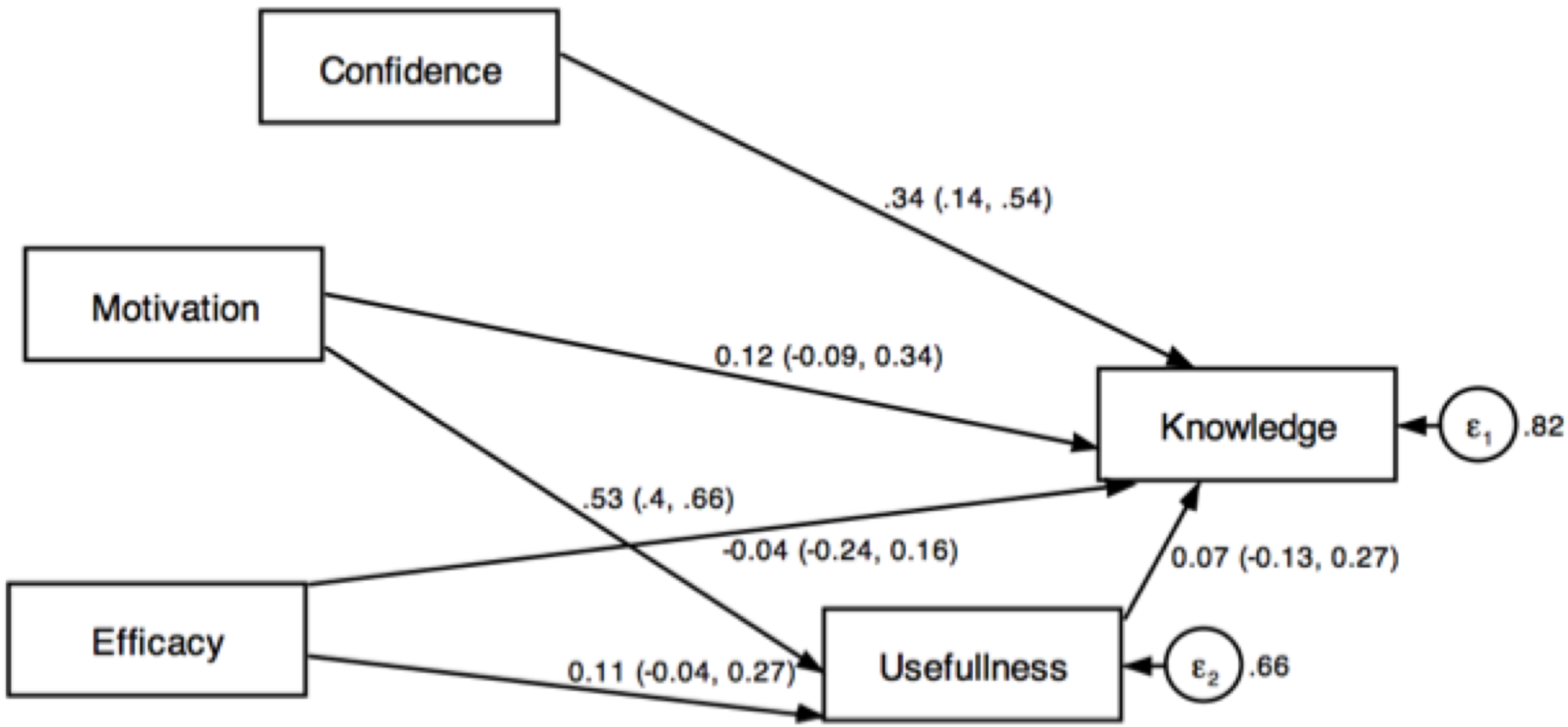
- Every cafe with a bus stop has busses arriving and busses departing; there are no "dead end" bus stops.
  - Let's take this one step by step. How would you express the set of cafes that are bus destinations? Hint: the composition operator . is your friend here: match against any cafe (the set Cafe) on the left hand side.
  - How about the set of cafes that have bus departures?
  - What has to be true about the two sets you've just defined (departures and destinations)?
- Let's assert that every cafe is reachable from every cafe within two "green travel" steps (walk-walk, walk-bus, bus-walk, bus-bus).
  - The pairs of cafes that have a one-step green path? That's easy: it's just greenTravel.
  - But how about two-step green paths? Hint: again, the composition operator . is your friend.
  - Zero-step green paths? I'll give that to you: it's the identity relation iden that relates each cafe to itself.
  - Now, how about within two green travel steps? That means zero, one, or two steps. Combine the sets you've defined above.
  - What do we want to say about the pairs of cafes with zero/one/two-step green paths; i.e. what has to be true of this collection of pairs? We have a predicate for that.

Guided inquiry steps point to a problem-solving process
(would be interesting to formulate a general process)

Preliminary results

- Course with Alloy labs significantly outperformed course without …
  - but some confounding factors: similarity of assignment/exam questions, student composition across courses
- (Cleaner longitudinal study underway)

How does it help?



Pathways to better performance:
via confidence, motivation, self-efficacy, perceived usefulness

Comments: (26/34)

The labs were a great experience. They ***forced you to sit down and try to solve the problems*** *with a partner while you were there to help. I would say that the labs were easily where I learned the most this semester.*

The labs were helpful to ***be able to visualize things****. I liked how in the labs* ***you didn't just give us the answers*** *if we didn't understand it. Instead you gave us hints to help us see the answers ourselves.*

The labs were helpful for understanding what we were learning about in class and ***applying topics to situations****.*

Having the labs (even though they were ***frustrating*** *for me), I think* ***helped me understand*** *many of the topics of discrete structures better.*

Comments: (5/34)

I'm very conflicted about the lab. I'm not sure if it really enforced my learning of the material. Although, the lab with the automata [JFLAP] was incredibly useful and a great learning booster, as it's the exact way you actually implement state-logic type of material. The rest of the labs were more of ``learn how to do the same problems in this **strange language**'' rather than ``this will bolster your knowledge of the material''.

Ideas for the future

- Block-based interface –
  avoid early syntax struggles
- Autograder – feedback for students as they work through problems
- Graphical scenario creator –
  encourage student testing
- Repository of worked problems
- Process(es) for design, debugging, testing

Publications

- L.C. Ureel. Discrete Mathematics for Computing Students: A Programming Oriented Approach with Alloy. FIE 2016.
- C. Wallace. Learning Discrete Structures Interactively with Alloy. SIGCSE 2018 Workshop.
- L.E. Brown, A. Feltz and C. Wallace. Lab Exercises for a Discrete Structures Course: Exploring Logic and Relational Algebra with Alloy. ITiCSE 2018.