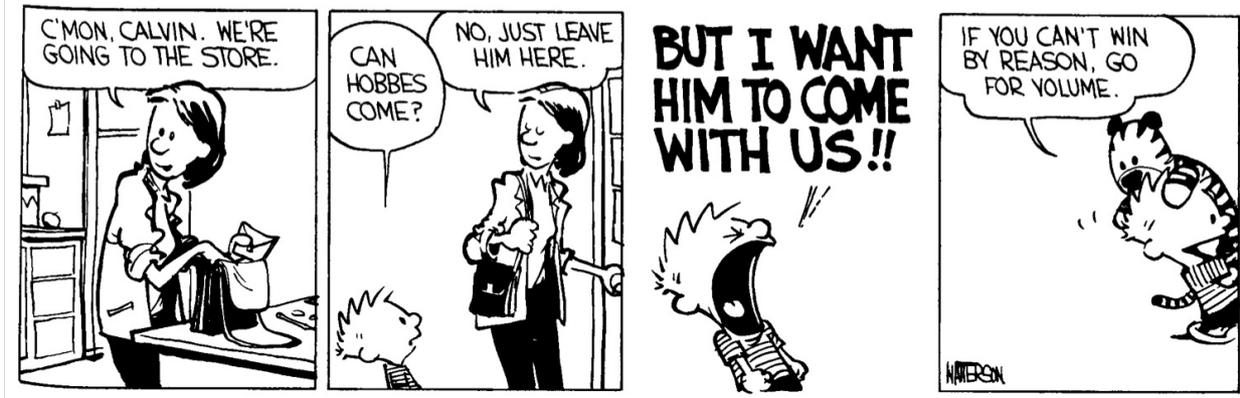


Meeting 18: Functions



Announcements

- HW5 due next Friday 11/3 -- start now!
- Project Pre-Proposal due this Friday
- HW4 COG re-opened until at Thursday 11:55pm
 - Note that COG points are not so significant.
 - Lots of folks working on improving HW4 after the deadline to get ready for the interviews and HW5.
 - But don't ignore HW5. HW5 has started!
 - Do you have lessons learned about how to approach the homeworks from HW4?

HW4 Discussion

Hours spent: mean 19 hours, stdev 9, median 16

Hardness: mean 5.4, stdev 0.6, median 6

Comments

- "Implementing polymorphism required many changes across all phases of the compiler, which were very difficult to test individually."
- "Edge cases are a terrible thing."
- "I didn't have enough time, I had so much other stuff this week (mandatory obligations, other homework assignments) that I only got like 2 days to work on it."
- "Keeping track of conversions between pyobj and untagged values was probably the most difficult. It might have been better if we had known we'd be switching between these so often so we could have put something in place to keep better track of them. I might

L-type checker

have also separated polymorphism and if statements into different assignments, they ended up being a lot of work implementing together."

- "While we didn't get everything done, we both worked hard and tried hard and that makes a huge difference."
- "Not knowing exactly what the requirements of the lab were. There were so many new pieces and concepts with this lab that it was hard to figure out what questions to ask to get clarification on. Also, not having explicit requirements on what was and wasn't supported, namely around type conversions/casting, made us miss some points on COG."
- "It was very time consuming. I did appreciate the ample warnings to start early, but I still felt a bit rushed at the end. I think I may have devoted too much time up front with carefully unit-testing each module. Switching to the suggested "horizontal testing" approach, whereby I focused on getting one simple program to run at a time, sped things up. Unfortunately, it also meant my unit tests quickly disintegrated and I didn't have time to fix them up. ... I really liked the challenge of this assignment and throwing weird P1 programs at my compiler, incrementally fixing the bugs. Unfortunately, I didn't get to this stage until the very last day -- I would have loved a few more days to find a comfortable rhythm."
- "a lot of time was spent understanding relation between meta and source languages, and how to use them effectively. another confusion was trying to determine if a given program is valid p1. it might be worthwhile to make output of reference compiler visible to the user so they can run different programs and check their validity."

Python —

Some Discussion on Testing

- Going from individual instances to sets of instances.
 - Checkers, interpreters for intermediate stages.
- Minimize ephemeral testing => persist tests.
 - Analogy: copy-and-paste coding => lift into an abstraction.

TDD

- input

→ expected

output = f(input)

if not expected == output

Today

- Midterm Questions?
- Explicate from HW4?
- Your Questions

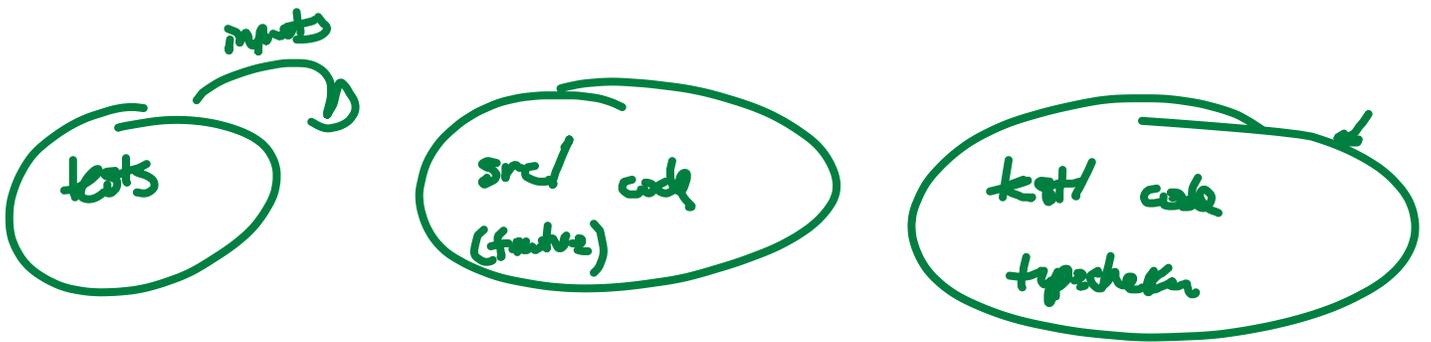
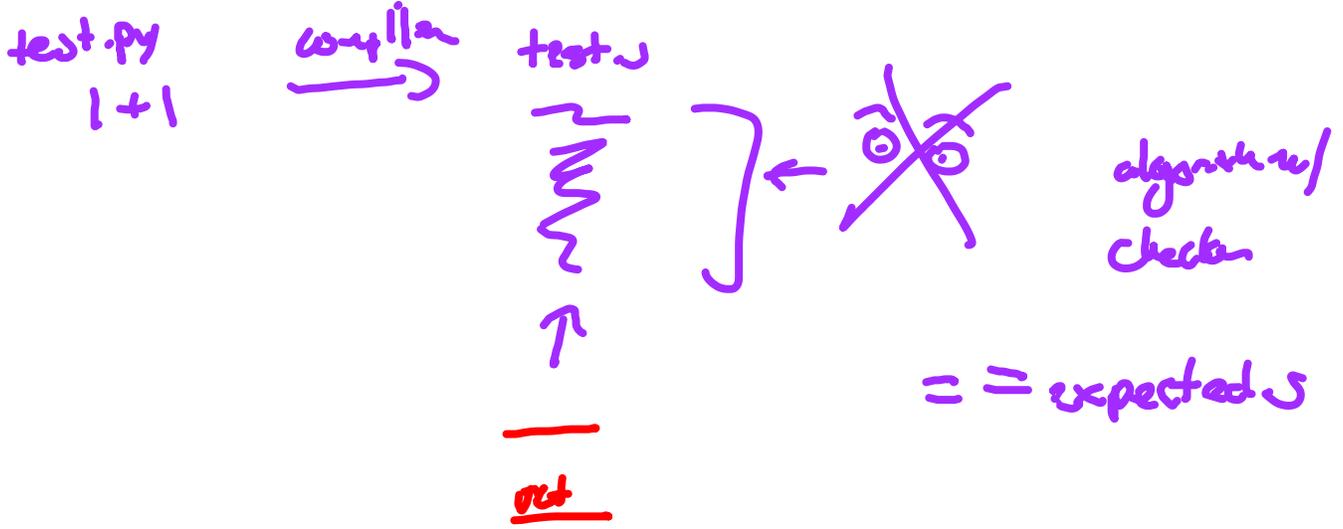
↳ assert ok(f(input))

↑
↑
typechecker

Thanks

you
😊
😊

✓



Questions

① ~~constant folding~~

Free variables

③ ~~explicitate~~

② ~~constant propagation~~ propagation

③ Building closures

④ runtime functions

