Module 12: Mutation

CPSC 110

Peyton Seigo

2018-11-28

Module 12: Mutation

Learning goals

- Explain the difference between non-mutable and mutable variables.
- Hand step the execution of functions operating on mutable variables.
- Design loops using ASL loop constructs similar to those found in languages like Java, Python, C, etc.
- Explain the interaction between using mutable state and using parallel computation to speed up large computations.

Motivation and Considerations

Mutable variables: Variables that can change their value after they are defined.

This mechanism is fundamental in almost any other language you program in.

- Despite being fundamental, it is surprisingly complex
- Overuse of it leads to programs that are not amenable to parallelization (running on multiple processors). Since multi-core computers are now common, the ability to use mutation only when needed is becoming more and more important
- Overuse of mutation can also make it difficult to understand programs, and difficult to test them well

Advanced Student Language (ASL) Differences

New expressions:

```
<expr> = (set! <var> <expr>)
         (begin <expr1> <expr2> ...)
```

Functions can now have 0 parameters.

Terminology

- x

Review: when did we write functions that accept anonymous functions as an argument?