Equational Reasoning

Pete Manolios
Northeastern

Logic and Computation, 2/13/2019

Context vs Theorems

- L1: $x=1 \Rightarrow 0=1$

  C1. $x=1$

  Proof:

```
   0
= { C1 }
   1
```

- L2. $0=1$

  Proof:

```
   0
= { L1|((x 1)), PL }
   1
```

- L3. φ (any conjecture)

  Proof: nil $\Rightarrow \varphi$, L2, MP, so φ

So, what went wrong?

You cannot instantiate context

You can only instantiate theorems!

Lessons Learned sum/fsum

- Algorithmic complexity is vitally important: consider big-data, Web
- Take algorithms as soon as possible
- As a computer scientist, *always* think about complexity
- But, correctness is most important: fast, but wrong is not good
  - Planes, trains and automobiles (not the movie) crash
  - Wrong simulation results for weather, nuclear testing, experiments…
  - Correctness is mostly what we care about in this class
- Powerful idea: define correctness using simplest definitions (the spec)
- Then define efficient implementation and prove equivalence
- Allows one to reason using the spec, but execute using efficient code

Comparison with C & Java

- Suppose that we write this code in an imperative language like C or Java
- Let's see a DEMO
- What happened?

Limited Precision!

- C, Java, etc. do not have arbitrary precision arithmetic
- So sum, fsum are not equivalent!
- We get a negative number because most languages use fixed-bit arithmetic

Finding Bugs

- You could have tested your program 1K times and not found errors
- We knew what we were looking for and so we found an error
- Is this a problem in practice? Yes. See http://googleresearch.blogspot.no/2006/06/extra-extra-read-all-about-it-nearly.html

Fixing Bugs

- How do we fix the bug?
- What is the bug?
- What is the specification?
- Is the spec is that fsum should be equal to sum?
  - Then don’t overflow when performing intermediate computations
- If the spec is that fsum should return the right value?
  - Then you have to use arbitrary precision arithmetic

Reasoning About C/Java

- Can we reason about C/Java code?
- We don’t have a theorem prover for these languages
- But, we can reason about them!
- Use ACL2s to model arithmetic in C/Java
    - Let’s say that the spec is that fsum should be equal to sum
    - We can use property-based testing
    - DEMO