Logic for Computer Science

16 – Wrapping up

Paige Randall North
(based on Wouter Swierstra's slides)

University of Utrecht

Review

- Proving with Hoare logic
- Recap of material covered
- Retrospective
- Looking ahead – logic in the rest of your degree
- Answering any questions you might have

Exam – practical matters

- **Exam:** 27 January at 13:30 in Educatorium Gamma.
- There is a 'low-stimulus room' at the same time. This is **only for students who are registered for it**.
- The exam will cover all the material from Chapter 8 onwards, including any dependencies on the previous chapters (such as propositional logic);
- This includes material on natural deduction, operational semantics and Hoare logic in the lecture notes.
- Material that we did not encounter in the second half will not be on the exam – such as circuit diagrams or boolean algebra, for instance.
- There is *no* lecture on Thursday; there is an exercise session – please ask any last questions you might have there!

Hoare logic & operational semantics

“I already know how to program, what is the point of all this?”

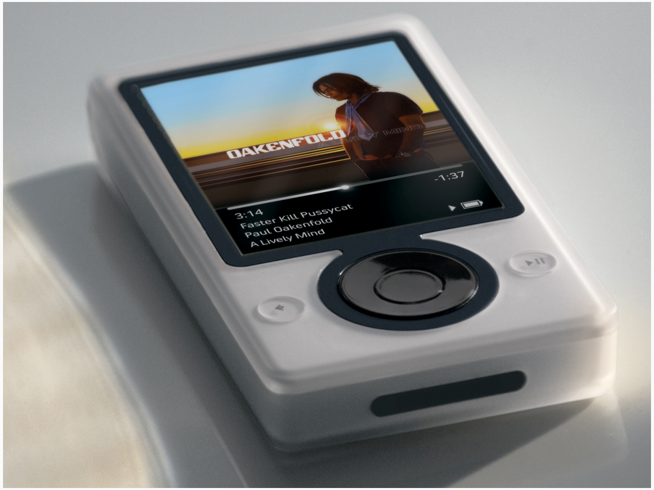

Figure 1: What is this thing?

The Zune bug

On January 1st 2009, every Zune player in the world froze.

No matter what users did, nothing brought them back to life.

The very next day, everything worked as normal.

What went wrong?

The problematic code

```
year = ORIGINYEAR; /* = 1980 */

while (days > 365) {
    if (IsLeapYear(year)) {
        if (days > 366){
            days -= 366;
            year += 1;
        }
    }
    else {
        days -= 365;
        year += 1;
    }
}
```

Dafny

There are tools to check for such bugs.

- Dafny is a ‘verification aware programming language’;
- We can decorate each of our methods with pre- and postconditions.
- These are checked *before a program is ever tested or executed*.
- Dafny itself can be compiled to C#, Java, JavaScript, Go and Python.
- It works seemlessly as a plugin for Visual Code (or emacs).

See

- `https://dafny.org`
- Verifying Euclid’s GCD algorithm in Dafny: `https://leino.science/papers/krml279.html`

Hoare logic

Verifying a program via Hoare logic means:

- you have a specification (a pair of a precondition and postcondition)
- you prove the program meets that specification

Dafny:

- gives you a formal language in which to write the specification
- helps you prove that the program meets that specification and checks the proof

But you have to do two things: programming and proving.

Correct-by-construction code

Next level: roll programming and proving together

- These programming languages are known as *dependent type theories*
- Include Coq, Agda, Lean

In these:

- The user writes the specification for the program
- Write the program
- The compiler checks that your program meets the specification
- Programming Euclid's GCD algorithm in Coq: `https://www.cs.princeton.edu/courses/archive/fall07/cos595/stdlib/html/Coq.ZArith.Znumtheory.html`

Recap

- Inductively defined sets and functions
- Proofs by induction
- Games
- Inductively defined relations by means of inference rules
- Natural deduction
- Operational semantics
- Hoare logic

Inductive definitions

- Inductively defined sets of natural numbers
- Inductively defined sets of words, trees, lists, programs, ...
- BNF notation
- Inductively defined functions
- Recursively defined functions

Inductive definitions – example questions

- Explain why some element – say 5 – is in some inductively defined set (or not);
- Give an inductive definition of some subset of the natural numbers;
- Or give an inductive definition of a sequence of numbers – such as the Fibonacci numbers;
- Execute a given inductively defined function;
- Define a function using induction/recursion.

Proofs by induction

- Proofs by induction over the natural numbers;
- Strong induction;
- Structural induction over trees, words, lists, etc.

Proofs by induction

- Proofs by induction over the natural numbers;
- Strong induction;
- Structural induction over trees, words, lists, etc.

Example questions

Prove X by induction on Y (for some suitable choice of X and Y).

Games

- Games of no chance;
- Concepts such as winning position and strategy;
- Examples of games and how to ‘solve’ them.

Games

- Games of no chance;
- Concepts such as winning position and strategy;
- Examples of games and how to ‘solve’ them.

Example questions

- Given some game, explain why player X has a winning strategy;
- Draw a game tree for some game X;
- . . .

Natural deduction

- Proof rules for propositional logic;
- How these are used to give a derivation;
- How *assumptions* are used in such derivations;
- Which rules add/remove assumptions from the context?
- The concepts of *soundness* and *completeness*.

Natural deduction

- Proof rules for propositional logic;
- How these are used to give a derivation;
- How *assumptions* are used in such derivations;
- Which rules add/remove assumptions from the context?
- The concepts of *soundness* and *completeness*.

Example questions

- Give the elimination/introduction rule for a certain logical operator;
- Give a natural deduction proof of the formula X from the assumptions Y;
- Identify what is wrong in a given proof.

Operational semantics and Hoare logic

- Operational semantics – showing how a program is executed;
- Hoare triples – specifying how a program behaves;
- Inference rules for Hoare logic – that can be used to prove a program correct;
- Soundness and completeness results of the Hoare logic with respect to the operational semantics.

Example questions

- Show how to execute a given program using the operational semantics we have seen;
- Identify/reproduce the rules for this semantics;
- Give a simple proof using Hoare logic;
- Understand the meaning of soundness and completeness;

Looking ahead. . .

Is this the last time you'll ever encounter logic?

- Databases
- Datastructures
- Concepts of programming languages
- Program semantics and verification
- Talen en compilers
- Functional programming
- Intelligent systems
- Logic and language/Logic and computation
- Discrete mathematics
- . . . .

Almost every CS course you will do in the rest of your academic career, you will probably need to do some (in)formal reasoning – for which you'll need logic.

Good luck on your exam!

Good luck on your exam!

And don't forget to fill out the Caracal evaluation ;)