Introduction to Specifications and Correctness

Musa Al-hassy

McMaster University
alhassm@mcmaster.ca

January 4, 2017

Who's Who

Instructor
Musa

TA
Curtis

Times and Places

- Classes in BSB-105 Mondays and Wednesdays at 8:30am, and Friday's at 10:30am.
- Tutorial in BSB-B154 Tuesday's at 2pm.
- TA Office Hours are in ITB-128 from 1pm till 3pm.

Two opportunities to get help with material!

Course web-site

`http://www.cas.mcmaster.ca/~alhassm/`

Quizzes and Participation

- Six quizzes over the term —intended to be brief!
- Every other Monday in-class; worth 9% of grade.

- 6% of grade is for participation
- A letter of introduction telling me about yourself is due next Friday.

Note Taking and Prerequisite Knowledge

- Slides outline the ideas discussed
- Details are found in lecture —for which participation counts!
- The intent is for the student to print them out and adjoin details as needed.

What do I need to know to get by?

Familiarity with basic arithmetic, programming, and elementary logic. The last item will be reviewed, as will other topics that arise.

The outline contains a formal 'specification' of what you ought to know and what you'll hopefully learn ;)

Frama-C

Example (First Pointer Is Set To Min)

```
/*@ requires \valid(p) && \valid(q);
ensures *p <= *q;
ensures (*p == \old(*p) && *q == \old(*q)) ||
(*p == \old(*q) && *q == \old(*p));
*/
void max_ptr(int* p, int* q)
{
  if (*p > *q) {  int tmp = *p; *p = *q; *q = tmp; }
}
```

This is C; later we'll switch to a more elegant learning language...

CalcCheck –verified proofs, limited

Example

```
Theorem (3.92) "if alternate definition":
  P if b then x else y fi  ≡  b ⇒ P x  ≡  ¬ b ⇒ P y
Proof:
    P if b then x else y fi
  ≡⟪ (3.92a) "if alternate definition" ⟫
    b ∧ P x  ≢  ¬ b ∧ P y
  ≡⟪ (3.15) and (3.14) ⟫
    false  ≡  b ∧ P x  ≡  ¬ b ∧ P y
  ≡⟪ (3.15) ⟫
    b  ≡  b ∧ P x  ≡  ¬ b ≡ ¬ b ∧ P y
  ≡⟪ "Definition of ⇒" ⟫
    b ⇒ P x  ≡  ¬ b ⇒ P y
```

We will use this system within Avenue ; theorem lists to come.

Hoare Logic —deriving code from math

Example (Celebrity Problem)

$\{$ there is a celebrity in the room $\}$
$c, j := 0, 1$
; **do** $j \neq n \rightarrow$
  if j knows c **then** $j := j + 1$ **else** $c, j := j, j + 1$ **fi**
od
$\{c$ is in the room and every one else is not a celebrity $\}$

Example (Proving a theorem)

$\{n \geq 1\}$
$i, m := 1, 1$
; **do** $i \neq n \rightarrow i, m := i + 1, x \cdot m + 1$ **od**
$\{x^n - 1$ is divisible by $x - 1\}$

Agda –verified proofs, unlimited; and actual programming

Example

```
&-preservation = λ {x} {y} → ≈-begin
      f  (g  (x & y))
  ≈⟨ g preserves-& even-under f ⟩
      f ((g x) & (g y))
  ≈⟨ f preserves-& ⟩
      f (g x) & f (g y)
    □
```

This is actual code! We'll only do a little bit of this.

Constructive Theorem Proving

Function Application

$$\text{if } f : A \to B \text{ and } a : A \text{ then } f\ a : B$$

Or ignoring 'f' and renaming,

Modus Ponens

$$(p \implies q) \wedge p \implies q$$

These both are "crossing the bridge" and more generally,

Curry-Howard Isomorphism

$$\text{Proving} \approx \text{Programming}$$

Class Discussions

Motivating Example

How to sort an array, a sequence of numbers?

Buzzwords: pre- and post-conditions, specification, requirement, testing

Daily Life

Finally, briefly discuss how specifications and correctness applies to daily non-computing life and the notion of goal-setting.

References

ACSL Mini-Tutorial

`https://frama-c.com/download/acsl-tutorial.pdf`

We will begin Frama-C next week, just to "get our hands dirty"

Agda by Example: Sorting

`http://mazzo.li/posts/AgdaSort.html`

Highly recommended read!

CalcCheck: A Proof-Checker for Gries and Schneider's "Logical Approach to Discrete Math"

`http://calccheck.mcmaster.ca/`

There is an in-browser version with code completion and other expected utilities that will be made available on Avenue. Our course builds on LADM and so the checker is useful for us.