Should We Teach Formal Methods at All?

Jayadev Misra

Department of Computer Science
University of Texas at Austin

Visions Lecture
Nov. 9, 2009

Dedicated

to the memory of Amir Pnueli
April 22, 1941 — Nov 2nd, 2009

Lecture Subject

- Should we teach formal methods to undergraduates?
- If so, why and how?

What this lecture is not about

- My latest research.
- I find formal methods delightful.
- Tales from a traveling salesman.

Who I am not

- Dana Scott.
- An apologist for formal methods.
- Candidate for any office.

Euclid's argument for the GCD algorithm

http://aleph0.clarku.edu/~djoyce/java/elements/boo

Russel's letter to Frege, circa 1902

Let w be the predicate: to be a predicate that can not be predicated of itself.

Can w be be predicated of itself?

From each answer its opposite follows. Therefore we must conclude that w is not a predicate.

...

The above contradiction, when expressed in Peano's ideography, reads as follows:

$$w = cls \cap x\, \varepsilon (x \sim_{\in} x).\supset: w \in w\,.=.\, w \sim_{\in} w$$

Russel Paradox in Modern Notation

$S = \{x | \ x \notin x\}$, or

for all x, $x \in S \equiv x \notin x$

Proof: Instantiate x by S

$S \in S \equiv S \notin S$

Recreational Math

- Lack of formal notation and symbol manipulation.
- Plenty of pictures, analogies and examples.
- Only common-sense reasoning.
- Proof of specialized cases, omission of boundary cases.

In our teaching of professionals, we do no better.

Cantor Diagonalization

Theorem:
There is no 1-1 correspondence between a set and its powerset.

Proof of Cantor's Theorem

Let f be a 1-1 correspondence from S to its powerset. Let T be the subset of S defined by

$$T = \{x \mid x \notin f(x)\}$$, or

for all x, $x \in T \equiv x \notin f(x)$

Instantiate x by $f^{-1}(T)$

$$f^{-1}(T) \in T \equiv f^{-1}(T) \notin f(f^{-1}(T))$$, or
$$f^{-1}(T) \in T \equiv f^{-1}(T) \notin T$$

But Computer Science is different!

- Computer science is about programming, not math.
- Our students have poor math background; they will never get it.
- No one else teaches this stuff; text books don't teach it.
- I can teach this material equally convincingly without using any formalism at all.
- Formal methods just waste a lot of time; you could cover much more material in the same time.
- The only formal methods I like are the ones I don't see.

Formal methods come in 3 Flavors

- Good
- Bad
- Ugly

Good Formal methods

- Good (effective) formalisms are few in number.
- Goal is not merely correctness, but minimizing mental effort.
- Goal is not merely verification; it is for specification, design, analysis and synthesis.

Our curriculum has become math-phobic[1]

Among 5 most popular books on data structures, each in the neighborhood of 800 pages

Only one book devoted 8 pages to program correctness.

Message to students: muddle through with words, pictures and plausible explanations than make an honest effort at a proof.

Do spend a lot of time muddling through.

---

[1] Allen B. Tucker, Charles F. Kelemen, and Kim B. Bruce. Our curriculum has become math-phobic! In H. Walker, R. McCauley, J. Gersting, and I. Russell, editors, SIGCSE Technical Symposium on Computer Science Education, pages 243–247, Charlotte, North Carolina, USA, 2001. ACM Press.

How do we explain without invariants

- Heapsort
- Quicksort
- Red-Black Trees
- Scan-Mark Algorithm for Garbage Collection
- ...

Euclid's GCD algorithm

$u, v := x, y$
$\{u \geq 0,\ v \geq 0,\ u \neq 0 \ \vee \ v \neq 0,\ \gcd(x, y) = \gcd(u, v)\}$

while $v \neq 0$ **do**
  $u, v := v, u \bmod v$
od

$\{\gcd(x, y) = \gcd(u, v),\ v = 0\}$
$\{\gcd(x, y) = u\}$

Extended Euclid Algorithm

Given nonnegative integers x and y, where both integers are not zero, there exist integers a and b such that

$$a \times x + b \times y = \gcd(x, y)$$

Extended Euclid Algorithm

invariant: $(a \times x + b \times y = u) \wedge (c \times x + d \times y = v)$

$u, v := x, y;\ \ a, b := 1, 0;\ \ c, d := 0, 1;$

while $v \neq 0$ **do**

$\{ (a \times x + b \times y = u) \ \wedge \ (c \times x + d \times y = v) \}$

$u, v := v, u \bmod v;$

$a, b, c, d := a', b', c', d'$ —- what are a', b', c', d' ?

$\{ (a \times x + b \times y = u) \ \wedge \ (c \times x + d \times y = v) \}$

od

What we must teach well

- Basic algebra: Sets, Functions, Relations, Partial Order
- Basic Logic: Propositional and Predicate Calculus
- Recursion and Induction
- Invariant and well-founded order

For advanced students

- Structural Induction
- Temporal Logic (Thanks to Amir Pnueli)

What we must teach well

- teach the basics relentlessly.
- teach when formal methods are inadequate.

Can UT take the lead?

In formal methods, UT boasts:

- Allen Emerson,
- Robert Van de Geijn,
- Warren Hunt,
- Vladimir Lifschitz,
- J Moore, and
- many others ...

Our courses could emphasize

- writing careful spcifications (contracts),
- analyzing specifications, manually and automatically,
- annotating programs with invariants,
- having termination arguments, and
- understanding where formal methods are not applicable.