Keynote

The Psychology of Programming and the Psychology of Mathematics

Henry Lieberman (work with Warren Robinett)
MIT Computer Science & AI Lab

Many fundamental results in mathematical logic and set theory, among other mathematical fields, were developed long before the advent of computers and modern computer science. Many of these proofs depend crucially on symbol-manipulation procedures, originally described in the proofs with prose or with equations that specify constraints on the procedures. We recast some of these procedures with modern computer science concepts, in modern programming languages, and discover (perhaps not surprisingly), that... some of them have bugs.

One reason that these bugs may have gone unnoticed for so long may be due to differences in how mathematicians and programmers think about procedures. Programmers think about control structures and temporal relationships, whereas mathematicians want to abstract away from time. The equation is ubiquitous in mathematical language, whereas programmers have discovered the "=" can have different roles: definition, equality testing, and assignment. Programmers think computationally, whereas mathematicians are comfortable with "there exists" descriptions, which may prove uncomputable. Finally, the two fields have different ideas about the notion of abstraction, which affect how their concepts are introduced and how they are used.