Lecture 1

1. What is this course about?

– Clearly it covers reasoning logically about programs, that entails a specification language for programming tasks and rules/principles of reasoning.

  This kind of reasoning seems clear for certain tasks such as well known data structures – numbers, lists, trees.

  It is less clear for others such as real numbers or distributed protocols.

– This course covers formal reasoning mentioned as in some of our courses on programming, “proving programs correct”.

– Nowadays formal reasoning must cover automated reasoning and proof assistants such as Agda, Coq, Nuprl; for type theory: HOL, HOL-light, PVS for higher order logic.

– A comprehensive approach to logic touches on many facits of computer science:

  Precise definition of a programming language.

  Defining data structures formally.

– Proof assistants are playing an increasing role in mathematics and theoretical computer science such as helping solve open problems [3, 4], and checking long hard proofs such as the Four Color Theorem or the Odd Order Theorem in finite simple group theory [2, 1].

– The logics of the major proof assistants may play a role as a unifying foundation for mathematics and computer science.

– Opportunity to see some first-class research problems.

  Foundations of HoTT.

  Protocol verification and synthesis.

  HACMS-2/next generation proof assistants; machine learning, NLP (logic of events).

– Opportunity to learn to use proof assistants: Coq, Nuprl.

– Opportunity to join formal methods research projects, learn proposal writing.

– Excellent thesis topics. There are fewer strong formal methods researchers than machine learning experts. Industrial opportunities: MSR, EMC, INRIA.
2. Course mechanics
(a) Small project/article/verification.
(b) Study an open problem.

3. Rough Schedule
- First-order logic.
- Higher-order logic.
- Type theories: CTT vs CIC.
- Constructivity, key examples: completeness, virtual evidence semantics.
- Open problems.
- More type theory, possibly elements of Coq model of Nuprl; possibly Constructive Euclidean Geometry.
- Logic of events.
- Protocol synthesis.

4. Example of possible new result even for the 136 year old first-order logic (FOL).
Consider this valid first-order formula:

\[(\forall x : D.F(x) \Rightarrow \exists v : D.G(v)) \Rightarrow \exists v : D.(F(v) \Rightarrow G(v))\]

It is easy to see that this is “true” using the Law of Excluded Middle on \(\forall x : D.F(x)\), that is \((\forall x : D.F(x) \lor \sim \forall x : D.F(x))\).

In the case of \(\forall x : D.F(x)\), we get \(\exists v : D.G(v)\), and we use the value \(v\) to show \(\exists v : D.(F(v) \Rightarrow G(v))\) since \(G(v)\) holds.

In the case \(\sim (\forall x : D.F(x))\), we know \(\exists x : D \sim F(x)\). We can use this \(x\) for \(v\) since \(\sim F(v)\) and \(F(v)\) gives False from which we know \(G(v)\).

We can probably show that for a class of domains \(D\) with decidable equality, we can find a constructive proof using one new rule of FOL. We will explore this topic later.

References

