Computer-Aided Reasoning for the Masses

Pete Manolios
Northeastern University

Seattle Microsoft Nov 15 2010

Thesis

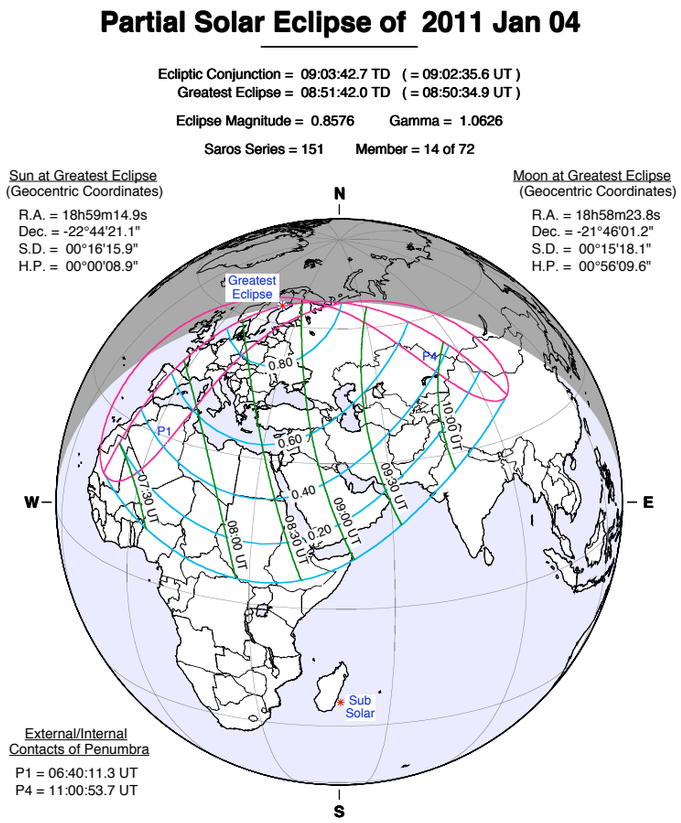
Usability =

- Can you integrate in undergraduate curriculum?
- Is there wide adoption?
- I'll tell you about our experience at Northeastern U.

Why Teach Formal Methods?

- Rules of computation?
- Predictive power?
- Specify conjectures?
- Reason about computation?

- Compare w/ physics
- Rules of the universe?
- Predictive power?
- Conjecture: 1st solar eclipse of 2011: Jan 4



Logic & Computation

- Freshman, first semester
  - How to Design Programs
  - Racket, Scheme-like language
  - Design Recipe
  - Data-Driven Definitions

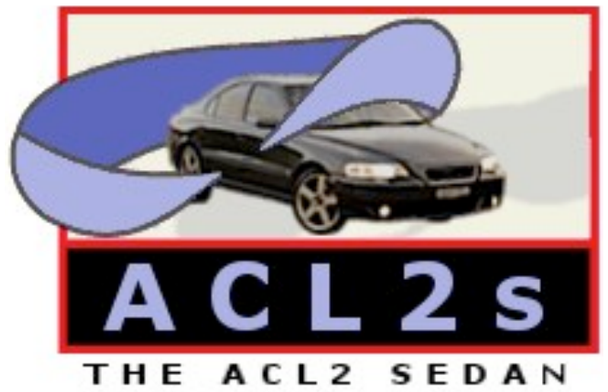
- Second semester
  - How to Specify, Reason about Programs
  - ACL2, Lisp-like language
  - Informal contracts become formal
  - Requirements and specification
  - Theorem Proving

Logic & Computation

- Material is new for almost everyone
- Start with propositional logic
  - Satisfiability, tautology, falsifiable
  - Proof by exhaustive testing
  - SAT solver & applications
- Reasoning about programs
  - Falsification as before: counterexamples (evaluation)
  - Proof is more elusive: exhaustive testing fails
  - Logic: finite work, infinite conclusions!
- Induction: Data-Recursion-Induction trinity
- Examples: circuits, data structures, compilers, efficiency, equivalence, video games, ...

ACL2 Sedan

- ACL2 theorem prover
  - Runs like a well-tuned race car in the hands of an expert
  - Unfortunately, novices don't have the same experience
- ACL2s: The ACL2 Sedan
  - Usability primary concern
  - From race car to sedan
  - Eclipse: modern development environment
  - Self-teaching, modes, debug
  - Visualize, interact with ACL2
  - Termination, Counterexamples
  - Open source



DEMO

Conclusions

- FM tools for the masses will not happen, unless we integrate their use in the undergraduate curriculum
- We need to train the next generation of engineers
- Freshmen + theorem proving works. Try it!
- FM tools should support the art of specification
- Provide counterexamples, explain failure
- Keep interfaces simple: focus on concepts not syntax
- Users are king: FM tools provide cognitive amplification