Axiomatic Events in ACL2(r)

Ruben Gamboa, John Cowles, and Nadya Kuzmina
University of Wyoming

Introduction

- ACL2(r) is a variant of ACL2 that supports the irrational numbers
- It is distributed with the ACL2 sources
- The foundations of ACL2(r) lie in non-standard analysis

The Big Problem

- Soundness of ACL2(r) has been argued before
- But the soundness argument was static
  - I.e., it is based on looking at a single theory
- The question remains: how does ACL2(r) interact with the dynamic aspects of ACL2?
  - e.g., defun, defchoose, encapsulate

Static? Dynamic?

- The real question is:
  - When is a formula X a theorem of a particular ACL2(r) theory T?
- This is complicated by the fact that the theory T changes as new function symbols are added
- The previous soundness argument did not address changes in the theory T

The ACL2 Story

- This question has been answered in the context of ACL2
- K&M proved the consistency of ACL2 by showing how ACL2 theories are really ordinary first-order theories
- What this means is that instead of thinking of inference methods (e.g., induction) for ACL2, we think of having special first-order axioms (e.g., an induction axiom schema)

The First Challenge: Inference Rules

- Thinking of ACL2 as a first-order theory with some special axioms results in a big challenge
- How do we make sure that the special "rule axioms" are in the theory when new functions are added?
- E.g., if T is a theory and we extend it by adding the new function symbol f, why should the induction axioms involving f be automatically included in the new theory?

The Second Challenge: Functional Instantiation

- Functional instantiation is another major inference rule of ACL2
- This can not be justified using an axiomatic approach
- Instead, the soundness of functional instantiation follows by proof transformation

Conservative Extensions

- K&M's proof of the correctness of ACL2 makes extensive use of "conservative extensions"
  - A theory T' is a conservative extension of a theory T if the theorems of T' that can be stated in T are precisely the theorems of T
  - I.e., no new theorems over the old language

Why Conservative?

- Suppose T' is a conservative extension of T
- Let X be a theorem of T', where X is in the language of T
- Then there is a proof of X completely in T
  - used to justify functional instantiation
  - order of definitions is unimportant

The Third Challenge: Definitional Axioms

- The ACL2 story depends on the fact that when a new function symbol is introduced, the new theory is a conservative extension of the old

- A large part of the story is concerned with showing that each of the definitional axioms are conservative

  - defun, defchoose, encapsulate

What's in ACL2(r)?

- Built-in support for realp and complexp
- Some numbers are "standard", and at least one number is not
- Functions can be classical or not
  - non-classicalness is infectious
- Non-classical functions can not be defined recursively

What else is in ACL2(r)?

- It is possible to create a new classical function using a non-classical body (seemingly violating the infectiousness of non-classical)
  - If so, we only know what the new function does for standard arguments

Dangerous things in ACL2(r)

- Suppose F(x) is a classical formula with free variable x
- To prove that F(x) is a theorem, we can assume that x takes on only standard values!
- This is called the *Transfer Principle*

More Dangerous Visions

- Induction has to be carefully controlled in ACL2(r)
- If P(x) is a non-classical formula, we can not use induction to prove that P(x) is true
- We can use induction to show that P(x) is true, but only for all standard values of x
- The remaining case must be handled separately

Basic Soundness of ACL2(r)

- The Transfer Principle and the basic machinery of "standard" was developed by Robinson in the context of model theory
- Nelson reformulated this non-standard analysis into an axiomatic setting called internal set theory

Basic Soundness of ACL2(r) (Cont'd)

- Internal set theory (IST) is a conservative extension of classical set theory (e.g, ZFC)
- A given ACL2(r) theory can be interpreted in an IST setting
- IST places some stringent syntactic restrictions on the use of induction and the transfer principle
  - ACL2(r) abides by these restrictions

End of story?

- Not quite....
- How does this reconcile with the correctness of ACL2?
- E.g., where does conservativity come in?
- What about encapsulate, include-book?
- We need a story of ACL2(r) that coexists with the story of ACL2

ACL2(r) Induction Axioms

- The ACL2 story uses “induction axioms” to justify the induction inference rule of ACL2
- In ACL2(r), we have similar induction axioms, but we take special care of non-classical formulas
- Induction in ACL2(r) is weaker than induction in ACL2 (for the “(r)” formulas)

ACL2(r) Transfer Axioms

- ACL2(r) introduces "transfer axioms" to justify the transfer principle in ACL2(r)
- These are completely analogous to the induction axioms

ACL2(r) Standardization Axioms

- ACL2(r) uses "standardization axioms" to justify the introduction of new classical functions from non-classical definitions
- These refer to function symbols that are not in the "user visible" language of ACL2(r)
  - There is one "non-visible" symbol for each formula in ACL2(r)
  - They name each definable function

Are these "rule axioms" sound?

- Yes!
  - At least in the initial ACL2(r) theory
- This follows from the basic soundness of ACL2(r)
- E.g., use IST to build a non-standard model of ACL2(r)

What happens when we defun?

- If we use defun to introduce a new function symbol, why are the corresponding “rule axioms” of the new function symbol true?
- We can show this by carefully considering each axiom type, and showing that each axiom is a logical consequence of the definitional axiom and the old rule axioms

What about defun-std?

- A similar story works for defun-std
- The rule axioms can be derived from the old rule axioms and the definitional axiom for the new symbol

What about defchoose?

- Well, we think we have an answer for that....
- ....but that's for the future

Functional Instantiation

- The trick to showing functional instantiation is sound is to consider each step in the proof of the original theorem
- Each step can be transformed using the functional instance
- It all works, as long as the functional instance converts axioms to axioms

Functional Instantiation (Cont'd)

- This almost works in ACL2(r)
- The biggest challenge has to do with the standardization axioms
- This is because the functional instance has to transform a formula and the non-visible funtion corresponding to that formula consistently
- This is worked out in the paper

Conservativity in ACL2(r)

- Finally, we can show that the definitional axioms in ACL2(r) are conservative
- The argument is similar to the one used in the story of ACL2

Looking back

- It is possible to tell a story of the soundness of ACL2(r) that is consistent with the story for ACL2
- This means that the "new" principles in ACL2(r) work nicely with the structured mechanisms of ACL2
- We now have a rigorous foundation for ACL2(r)

Looking forward

- We can use the new, rigorous foundation for ACL2(r) to evaluate possible enhancements
- We are in the process of extending ACL2(r) to make it more powerful
  - recursive, non-classical functions
  - easier to prove a term is standard
  - classical, internal, and external terms