Proposal: Extending ACL2's Arithmetic Lemma Library

Guillermo Cabrera
The University of Texas at Austin
Department of Computer Science
1 University Station M/S C0500
gcabrera@cs.utexas.edu

ABSTRACT

The proposed project involves the analysis of the current state of ACL2's arithmetic libraries, as literature in this area dates back to version 2.7, whereas current version is 4.1. Furthermore, lemmas will be added to the library, in an effort to reduce the time consumed by users in proving arithmetic proofs.

1. INTRODUCTION

The ACL2 theorem prover has been used in many fields ranging from algorithm correctness to verification of correct functionality of hardware components. Some of these industrial proofs can extend to thousands of pages if they were to be printed and in this process there could be a need to prove arithmetic lemmas. As an example, the work in [2] mentions some arithmetic problems involving the correctness of a microprocessor and the correctness of a 6502 assembly program to multiply two 8-bit numbers.

Until now, in class we have focused on using the list as the principal data structure and have proved theorems that are the result of formalizing concepts and properties related to lists, sets and trees. In this process of theorem proving with ACL2, we have been guided by the mechanical nature of its process, starting from our input and sequentially getting the help of several proof techniques, including: Simplification, destructor elimination, equivalence use, generalization, irrelevance elimination and finally induction.

The ACL2 theorem prover works with the user and a logical world, the latter comprises the steps outlined above. For the purpose of this work, our focus will be on the simplification step; it processes formulae by first applying propositional calculus, then using equivalence relations does congruence closure, and finally linear arithmetic is used. This last step encompasses among others, a special simplifier, divided into the lemma library and the simplifier proper.

The lemma library constitutes an important part of this component, as it will serve as the knowledge base that will help ACL2 arrive at correct proofs. The problem arises when the lack of lemmas leaves the burden to the user, for him/her to introduce additional lemmas, in order to arrive at a correct proof

2. MOTIVATION

The effort by users in dealing with arithmetic proofs can be considerable, and is the main problem I plan to undertake in this project.

It is worth mentioning a succinct overview of how ACL2 handles arithmetic as clearly presented in [1]. First, the conclusion in an arithmetic proof is negated, then, the hypothesis which take the form of polynomials are processed through a series of steps involving multiplication and addition. These polynomials are combined and reduced so that a contradiction can be obtained, thus, proving the theorem correct.

The entire process can be abstracted into three main stages that depend on each other: Starting with linear arithmetic, then partial interpretation and finally nonlinear arithmetic; the implementation involves all three as nested loops. If a formula contains a single polynomial and use of functions, then linear arithmetic cannot do anything, yet, the partial interpretation loop takes over in an effort to find a contradiction. Finally, the nonlinear loop will derive polynomials that might be apparent by properties such as factoring.

This above process is highly dependent on the lemma library mentioned earlier, it contains rules that extend and guide the operation of three packages: rewriter,, a type-reasoning package and the arithmetic package. Providing this library with any potential intermediate lemmas will add to the power of ease of use of ACL2's arithmetic proof process.

3. APPROACH

As a first step, a meeting has been scheduled with Robert Krug from the ACL2 group, in order to get the status of the arithmetic package as of version 4.1 Thereafter, I will focus on an area needing work, such as the case of translating existing lemmas into the current version of ACL2, and adding necessary intermediate lemmas that might be essential in obtaining a correct proof.

Moreover, a key part in undertaking this project will require knowledge on the details of how arithmetic proofs are treated, thus, further literature will be added that explains this process.

The expected contribution is twofold, first report on the current state on linear and nonlinear arithmetic in ACL2, as well as add to the database of lemmas available for the arithmetic package.

4. REFERENCES

[1] W. A. Hunt, R. B. Krug, and J. Moore. Integrating nonlinear arithmetic into ACL2. ACL2-2004.

[2] W. A. Hunt, R. B. Krug, and J. Moore. Linear and nonlinear arithmetic in ACL2. In D. Geist, editor, Proceedings of the 12th Advanced Research Working Conference on Correct Hardware Design and Verification Methods, CHARME 2003, volume 2860 of Lecture Notes in Computer Science, pages 319–333. Springer-Verlag.