COSC 5010. Formalizing the JVM in ACL2.   Spring 2004

Homework Assignment 3

1. Start reading Chapters 6 and 7 of Computer-Aided Reasoning: An Approach

2. Start learning about ACL2 proofs by doing the following 5 easy lessons (no credit).

   ACL2 Proofs – Lesson 1
   > http://www.cs.uwyo.edu/~cowles/jvm-acl2/proof-lesson1.lisp

   ACL2 Proofs – Lesson 2
   > http://www.cs.uwyo.edu/~cowles/jvm-acl2/proof-lesson2.lisp

   ACL2 Proofs – Lesson 3
   > http://www.cs.uwyo.edu/~cowles/jvm-acl2/proof-lesson3.lisp

   ACL2 Proofs – Lesson 4
   > http://www.cs.uwyo.edu/~cowles/jvm-acl2/proof-lesson4.lisp

   ACL2 Proofs – Lesson 5
   > http://www.cs.uwyo.edu/~cowles/jvm-acl2/proof-lesson5.lisp

3. In class I claimed that Ackermann's function could be implemented iteratively (using tail recursion). The file

   > http://www.cs.uwyo.edu/~cowles/jvm-acl2/iter-ack.lisp

   contains an ACL2 proof of that claim. The last event shows that recursive Ackermann computes the same result as iterative Ackermann for nonneg. int. inputs. The measure, showing iterative Ackermann terminates, is somewhat complicated and explained at the beginning.