ETH zürich

Alexander J. Summers

Program Verification

Exercise Sheet 2: Encoding Problems to SAT

Assignment 1 (Eliminating Equality)

The variation of Ackermannization should in the lectures to eliminate equality potentially generates many variables and extra conjuncts to express instances of the reflexivity, symmetry and transitivity properties of equality.

1. How many variables are introduced, given n term variables in the original formula?
2. How many instances of reflexivity, symmetry and transitivity properties will be added?
3. Can you think of ways to reduce the necessary additions to the original formula?

Assignment 2 (Eliminating Uninterpreted Functions)

Apply the techniques explained in the lectures, to convert the following formula into a formula without the functions f and g (you don't have to convert it all the way to a propositional one, unless you want to):

$$f(g(x)) = x \wedge f(y) = x \wedge \neg(y = g(x))$$

The resulting formula should be satisfiable - what is a model for it? What model does this describe for the *original* formula above?

Assignment 3 (Sudoku Encodings)

Consider the two different encodings of Sudoku rules from the lectures. How many clauses do each generate? What size are the clauses themselves? Which would you expect to perform better, using the SAT algorithms covered in the lectures?

Assignment 4 (Bit-blasting)

Write down a propositional encoding (via bit-vectors) of a multiplier which takes two, 2-bit input integers, and produces a 4-bit output value equal to their product.