EECS 219C: Formal Methods — Assignment 2

Parker Ziegler

February 23, 2022

1 Sum-Sudoku

1.1 Bit Twiddling Hacks

1.1.1 (a) Equivalence of `f1` and `f2`

My SMT-LIB encoding to check equivalence of `f1` and `f2` is located in `1a.ascii`. Based on my encoding, `f1` and `f2` **are not equivalent**. Z3 provides the counterexample $x = 1$, in which `f1(x)` evlauates to -1 while `f2(x)` evaluates to 1. The full output returned by Z3 is:

```
(
  (define-fun v_1 () (_ BitVec 32)
    #x00000000)
  (define-fun x () (_ BitVec 32)
    #x00000001)
  (define-fun v_0 () (_ BitVec 32)
    #xffffffff)
  (define-fun ret_2 () (_ BitVec 32)
    #x00000001)
  (define-fun v_2 () (_ BitVec 32)
    #x00000001)
  (define-fun ret_1 () (_ BitVec 32)
    #xffffffff)
)
```

Cleaning this up to use decimal notation, the full assignment is:

$$[x = 1,\ v\_0 = -1,\ ret\_1 = -1,\ v\_1 = 0,\ v\_2 = 1,\ ret\_2 = 1]$$

1.1.2 (b) Equivalence of `f3` and `f4`

My SMT-LIB encoding to check equivalence of `f3` and `f4` is located in `1b.ascii`. Based on my encoding, `f3` and `f4` **are equivalent**.

1.2 Formulate an SMT instance that finds a solution to Sum-Sudoku puzzles