EE219C HW2: SMT

Vighnesh Iyer

1 Bit-Twiddling Hacks

(a) Are the functions `f1` and `f2` in Figure 1 equivalent?

```
int f1(int x) {
  int v0;
  if (x > 0) v0 = x;
  else v0 = -x;
  return v0;
}
```

```
int f2(int x) {
  int v1, v2;
  v1 = x >> 31;
  v2 = x ^ v1;
  return (v2 - v1);
}
```

I encoded this problem with the Z3 Python API:

```
x, v0, v1, v2 = BitVecs('x v0 v1 v2', 32)
s = Solver()
s.add(v0 != v2 - v1, v0 == If(x > 0, x, -x), v1 == x >> 32, v2 == x ^ v1)
print(s.check())
print(s.sexpr())
```

The equality between the return values of `f1` and `f2` was inverted to check for validity. The results were:

```
unsat
(declare-fun v0 () (_ BitVec 32))
(declare-fun x () (_ BitVec 32))
(declare-fun v2 () (_ BitVec 32))
(declare-fun v1 () (_ BitVec 32))
(assert (distinct v0 (bvsub v2 v1)))
(assert (= v0 (ite (bvsgt x #x00000000) x (bvneg x))))
(assert (= v1 (bvashr x #x00000020)))
(assert (= v2 (bvxor x v1)))
(model-add v0
           ()
           (_ BitVec 32)
           (bvmul x (ite (bvsle x #x00000000) #xffffffff #x00000001)))
(model-add v2 () (_ BitVec 32) (bvxor x v1))
```

Showing that `f1` and `f2` are functionally equivalent.