Selected Solutions 12 June 21, 2019

1 (a) The ground solver will return the model M containing $\neg P(a)$, $R(b)$, $S(c)$ together with the quantified literals $\forall x.R(x) \vee S(x)$ $\forall x.\neg R(x) \vee P(x)$ $\forall x.P(x) \vee \neg S(x)$.

We first consider E-matching using the instantiation patterns $P(x)$. This generates the instances

$$R(a) \vee S(a) \qquad \neg R(a) \vee P(a) \qquad \neg P(a) \vee \neg S(a)$$

such that in the next iteration the ground solver can conclude unsatisfiability.

Second, for enumerative instantiation we can consider the order $a < b < c$. The smallest instances of the three literals with quantifiers are thus $R(a) \vee S(a)$, $\neg R(a) \vee P(a)$, and $\neg P(a) \vee \neg S(a)$. None of these clauses follows from M, so they are returned by the instantiation module. In fact these instances suffice for the ground solver to conclude unsatisfiability.

(b) This problem is satisfiable. Suppose an order where $a < f(a) < f(f(a)) < \dots$ are smaller than all terms with b or g.

We assume the ground solver returned the model $a = b$, $g(a) = g(b)$, $f(a) = g(a)$ together with $\forall x.f(f(x)) \neq x$. Consider the series

$$E_0 = \{a = b,\ g(a) = g(b),\ f(a) = g(a)\}$$
$$Q_i = \{f^{i+2}(a) \neq f^i(a)\}$$
$$E_{i+1} = E_i \cup Q_i$$

where all E_i are satisfiable, e.g. by taking the model $a_{\mathbb{N}} = b_{\mathbb{N}} = 0$, $g_{\mathbb{N}}(x) = 1$, and $f_{\mathbb{N}}(x) = x + 1$ for all x. Thus the series satisfies the conditions of the lemma on Slide 19 of Week 12 and hence witnesses satisfiability of the quantified problem.

2 For instance, consider the problem $a = b$, $g(a) = a$, $f(a) \neq b$, $\forall x.f(x) = x$. Using instantiation pattern $g(x)$, no instance is generated that can be used to show unsatisfiability.

On the other hand, enumerative instantiation with $a < b < f(a) < \dots$ will immediately generate $f(a) = a$ such that unsatisfiability can be concluded.

3 (a) Using a quantified formula, one can check

```
(assert (forall ((x (_ BitVec 4)))
  (= (bvsub (bvxor (bvadd x #x2) #x3) #x2) (bvadd (bvxor (bvsub x #x2) #x3) #x2))
))
(check-sat)
```

(b) There are 128 solutions, see `knuth.py`.

5 See `rushhour.py`.