1 High-level Algorithm

Algorithm 1: High-level Quantifier Instantiation

Input: Quantified axioms: $\forall \boldsymbol{x}.A_1, \ldots, \forall \boldsymbol{x}.A_n$
Input: Quantifier free: φ
Output: Instantiations $A_{i_1}[\boldsymbol{t_1}], \ldots, A_{i_m}[\boldsymbol{t_m}]$ s.t. $\varphi \wedge \bigwedge_j A_{i_j}[\boldsymbol{t_j}]$ is unsat

```
1  𝓜 := ∅ // Initially, empty set of models
2  𝓘 := ∅ // and empty set of instantiations
3  while φ ∧ 𝓘 is sat do
4      Get model for φ ∧ 𝓘 and add to 𝓜
5      Find a small* set of instantiations 𝓘′ s.t. ∀M ∈ 𝓜. ∃I ∈ 𝓘′.M ⊭ I
6      𝓘 := 𝓘 ∪ 𝓘′
7      if 𝓜 is too large and line 5 takes too long then
8          Reset 𝓜 ← ∅ or optimize 𝓜 ← 𝓜′ where 𝓜′ ⊂⊂ 𝓜
9      end
10 end
11 return 𝓘
```

2 Finding Instantiations

Let there be an axiom $A = \forall \boldsymbol{x}.\psi$. An instantiation of that quantifier is a tuple of ground terms $\boldsymbol{t}$: $\psi[\boldsymbol{t}]$. Given a model M, violated instantiations are determined by checking $M \models \psi[\boldsymbol{t}]$. We can think of a quantifier as a relation R_A, and then violations are $\boldsymbol{t}^M \notin R_A^M$. We can define the set of violations as $\text{Violations}(M, A) := \{\boldsymbol{t} \mid \boldsymbol{t}\}$.

Given a set of axioms $A_1, \ldots, A_n$ we can define the set of all violations as $\text{Violations}(M) := \{(A_i, \boldsymbol{t}) \mid \boldsymbol{t} \in \text{Violations}(M, A_i)\}$.

Given a set of models $M_1, \ldots, M_m$ we want a set of

3 Points

- Lemma pollution solution: models hitting sets to choose "good" lemmas.
- Running CVC5 with Dafny examples.