Complete Trigger Selection in Satisfiability modulo First-Order Theories

Christopher Lynch and Stephen Miner (Clarkson University)

SMT Workshop
Rome, Italy
July 2023

Background

- SMT solvers efficiently handle many built-in theories
- Other theories can be handled by quantified first-order theories
  - Represented by set of clauses, with universally quantified variables
- SMT solvers use *triggers* to decide which variables to instantiate, with one of the following results:
  1. Creating an unsatisfiable set of ground clauses
  2. Creating all possible instantiations according to the triggers
     - But does that mean satisfiable?
  3. Instantiation never halts

Example

First Order Theory

$$\neg p(X) \vee q(X)$$

$$\neg q(Y) \vee r(Y)$$

SAT Problem (ground)

$$p(a)$$

$$\neg r(a)$$

This is unsatisfiable

Triggers

- A *trigger function* maps a FO clause to a subset of its literals
  - We underline those literals in examples
- If each of those literals matches a literal in ground model then instantiate FO clause
- Note: In this paper there are no equalities or other theories
  - More on that later

Example 1 with Triggers

First Order Theory

$$\underline{\neg p(X)} \vee q(X)$$

$$\underline{\neg q(Y)} \vee r(Y)$$

SAT Problem (ground)

$$p(a)$$

$$\neg r(a)$$

Ground model: $p(a), \neg r(a)$

Example 1 continued

First Order Theory

$$\underline{\neg p(X)} \vee q(X)$$

$$\underline{\neg q(Y)} \vee r(Y)$$

SAT Problem (ground)

$$p(a)$$

$$\neg r(a)$$

$$\neg p(a) \vee q(a)$$

New ground model: $p(a), q(a), \neg r(a)$

Example 1 continued

First Order Theory

$$\underline{\neg p(X)} \vee q(X)$$

$$\underline{\neg q(Y)} \vee r(Y)$$

SAT Problem (ground)

$$p(a)$$

$$\neg r(a)$$

$$\neg p(a) \vee q(a)$$

$$\neg q(a) \vee r(a)$$

UNSAT

Example 2 with Triggers

First Order Theory

$$\underline{\neg p(X)} \vee q(X)$$

$$\underline{\neg q(Y)} \vee r(Y)$$

SAT Problem (ground)

$$p(a)$$

Ground model: $p(a)$

Example 2 continued

First Order Theory

$$\underline{\neg p(X)} \vee q(X)$$

$$\underline{\neg q(Y)} \vee r(Y)$$

SAT Problem (ground)

$$p(a)$$

$$\neg p(a) \vee q(a)$$

New ground model: $p(a), \color{red}{q(a)}$

Example 2 continued

First Order Theory

$$\underline{\neg p(X)} \vee q(X)$$

$$\underline{\neg q(Y)} \vee r(Y)$$

SAT Problem (ground)

$$p(a)$$

$$\neg p(a) \vee q(a)$$

$$\neg q(a) \vee r(a)$$

New ground model: $p(a), q(a), r(a)$
No more instantiations
SAT

Example 3 with Triggers

First Order Theory

$$\neg p(X) \vee \underline{q(X)}$$

$$\underline{\neg q(Y)} \vee r(Y)$$

SAT Problem (ground)

$$p(a)$$

$$\neg r(a)$$

Ground model: $p(a), \neg r(a)$
No instantiations
UNSAT but could not show it because of bad choice of triggers

Example 3 continued

First Order Theory

$$\neg p(X) \vee \underline{q(X)}$$

$$\underline{\neg q(Y)} \vee r(Y)$$

$$\color{red}{p(Z) \vee r(Z)}$$

SAT Problem (ground)

$$p(a)$$

$$\neg r(a)$$

Those triggers are ok if we add a new clause
We can show UNSAT

Example 4 with Triggers

First Order Theory

$$\underline{gt(s(X), X)}$$

$$\underline{\neg gt(Y, Y)}$$

SAT Problem (ground)

$$gt(a, b)$$

- z3 (With or without mbqi), cvc4, veriT, SMTInterpol all returned UNKNOWN
- We show that SMT solver can return SAT

Example 5

First Order Theory

$$\neg p(X, Y) \vee q(f(X), Y)$$

$$\neg q(X, Y) \vee p(X, f(Y))$$

SAT Problem (ground)

$$p(a, b)$$

z3 (With or without mbqi), cvc4, veriT, SMTInterpol all returned UNKNOWN

Example 5 trigger 1

First Order Theory

$$\neg p(X, Y) \vee \underline{q(f(X), Y)}$$

$$\neg q(X, Y) \vee \underline{p(X, f(Y))}$$

SAT Problem (ground)

$$p(a, b)$$

No instatiations
We show this is SAT

Example 5 trigger 2

First Order Theory

$$\neg p(X, Y) \vee \underline{q(f(X), Y)}$$

$$\underline{\neg q(X, Y)} \vee p(X, f(Y))$$

SAT Problem (ground)

$$p(a, b), \neg p(f(a), f(b))$$

No instantiation but UNSAT
We can add new first order clause to guarantee SAT

Example 5 trigger 3

First Order Theory

$$\underline{\neg p(X, Y)} \vee q(f(X), Y)$$

$$\underline{\neg q(X, Y)} \vee p(X, f(Y))$$

SAT Problem (ground)

$$p(a, b)$$

We show this is SAT if it halts
Unfortunately it has infinitely many instantiations

Finding the right triggers is important

- Bad triggers may
  - Cause infinite loop
  - Not find right instances for UNSAT
- Not knowing if your triggers are good will
  - Not allow you to decide SAT

Results of paper

We define a trigger function such that

- If FO theory is *saturated by Resolution* (to be defined) then SAT solving plus instantiations is complete, i.e.,
  - If UNSAT then it returns UNSAT
  - if it halts without returning UNSAT then it is SAT
- Guaranteed to halt under certain conditions
- Guaranteed to halt in polynomial time under certain conditions

Question from heckler in audience

- Q: So you are suggesting a new inference system that you claim is superior to the current triggers
- A: No. Think of this as a pre-processor
- Q: Good, because this seems interesting, but I don't want to re-implement my SMT solver
- A: You have anticipated what I was going to say next

Implications of paper

You can add a pre-processor to your SMT solver

1. Check if FO theory is saturated under Resolution
2. If not saturated, try to saturate it
3. If you fail to saturate, run SMT solver as usual
4. If you saturate, run SMT solver as usual and you never need to return UNKNOWN
    - Unless you time out

I still need to define *saturated*

Question from heckler in audience

- Q: Did you say there is no equality?
- A: Yes, I hoped to slip that by you
- Q: But SMT solvers are based on equality
- A: Ok, but this is just the start of our research, and we have already started extending it to equality. More at the end of the talk.
- Q: Fine. I will wait.

Example of Resolution

$$\neg p(U, V) \lor q(f(U), V)$$

$$\neg q(X, Y) \lor p(X, f(Y))$$

- q-predicates are identical with substitution $\sigma = [X \mapsto f(U), Y \mapsto V]$
- So we resolve on those literals
- Result is $\neg p(U, V) \lor p(f(U), f(V))$

Example of Resolution

$$\neg p(U, V) \vee q(f(U), V)$$

$$\neg q(X, Y) \vee p(X, f(Y))$$

- q-predicates are identical using mgu $\sigma = [X \mapsto f(U), Y \mapsto V]$
- So we resolve on those literals
- Result is $\neg p(U, V) \vee p(f(U), f(V))$
- This new clause self-resolves to $\neg p(U, V) \vee P(f(f(U)), f(f(V)))$
- Problem: This leads to infinite derivation: $\neg p(U, V) \vee p(f^n(U), f^n(V))$

Controlling Resolution with Literal Selection

- A *literal selection function* maps a FO clause to a subset of its literals
  - We underline those literals in examples
  - Yes, this is the same definition as trigger
- The only necessary resolutions are those among selected literals
- The literal selection function depends on a literal ordering

Atom Ordering

- Ordering must be well-founded
  - No infinite chain $A_1 > A_2 > \cdots$
- Ordering must be stable under substitution
  - $A > B$ implies $A\sigma > B\sigma$
- L must be totalizable on ground atoms
- Extend to literals so that $\neg A > A$
- Suppose $L \in C$ then
  - L is maximum in C if L is larger than all other literals in C
  - L is maximal in C if no literal in C is larger than L

Classical Literal Selection function

- Given an ordering $>$
- A literal selection function is *valid* if for each clause C either
  - all maximal literals are selected
  - some negative literal is selected

Our updated Literal Selection function

- Given an ordering $>$
- A literal selection function is *valid* if for each clause C whenever we remove a subset of the selected literals which does not contain all the clause variables either
  - all maximal literals are selected
  - some negative literal is selected
- Implication: Selected literals must contain all variables in clause

Saturation

- Assume a valid literal selection function
- A set of clauses is saturated if the conclusion of all Resolution and Factoring inferences already exists
  - Or is subsumed by an existing clause
  - Or is a tautology
- Completeness Theorem: If S is saturated then S is unsatisfiable iff $\bot \in S$

Saturation Example 1

$$\neg p(U, V) \vee \underline{q(f(U), V)}$$

$$\neg q(X, Y) \vee \underline{p(X, f(Y))}$$

- Valid selection because maximal selected in each clause
- Saturated because no inference among selected literals

Saturation Example 2

$$\neg p(U, V) \lor \underline{q(f(U), V)}$$

$$\underline{\neg q(X, Y)} \lor p(X, f(Y))$$

$$\neg p(U, V) \lor \underline{p(f(U), f(V))}$$

- Valid selection because maximal selected in first and third clause, and negative selected in second clause
- Saturated because the conclusion of the only inference already exists

Saturation Example 3: Intersection Theory

$$\neg elem(X, Y) \lor \neg elem(X, Z) \lor \underline{elem(X, int(Y, Z))}$$

$$\underline{\neg elem(X, int(Y, Z))} \lor elem(X, Y)$$

$$\underline{\neg elem(X, int(Y, Z))} \lor elem(X, Z)$$

- Valid selection rule because maximal selected in each clause
- Saturated because only inference yields tautology

Saturation under invalid Literal Selection rule

$$\underline{p} \vee q$$

$$\neg p \vee \underline{q}$$

$$p \vee \underline{\neg q}$$

$$\underline{\neg p} \vee \neg q$$

- Unsatisfiable but no empty clause
- Saturated because all inferences are tautologies
- Invalid Literal Selection

Our result

- Assume Trigger function = Literal Selection Function
- If FO theory is *saturated by Resolution* then SAT solving plus instantiations is complete, i.e.,
  - If UNSAT then it returns UNSAT
  - it halts without returning UNSAT then it is SAT

Instantiation rule

***I*-Instantiation:**

$$\frac{L_1 \vee \cdots \vee L_n \vee \Gamma}{(L_1 \vee \cdots \vee L_n \vee \Gamma)\theta}$$

where

1. $L_1 \vee \cdots \vee L_n \vee \Gamma$ is in FO theory
2. $L_1, \cdots, L_n$ are triggers
3. there exists $L'_1 \cdots L'_n$ in ground model such that $\bar{L_i}\theta = L'_i$ for all $1 \leq i \leq n$

- Notes:
  - We only require to instantiate onto complements
  - We only need matching not *E*-matching since there are no equalities

Question from audience (not heckler)

- Q: This is a bit abstract. How do I actually find this selection function?
- A: Good question. We did not focus on this in our paper. But here are some points
  1. Some literals are larger in any ordering, and those should be selected, like in my examples
  2. You can quickly try to saturate under different orderings if you are not sure
  3. There has been experimental research on this in the context of FO theorem proving

Another good question about models

- Q: If the SMT solver returns SAT, does it give you a model?
- : A: It gives you a model of your ground clauses that can be extended to a model of the FO clauses
- Q: Why not just give the entire model?
- A: It may be infinite
- Q: Then how do you know there is a model?
- A: We prove theoretically that the ground model can be extended to a Herbrand model of the FO clauses

Another good question

- Q: What happens if I select more literals than necessary?
- A: The FO clauses may not saturate.
- Q: But suppose the FO clauses do saturate
- A: You may get lots of (possibly infinitely many) instantiations.
- Q: How can we be sure that will not happen?
- A: We give some conditions next

Decision Problem Case 1

- Suppose the FO clauses are saturated
- and a single maximum literal is selected in each clause
- Then SAT solving plus Instantiation is guaranteed to halt

Decision Problem Case 2

- Suppose the FO clauses are saturated
- and all maximal literals are selected in each clause
- and there are only finitely many atoms smaller than each atom
  - Not the same as well-founded
- Then SAT solving plus Instantiation is guaranteed to halt

Polynomial Decision Problem for Horn Clauses

- Suppose all clauses are Horn
- Suppose the FO clauses are saturated
- and all maximal literals are selected in each clause
- and there are only polynomially many atoms smaller than each atom
- Then SAT solving plus Instantiation is guaranteed to halt in polynomial time for CDCL SAT solving
  - Assuming a negative literal is made true at each decision point
- I have not seen a proof of this for CDCL SAT solving without quantifiers

Polynomial Decision Problem for 2SAT

- Suppose all clauses have at most 2 literals
- Suppose the FO clauses are saturated
- and all maximal literals are selected in each clause
- and there are only polynomially many atoms smaller than each atom
- Then SAT solving plus Instantiation is guaranteed to halt in polynomial time for CDCL SAT solving
- I have not seen a proof of this for CDCL SAT solving without quantifiers

Intersection Theory example

$$\neg elem(X, Y) \lor \neg elem(X, Z) \lor \underline{elem(X, int(Y, Z))}$$

$$\underline{\neg elem(X, int(Y, Z))} \lor elem(X, Y)$$

$$\underline{\neg elem(X, int(Y, Z))} \lor elem(X, Z)$$

- There is an ordering with only polynomially many atoms smaller than each atom
- So this theory has a poly time decision procedure if all ground clauses are Horn
- Also this theory has a poly time decision procedure if all ground clauses have at most 2 literals

Summary

- If a FO theory is saturated by Resolution (pre-processing step) then
  - SAT solving + Instantiation will halt for UNSAT
  - if it halts without UNSAT then it is SAT
- Trigger function = Literal Selection function
- We also gave conditions when it is guaranteed to halt
- and when it is guaranteed to halt in polynomial time

Future work: Equality and other theories

1. Extend to ground equalities - needed for SMT
   - Already done
2. Extend to FO theories with equality
   - Currently working on
3. Extend to quantification over variables of other theories

Future Work: Selection Function

- Allow for less restrictive Literal Selection and Trigger function
  - Important for Equality
  - Currently working on

What if FO theory does not saturate

- May still work depending on ground theory
  - Requires more than pre-processing
- Make saturation depend on other models besides Herbrand model
  - E.g., combine with other methods like mbqi
  - This approach seems promising to me

Science vs Engineering

- This paper is more a work of science than a work of engineering
- It helps us understand when the trigger selection is complete
- In order to determine SAT when using triggers, this is crucial
- But obviously we want to use this to develop better SMT solvers
- I will let the heckler have the last word

The heckler returns

- Q: Where are your experimental results?
- A: We showed some examples where this succeeds and other methods do not
- Q: But those were toy examples. What about, for example, like SMTLIB
- A: Well, we partially implemented, but have no experimental results
- Q: I think the most important future work is to implement this and apply it to some real examples
- A: It seems a lot of work to implement this and it won't ever be as fast as other implementations
- Q: Maybe you could build a pre-processor that will saturate and select triggers and pass this to current SMT solver in verbose mode
- A: One reason I wanted to present this at SMT was exactly for these ideas