Z3 for iProver-Eq: Efficient Ground Solving for Instantiation-based First-order Reasoning

Christoph Sticksel

Joint work with Konstantin Korovin

Z3 Special Interest Group Meeting
3rd November 2011

Motivation

The main application of automated reasoning is verification of software, hardware, protocols etc.

Reasoning should

- scale to industrial-size problems and
- provide succinct formalisation.

First-order Logic

- High expressivity: quantifiers $\forall$, $\exists$
- Decidable fragments
- Resolution/superposition
- Weak ground reasoning and modulo theories

SAT / Quantifier-free SMT

- High efficiency
- Modulo theories
- DPLL/congruence closure
- Weak quantifier reasoning

Motivation

The main application of automated reasoning is verification of software, hardware, protocols etc.

Reasoning should

- scale to industrial-size problems and
- provide succinct formalisation.

First-order Logic

- High expressivity: quantifiers $\forall$, $\exists$
- Decidable fragments
- Resolution/superposition
- Weak ground reasoning and modulo theories

SAT / Quantifier-free SMT

- High efficiency
- Modulo theories
- DPLL/congruence closure
- Weak quantifier reasoning

Instantiation-based Methods: The Idea

Is a given closed formula $\forall \bar{x}\ \varphi(\bar{x})$ a theorem?

A refutationally complete method:

1. Guess finite number of ground instances of $\varphi(\bar{x})$
2. Test ground satisfiability

Benefits:

- Keep expressivity of first-order logic
- Exploit efficiency of SAT and *QF_SMT*

Instantiation-based Methods: The Main Problem

Core question in instantiation-based reasoning

How do we find the ground instances to witness first-order unsatisfiability?

- Decidable if there are finitely many ground instances
- Harder the "more" ground instances there are
- Differences between calculi:
  - generation of instances
  - integration of propositional solving

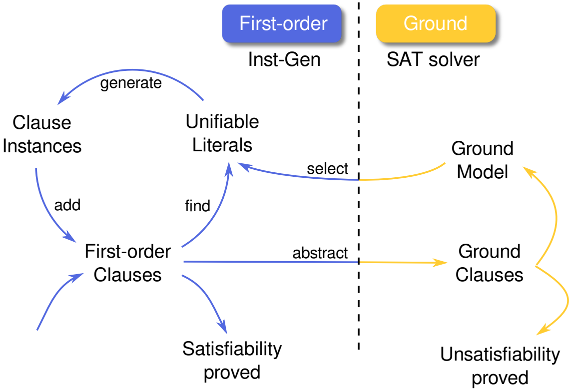
Features of Inst-Gen

- Modular combination of first-order and ground reasoning
- Ground reasoning delegated to off-the-shelf solver
- Very efficient for the EPR fragment
- Applied for hardware verification with bounded model checking (Intel)

- Non-equational variant related to Resolution
- Superposition-style equational reasoning
- Theory reasoning possible

- Implemented in *iProver* and *iProver-Eq*
- *[Korovin & Sticksel IJCAR 2010]* and *[Korovin & Sticksel LPAR 2010]*

The Inst-Gen Method



Inst-Gen: Ground Abstraction and Selection

First-order clauses

$$\neg Q(f(x))$$

$$\neg P(f(f(y)))$$

$$P(f(z)) \lor Q(z)$$

Ground abstraction with $\perp$

$$\neg Q(f(\perp))$$

$$\neg P(f(f(\perp)))$$

$$P(f(\perp)) \lor Q(\perp)$$

- Select literals which are true in ground abstraction

Fail to extend ground model to first-order

$$\neg P(f(f(y))) \models \neg P(f(f(a)))$$

$$P(f(z)) \models P(f(f(a)))$$

- Model has to be refined on the conflict

Inst-Gen: Ground Abstraction and Selection

First-order clauses

$$\neg Q(f(x))$$

$$\neg P(f(f(y)))$$

$$P(f(z)) \vee Q(z)$$

Ground abstraction with $\perp$

$$\neg Q(f(\perp))$$

$$\neg P(f(f(\perp)))$$

$$P(f(\perp)) \vee Q(\perp)$$

- Select literals which are true in ground abstraction

Fail to extend ground model to first-order

$$\neg P(f(f(y))) \models \neg P(f(f(a)))$$

$$P(f(z)) \models P(f(f(a)))$$

- Model has to be refined on the conflict

Inst-Gen: Ground Abstraction and Selection

First-order clauses

$$\neg Q(f(x))$$

$$\neg P(f(f(y)))$$

$$P(f(z)) \vee Q(z)$$

Ground abstraction with $\perp$

$$\underline{\neg Q(f(\perp))}$$

$$\underline{\neg P(f(f(\perp)))}$$

$$\underline{P(f(\perp))} \vee Q(\perp)$$

- Select literals which are true in ground abstraction

Fail to extend ground model to first-order

$$\neg P(f(f(y))) \models \neg P(f(f(a)))$$

$$P(f(z)) \models P(f(f(a)))$$

- Model has to be refined on the conflict

Inst-Gen: Ground Abstraction and Selection

First-order clauses

$$\underline{\neg Q(f(x))}$$

$$\underline{\neg P(f(f(y))}$$

$$\underline{P(f(z))} \vee Q(z)$$

Ground abstraction with $\perp$

$$\underline{\neg Q(f(\perp))}$$

$$\underline{\neg P(f(f(\perp))}$$

$$\underline{P(f(\perp))} \vee Q(\perp)$$

- Select literals which are true in ground abstraction

Fail to extend ground model to first-order

$$\neg P(f(f(y))) \models \neg P(f(f(a))$$

$$P(f(z)) \models P(f(f(a))$$

- Model has to be refined on the conflict

Inst-Gen: Ground Abstraction and Selection

First-order clauses

$$\underline{\neg Q(f(x))}$$
$$\underline{\neg P(f(f(y))}$$
$$\underline{P(f(z))} \vee Q(z)$$

Ground abstraction with $\perp$

$$\underline{\neg Q(f(\perp))}$$
$$\underline{\neg P(f(f(\perp))}$$
$$\underline{P(f(\perp))} \vee Q(\perp)$$

- Select literals which are true in ground abstraction

Fail to extend ground model to first-order

$$\neg P(f(f(y))) \models \neg P(f(f(a))$$
$$P(f(z)) \models P(f(f(a))$$

- Model has to be refined on the conflict

Inst-Gen: Instance Generation Inference

Inst-Gen Inference

$$\frac{\neg P(f(f(y))) \qquad P(f(z)) \vee Q(z)}{\neg P(f(f(y))) \qquad P(f(f(y))) \vee Q(f(y))}\,[z \to f(y)]$$

- Inference with most general unifier on $\neg P(f(f(y)))$ and $P(f(z))$ which are are selected and complementary.

First-order clauses

$$\neg Q(f(x))$$
$$\neg P(f(f(y)))$$
$$P(f(z)) \;\vee\; Q(z)$$
$$P(f(f(u))) \;\vee\; Q(f(u))$$

Ground abstraction with $\perp$

$$\neg Q(f(\perp))$$
$$\neg P(f(f(\perp)))$$
$$P(f(\perp)) \;\vee\; Q(\perp)$$
$$P(f(f(\perp))) \;\vee\; Q(f(\perp))$$

Inst-Gen: Instance Generation Inference

Inst-Gen Inference

$$\frac{\neg P(f(f(y))) \qquad P(f(z)) \vee Q(z)}{\neg P(f(f(y))) \qquad P(f(f(y))) \vee Q(f(y))}\,[z \to f(y)]$$

- Inference with most general unifier on $\neg P(f(f(y)))$ and $P(f(z))$ which are are selected and complementary.

First-order clauses

$$\neg Q(f(x))$$
$$\neg P(f(f(y)))$$
$$P(f(z)) \ \vee\ Q(z)$$
$$P(f(f(u))) \ \vee\ Q(f(u))$$

Ground abstraction with $\perp$

$$\neg Q(f(\perp))$$
$$\neg P(f(f(\perp)))$$
$$P(f(\perp)) \ \vee\ Q(\perp)$$
$$P(f(f(\perp))) \ \vee\ Q(f(\perp))$$

Inst-Gen: Instance Generation Inference

Inst-Gen Inference

$$\frac{\neg P(f(f(y))) \qquad P(f(z)) \vee Q(z)}{\neg P(f(f(y))) \qquad P(f(f(y))) \vee Q(f(y))}\,[z \to f(y)]$$

- Inference with most general unifier on $\neg P(f(f(y)))$ and $P(f(z))$ which are are selected and complementary.

First-order clauses

$$\neg Q(f(x))$$
$$\neg P(f(f(y)))$$
$$P(f(z)) \ \vee \ Q(z)$$
$$P(f(f(u))) \ \vee \ Q(f(u))$$

Ground abstraction with $\perp$

$$\neg Q(f(\perp))$$
$$\neg P(f(f(\perp)))$$
$$P(f(\perp)) \ \vee \ Q(\perp)$$
$$P(f(f(\perp))) \ \vee \ Q(f(\perp))$$

Inst-Gen: Instance Generation Inference

Inst-Gen Inference

$$\frac{\neg P(f(f(y))) \qquad P(f(z)) \lor Q(z)}{\neg P(f(f(y))) \qquad P(f(f(y))) \lor Q(f(y))}\,[z \to f(y)]$$

- Inference with most general unifier on $\neg P(f(f(y)))$ and $P(f(z))$ which are are selected and complementary.

First-order clauses

$$\neg Q(f(x))$$

$$\neg P(f(f(y)))$$

$$P(f(z)) \;\lor\; Q(z)$$

$$P(f(f(u))) \;\lor\; Q(f(u))$$

Ground abstraction with $\perp$

$$\neg Q(f(\perp))$$

$$\neg P(f(f(\perp)))$$

$$P(f(\perp)) \;\lor\; Q(\perp)$$

$$P(f(f(\perp))) \;\lor\; Q(f(\perp))$$

Inst-Gen Modulo Equality

- Inst-Gen inference rule not sufficient
- Obvious step from Resolution to Superposition to generate instances is incomplete
- Set of literals of any size can be contradictory

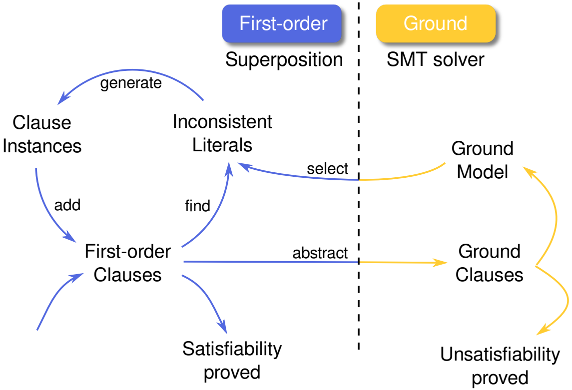
$$\{\, f(x) \not\simeq f(a) \,\}$$

$$\{\, f(x) \simeq a, \quad f(a) \not\simeq a \,\}$$

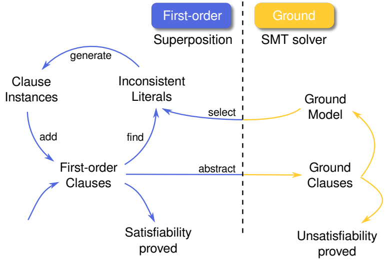
$$\{\, h(y) \simeq y, \quad f(h(x)) \simeq c, \quad f(a) \not\simeq c \,\}$$

- Labelled unit superposition calculus
- Instance generation from labels of contradictions
- Ground solver modulo equality (*QF_UF* solver)

The Inst-Gen-Eq Method



Efficient Ground Solving in Inst-Gen



Cooperation with SMT solver

- Incrementally add clauses
- Test unsatisfiability
- Query truth value of literals

Beyond the basics

- Global propositional subsumption
- Minimise changes to selection
- Auxiliary “soft” assertions

Statistics and experimental results

Global Propositional Subsumption

Generalise grounding by $\perp$ to a set of constants Σ_c, consider substitutions $\gamma \in \Omega$, e.g. $[x \to c_1, y \to c_2, \dots]$

Propositional Simplification

$$\frac{D \vee D'}{D'} \qquad \text{if } C_1\gamma_1, \dots C_k\gamma_k \models D\gamma$$

- Finding a minimal D' is linear in length of $D \vee D'$
- Individual ground constant for each variable
- Separate instance of Z3
- Approximation is sufficient:
  - consider only one $\gamma \in \Omega$ and
  - limit runtime of solver

Z3 Models and iProver Selection

- Semantic selection: in each clause one literal L such that $L\bot$ is true in the ground model
- Saturation process: Changing selection removes L and enters L', inferences with clause to be repeated

Tweak model to preserve selection

Local Is there is a model such that the previous selection for this clause can be kept?

Global Which model requires the least changes across all clauses to the current selection?

Inst-Gen(-Eq) calculus is complete for any model, hence approximate answers suffice

Soft Contraints and Unsatisfiable Cores

- Auxiliary literals for tracking purposes

  Proofs: Input clauses in unsatisfiable core

  Answers: Transform

  $$\exists x \ \varphi(x)$$

  to

  $$\exists x \ \varphi(x) \wedge answer(x)$$

- Soft constraints and unsat cores for incremental solving

  Bounded model checking: Enumerate states, transfer information from one bound to next

  Finite model finding: Enumerate domain constants

Two Incarnations of Z3

Satisfiability solver

- Witness unsatisfiability and select literals
- Full solving with model
- Grounding with $\perp$
- Tweak model to preserve previous selections (soft constraints, unsat cores)

Simplification solver

- Global propositional subsumption
- Fast and incomplete
- Unit propagation, bound number of decisions
- Grounding with $\perp$ and $[x \to c_1, y \to c_2, \ldots]$

Specifics of Ground Reasoning in iProver-Eq

- ground problems are typically simple $< 1s$
- frequent solver calls typically > 1000
- Incrementality: clauses are added inrementally, hundreds of thousands in some applications

Experimental Results

iProver-Eq with CVC3 vs. Z3 on
TPTP v5.2.0 problems with equations only (9,507 total)

Number of problems

	solved	only	faster	by 50%	by 100%
CVC3	2,468	87	663	57	30
Z3	2,510	129	1,718	551	317

- Ground model strongly influences first-order reasoning
- Problems for solvers are structurally simple
- Most effort on adaption of selection to model

iProver-Eq and Z3

Features used:

- Incrementally assert clauses
- Push and pop to find different model
- Check satisfiability, also with assumptions
- Evaluate literals in calculated model

Wishlist/To-Do:

- Access to learnt clauses
- Default decision value for literals
- Soft enforcing of truth values
- Fast solving for simplifications
- Use tactics to our advantage

Summary

Instantiation-based reasoning à la Inst-Gen and Inst-Gen-Eq

- Sound and complete first-order method
- Modulo equality and modulo theories
- SMT solver for ground reasoning

Future Work

- Improve efficiency of cooperation with solver
- Tune to applications
- First-order reasoning modulo theories