The University of Manchester

# Instantiation-based Methods for Equational Reasoning and Towards Theories Beyond

Christoph Sticksel

Joint work with Konstantin Korovin and Renate Schmidt

MPI Saarbrücken
14th January 2011

# Automated Reasoning for First-order Logic

- **First-order logic** Prove unsatisfiability of a set of clauses
- **Equational reasoning** Have a predicate $\simeq$ which is reflexive, symmetric, transitive and monotone
- **Reasoning modulo theories** Have a background theory (integer arithmetic, arrays, bitvectors)

In this talk:

- Paradigm of instantiation-based methods
- Inst-Gen calculus [Ganzinger and Korovin, 2003] with equality [Ganzinger and Korovin, 2004]
- Implemented in *iProver* and *iProver-Eq*

# Instantiation-based Methods: The Idea

**Herbrand Theorem**

Let $\varphi(\bar{x})$ be a quantifier free formula, then $\forall \bar{x}\, \varphi(\bar{x})$ is unsatisfiable if and only if there exist ground terms $\bar{t_1}, \ldots, \bar{t_n}$ such that $\bigwedge_i \varphi(\bar{t_i})$ is unsatisfiable.

A refutationally complete method:

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

Good news

- Propositional satisfiability (modulo equality) is decidable
- SAT solving techniques well explored
- SMT for quantifier-free formulae

# Instantiation-based Methods: The Main Problem

**Core question in instantiation-based reasoning**

How do we find a set of ground instances to witness first-order unsatisfiability?

- Easier if there are finitely many ground instances (Bernays-Schönfinkel)
- Harder the “more” ground instances there are, i.e. the more prolific the clause set is
- Calculi differ in the way instances are generated and how propositional solving is integrated

# Instantiation-based Methods: The Potential

- Decision procedure for Bernays-Schönfinkel
- Different search space than traditional methods
- Guided by model instead of logical conclusions
- Clause length remains unchanged when instantiating
- Resolution weak for propositional reasoning
- Employ SAT solving techniques
- A way to lift SMT to first-order?

# Some Implemented Instantiation-based Calculi

- Model Evolution: Darwin/E-Darwin
  (Baumgartner, Fuchs, Pelzer, Tinelli)
- Hyper Tableaux: E-KRHyper
  (Baumgartner, Furbach, Pelzer, Wernhard)
- Disconnection Calculus: DCTP
  (Billon, Letz, Stenz)
- Hyperlinking (OSHL): CLIN
  (Chu, Lee, Plaisted, Zhu)
- Inst-Gen: iProver, iProver-Eq
  (Ganzinger, Korovin, Sticksel)

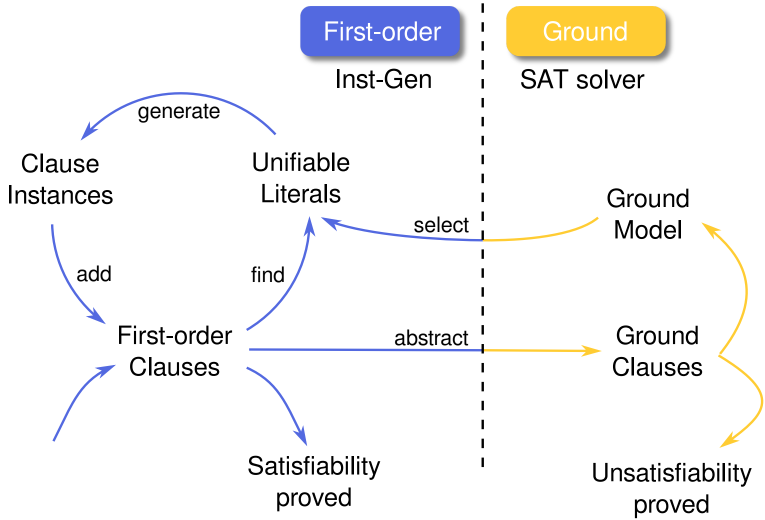
# Features of Inst-Gen

- Combination of first-order and ground reasoning
- Ground reasoning delegated to off-the-shelf solver

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

- Implemented in *iProver* and *iProver-Eq*
- [Korovin and Sticksel, IJCAR 2010] and [Korovin and 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 $\bot$**

$$\neg Q(f(\bot))$$

$$\neg P(f(f(\bot)))$$

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

- 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 $\bot$**

$$\neg Q(f(\bot))$$

$$\neg P(f(f(\bot)))$$

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

- 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 $\bot$**

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

- 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**

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

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

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

**Ground abstraction with $\bot$**

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

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

$$\underline{P(f(\bot))} \lor Q(\bot)$$

- 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

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

- Inference on $\neg P(f(f(y))$ and $P(f(z))$ which are are selected, unifiable 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

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

- Inference on $\neg P(f(f(y))$ and $P(f(z))$ which are are selected, unifiable 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 $\bot$

$\neg Q(f(\bot))$

$\neg P(f(f(\bot))$

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

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

# Inst-Gen: Instance Generation Inference

## Inst-Gen Inference

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

- Inference on $\neg P(f(f(y))$ and $P(f(z))$ which are are selected, unifiable 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

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

- Inference on $\neg P(f(f(y))$ and $P(f(z))$ which are are selected, unifiable 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 $\bot$

$$\neg Q(f(\bot))$$

$$\neg P(f(f(\bot))$$

$$P(f(\bot)) \;\vee\; Q(\bot)$$

$$P(f(f(\bot))) \;\vee\; Q(f(\bot))$$

# Inst-Gen Modulo Equality

- Unifiable complementary literal pairs not sufficient
- Set of literals of any size can be contradictory

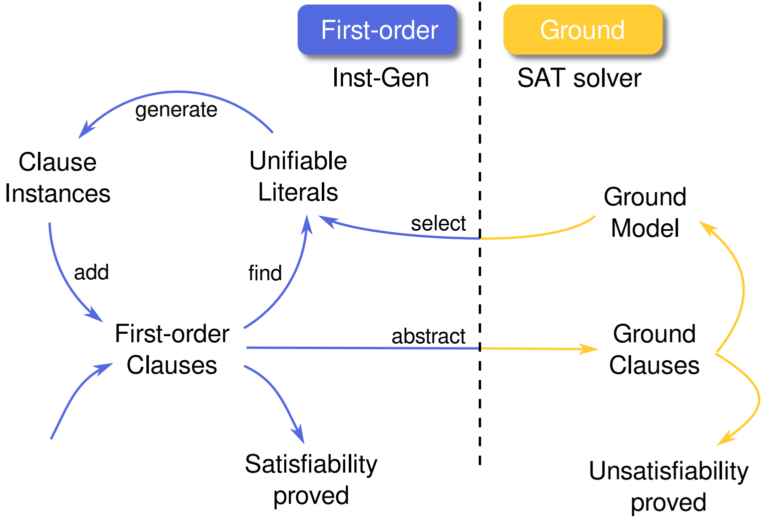
$$\{\, f(x) \not\simeq f(a) \,\}$$

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

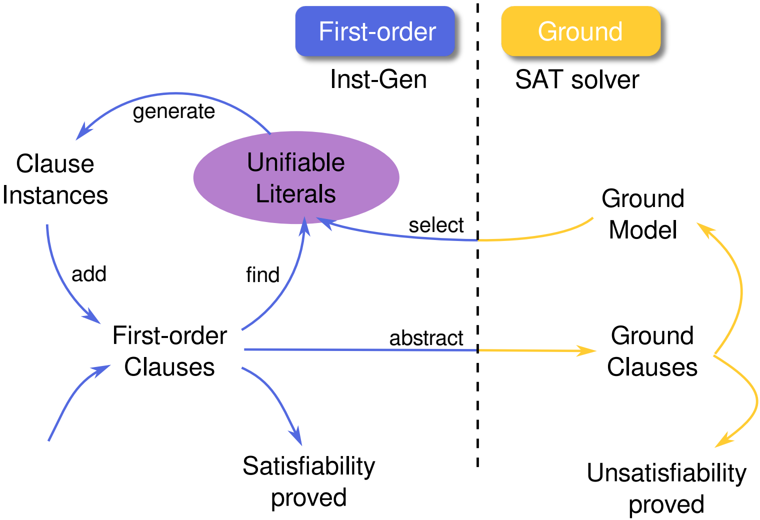
$$\{\, h(y) \simeq y, \quad f(h(x)) \simeq c, \quad f(a) \not\simeq c \,\}$$

- Obvious step from Resolution to Paramodulation to generate instances is incomplete
- Instance generation from superposition-style proofs instead of atomic Inst-Gen inference rule
- Ground solver modulo equality (SMT solver)

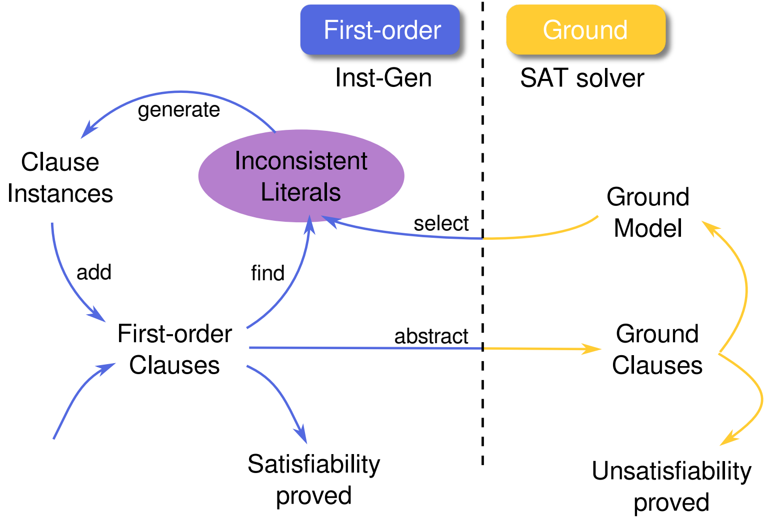
# The Inst-Gen Method



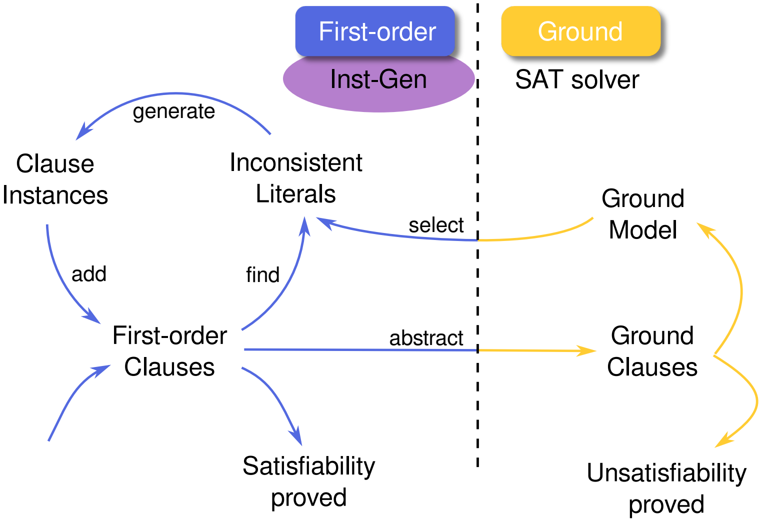
# From Inst-Gen to Inst-Gen-Eq



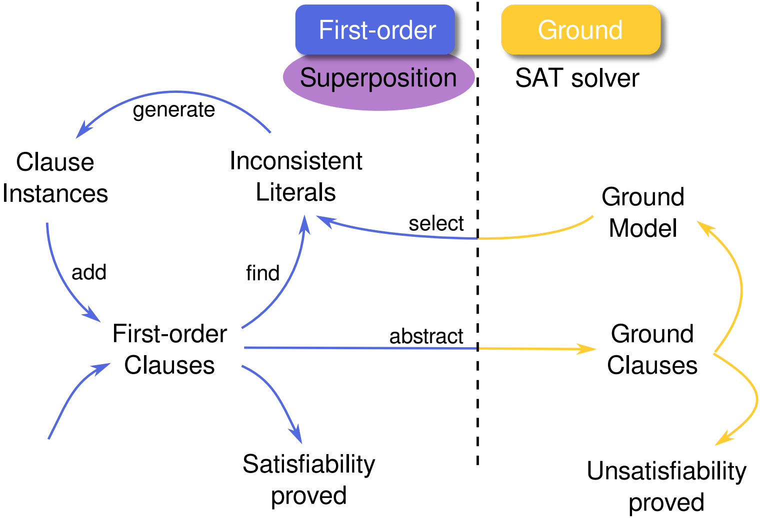
# From Inst-Gen to Inst-Gen-Eq



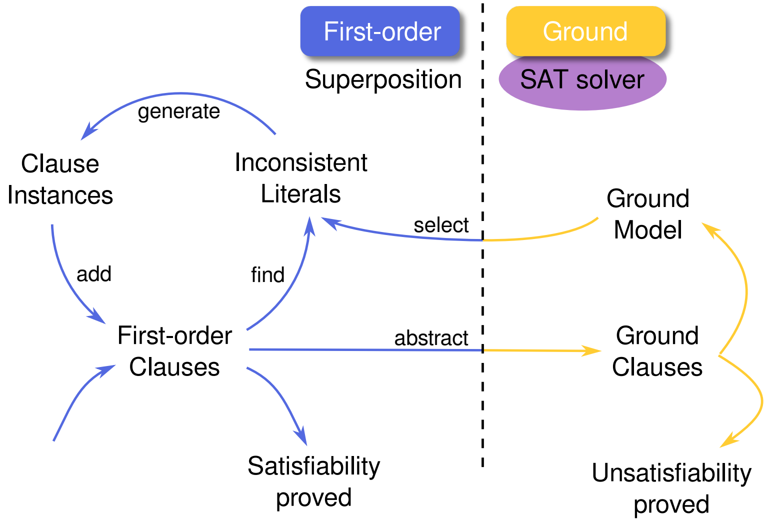
# From Inst-Gen to Inst-Gen-Eq



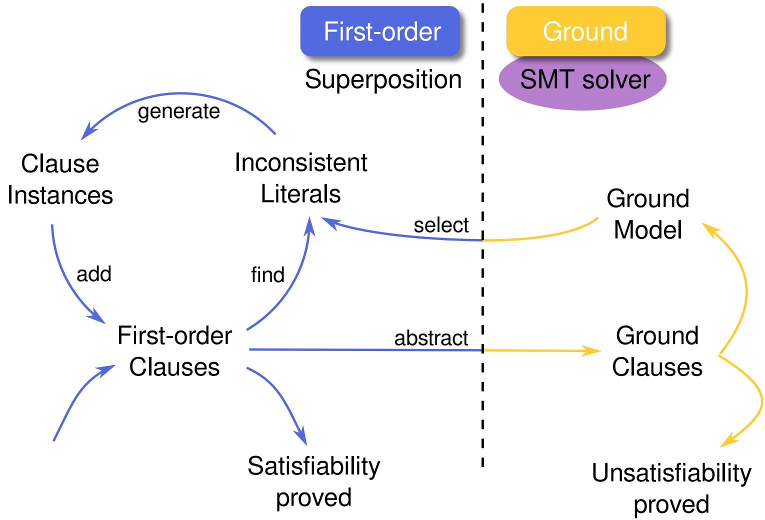
# From Inst-Gen to Inst-Gen-Eq



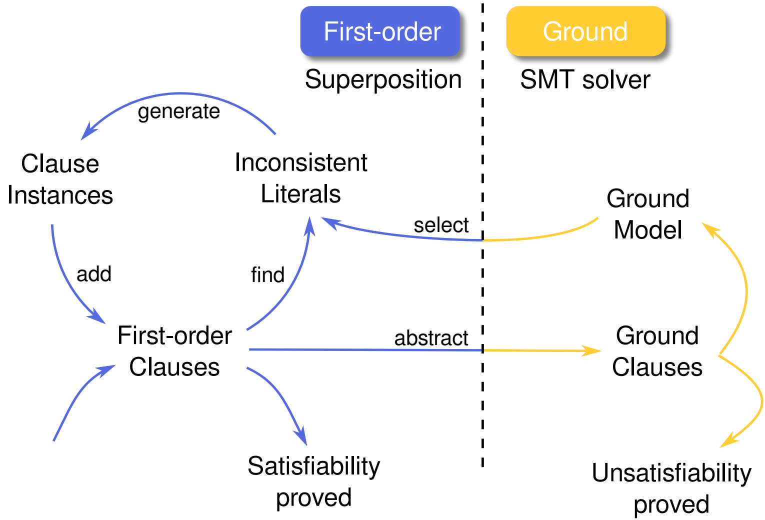
# From Inst-Gen to Inst-Gen-Eq



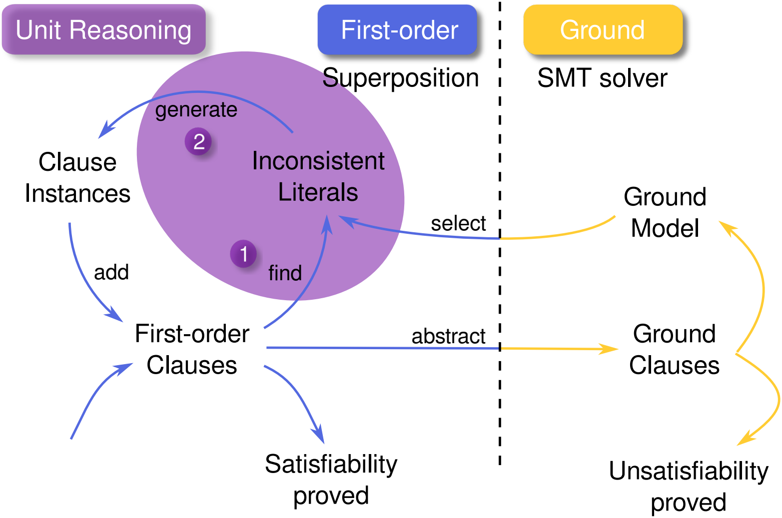
# From Inst-Gen to Inst-Gen-Eq



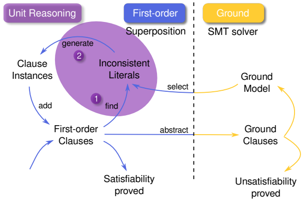
# The Inst-Gen-Eq Method



# The Inst-Gen-Eq Method



# Efficient Unit Reasoning with Selected Literals




Main problems

1. Find inconsistent literals with superposition reasoning
2. Generate clause instances from superposition proofs
3. All (non-redundant) proofs needed for completeness

Our solution

- Labelled Unit Superposition
  - Set labels
  - AND/OR tree labels
  - OBDD labels

# Inst-Gen-Eq: (1) Finding Inconsistencies

**First-order clauses**

$$f(x,y) \simeq f(y,x)$$

$$f(u,v) \not\simeq g(z) \ \lor \ u \simeq z$$

$$f(a,b) \simeq g(c)$$

$$a \not\simeq b$$

**Ground abstraction with ⊥**

$$\underline{f(\bot,\bot) \simeq f(\bot,\bot)}$$

$$\underline{f(\bot,\bot) \not\simeq g(\bot)} \ \lor \ \bot \simeq \bot$$

$$\underline{f(a,b) \simeq g(c)}$$

$$\underline{a \not\simeq b}$$

**Unit superposition proof: Selected literals inconsistent**

$$\dfrac{\dfrac{f(a,b) \simeq g(c) \quad \dfrac{f(x,y) \simeq f(y,x) \quad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}[u/x, v/y]}{g(c) \not\simeq g(z)}[a/v, b/u]}{\square}[c/z]$$

# Inst-Gen-Eq: (1) Finding Inconsistencies

## First-order clauses

$$f(x,y) \simeq f(y,x)$$

$$f(u,v) \not\simeq g(z) \ \vee \ u \simeq z$$

$$f(a,b) \simeq g(c)$$

$$a \not\simeq b$$

## Ground abstraction with $\bot$

$$\underline{f(\bot,\bot) \simeq f(\bot,\bot)}$$

$$\underline{f(\bot,\bot) \not\simeq g(\bot)} \ \vee \ \bot \simeq \bot$$

$$\underline{f(a,b) \simeq g(c)}$$

$$\underline{a \not\simeq b}$$

## Unit superposition proof: Selected literals inconsistent

$$\cfrac{\cfrac{f(a,b) \simeq g(c) \qquad \cfrac{f(x,y) \simeq f(y,x) \qquad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{g(c) \not\simeq g(z)}\,[a/v, b/u]}{\square}\,[c/z]$$

# Inst-Gen-Eq: (1) Finding Inconsistencies

**First-order clauses**

$$\underline{f(x,y) \simeq f(y,x)}$$

$$\underline{f(u,v) \not\simeq g(z)} \lor u \simeq z$$

$$\underline{f(a,b) \simeq g(c)}$$

$$\underline{a \not\simeq b}$$

**Ground abstraction with $\bot$**

$$\underline{f(\bot,\bot) \simeq f(\bot,\bot)}$$

$$\underline{f(\bot,\bot) \not\simeq g(\bot)} \lor \bot \simeq \bot$$

$$\underline{f(a,b) \simeq g(c)}$$

$$\underline{a \not\simeq b}$$

**Unit superposition proof: Selected literals inconsistent**

$$\dfrac{\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \qquad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{g(c) \not\simeq g(z)}\,[a/v, b/u]}{\square}\,[c/z]$$

# Inst-Gen-Eq: (1) Finding Inconsistencies

### First-order clauses

$$\underline{f(x,y) \simeq f(y,x)}$$

$$\underline{f(u,v) \not\simeq g(z)} \;\vee\; u \simeq z$$

$$\underline{f(a,b) \simeq g(c)}$$

$$\underline{a \not\simeq b}$$

### Ground abstraction with $\perp$

$$\underline{f(\perp,\perp) \simeq f(\perp,\perp)}$$

$$\underline{f(\perp,\perp) \not\simeq g(\perp)} \;\vee\; \perp \simeq \perp$$

$$\underline{f(a,b) \simeq g(c)}$$

$$\underline{a \not\simeq b}$$

### Unit superposition proof: Selected literals inconsistent

$$\dfrac{\dfrac{\dfrac{f(x,y) \simeq f(y,x) \quad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y] \qquad f(a,b) \simeq g(c)}{g(c) \not\simeq g(z)}\,[a/v, b/u]}{\Box}\,[c/z]$$

# Inst-Gen-Eq: (2) Generating Instances

## Unit superposition proof: Substitution extraction

$$\dfrac{\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \quad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{g(c) \not\simeq g(z)}\,[a/v, b/u]}{\square}\,[c/z]$$

## First-order clauses

$$\underline{f(x,y) \simeq f(y,x)}$$

$$\underline{f(u,v) \not\simeq g(z)} \;\vee\; u \simeq z$$

$$\underline{f(a,b) \simeq g(c)}$$

$$\underline{a \not\simeq b}$$

## New first-order instances

$$f(b,a) \simeq f(a,b)$$

$$f(b,a) \not\simeq g(c) \;\vee\; b \simeq c$$

# Inst-Gen-Eq: (2) Generating Instances

## Unit superposition proof: Substitution extraction

$$\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \qquad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{\dfrac{g(c) \not\simeq g(z)}{\square}\,[c/z]}\,[a/v, b/u]$$

## First-order clauses

$$\underline{f(x,y) \simeq f(y,x)}$$

$$\underline{f(u,v) \not\simeq g(z)} \;\vee\; u \simeq z$$

$$\underline{f(a,b) \simeq g(c)}$$

$$\underline{a \not\simeq b}$$

## New first-order instances

$$f(b,a) \simeq f(a,b)$$

$$f(b,a) \not\simeq g(c) \;\vee\; b \simeq c$$

# Inst-Gen-Eq: (2) Generating Instances

## Unit superposition proof: Substitution extraction

$$
\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \qquad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{\dfrac{g(c) \not\simeq g(z)}{\square}\,[c/z]}\,[a/v, b/u]
$$

## First-order clauses

$\underline{f(x,y) \simeq f(y,x)}$

$\underline{f(u,v) \not\simeq g(z)} \;\vee\; u \simeq z$

$\underline{f(a,b) \simeq g(c)}$

$\underline{a \not\simeq b}$

## New first-order instances

$f(b,a) \simeq f(a,b)$

$f(b,a) \not\simeq g(c) \;\vee\; b \simeq c$

# Inst-Gen-Eq: (2) Generating Instances

## Unit superposition proof: Substitution extraction

$$\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \qquad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{\dfrac{g(c) \not\simeq g(z)}{\square}\,[c/z]}\,[a/v, b/u]$$

## First-order clauses

$\underline{f(x,y) \simeq f(y,x)}$

$\underline{f(u,v) \not\simeq g(z)} \lor u \simeq z$

$\underline{f(a,b) \simeq g(c)}$

$\underline{a \not\simeq b}$

## New first-order instances

$f(b,a) \simeq f(a,b)$

$f(b,a) \not\simeq g(c) \lor b \simeq c$

# Inst-Gen-Eq: (2) Generating Instances

## Unit superposition proof: Substitution extraction

$$\dfrac{\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \quad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{g(c) \not\simeq g(z)}\,[a/v, b/u]}{\square}\,[c/z]$$

## First-order clauses

$\underline{f(x,y) \simeq f(y,x)}$

$\underline{f(u,v) \not\simeq g(z)} \;\vee\; u \simeq z$

$\underline{f(a,b) \simeq g(c)}$

$\underline{a \not\simeq b}$

## New first-order instances

$f(b,a) \simeq f(a,b)$

$f(b,a) \not\simeq g(c) \;\vee\; b \simeq c$

# Inst-Gen-Eq: (3) Instances from Each Inconsistency

**Proof of inconsistency (1)**

$$\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \qquad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{\dfrac{g(c) \not\simeq g(z)}{\square}\,[c/z]}\,[a/v, b/u]$$

**Proof of inconsistency (2)**

$$\dfrac{\dfrac{f(a,b) \simeq g(c) \qquad f(u,v) \not\simeq g(z)}{g(c) \not\simeq g(z)}\,[a/u, b/v]}{\square}\,[c/z]$$

**Instances from proof (1)**

$$f(b,a) \simeq f(a,b)$$
$$f(b,a) \not\simeq g(c) \lor b \simeq c$$

**Instances from proof (2)**

$$f(a,b) \not\simeq g(c) \lor a \simeq c$$

# Inst-Gen-Eq: (3) Instances from Each Inconsistency

## Proof of inconsistency (1)

$$\dfrac{\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \qquad \color{magenta}{f(u,v) \not\simeq g(z)}}{\color{magenta}{f(v,u) \not\simeq g(z)}}\,[u/x, v/y]}{g(c) \not\simeq g(z)}\,[a/v, b/u]}{\square}\,[c/z]$$

## Proof of inconsistency (2)

$$\dfrac{\dfrac{f(a,b) \simeq g(c) \qquad f(u,v) \not\simeq g(z)}{g(c) \not\simeq g(z)}\,[a/u, b/v]}{\square}\,[c/z]$$

## Instances from proof (1)

$$f(b,a) \simeq f(a,b)$$
$$f(b,a) \not\simeq g(c) \vee b \simeq c$$

## Instances from proof (2)

$$f(a,b) \not\simeq g(c) \vee a \simeq c$$

# Inst-Gen-Eq: (3) Instances from Each Inconsistency

## Proof of inconsistency (1)

$$\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \qquad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{\dfrac{g(c) \not\simeq g(z)}{\square}\,[c/z]}\,[a/v, b/u]$$

## Proof of inconsistency (2)

$$\dfrac{f(a,b) \simeq g(c) \qquad f(u,v) \not\simeq g(z)}{\dfrac{g(c) \not\simeq g(z)}{\square}\,[c/z]}\,[a/u, b/v]$$

## Instances from proof (1)

$$f(b,a) \simeq f(a,b)$$
$$f(b,a) \not\simeq g(c) \vee b \simeq c$$

## Instances from proof (2)

$$f(a,b) \not\simeq g(c) \vee a \simeq c$$

# Inst-Gen-Eq: (3) Instances from Each Inconsistency

## Proof of inconsistency (1)

$$\dfrac{f(a,b) \simeq g(c) \qquad \dfrac{f(x,y) \simeq f(y,x) \qquad f(u,v) \not\simeq g(z)}{f(v,u) \not\simeq g(z)}\,[u/x, v/y]}{\dfrac{g(c) \not\simeq g(z)}{\square}\,[c/z]}\,[a/v, b/u]$$

## Proof of inconsistency (2)

$$\dfrac{\dfrac{f(a,b) \simeq g(c) \qquad f(u,v) \not\simeq g(z)}{g(c) \not\simeq g(z)}\,[a/u, b/v]}{\square}\,[c/z]$$

## Instances from proof (1)

$f(b,a) \simeq f(a,b)$

$f(b,a) \not\simeq g(c) \vee b \simeq c$

## Instances from proof (2)

$f(a,b) \not\simeq g(c) \vee a \simeq c$

# The Labelling Approach

- Distinguish literal variants by labels
- Explicit *merging* inference to combine variants
  Components
- *Closure* $C \cdot \theta$: clause $C$ and substitution $\theta$
- Initially $\{C \cdot []\} : L$ where $L$ is selected in $C$
- Label of contradiction $\Box$ contains instances to be added

Advantages

- Eager extraction of instances after each inference in label
- Uniform treatment of literal variants
- Preserve proof structure for redundancy elimination

# Inference Rules in Labelled Unit Superposition

Superposition

$$\frac{\mathcal{T} : l \simeq r \qquad \mathcal{T}' : L[l']}{(\mathcal{T} \sqcap \mathcal{T}')\sigma : L[r]\sigma} \; (\sigma)$$

$\sigma$ is mgu of $l$ and $l'$

Variant merging

$$\frac{\mathcal{T} : L \qquad \mathcal{T}' : L'}{\mathcal{T} \sqcup \mathcal{T}'\sigma : L} \; (\sigma)$$

$L = L'\sigma$, $\sigma$ is a renaming

Equality resolution

$$\frac{\mathcal{T} : (l \not\simeq r)}{\mathcal{T}\sigma : \Box} \; (\sigma)$$

$\sigma$ is mgu of $l$ and $r$

- No labels in side conditions
- $\sqcap$ and $\sqcup$ dependant on implementation of labels
- Label $\mathcal{T}$ is either a set, an AND/OR tree or an OBDD

# Set Labelled Unit Superposition

- Label is a set of closures
- Set union $\cup$ in both merging $\sqcup$ and superposition $\sqcap$

**Superposition**

$$\frac{\{C \cdot []\} : f(x,y) \simeq f(y,x) \quad \{D \cdot []\} : f(u,v) \not\simeq g(z)}{\{C \cdot [u/x, v/y], D \cdot []\} : f(v,u) \not\simeq g(z)} \; [u/x, v/y]$$

**Merging $f(u,v) \not\simeq g(z)$ and $f(v,u) \not\simeq g(z)$ with $[u/v, v/u]$**

$$\{D \cdot [], C \cdot [v/x, u/y], D \cdot [u/v, v/u]\} : f(u,v) \not\simeq g(z)$$

**Label of the contradiction $\square$**

$$\{D \cdot [a/u, b/v, c/z], E \cdot [], C \cdot [b/x, a/y], D \cdot [b/u, a/v, c/z]\}$$

# Set Labelled Unit Superposition

- Label is a set of closures
- Set union $\cup$ in both merging $\sqcup$ and superposition $\sqcap$

**Superposition**

$$\frac{\{C \cdot []\} : f(x,y) \simeq f(y,x) \quad \{D \cdot []\} : f(u,v) \not\simeq g(z)}{\{C \cdot [u/x, v/y], D \cdot []\} : f(v,u) \not\simeq g(z)} \; [u/x, v/y]$$

**Merging $f(u,v) \not\simeq g(z)$ and $f(v,u) \not\simeq g(z)$ with $[u/v, v/u]$**

$$\{D \cdot [], C \cdot [v/x, u/y], D \cdot [u/v, v/u]\} : f(u,v) \not\simeq g(z)$$

**Label of the contradiction $\square$**

$$\{D \cdot [a/u, b/v, c/z], E \cdot [], C \cdot [b/x, a/y], D \cdot [b/u, a/v, c/z]\}$$

# Set Labelled Unit Superposition

- Label is a set of closures
- Set union $\cup$ in both merging $\sqcup$ and superposition $\sqcap$

**Superposition**

$$\frac{\{C \cdot []\} : f(x,y) \simeq f(y,x) \quad \{D \cdot []\} : f(u,v) \not\simeq g(z)}{\{C \cdot [u/x, v/y], D \cdot []\} : f(v,u) \not\simeq g(z)} \; [u/x, v/y]$$

**Merging $f(u,v) \not\simeq g(z)$ and $f(v,u) \not\simeq g(z)$ with $[u/v, v/u]$**

$$\{D \cdot [], C \cdot [v/x, u/y], D \cdot [u/v, v/u]\} : f(u,v) \not\simeq g(z)$$

**Label of the contradiction $\Box$**

$$\{D \cdot [a/u, b/v, c/z], E \cdot [], C \cdot [b/x, a/y], D \cdot [b/u, a/v, c/z]\}$$

# Set Labelled Unit Superposition

- Label is a set of closures
- Set union $\cup$ in both merging $\sqcup$ and superposition $\sqcap$

**Superposition**

$$\frac{\{C \cdot []\} : f(x,y) \simeq f(y,x) \quad \{D \cdot []\} : f(u,v) \not\simeq g(z)}{\{C \cdot [u/x, v/y], D \cdot []\} : f(v,u) \not\simeq g(z)} \; [u/x, v/y]$$

**Merging $f(u,v) \not\simeq g(z)$ and $f(v,u) \not\simeq g(z)$ with $[u/v, v/u]$**

$$\{D \cdot [], C \cdot [v/x, u/y], D \cdot [u/v, v/u]\} : f(u,v) \not\simeq g(z)$$

**Label of the contradiction $\Box$**

$$\{D \cdot [a/u, b/v, c/z], E \cdot [], C \cdot [b/x, a/y], D \cdot [b/u, a/v, c/z]\}$$

# Set Labelled Unit Superposition

- Label is a set of closures
- Set union $\cup$ in both merging $\sqcup$ and superposition $\sqcap$

**Superposition**

$$\frac{\{C \cdot []\} : f(x,y) \simeq f(y,x) \quad \{D \cdot []\} : f(u,v) \not\simeq g(z)}{\{C \cdot [u/x, v/y], D \cdot []\} : f(v,u) \not\simeq g(z)}\ [u/x, v/y]$$

**Merging $f(u,v) \not\simeq g(z)$ and $f(v,u) \not\simeq g(z)$ with $[u/v, v/u]$**

$$\{D \cdot [], C \cdot [v/x, u/y], D \cdot [u/v, v/u]\} : f(u,v) \not\simeq g(z)$$

**Label of the contradiction $\Box$**

$$\{D \cdot [a/u, b/v, c/z], E \cdot [], C \cdot [b/x, a/y], D \cdot [b/u, a/v, c/z]\}$$

# Set Labelled Unit Superposition

- Label is a set of closures
- Set union $\cup$ in both merging $\sqcup$ and superposition $\sqcap$

**Superposition**

$$\frac{\{C \cdot []\} : f(x,y) \simeq f(y,x) \quad \{D \cdot []\} : f(u,v) \not\simeq g(z)}{\{C \cdot [u/x, v/y], D \cdot []\} : f(v,u) \not\simeq g(z)} \; [u/x, v/y]$$

**Merging $f(u,v) \not\simeq g(z)$ and $f(v,u) \not\simeq g(z)$ with $[u/v, v/u]$**

$$\{D \cdot [], C \cdot [v/x, u/y], D \cdot [u/v, v/u]\} : f(u,v) \not\simeq g(z)$$

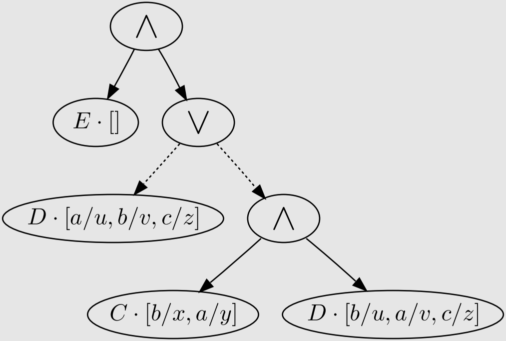
**Label of the contradiction $\square$**

$$\{D \cdot [a/u, b/v, c/z], E \cdot [], E \cdot [], C \cdot [b/x, a/y], D \cdot [b/u, a/v, c/z]\}$$

# Tree Labelled Unit Superposition

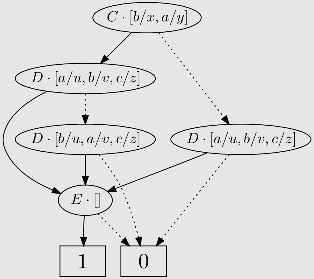
- Preserve Boolean structure of proofs
- Closure is a propositional variable in an AND/OR tree
- Conjunction $\wedge$ in superposition, disjunction $\vee$ in merging

## Label of the Contradiction $\Box$



# OBDD Labelled Unit Superposition

**Label of the contradiction □**



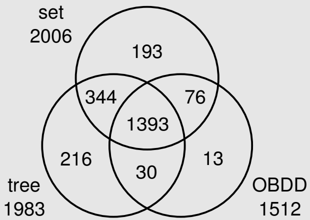

Disadvantages of trees

- Not produced in normal form
- Sequence of inferences determines shape
- Potential growth *ad infinitum*

- OBDD as normal form
- Maintenance effort
- Reordering required

# Evaluation: Sets vs. Trees vs. OBDDs

**Number of solved problems**




**Features**

| | Normal form | Precise elim. |
|---|---|---|
| Sets | yes | no |
| Trees | no | yes |
| OBDDs | yes | yes |

- TPTP v4.0.1
- Equational problems only

# Towards Theory Reasoning

- Superposition and rewriting approaches to reasoning modulo theories exist
- Unit reasoning modulo theory $\mathcal{T}$ on selected literals to generate instances:

$$L_1, \ldots, L_n \models_{\mathcal{T}} \Box$$

  find substitutions $\sigma_i$ such that

$$L_1\sigma_1\bot, \ldots, L_n\sigma_n\bot \models_{\mathcal{T}} \Box$$

- Use ground solver modulo $\mathcal{T}$
- Satisfiability of ground abstraction may be undecidable

# Lemma Generation from Labels

- Relax requirement on ground solver:

$$L_1\sigma_1\bot, \ldots, L_n\sigma_n\bot \models_{\mathcal{T}} \Box$$

- Add lemma to preclude this
- Ground solver modulo a weaker theory than unit reasoning
- More burden on unit reasoning: selected literals can be ground inconsistent modulo $\mathcal{T}$

Encouraging results

- iProver-Eq with SAT solver
- Generate lemma from tree or OBDD label
- Incomplete with set labels

# Summary

Instantiation-based reasoning à la Inst-Gen

Contributions in current work

- Labelled unit superposition for efficient unit reasoning
- Different label structures: sets, trees, OBDDs
- Implementation in *iProver-Eq*
- Evaluation on TPTP v4.0.1

Future Work

- Hybrid labels
- Unit calculi for theory reasoning
- Lemma generation