Complete Instantiation for Quantified Formulas in SMT

CAV 2009

Yeting Ge
New York University

Leonardo de Moura
Microsoft Research

Satisfiability Modulo Theories (SMT)

$$a > 3, (a = b \vee a = b + 1), f(a) = 0, f(b) = 1$$

Many Applications

- Dynamic symbolic execution (DART)
- Extended static checking
- Test-case generation
- Bounded model checking (BMC)
- Equivalence checking
- …

What is a Theory?

A theory T is a set of sentences.

F is satisfiable modulo T

iff

$T \cup F$ is satisfiable.

Theory: Examples

- Array Theory:
  $\forall a,i,v$: read(write(a,i,v), i) = v
  $\forall a,i,v$: i = j $\vee$ read(write(a,i,v), j) = read(a,j)

- Linear Arithmetic
- Bit-vectors
- Inductive datatypes
- …

SMT: Example

$$a > 3,\ (a = b \vee a = b + 1),\ f(a) = 0,\ f(b) = 1$$

f,g,h	Uninterpreted functions
a,b,c	Uninterpreted constants
+,-,<,≤,0,1,…	Interpreted symbols

SMT: Example

$a > 3, (a = b \vee a = b + 1), f(a) = 0, f(b) = 1$

Model/Structure:

$a \rightarrow 4$

$b \rightarrow 3$

$f \rightarrow \{ 4 \rightarrow 0,\ 3 \rightarrow 1,\ \ldots \}$

SMT: Example

$a > 3, (a = b \lor a = b + 1), f(a) = 0, f(b) = 1$

Model M:

$M(a) = 4$

$M(b) = 3$

$M(f) = \{ 4 \to 0, 3 \to 1, \ldots \}$

SMT Solvers

Many SMT Solvers:

- Barcelogic, Beaver, Boolector,
- CVC3, MathSAT, OpenSMT,
- Sateen, Yices, Z3, …

They are very efficient for quantifier-free formulas.

Many applications need quantifiers

- Modeling the runtime

$$\forall\, h,o,f:$$
$$\quad IsHeap(h) \land o \neq null \land read(h, o, alloc) = t$$
$$\quad \Rightarrow$$
$$\quad read(h,o, f) = null \lor read(h, read(h,o,f),alloc) = t$$

Many applications need quantifiers

- Modeling the runtime
- User provided assertions

  $\forall$ i,j: i $\leq$ j $\Rightarrow$ read(a,i) $\leq$ read(b,j)

Many applications need quantifiers

- Modeling the runtime
- User provided assertions
- Unsupported theories
  - $\forall\ x\colon p(x,x)$
  - $\forall\ x,y,z\colon p(x,y),\ p(y,z) \Rightarrow p(x,z)$
  - $\forall\ x,y\colon p(x,y),\ p(y,x) \Rightarrow x = y$

Many applications need quantifiers

- Modeling the runtime
- User provided assertions
- Unsupported theories
- Solver must be fast in satisfiable instances.

We want to find bugs!

Many Approaches

- Superposition Calculus + SMT.
- Instantiation Based Methods
  - Implemented on top of “regular” SMT solvers.
  - Heuristic quantifier instantiation (E-Matching).
  - Complete quantifier instantiation.

Instantiation Based Methods: Related work

- Bernays-Schönfinkel class.
- Stratified Many-Sorted Logic.
- Array Property Fragment.
- Local theory extensions.

Simplifying Assumption: CNF

$\forall x_1, x_2 : \neg p(x_1, x_2) \vee f(x_1) = f(x_2) + 1,$
$p(a,b),\ a < b + 1$

Simplifying Assumption: CNF

$\neg p(x_1, x_2) \lor f(x_1) = f(x_2) + 1,$

$p(a,b),\ a < b + 1$

Essentially uninterpreted fragment

- Variables appear only as arguments of uninterpreted symbols.

$$f(g(x_1) + a) < g(x_1) \vee h(f(x_1), x_2) = 0$$

$$f(x_1+x_2) \leq f(x_1) + f(x_2)$$

Basic Idea

Given a set of formulas F,
build an equisatisfiable set of quantifier-free formulas F*

"Domain" of f is the set of ground terms A_f

$t \in A_f$ if there is a ground term f(t)

Suppose

1. We have a clause C[f(x)] containing f(x).
2. We have f(t).

➔

Instantiate x with t: C[f(t)].

Example

F **F***

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$

$g(f(x_1),b) + 1 \leq f(x_1),$

$h(c) = 1,$

$f(a) = 0$

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$

$g(f(x_1),b) + 1 \leq f(x_1),$

$h(c) = 1,$

$f(a) = 0$

F*

$h(c) = 1,$

$f(a) = 0$

Copy quantifier-free formulas

"Domains":

A_f: { a }

A_g: { }

A_h: { c }

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$

$g(f(x_1),b) + 1 \leq f(\color{red}{x_1}),$

$h(c) = 1,$

$f(a) = 0$

F*

$h(c) = 1,$

$f(\color{red}{a}) = 0,$

"Domains":

$A_f : \{ \color{red}{a} \}$

$A_g : \{ \}$

$A_h : \{ c \}$

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$

$g(f(x_1),b) + 1 \leq f(x_1),$

$h(c) = 1,$

$f(a) = 0$

F*

$h(c) = 1,$

$f(a) = 0,$

$g(f(a),b) + 1 \leq f(a)$

"Domains":

$A_f : \{ a \}$

$A_g : \{ [f(a), b] \}$

$A_h : \{ c \}$

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$
$g(f(x_1),b) + 1 \leq f(x_1),$
$h(c) = 1,$
$f(a) = 0$

$\Rightarrow$

F*

$h(c) = 1,$
$f(a) = 0,$
$g(f(a),b) + 1 \leq f(a),$

"Domains":

$A_f : \{ a \}$
$A_g : \{ [f(a), b] \}$
$A_h : \{ c \}$

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$

$g(f(x_1),b) + 1 \leq f(x_1),$

$h(c) = 1,$

$f(a) = 0$

F*

$h(c) = 1,$

$f(a) = 0,$

$g(f(a),b) + 1 \leq f(a),$

$g(f(a), b) = 0 \vee h(b) = 0$

"Domains":

$A_f : \{ a \}$

$A_g : \{ [f(a), b] \}$

$A_h : \{ c, b \}$

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$

$g(f(x_1),b) + 1 \leq f(x_1),$

$h(c) = 1,$

$f(a) = 0$

F*

$h(c) = 1,$

$f(a) = 0,$

$g(f(a),b) + 1 \leq f(a),$

$g(f(a), b) = 0 \vee h(b) = 0$

“Domains”:

$A_f : \{ a \}$

$A_g : \{ [f(a), b] \}$

$A_h : \{ c, b \}$

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$
$g(f(x_1),b) + 1 \leq f(x_1),$
$h(c) = 1,$
$f(a) = 0$

F*

$h(c) = 1,$
$f(a) = 0,$
$g(f(a),b) + 1 \leq f(a),$
$g(f(a), b) = 0 \vee h(b) = 0,$
$g(f(a), c) = 0 \vee h(c) = 0$

"Domains":

$A_f : \{ a \}$
$A_g : \{ [f(a), b], [f(a), c] \}$
$A_h : \{ c, b \}$

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$
$g(f(x_1),b) + 1 \leq f(x_1),$
$h(c) = 1,$
$f(a) = 0$

$\Rightarrow$

F*

$h(c) = 1,$
$f(a) = 0,$
$g(f(a),b) + 1 \leq f(a),$
$g(f(a), b) = 0 \vee h(b) = 0,$
$g(f(a), c) = 0 \vee h(c) = 0$

$\Downarrow$

M

$a \rightarrow 2, b \rightarrow 2, c \rightarrow 3$
$f \rightarrow \{ 2 \rightarrow 0, \ldots\}$
$h \rightarrow \{ 2 \rightarrow 0, 3 \rightarrow 1, \ldots\}$
$g \rightarrow \{ [0,2]\rightarrow -1, [0,3]\rightarrow 0, \ldots\}$

Basic Idea (cont.)

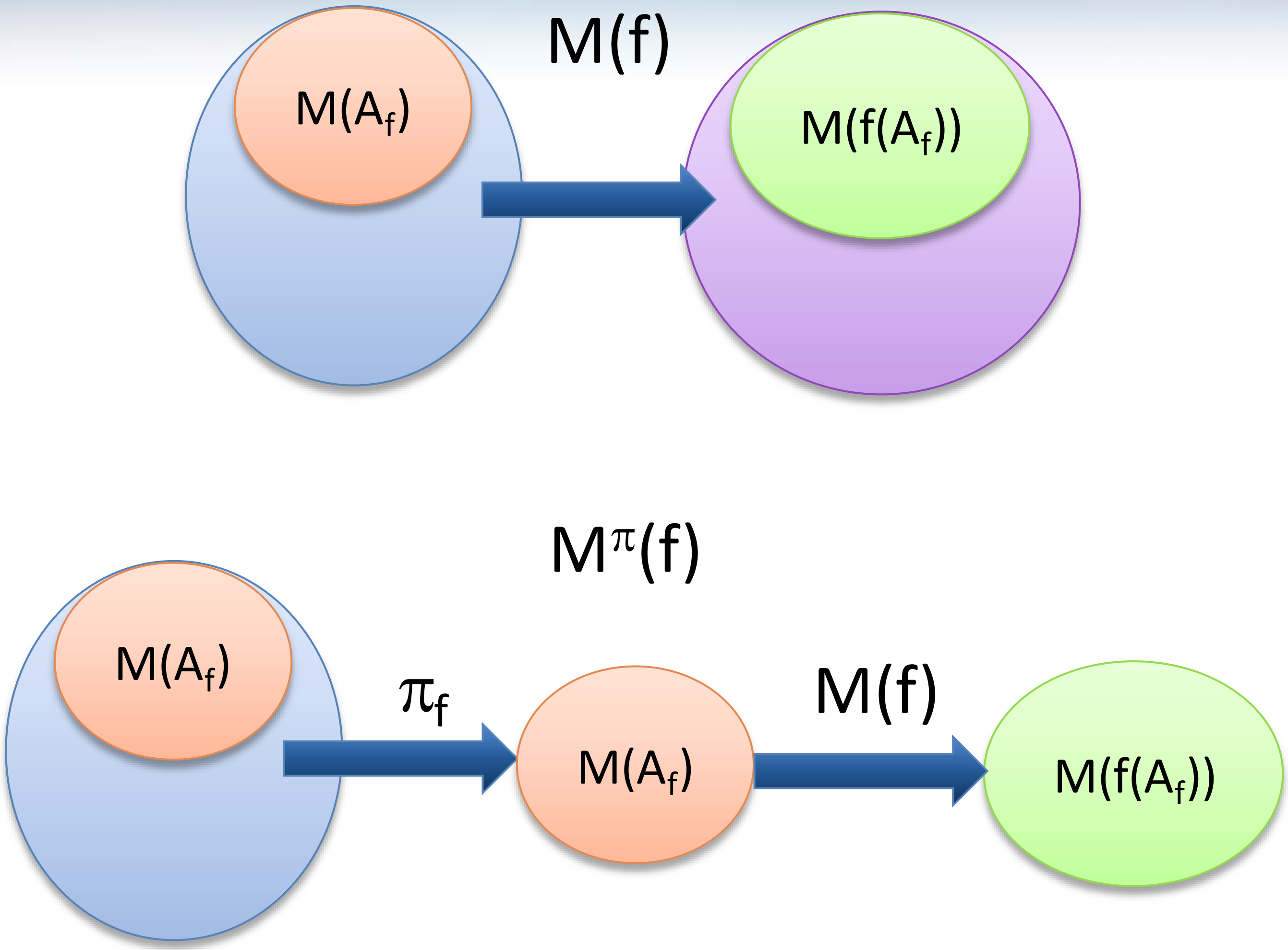
Given a model M for F^*,
Build a model M^π for F

Define a projection function π_f s.t.
range of π_f is $M(A_f)$, and
$\pi_f(v) = v \quad \text{if} \quad v \in M(A_f)$

Then,
$M^\pi(f)(v) = M(f)(\pi_f(v))$

Microsoft Research

Basic Idea (cont.)



Microsoft Research

Basic Idea (cont.)

Given a model M for F^*,
Build a model M^π for F

In our example, we have: $h(b)$ and $h(c)$
$\rightarrow A_h = \{ b, c \}$, and $M(A_h) = \{ 2, 3 \}$

$$\pi_h = \{ 2 \rightarrow 2, 3 \rightarrow 3, \text{else} \rightarrow 3 \}$$

$$\begin{array}{ccc} M(h) & & M^\pi(h) \\ \{ 2 \rightarrow 0, 3 \rightarrow 1, \ldots\} & \Rightarrow & \{ 2 \rightarrow 0, 3 \rightarrow 1, \text{else} \rightarrow 1\} \end{array}$$

$$M^\pi(h) = \lambda x.\ \text{if}(x=2, 0, 1)$$

Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$
$g(f(x_1),b) + 1 \leq f(x_1),$
$h(c) = 1,$
$f(a) = 0$

F*

$h(c) = 1,$
$f(a) = 0,$
$g(f(a),b) + 1 \leq f(a),$
$g(f(a), b) = 0 \vee h(b) = 0,$
$g(f(a), c) = 0 \vee h(c) = 0$

M

$a \to 2, b \to 2, c \to 3$
$f \to \{ 2 \to 0, \ldots\}$
$h \to \{ 2 \to 0, 3 \to 1, \ldots\}$
$g \to \{ [0,2] \to -1, [0,3] \to 0, \ldots\}$

M^π

$a \to 2, b \to 2, c \to 3$
$f \to \lambda x.\ 2$
$h \to \lambda x.\ if(x=2, 0, 1)$
$g \to \lambda x,y.\ if(x=0 \wedge y=2, -1, 0)$

Example: Model Checking

M^π

$a \to 2,\ b \to 2,\ c \to 3$

$f \to \lambda x.\ 2$

$h \to \lambda x.\ \text{if}(x=2,\ 0,\ 1)$

$g \to \lambda x,y.\ \text{if}(x=0 \wedge y=2, -1,\ 0)$

Does M^π satisfies?

$\forall x_1, x_2 : g(x_1, x_2) = 0 \vee h(x_2) = 0$

$\forall x_1, x_2\text{: if}(x_1=0 \wedge x_2=2,-1,0) = 0 \vee \text{if}(x_2=2,0,1) = 0$ is valid

$\exists x_1, x_2\text{: if}(x_1=0 \wedge x_2=2,-1,0) \neq 0 \wedge \text{if}(x_2=2,0,1) \neq 0$ is unsat

$\text{if}(s_1=0 \wedge s_2=2,-1,0) \neq 0 \wedge \text{if}(s_2=2,0,1) \neq 0$ is unsat

Why does it work?

Suppose M^{π} does not satisfy $C[f(x)]$.

Then for some value v,
$M^{\pi}\{x \rightarrow v\}$ falsifies $C[f(x)]$.

$M^{\pi}\{x \rightarrow \pi_f(v)\}$ also falsifies $C[f(x)]$.

But, there is a term $t \in A_f$ s.t. $M(t) = \pi_f(v)$
Moreover, we instantiated $C[f(x)]$ with t.

So, M must not satisfy $C[f(t)]$.
Contradiction: M is a model for F^*.

Refinement 1: Lazy construction

- F^* may be very big (or infinite).
- Lazy-construction
  - Build F^* incrementally, F^* is the limit of the sequence

$$F^0 \subset F^1 \subset \ldots \subset F^k \subset \ldots$$

  - If F^k is unsat then F is unsat.
  - If F^k is sat, then build (candidate) M^π
  - If M^π satisfies all quantifiers in F then return sat.

Refinement 2: Model-based instantiation

Suppose M^π does not satisfy a clause $C[f(x)]$ in F.

Add an instance $C[f(t)]$ which "blocks" this spurious model.
Issue: how to find t?

Use model checking,
and the "inverse" mapping π_f^{-1} from values to terms (in A_f).
$\pi_f^{-1}(v) = t$ if $M^\pi(t) = \pi_f(v)$

Model-based instantiation: Example

F

$\forall x_1: f(x_1) < 0,$
$f(a) = 1,$
$f(b) = -1$

F^0

$f(a) = 1,$
$f(b) = -1$

M^π

$a \rightarrow 2, b \rightarrow 3$
$f \rightarrow \lambda x.\ if(x = 2, 1, -1)$

Model Checking $\forall x_1: f(x_1) < 0$

$\text{not } if(s_1 = 2, 1, -1) < 0$

$s_1 \rightarrow 2$
$\pi_f^{-1}(2) = a$

F^1

$f(a) = 1,$
$f(b) = -1$
$f(a) < 0$

unsat

Infinite F*

- Is our procedure refutationally complete?

- FOL Compactness

  A set of sentences is unsatisfiable

  iff

  it contains an unsatisfiable finite subset.

- A theory T is a set of sentences, then apply compactness to $F^* \cup T$

Infinite F*: Example

F

$\forall x_1: f(x_1) < f(f(x_1))$,
$\forall x_1: f(x_1) < a$,
$1 < f(0)$.

Unsatisfiable

F*

$f(0) < f(f(0))$, $f(f(0)) < f(f(f(0)))$, ...
$f(0) < a$, $f(f(0)) < a$, ...
$1 < f(0)$

Every finite subset of F* is satisfiable.

Infinite F*: What is wrong?

- Theory of linear arithmetic T_Z is the set of all first-order sentences that are true in the standard structure Z.
- T_z has non-standard models.
- F and F* are satisfiable in a non-standard model.

- Alternative: a theory is a class of structures.
- Compactness does not hold.
- F and F* are still equisatisfiable.

Δ_F and Set Constraints

Given a clause $C_k[x_1, \ldots, x_n]$

Let

$S_{k,i}$ be the set of ground terms used to instantiate x_i in clause $C_k[x_1, \ldots, x_n]$

How to characterize $S_{k,i}$?

F **j-th argument of f in C_k**	**Δ_F** **system of set constraints**
a ground term t	$t \in A_{f,j}$
$t[x_1, \ldots, x_n]$	$t[S_{k,1}, \ldots, S_{k,n}] \subseteq A_{f,j}$
x_i	$S_{k,i} = A_{f,j}$

Microsoft Research

Δ_F: Example

F

$g(x_1, x_2) = 0 \vee h(x_2) = 0,$
$g(f(x_1),b) + 1 \leq f(x_1),$
$h(c) = 1,$
$f(a) = 0$

Δ_F

$S_{1,1} = A_{g,1},\ S_{1,2} = A_{g,2},\ S_{1,2} = A_{h,1}$
$S_{2,1} = A_{f,1},\ f(S_{2,1}) \subseteq A_{g,1},\ b \in A_{g,2}$
$c \in A_{h,1}$
$a \in A_{f,1}$

Δ_F: least solution

$S_{1,1} = \{ f(a) \},\ \ S_{1,2} = \{ b, c \}$
$S_{2,1} = \{ a \}$

Use Δ_F to generate F*

Complexity

- Δ_F is stratified then the least solution (and F*) is finite

$t[S_{k,1}, \ldots, S_{k,n}] \subseteq A_{f,j}$	$\text{level}(S_{k,i}) < \text{level}(A_{f,j})$
$S_{k,i} = A_{f,j}$	$\text{level}(S_{k,i}) = \text{level}(A_{f,j})$

- New decidable fragment: NEXPTIME-Hard.
- The least solution of Δ_F is exponential in the worst case.
  $a \in S_1$, $b \in S_1$, $f_1(S_1, S_1) \subseteq S_2$, …, $f_n(S_n, S_n) \subseteq S_{n+1}$
- F* can be doubly exponential in the size of F.

Extensions

- Arithmetical literals: π_f must be monotonic.

Literal of C_k	Δ_F
$\neg(x_i \leq x_j)$	$S_{k,i} = S_{k,j}$
$\neg(x_i \leq t), \neg(t \leq x_i)$	$t \in S_{k,i}$
$x_i = t$	$\{t+1, t-1\} \subseteq S_{k,i}$

- Offsets:

j-th argument of f in C_k	Δ_F
$x_i + r$	$S_{k,i}+r \subseteq A_{f,j}$ $A_{f,j}+(-r) \subseteq S_{k,i}$

Extensions: Example

Shifting

$$\neg(0 \leq x_1) \vee \neg(x_1 \leq n) \vee f(x_1) = g(x_1+2)$$

Microsoft Research

More Extensions

- Many-sorted logic
- Pseudo-Macros

$0 \leq g(x_1) \vee f(g(x_1)) = x_1,$
$0 \leq g(x_1) \vee h(g(x_1)) = 2x_1,$
$g(a) < 0$

Conclusion

- SMT solvers usually return unsat or unknown for quantified SMT formulas.
- Z3 was the only SMT-solver in SMT-COMP'08 to correctly answer satisfiable quantified formulas.
- New decidable fragments.
- Model-based instantiation and Model checking.
- Conditions for refutationally complete procedures.
- Future work: more efficient model checking techniques.

Thank you!