Theory Combination

Bruno Dutertre

SRI International

SAT/SMT/AR Summer School, Lisbon, July 2019

SMT Background

Basic SMT Problem

- Given a formula Φ in some logical theory T, determine whether Φ is satisfiable or not.
- In addition, if Φ is satisfiable, provide a model of Φ

CDCL(T) Approach

- Combine a CDCL-based SAT Solver with a theory solver for T
- The theory solver works on conjunctions of literals of T

Our Focus

- Quantifier-free theories

Theory Combination

Many Applications Involve Multiple Theories

$$x \leqslant y \ \wedge \ 2y \leqslant x \ \wedge \ f(h(x) - h(y)) > f(0)$$

- This formula is unsat
- To show this, we need to reason about linear arithmetic and uninterpreted functions

Combining Decision Procedures for Modularity

- We don't want to write a global decision procedure
- We have decision procedures for basic theories
- We want to combine them to get a decision procedure for the combined theory.

Common Base Theories

Uninterpreted functions QF_UF

$$\begin{aligned} f(f(x)) &= a \\ g(a) &\neq f(b) \end{aligned}$$

Arithmetic
QF_LRA, QF_LIA, ...

$$\begin{aligned} 2x + y &\geqslant 3 \\ x - y &> 1 \end{aligned}$$

Bitvectors
QF_BV

$$\begin{aligned} \mathtt{bvnot}(x) + 1 &= x \\ \mathtt{bvuge}(x, 0b000..0) & \end{aligned}$$

Arrays
QF_AX

$$\begin{aligned} b &= \mathtt{store}(a, i, v) \\ x &= \mathtt{select}(b, j) \end{aligned}$$

Important: These theories have no non-logical symbol in common

(the only thing they share is equality)

Purification

If Φ is a formula in theory $T_1 \cup T_2$, we can always transform Φ into two parts

- Φ_1 is in theory T_1
- Φ_2 is in theory T_2
- Φ is satisfiable in $T_1 \cup T_2$ iff $\Phi_1 \wedge \Phi_2$ is satisfiable (also in $T_1 \cup T_2$)

This is called purification.

It's done by introducing new variables to remove mixed terms.

Purification Example

Formula with mixed terms:

$$x \leqslant y \ \wedge \ 2y \leqslant x \ \wedge \ f(h(x) - h(y)) > f(0)$$

Purification: separate the uninterpreted function part and the arithmetic part

QF_UF	QF_LRA
$$\begin{aligned} a &= h(x) \\ b &= h(y) \\ d &= f(c) \\ g &= f(e) \end{aligned}$$	$$\begin{aligned} x &\leqslant y \\ 2y &\leqslant x \\ c &= a - b \\ e &= 0 \\ d &> g \end{aligned}$$

After Purification

Purification of Φ produces formulas Φ_1 in T_1 and Φ_2 in T_2

- Unsat Case:
  If Φ_1 is unsat in T_1 or Φ_2 is unsat in T_2 then Φ is unsat in $T_1 \cup T_2$.

- Sat Case:
  If Φ_1 is sat in T_1 and Φ_2 is sat in T_2, is Φ satisfiable in $T_1 \cup T_2$?
  - Φ_1 has a model M_1: $M_1 \models_{T_1} \Phi_1$
  - Φ_2 has a model M_2: $M_2 \models_{T_2} \Phi_2$
  - Can we construct a model M such that $M \models_{T_1 \cup T_2} \Phi$?

Back to Our Example

Formula $x \leqslant y \ \wedge \ 2y \leqslant x \ \wedge \ f(h(x) - h(y)) > f(0)$ is UNSAT

QF_UF part is SAT

$$a = h(x) \ \wedge \ b = h(y) \ \wedge \ d = f(c) \ \wedge \ g = f(e)$$

Possible model with domain = $\{\alpha, \beta\}$

x	α
y	β
a	α
b	β
c	α
d	β

	α	β
f	β	β
h	α	β

QF_LRA part is SAT

$$x \leqslant y \wedge 2y \leqslant x \wedge c = a - b \wedge e = 0 \wedge d > g$$

Possible model (with domain = $\mathbb{R}$)

x	0
y	0
a	0
b	0

c	0
d	1
e	0
g	0

The two models are not consistent

- One says $x \neq y$, the other says $x = y$
- Their domains have different cardinalities

Another Example

In QF_UF + QF_BV:

- a, b, c, d, e are vectors of two bits (type $\texttt{bv}[2]$)
- f is a function from $\texttt{bv}[2]$ to $\texttt{bv}[2]$

Formula $\texttt{distinct}(f(a), f(b), f(c), f(d), f(e))$ is UNSAT

QF_UF part

$\texttt{distinct}(f(a), f(b), f(c), f(d), f(e))$

Satisfiable with models of cardinality at least 5.

QF_BV part

`true`

Satisfiable, but all models have cardinality 4.

Central Problem in Theory Combination

Search for consistent models

- Start with Φ in $T_1 \cup T_2$
- Purify to get Φ_1 in T_1 and Φ_2 in T_2
- Search for two models M_1 and M_2 such that:

$$M_1 \models_{T_1} \Phi_1 \quad \text{and} \quad M_2 \models_{T_2} \Phi_2$$

  M_1 and M_2 have the same cardinality

  M_1 and M_2 agree on equalities between shared variables

Nelson-Oppen Method

- A general framework for solving this problem
- Originally proposed by Nelson and Oppen, 1979
- Give sufficient conditions for consistent models to exist
- Many extensions and variations

Non-Deterministic Nelson-Oppen (Tinelli & Harandi, 1996)

Assumptions

- Two theories T_1 and T_2 that share no non-logical symbol and are stably infinite
- Φ is a conjunction of literals of $T_1 \cup T_2$
- Φ is purified to Φ_1 in T_1 and Φ_2 in T_2

Stably Infinite Theories

- A theory T is stably infinite if every formula that's satisfiable in T has an infinite model
- Examples: QF_UF and QF_LRA are stably infinite, QF_BV is not

Variable Arrangements

Definition

- Let V be the set of all variables that are shared by Φ_1 and Φ_2
- An arrangement of V is a conjunction of variable equalities and disequalities that define a partition of V

Example

- If $V = \{x_0, x_1, x_2, x_3\}$ and we partition V into three subsets $\{x_0, x_1\}$, $\{x_2\}$, and $\{x_3\}$ then the corresponding arrangement is

$$x_0 = x_1 \wedge x_0 \neq x_2 \wedge x_1 \neq x_2 \wedge x_0 \neq x_3 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3$$

Non-Deterministic Nelson-Oppen (continued)

Procedure

- Guess a partition of the variables V and let A be the corresponding arrangement
- Check whether $\Phi_1 \wedge A$ is satisfiable in T_1 and $\Phi_2 \wedge A$ is satisfiable in T_2

Theorem

- If $\Phi_1 \wedge A$ is satisfiable in T_1 and $\Phi_2 \wedge A$ is satisfiable in T_2 then Φ is satisfiable in $T_1 \cup T_2$.

Why this works (informally)

- T_1 and T_2 are stably infinite. This implies that they have models of the same infinite cardinality.
- The arrangement A forces the two models to agree on equalities between shared variables.

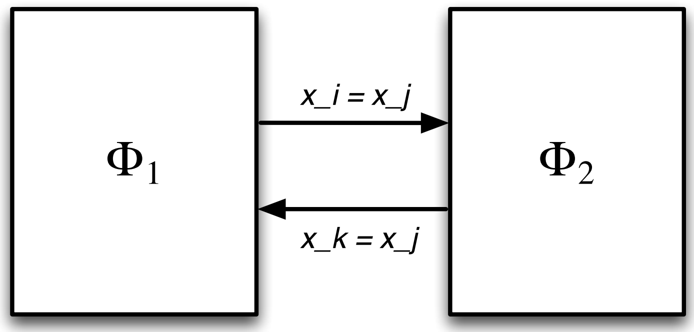
Issues

How do we find the right arrangement?

- The number of possible partitions of a set of n variables is known as Bell's number (B_n)
- This grows very fast with n (e.g., B_{11} is 27644437)
- We can't possibly try them all

How do we handle theories that are not stably infinite?

The Nelson-Oppen Method (Nelson & Oppen, 1979)



Method

- The theory solvers propagate implied equalities between shared variables.
- If both sides are satisfiable and no-more equalities can be propagated, then Φ is satisfiable.

Nelson-Oppen Example

Input

QF_UF

$$
\begin{aligned}
a &= h(x) \\
b &= h(y) \\
d &= f(c) \\
g &= f(e)
\end{aligned}
$$

QF_LRA

$$
\begin{aligned}
x &\leqslant y \\
2y &\leqslant x \\
c &= a - b \\
e &= 0 \\
d &> g
\end{aligned}
$$

Nelson-Oppen Example

QF_LRA deduces and propagates $x = y$

QF_UF	QF_LRA
$a = h(x)$	$x \leqslant y$
$b = h(y)$	$2y \leqslant x$
$d = f(c)$	$c = a - b$
$g = f(e)$	$e = 0$
	$d > g$
$x = y$	$x = y$

Nelson-Oppen Example

QF_UF propagates $a = b$

QF_UF	QF_LRA
$a = h(x)$	$x \leqslant y$
$b = h(y)$	$2y \leqslant x$
$d = f(c)$	$c = a - b$
$g = f(e)$	$e = 0$
	$d > g$
$x = y$	$x = y$
$a = b$	$a = b$

Nelson-Oppen Example

QF_LRA propagates $e = c$

QF_UF	QF_LRA
$$\begin{aligned} a &= h(x) \\ b &= h(y) \\ d &= f(c) \\ g &= f(e) \end{aligned}$$	$$\begin{aligned} x &\leqslant y \\ 2y &\leqslant x \\ c &= a - b \\ e &= 0 \\ d &> g \end{aligned}$$
$$\begin{aligned} x &= y \\ a &= c \\ e &= c \end{aligned}$$	$$\begin{aligned} x &= y \\ a &= c \\ e &= c \end{aligned}$$

Nelson-Oppen Example

QF_UF propagates $d = g$

QF_UF	QF_LRA
$a = h(x)$	$x \leqslant y$
$b = h(y)$	$2y \leqslant x$
$d = f(c)$	$c = a - b$
$g = f(e)$	$e = 0$
	$d > g$
$x = y$	$x = y$
$a = b$	$a = b$
$e = c$	$e = c$
$d = g$	$d = g$

Nelson-Oppen Example

QF_LRA concludes unsat

QF_UF	QF_LRA
$$\begin{aligned} a &= h(x) \\ b &= h(y) \\ d &= f(c) \\ g &= f(e) \end{aligned}$$	$$\begin{aligned} x &\leqslant y \\ 2y &\leqslant x \\ c &= a - b \\ e &= 0 \\ d &> g \end{aligned}$$
$$\begin{aligned} x &= y \\ a &= b \\ e &= c \\ d &= g \end{aligned}$$	$$\begin{aligned} x &= y \\ a &= b \\ e &= c \\ d &= g \end{aligned}$$

Properties of Nelson-Oppen

Soundness and Completeness

- propagating implied equalities is sufficient for some theories but not others
- the theories for which this is sufficient are called convex theories
- for these theories, the method is sound and complete

Termination

- obvious if the number of shared variables is fixed
- this is usually the case
- some theory solvers (e.g., arrays) may dynamically add more variables but this can be bounded

Convex Theories

Definition

- T is convex if, for every set of literals Γ, and every disjunction of variable equalities $x_1 = y_1 \vee \ldots \vee x_n = y_n$, such that

$$\Gamma \models x_1 = y_1 \vee \ldots \vee x_n = y_n,$$

we have

$$\Gamma \models x_i = y_i$$

for some index i.

Examples

- QF_UF and QF_LRA are convex
- QF_LIA, QF_BV, and QF_AX are not convex

Non-Convex Examples

QF_LIA: linear arithmetic over the integers

$$0 \leqslant x \;\wedge\; x \leqslant y \;\wedge\; y \leqslant z \;\wedge\; z \leqslant 1 \quad \models \quad x = y \vee y = z$$

QF_AX: array theory

$$b = \mathtt{store}(a, i, v) \;\wedge\; x = \mathtt{select}(b, j) \;\wedge\; y = \mathtt{select}(a, j) \quad \models \quad x = v \vee x = y$$

More on Nelson-Oppen

Can be extended to non-convex theories

- the theory solvers propagate disjunctions of equalities

Finding Implied Equalities

- For QF_UF, decision procedures based on congruence closure give implied equalities for free.
- It's harder and more expensive for other theories (e.g., linear arithmetic).
- It gets worse for non-convex theories.

Delayed Theory Combination

- Attempt to construct an arrangement lazily in the CDCL(T) framework
- Create interface equalities and let the SAT solver do the search
- Different heuristics to decide when and what equalities to create

Model-Based Theory Combination

Models are available

- The theory solvers for T_1 and T_2 produce models when Φ_1 and Φ_2 are sat:

$$M_1 \models_{T_1} \Phi_1 \text{ and } M_2 \models_{T_2} \Phi_2$$

- The Nelson-Oppen methods do not use these models

Model-based theory combination

- Make use of the models M_1 and M_2:
  - if M_1 and M_2 are consistent, done
  - optionally, attempt to modify M_1 and M_2 to make them consistent
  - if that fails, add constraints to cause CDCL(T) to backtrack and search for other models

Combining a Theory with QF_UF

Very Common Case

- One theory is QF_UF and the other is either an arithmetic theory or QF_BV

QF_UF has good properties

- Deciding satisfiability is cheap (fast congruence closure algorithms)
- These algorithms give the implied equalities for free
- It's stably infinite

Model-Based Combination With QF_UF

- Works with an arbitrary theory T (non-convex, non-stably infinite)
- Main components:
  - congruence closure
  - interface lemmas
  - model mutation and reconciliation

Congruence Closure

Key problem in QF_UF

- Given a finite set of terms and some equalities between them

$$t_1 = u_1, \ldots, t_m = u_m$$

  find all the implied equalities

Congruence Closure Algorithms

- Construct an equivalence relation $\sim$ between terms such that
  - If $t_i = u_i$ is an original equality then $t_i \sim u_i$
  - $\sim$ is closed under the congruence rule:

$$v_1 \sim w_1, ..., v_k \sim w_k \Rightarrow f(v_1, \ldots, v_k) \sim f(w_1, \ldots, w_k)$$

- The $\sim$ relation contains all the implied equalities:

$$t_1 = u_1, \ldots, t_n = u_n \Rightarrow t = u \quad \text{iff} \quad t \sim u$$

Congruence Closure Example

Terms: $a, b, f(a), f(f(a)), f(f(f(a)), f(b)$

Initial Equalities: $f(f(a)) = a, f(a) = b$

Equivalence Relation

- Initially

$$\{a, f(f(a))\} \ \ \{b, f(a)\} \ \ \{f(b)\} \ \ \{f(f(f(a))\}$$

Congruence Closure Example

Terms: $a, b, f(a), f(f(a)), f(f(f(a)), f(b)$

Initial Equalities: $f(f(a)) = a, f(a) = b$

Equivalence Relation

- Congruence: $f(a) = f(f(f(a))$

$$\{a, f(f(a))\} \;\; \{b, f(a), f(f(f(a)))\} \;\; \{f(b)\}$$

Congruence Closure Example

Terms: $a, b, f(a), f(f(a)), f(f(f(a)), f(b)$

Initial Equalities: $f(f(a)) = a$, $f(a) = b$

Equivalence Relation

- Congruence: $f(b) = f(f(a))$

$$\{a, f(f(a)), f(b)\} \;\; \{b, f(a), f(f(f(a)))\}$$

Congruence Closure Example

Terms: $a, b, f(a), f(f(a)), f(f(f(a)), f(b)$

Initial Equalities: $f(f(a)) = a, f(a) = b$

Equivalence Relation

- Done

$$\{a, f(f(a)), f(b)\} \ \ \{b, f(a), f(f(f(a)))\}$$

Checking Satisifiability in QF_UF

A QF_UF formula can be written as a conjunction of equalities and disequalities:

$$(t_1 = u_1 \land \ldots \land t_n = u_n) \ \land \ (v_1 \neq w_1 \land \ldots \land v_m \neq w_m)$$

To check satisfiability

- compute the congruence closure $\sim$ of the equalities
- if $v_i \sim w_i$ for some i then return UNSAT else return SAT

Example

- Formula: $f(f(a)) = a \land f(a) = b \land b \neq f(f(f(a)))$
- Congruence closure: $\{a, f(f(a)), f(b)\} \ \{b, f(a), f(f(f(a)))\}$
- So the formula is UNSAT

Building Models in QF_UF

From A Congruence Closure

- Basic idea: one element in the domain per equivalence class in the congruence closure
- We can always ensure that every term t is interpreted as its class representative

Example

- Formula: $f(b) = a \wedge b = f(a) \wedge a \neq f(c)$
- Congruence closure: $\{a, f(b)\}\ \{b, f(a)\}\ \{c\}\ \{f(c))\}$
- Model:

domain = $\{\alpha, \beta, \gamma, \delta\}$

a	α
b	β
c	γ

	α	β	γ	δ
f	β	α	δ	α

Flexibility in QF_UF Models

Enlarging the domain

- Let Φ be a satisfiable QF_UF formula and M a model of Φ
- For any cardinal $\kappa > |M|$, we can construct a new model M' of cardinality κ that satisfies Φ
- This implies that QF_UF is stably infinite

Shrinking the domain

- We can sometimes make the domain smaller by modifying the congruence closure
- Previous example: Φ is $f(b) = a \land b = f(a) \land a \neq f(c)$
  - Congruence closure: $\{a, f(b)\}\ \{b, f(a)\}\ \{c\}\ \{f(c)\}$
- We could merge $\{f(c)\}$ and $\{b, f(a)\}$ to get a new relation $\sim'$

$$\{a, f(b)\}\ \{b, f(a), f(c)\}\ \{c\}$$

- A model built from $\sim'$ still satisfies Φ

Basic Model-Based Combination With QF_UF

Assumptions

- A formula Φ in QF_UF $\cup\, T$
- After purification: Φ_1 in QF_UF and Φ_2 in T
- V denotes the set of variables shared by Φ_1 and Φ_2
- $\sim$ is the equivalence relation computed by congruence closure from Φ_1

Procedure

- If Φ_1 is not satisfiable, return UNSAT
- Get all equalities implied by Φ_1
- Let H be the set of implied equalities that are between variables of V
- Check whether $\Phi_2 \wedge H$ is satisfiable in T; if not return UNSAT
- Otherwise, get a model M for $\Phi_2 \wedge H$.
- If M does not conflict with relation $\sim$ return SAT
- Otherwise, add interface lemmas to force backtracking

Properties

Conflicts

- M conflicts with E if there are two shared variables x and y such that

$$M \models x = y \quad \text{but} \quad x \not\sim y$$

- conflicts in the other direction are not possible (since $M \models H$)

If there are no conflicts

- M and $\sim$ agree on equalities between shared variables
- We can extend M by adding an interpretation for all the uninterpreted functions in the QF_UF part
- We get a new model M' that satisfies Φ_2 and Φ_1

Interface Lemmas

Interface lemma for x and y

- A formula that encodes “$x = y$ in T” $\Rightarrow$ “$x = y$ in QF_UF”
- The exact formulation depends on the implementation and theory involved
- Examples
  - T is QF_LRA: we add the clause $x = y \vee x > y \vee y > x$
  - T is QF_BV: we add the clause $\neg(\text{bveq } x\ y) \vee x = y$

  in these clauses, $(x = y)$ must be an atom handled by the QF_UF solver

If M conflicts with $\sim$ on $x = y$, this lemma forces the SMT solver to backtrack and search for different models

Improvements

Model Mutation (de Moura & Bjørner, 2007)

- Exploit flexibility in the Simplex-based arithmetic solver.
- There may be many solutions to a set of linear arithmetic constraints.
- Mutation: modify the Simplex model to give distinct values to distinct interface variables.
- This reduces the risk of *accidental conflicts*

Improvements (continued)

Model Reconciliation

- Exploit flexibility in QF_UF to eliminate conflicts while keeping M fixed
- If x and y are in conflict: $M \models x = y$ and $x \not\sim y$
- To try to resolve this conflict:
  - tentatively merge the equivalence classes of x and y
  - propagate the consequences by congruence closure
  - accept the merge unless if makes the QF_UF part unsat or it would propagate new equalities to theory T

Conclusion

Combining decision procedures and theories is central to SMT

Nelson-Oppen is the most common framework for this

- Another method due to Shostak has lost popularity

Nelson-Oppen method has limitations

- require stably infinite, convex theories
- propagating equalities can be expensive

Model-based theory combination methods overcome these limitations

- well-suited for the common case: QF_UF + T
- model mutation or reconciliation can eliminate conflicts
- search for consistent models use dynamic lemmas and backtracking
- more efficient in practice

Related Topics

More on theory combination

- Extensions of Nelson-Oppen to theories that are not stably infinite
- Theory combination in MC-SAT (an alternative to CDCL(T))
- Combination of theories that share logical symbols

Model-based techniques in SMT

- array solvers
- model-based instantiation for problems with quantifiers
- model-based projection

References

Greg Nelson and Derek C. Oppen, Simplification by Cooperating Decision Procedures, ACM Transactions on Programming Languages and Systems, Vol 1, No 2, October 1979.

Greg Nelson and Derek C. Oppen, Fast Decision Procedures Based on Congruence Closure, Journal of the Association for Computing Machinery, Vol 27, No 2, April 1980.

David Detlefs, Greg Nelson, and James B. Saxe, Simplify: A Theorem Prover for Program Checking, Journal of the ACM, Vol 52, No 3, May 2005.

Cesare Tinelli and Mehdi Harandi, A New Correctness Proof of the Nelson-Oppen Combination Procedure, in Frontier of Combining Systems (FROCOS 1996).

Leonardo de Moura and Nikolaj Bjørner, Model-based Theory Combination, SMT Workshop 2007, Electronic Notes in Theoretical Computer Science, 2007.