Algorithms for Satisfiability Modulo Theories

IA085: Satisfiability and Automated Reasoning

Martin Jonáš

FI MUNI, Spring 2024

Last time

- overview of basic notions of first-order logic and satisfiabilty modulo theories
- overview of practically used theories

Terminology

- T-valid formula = T-lemma
- T-satisfiable formula = T-consistent formula

Two approaches

eager encode the input SMT formula into an equisatisfiable SAT formula and use a SAT solver

lazy try checking individual Boolean assignments to the input SMT formula one by one

Eager algorithms

Eager algorithms

Encode the input SMT formula into an equisatisfiable SAT formula and use a SAT solver.

Small-domain encoding

- prove a result *"if φ has a model, it has a model of size at most $k = f(|\varphi|)$"*
- express the set $\{1, \ldots, k\}$ and all the operations by a SAT formula
- example: equality (f = linear), linear arithmetic (f = exponential)

Encoding of axioms

- instantiate all the necessary axioms of the theory and add them to the formula

Encoding axioms: Theory of Equality

$$a = b \ \wedge \ (b = c \vee b \neq d) \ \wedge \ a \neq c \ \wedge \ b = d$$

$$eq_{\{a,b\}} \ \wedge \ (eq_{\{b,c\}} \vee \neg eq_{\{b,d\}}) \ \wedge \ \neg eq_{\{a,c\}} \ \wedge \ eq_{\{b,d\}}$$

where

- $eq_{\{x,y\}}$ are Boolean variables and
- $eq_{\{x,y\}}$ and $eq_{\{y,x\}}$ are the same variable.

Encoding axioms: Theory of Equality

$$a = b \;\wedge\; (b = c \vee b \neq d) \;\wedge\; a \neq c \;\wedge\; b = d$$

$$eq_{\{a,b\}} \;\wedge\; (eq_{\{b,c\}} \vee \neg eq_{\{b,d\}}) \;\wedge\; \neg eq_{\{a,c\}} \;\wedge\; eq_{\{b,d\}}$$

where

- $eq_{\{x,y\}}$ are Boolean variables and
- $eq_{\{x,y\}}$ and $eq_{\{y,x\}}$ are the same variable.

Are we done?

Encoding axioms: Theory of Equality

$$a = b \;\wedge\; (b = c \vee b \neq d) \;\wedge\; a \neq c \;\wedge\; b = d$$

$$eq_{\{a,b\}} \;\wedge\; (eq_{\{b,c\}} \vee \neg eq_{\{b,d\}}) \;\wedge\; \neg eq_{\{a,c\}} \;\wedge\; eq_{\{b,d\}}$$

where

- $eq_{\{x,y\}}$ are Boolean variables and
- $eq_{\{x,y\}}$ and $eq_{\{y,x\}}$ are the same variable.

Are we done?

Add transitivity and reflexivity

- for each added $eq_{\{x,y\}}$ and $eq_{\{y,z\}}$, add conjunct $(eq_{\{x,y\}} \wedge eq_{\{y,z\}}) \rightarrow eq_{\{x,z\}}$
- replace each added $eq_{\{x,x\}}$ by $\top$

Encoding axioms: Theory of Equality and Uninterpreted Functions

$$x = v \ \wedge \ y = g(z) \quad \wedge \quad f(g(x)) \neq f(y) \quad \wedge \quad z = v$$

$$x = v \ \wedge \ y = res_{g(z)} \quad \wedge \quad res_{f(g(x))} \neq res_{f(y)} \quad \wedge \quad z = v$$

where $res_{f(t)}$ and $res_{g(t)}$ are new variables

Are we done?

Encoding axioms: Theory of Equality and Uninterpreted Functions

$$x = v \ \wedge \ y = g(z) \quad \wedge \quad f(g(x)) \neq f(y) \quad \wedge \quad z = v$$

$$x = v \ \wedge \ y = res_{g(z)} \quad \wedge \quad res_{f(g(x))} \neq res_{f(y)} \quad \wedge \quad z = v$$

where $res_{f(t)}$ and $res_{g(t)}$ are new variables

Are we done?

Add congruences

- for each added $res_{f(t_1)}$ and $res_{f(t_2)}$, add conjunct $(t_1 = t_2) \rightarrow (res_{f(t_1)} = res_{f(t_2)})$
- similarly for functions of higher arity: $(t_1 = t_2 \wedge s_1 = s_2) \rightarrow (res_{h(t_1,s_1)} = res_{h(t_2,s_2)})$
- repeat until fixed point

Encoding axioms: Theory of Equality and Uninterpreted Functions

The above procedure

- removes uninterpreted functions by adding new variables and congruences
- reduction of UF to the theory of equality
- known as Ackermann's reduction

Eager algorithms

- usually poor performance, interesting only theoretically
- nowadays almost never used in practice
- one exception: theory of fixed-size bit-vectors (next time)

Lazy algorithms

Lazy algorithms

SMT formula = Boolean structure + theory literals

Combine

- SAT solver to perform the Boolean search
- Theory solver (T-solver) to check satisfiability of conjunctions of T-literals

In the rest of the lecture assume that we have a T-solver for the theory T.

Lazy algorithms

Note

- all following examples use the LRA theory
- because the structure is fixed, instead of $(\mathcal{A}, \mu) \models \varphi$, write only $\mu \models \varphi$ (and similar)

Propositional abstraction

- replace each atomic subformula ψ in the formula φ by a new Boolean variable
- resulting formula φ^P
- denote the mapping by two functions $\mathcal{T}2\mathcal{B}$ and $\mathcal{B}2\mathcal{T}$

Example

$$\begin{aligned} \varphi &= x = 1 \;\wedge\; (y < 3 \vee x + y = 4) \;\wedge\; (\neg(y < 3) \vee x + y = 10) \\ \varphi^P &= A_1 \wedge (A_2 \vee A_3) \wedge (\neg A_2 \vee A_4) \end{aligned}$$

where $\mathcal{T}2\mathcal{B}(x = 1) = A_1$ and $\mathcal{B}2\mathcal{T}(\neg A_2) = \neg(y < 3)$

Propositional abstraction

Theorem
If the propositional abstraction φ^P *is unsatisfiable, the original formula* φ *is* T*-unsatisfiable.*

Proof.
If μ is a T-model of the original formula φ, then μ^P defined by $\mu(A_i) = [\![\mathcal{B}2\mathcal{T}(A_i)]\!]^\mu$ is a propositional model of φ^P. □

The converse does not hold.

Propositional abstraction

Each propositional assignment μ of φ^P corresponds to a conjunction of T-literals

$$\mu^T = \bigwedge_{v \in Vars, \mu(v)=\top} \mathcal{B}2\mathcal{T}(v) \wedge \bigwedge_{v \in Vars, \mu(v)=\bot} \neg\mathcal{B}2\mathcal{T}(v)$$

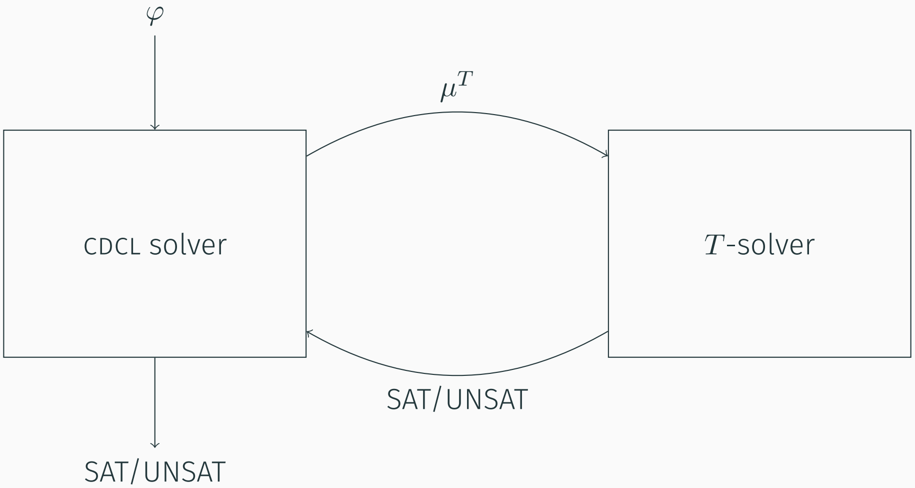
Example
For

$$\begin{aligned} \varphi &= x = 1 \;\wedge\; (y < 3 \vee x + y = 4) \;\wedge\; (\neg(y < 3) \vee x + y = 10) \\ \varphi^P &= A_1 \wedge (A_2 \vee A_3) \wedge (\neg A_2 \vee A_4) \end{aligned}$$

and $\mu(A_1) = \top, \mu(A_2) = \bot, \mu(A_3) = \top$

$$\mu^T = (x = 1) \wedge \neg(y < 3) \wedge (x + y = 4)$$

Offline Lazy SMT solving – schema



Offline Lazy SMT solving – algorithm

```
offline_smt(formula φ):
    φ^P ← T2B(φ)
    while check_sat(φ^P) == SAT {
        μ = get_model(φ^P)
        if check_theory(μ^T) == SAT {
            return SAT
        } else {
            φ^P ← φ^P ∧ ¬μ
        }
    }
    return UNSAT
```

Offline Lazy SMT solving – example

$$\varphi \quad = \quad x = 1 \;\;\wedge\;\; (y < 3 \;\vee\; y > 5) \;\;\wedge\;\; (x + y = 4 \;\vee\; y = 6)$$

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\},$$
$$\{A_2, A_3\},$$
$$\{A_4, A_5\}$$

Offline Lazy SMT solving – example

$$\varphi \quad = \quad x = 1 \quad \wedge \quad (y < 3 \ \vee \ y > 5) \quad \wedge \quad (x + y = 4 \ \vee \ y = 6)$$

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$

$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\}, \\ \{A_2, A_3\}, \\ \{A_4, A_5\}$$

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \;\vee\; y > 5) \;\wedge\; (x + y = 4 \;\vee\; y = 6)$$

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\},\ \{A_2, A_3\},\ \{A_4, A_5\}$$

$$\mu = \{A_1, A_2, \neg A_3, A_4, \neg A_5\}$$
$$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge \neg(y = 6)$$

Offline Lazy SMT solving – example

$$\varphi \quad = \quad x = 1 \;\; \wedge \;\; (y < 3 \;\vee\; y > 5) \;\; \wedge \;\; (x + y = 4 \;\vee\; y = 6)$$

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\}\}$$

$\mu = \{A_1, A_2, \neg A_3, A_4, \neg A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge \neg(y = 6)$

T-unsatisfiable ☹

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \;\vee\; y > 5) \;\wedge\; (x + y = 4 \;\vee\; y = 6)$$

$\mathcal{B}2\mathcal{T}(A_1) = x = 1$, $\mathcal{B}2\mathcal{T}(A_2) = y < 3$, $\mathcal{B}2\mathcal{T}(A_3) = y > 5$,

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4$, $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\}$$

$\mu = \{A_1, A_2, \neg A_3, A_4, \neg A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge \neg(y = 6)$

T-unsatisfiable ☹

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \;\vee\; y > 5) \;\wedge\; (x + y = 4 \;\vee\; y = 6)$$

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\begin{aligned}\varphi^P = \{&\{A_1\},\\ &\{A_2, A_3\},\\ &\{A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\}\end{aligned}$$

$$\mu = \{A_1, A_2, \neg A_3, A_4, A_5\}$$
$$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge y = 6$$

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \;\vee\; y > 5) \;\wedge\; (x + y = 4 \;\vee\; y = 6)$$

$$\mathcal{B2T}(A_1) = x = 1, \qquad \mathcal{B2T}(A_2) = y < 3, \qquad \mathcal{B2T}(A_3) = y > 5,$$
$$\mathcal{B2T}(A_4) = x + y = 4, \qquad \mathcal{B2T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\}$$

$\mu = \{A_1, A_2, \neg A_3, A_4, A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge y = 6$

T-unsatisfiable ☹

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \vee y > 5) \;\wedge\; (x + y = 4 \vee y = 6)$$

$\mathcal{B2T}(A_1) = x = 1$, $\mathcal{B2T}(A_2) = y < 3$, $\mathcal{B2T}(A_3) = y > 5$,

$\mathcal{B2T}(A_4) = x + y = 4$, $\mathcal{B2T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\} \{\neg A_1, \neg A_2, A_3, \neg A_4, \neg A_5\}$$

$\mu = \{A_1, A_2, \neg A_3, A_4, A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge y = 6$

T-unsatisfiable ☹

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \,\vee\, y > 5) \;\wedge\; (x + y = 4 \,\vee\, y = 6)$$

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\begin{aligned}\varphi^P = \{&\{A_1\},\\ &\{A_2, A_3\},\\ &\{A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, \neg A_5\}\end{aligned}$$

$$\mu = \{A_1, A_2, \neg A_3, \neg A_4, A_5\}$$
$$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge \neg(x + y = 4) \wedge y = 6$$

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \;\vee\; y > 5) \;\wedge\; (x + y = 4 \;\vee\; y = 6)$$

$\mathcal{B}2\mathcal{T}(A_1) = x = 1$, $\mathcal{B}2\mathcal{T}(A_2) = y < 3$, $\mathcal{B}2\mathcal{T}(A_3) = y > 5$,

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4$, $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\begin{aligned}\varphi^P = \{\{&A_1\},\\ &\{A_2, A_3\},\\ &\{A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, \neg A_5\}\end{aligned}$$

$\mu = \{A_1, A_2, \neg A_3, \neg A_4, A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge \neg(x + y = 4) \wedge y = 6$

T-unsatisfiable ☹

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \;\vee\; y > 5) \;\wedge\; (x + y = 4 \;\vee\; y = 6)$$

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\} \{\neg A_1, \neg A_2, A_3, \neg A_4, \neg A_5\} \{\neg A_1, \neg A_2, A_3, A_4, \neg A_5\}$$

$\mu = \{A_1, A_2, \neg A_3, \neg A_4, A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge \neg(x + y = 4) \wedge y = 6$

T-unsatisfiable ☹

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \;\vee\; y > 5) \;\wedge\; (x + y = 4 \;\vee\; y = 6)$$

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\mu = \{A_1, \neg A_2, A_3, \neg A_4, A_5\}$$
$$\mu^P = x = 1 \wedge \neg(y < 3) \wedge y > 5 \wedge \neg(x + y = 4) \wedge y = 6$$

$$\begin{aligned}\varphi^P = \{&\{A_1\},\\ &\{A_2, A_3\},\\ &\{A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, \neg A_5\}\\ &\{\neg A_1, \neg A_2, A_3, A_4, \neg A_5\}\end{aligned}$$

Offline Lazy SMT solving – example

$$\varphi \;=\; x = 1 \;\wedge\; (y < 3 \vee y > 5) \;\wedge\; (x + y = 4 \vee y = 6)$$

$\mathcal{B}2\mathcal{T}(A_1) = x = 1$, $\mathcal{B}2\mathcal{T}(A_2) = y < 3$, $\mathcal{B}2\mathcal{T}(A_3) = y > 5$,

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4$, $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\begin{aligned}\varphi^P = \{&\{A_1\},\\ &\{A_2, A_3\},\\ &\{A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, A_5\}\\ &\{\neg A_1, \neg A_2, A_3, \neg A_4, \neg A_5\}\\ &\{\neg A_1, \neg A_2, A_3, A_4, \neg A_5\}\}\end{aligned}$$

$\mu = \{A_1, \neg A_2, A_3, \neg A_4, A_5\}$

$\mu^P = x = 1 \wedge \neg(y < 3) \wedge y > 5 \wedge \neg(x + y = 4) \wedge y = 6$

T-satisfiable ☺

Downsides

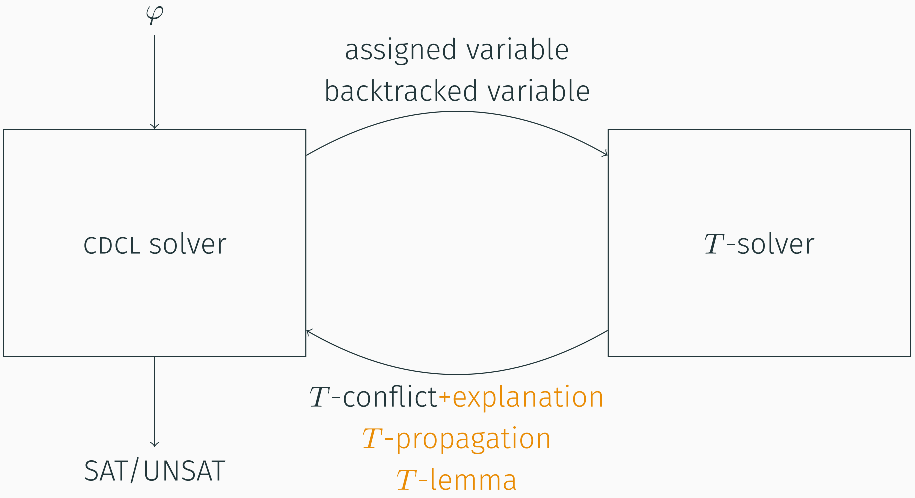
- the SAT solver is executed from scratch every time
- propositional models are blocked one at time
- theory reasoning is applied only for complete assignments

CDCL(T)

CDCL(T)

- tight integration of a CDCL-based SAT solver and a theory solver
- theory solver can explain conflicts and guide the search of the SAT solver
- basis of most of modern SMT solvers (CVC5, MathSAT, Yices, Z3, . . .)

CDCL(T) – schema



Conflict Explanation

- if the T-solver detects a conflict in the Boolean assignment $\mu = \{l_1, \ldots, l_k\}$, it can compute its subset $\mu' \subseteq \mu$ such that $\mu' \models_T \bot$
- instead of learning $\vee_{l \in \mu} \neg l$, the SAT solver can learn $\vee_{l \in \mu'} \neg l$

Conflict Explanation: Example

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\begin{aligned}\varphi^P = \{&\{A_1\},\\ &\{A_2, A_3\},\\ &\{A_4, A_5\}\end{aligned}$$

Conflict Explanation: Example

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\},\ \{A_2, A_3\},\ \{A_4, A_5\}\}$$

Conflict Explanation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\}\}$$

$$\mu = \{A_1, A_2, \neg A_3, A_4, \neg A_5\}$$

$$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge \neg(y = 6)$$

Conflict Explanation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\}$$

$\mu = \{A_1, A_2, \neg A_3, A_4, \neg A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge \neg(y = 6)$

T-unsatisfiable ☹

reason $\{A_1, A_2, A_4\}$

Conflict Explanation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1$, $\mathcal{B}2\mathcal{T}(A_2) = y < 3$, $\mathcal{B}2\mathcal{T}(A_3) = y > 5$,

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4$, $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2, \neg A_4\}$$

$\mu = \{A_1, A_2, \neg A_3, A_4, \neg A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge x + y = 4 \wedge \neg(y = 6)$

T-unsatisfiable ☹

reason $\{A_1, A_2, A_4\}$

Conflict Explanation: Example

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \qquad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \qquad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \qquad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\begin{aligned}\varphi^P = \{&\{A_1\},\\ &\{A_2, A_3\},\\ &\{A_4, A_5\}\\ &\{\neg A_1, \neg A_2, \neg A_4\}\end{aligned}$$

$$\mu = \{A_1, A_2, \neg A_3, \neg A_4, A_5\}$$
$$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge \neg(x+y = 4) \wedge y = 6$$

Conflict Explanation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\}, \{\neg A_1, \neg A_2, \neg A_4\}$$

$\mu = \{A_1, A_2, \neg A_3, \neg A_4, A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge \neg(x + y = 4) \wedge y = 6$

T-unsatisfiable ☹

reason $\{A_2, A_5\}$

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \quad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \quad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \quad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2, \neg A_4\} \{\neg A_2, \neg A_5\}$$

$\mu = \{A_1, A_2, \neg A_3, \neg A_4, A_5\}$

$\mu^P = x = 1 \wedge y < 3 \wedge \neg(y > 5) \wedge \neg(x + y = 4) \wedge y = 6$

T-unsatisfiable ☹

reason $\{A_2, A_5\}$

Conflict Explanation: Example

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \quad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \quad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \quad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2, \neg A_4\} \{\neg A_2, \neg A_5\}$$

$$\mu = \{A_1, \neg A_2, A_3, \neg A_4, A_5\}$$
$$\mu^P = x = 1 \wedge \neg(y < 3) \wedge y > 5 \wedge \neg(x + y = 4) \wedge y = 6$$

Conflict Explanation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2, \neg A_4\} \{\neg A_2, \neg A_5\}\}$$

$\mu = \{A_1, \neg A_2, A_3, \neg A_4, A_5\}$

$\mu^P = x = 1 \wedge \neg(y < 3) \wedge y > 5 \wedge \neg(x+y = 4) \wedge y = 6$

T-satisfiable ☺

Theory propagation

- SAT solver notifies the T-solver about all variable assignments/backtracking
- T-solver knows the currently assigned literals μ^T
- T-solver can detect T-entailed literals $\mu^T \models_T l$ and propagate them

For the backtracking

- T-solver must be able to provide explanations of the propagations
- for each T-propagated literal $\mu^T \models l$, an explanation $\mu' \subseteq \mu^T$ such that $\mu' \models_T l$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1$, $\mathcal{B}2\mathcal{T}(A_2) = y < 3$, $\mathcal{B}2\mathcal{T}(A_3) = y > 5$,
$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4$, $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \}$$

SAT solver trail

T-solver assignment

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,\quad \mathcal{B}2\mathcal{T}(A_2) = y < 3,\quad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,\quad \mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\},\ \{A_2, A_3\},\ \{A_4, A_5\}\ \}$$

SAT solver trail

A_1^{up}

T-solver assignment

$x = 1$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \}$$

SAT solver trail

A_1^{up}, A_2^d

T-solver assignment

$x = 1$

$y < 3$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1$, $\mathcal{B}2\mathcal{T}(A_2) = y < 3$, $\mathcal{B}2\mathcal{T}(A_3) = y > 5$,
$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4$, $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \}$$

SAT solver trail

A_1^{up} , A_2^{d} , $\neg A_3^{tp}$

T-solver assignment

$x = 1$

$y < 3$

$\neg(y > 5)$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$
$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \}$$

SAT solver trail

$A_1^{up}, A_2^{d}, \neg A_3^{tp}, \neg A_4^{tp}$

T-solver assignment

$x = 1$

$y < 3$

$\neg(y > 5)$

$\neg(x + y = 4)$

Theory propagation: Example

$$\mathcal{B}2\mathcal{T}(A_1) = x = 1, \quad \mathcal{B}2\mathcal{T}(A_2) = y < 3, \quad \mathcal{B}2\mathcal{T}(A_3) = y > 5,$$
$$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4, \quad \mathcal{B}2\mathcal{T}(A_5) = y = 6$$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \}$$

SAT solver trail

A_1^{up} , A_2^{d} , $\neg A_3^{tp}$, $\neg A_4^{tp}$, $\neg A_5^{tp}$

T-solver assignment

$x = 1$

$y < 3$

$\neg(y > 5)$

$\neg(x + y = 4)$

$\neg(y = 6)$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1$, $\mathcal{B}2\mathcal{T}(A_2) = y < 3$, $\mathcal{B}2\mathcal{T}(A_3) = y > 5$,
$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4$, $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\},\ \{A_2, A_3\},\ \{A_4, A_5\}\ \}$$

SAT solver trail

A_1^{up} , A_2^{d} , $\neg A_3^{tp}$, $\neg A_4^{tp}$, $\neg A_5^{tp}$

T-solver assignment

$x = 1$

$y < 3$

$\neg(y > 5)$

$\neg(x + y = 4)$

$\neg(y = 6)$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1$, $\mathcal{B}2\mathcal{T}(A_2) = y < 3$, $\mathcal{B}2\mathcal{T}(A_3) = y > 5$,
$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4$, $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\}, \{\neg A_1, \neg A_2\}\}$$

SAT solver trail

$A_1^{up}, A_2^{d}, \neg A_3^{tp}, \neg A_4^{tp}, \neg A_5^{tp}$

T-solver assignment

$x = 1$

$y < 3$

$\neg(y > 5)$

$\neg(x + y = 4)$

$\neg(y = 6)$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2\} \}$$

SAT solver trail

$A_1^{up}, \neg A_2^{bj}$

T-solver assignment

$x = 1$

$\neg(y < 3)$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2\} \}$$

SAT solver trail

$A_1^{up}, \neg A_2^{bj}, A_3^{up}$

T-solver assignment

$x = 1$

$\neg(y < 3)$

$(y > 5)$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2\} \}$$

SAT solver trail

$A_1^{up}, \neg A_2^{bj}, A_3^{up}, \neg A_4^{tp}$

T-solver assignment

$x = 1$

$\neg(y < 3)$

$(y > 5)$

$\neg(x + y = 4)$

Theory propagation: Example

$\mathcal{B}2\mathcal{T}(A_1) = x = 1,$ $\mathcal{B}2\mathcal{T}(A_2) = y < 3,$ $\mathcal{B}2\mathcal{T}(A_3) = y > 5,$

$\mathcal{B}2\mathcal{T}(A_4) = x + y = 4,$ $\mathcal{B}2\mathcal{T}(A_5) = y = 6$

$$\varphi^P = \{\{A_1\}, \{A_2, A_3\}, \{A_4, A_5\} \{\neg A_1, \neg A_2\} \}$$

SAT solver trail

$A_1^{up}, \neg A_2^{bj}, A_3^{up}, \neg A_4^{tp}, A_5^{up}$

T-solver assignment

$x = 1$

$\neg(y < 3)$

$(y > 5)$

$\neg(x + y = 4)$

$y > 5$

Early pruning

- T-solver knows the currently assigned literals μ^T
- if $\mu^T \models_T \bot$, declare conflict before setting all literals

For correctness

- needs to provide explanations of the conflicts
- can perform cheaper approximate check $\rightarrow$ does not have to detect all inconsistencies
- the expensive full check needs to be performed only for the complete assignments

Interface of T-solver

T-solver can be instantiated arbitrarily, but it should

- handle assignment of literal values efficiently
- backtrack efficiently
- provide reasons for theory conflicts

It further can

- perform theory propagation (identify implied literals)
- perform early pruning (identify theory conflicts during the search)

Interface of T-solver

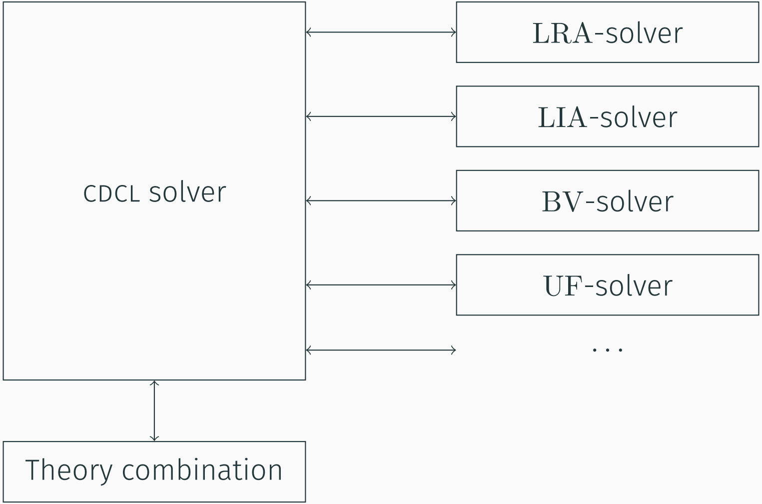
Possible interface of the T-solver

- void notifyAtom(lit)
- void assignLiteral(lit)
- void push()
- void pop()
- result checkSat()
- option<result> checkSat_approx()
- list<lit> getConflictReason()
- option<lit> getPropagatedLiteral()
- list<lit> getExplanation(lit)

Other improvements

- **normalize T-literals**
  - $(x > y) \rightsquigarrow \neg(x \leq y)$
  - $(y + 3 + x) \rightsquigarrow x + y + 3$
- **eagerly learn some interesting T-lemmas**
  - if the formula contains $x = 0$ and $x = 1$
  - add T-lemma $\neg(x = 0) \vee \neg(x = 1)$ before solving
- **pure literal filtering**
  - if the formula contains a literal l only positively and the current assignment contains $\neg l$, do not send $\neg l$ to the T-solver
- **splitting on demand**
  - when T-solver wants to do a case split, it can add a new T-lemma corresponding to the split to the SAT solver
  - can introduce new T-literals and new Boolean variables
  - $(x + y < 0) \vee (x + y \geq 0)$
  - case split will be performed as part of the propositional search

Modern SMT solvers



Next time

- theory solvers for selected theories