CS5733 Program Synthesis

#8. First Order Theories and SMT Solvers

Ashish Mishra, August 27, 2024

EUPhony

- Q. What does Euphony use as behavioral constraints? Structural constraint? Search strategy? How are they different from EUSolver?
  - Logical formula capturing input/output examples
  - Probabilistic Higher Order Grammar (PHOG)
  - A* variant for weighted top-down search.

EUPhony

- Q. Consider Fig 2b, where the synthesizer is unrolling the sentential form Rep(x,"-",S). When the search is guided by a PHOG, it considers the weighted productions shown in Fig 2a (top). What would these productions look like if we replaced the PHOG with a PCFG? With 3-grams? Do you think these other probabilistic models would work as well as a PHOG?

    - For this question, we missed one of the important topics, so I will cover that in the next class.

EUPhony

- Q. Consider Theorem 3.7. Give an example of sentential forms n_i , n_j and set of points pts such that n_i and n_j are equivalent on pts but not weakly equivalent.

```
n1 = Rep(”-”, ”.”, x)  n2 = “-” pts = [“-.”]
```

only P2 < n2 is \epsilon

```
n1 = x + 1 + S    n1 = x + 2 + S     pts = [1]
```

```
n1 = ”-” + ”.”    n1 = ”-” + S     pts = [“-.”]
```

Last lecture on Verification

Roadmap

- Previously
  - PL
  - SAT Solving
  - FOL
- Today
  - Overview FOT
  - Satisfiability Modulo Theories

Semi-decidability of FOL

A problem is semi-decidable iff there exists a procedure that, for any input:
1. halts and says “yes” if answer is positive, and
2. may not terminate if answer is negative.

Semi-decidability of FOL:
For every valid FOL formula, there exists a procedure (semantic argument method) that always terminates and says “yes”.
If an FOL formula is invalid, there exists no procedure that is guaranteed to terminate.

Motivation FOT

- FOL is very expressive, powerful and undecidable in general
- Some application domains do not need the full power of FOL.
- First-order theories are useful for reasoning about specific applications
  - We have structure in mind while reasoning about certain problems.
  - e.g., programs with arithmetic operations over integers
- FOT formalize these structures to help reasoning about them.
- Specialized, efficient decision procedures!

First-Order Theories I

First-order theory T consists of

- Signature Σ_T - set of constant, function, and predicate symbols
- Set of axioms A_T - set of closed (no free variables) Σ_T-formulae

A Σ_T-formula is a formula constructed of constants, functions, and predicate symbols from Σ_T, and variables, logical connectives, and quantifiers.

The symbols of Σ_T are just symbols without prior meaning — the axioms of T provide their meaning.

First-Order Theories II

A Σ_T-formula F is valid in theory T (T-valid, also $T \models F$),
iff every interpretation I that satisfies the axioms of T,
i.e. $I \models A$ for every $A \in A_T$ (T-interpretation)
also satisfies F,
i.e. $I \models F$

A Σ_T-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation (i.e. satisfies all the axioms of T) that satisfies F

Two formulae F_1 and F_2 are equivalent in T (T-equivalent),
iff $T \models F_1 \leftrightarrow F_2$,
i.e. if for every T-interpretation I, $I \models F_1$ iff $I \models F_2$

Note:

- $I \models F$ stands for "F true under interpretation I"
- $T \models F$ stands for "F is valid in theory T"

Fragments of Theories

A fragment of theory T is a syntactically-restricted subset of formulae of the theory.

Example: a quantifier-free fragment of theory T is the set of quantifier-free formulae in T.

A theory T is decidable if $T \models F$ (T-validity) is decidable for every Σ_T-formula F;

i.e., there is an algorithm that always terminate with "yes", if F is T-valid, and "no", if F is T-invalid.

A fragment of T is decidable if $T \models F$ is decidable for every Σ_T-formula F obeying the syntactic restriction.

Common first-order theories

- Theory of equality (with uninterpreted functions)
- Peano arithmetic (first-order arithmetic)
- Presburger arithmetic
- Theory of reals
- Theory of rationals
- Theory of arrays

Theory of Equality T_E I

Signature:

$$\Sigma_= : \{=, a, b, c, \cdots, f, g, h, \cdots, p, q, r, \cdots\}$$

consists of

- $=$, a binary predicate, interpreted with meaning provided by axioms
- all constant, function, and predicate symbols

Axioms of T_E

1. $\forall x.\ x = x$ (reflexivity)
2. $\forall x, y.\ x = y \ \rightarrow\ y = x$ (symmetry)
3. $\forall x, y, z.\ x = y \wedge y = z \ \rightarrow\ x = z$ (transitivity)
4. for each positive integer n and n-ary function symbol f,
   $\forall x_1, \ldots, x_n, y_1, \ldots, y_n.\ \bigwedge_i x_i = y_i$
   $\rightarrow\ f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$ (function congruence)

Theory of Equality T_E II

5. for each positive integer n and n-ary predicate symbol p,
   $$\forall x_1, \ldots, x_n, y_1, \ldots, y_n.\ \bigwedge_i x_i = y_i \\ \rightarrow\ (p(x_1, \ldots, x_n) \leftrightarrow p(y_1, \ldots, y_n))$$ (predicate congruence)

(function) and (predicate) are axiom schemata.

Example:

(function) for binary function f for $n = 2$:

$$\forall x_1, x_2, y_1, y_2.\ x_1 = y_1 \wedge x_2 = y_2 \ \rightarrow\ f(x_1, x_2) = f(y_1, y_2)$$

(predicate) for unary predicate p for $n = 1$:

$$\forall x, y.\ x = y \ \rightarrow\ (p(x)\ \leftrightarrow\ p(y))$$

Note: we omit "congruence" for brevity.

Decidability of T_E I

T_E is undecidable.

The quantifier-free fragment of T_E is decidable. Very efficient algorithm.

Semantic argument method can be used for T_E

Example: Prove

$$F: \; a = b \land b = c \; \rightarrow \; g(f(a), b) = g(f(c), a)$$

is T_E-valid.

Decidability of T_E II

Suppose not; then there exists a T_{E}-interpretation I such that $I \not\models F$. Then,

1.	I	$\not\models$	F	assumption
2.	I	$\models$	$a = b \wedge b = c$	1, $\rightarrow$
3.	I	$\not\models$	$g(f(a), b) = g(f(c), a)$	1, $\rightarrow$
4.	I	$\models$	$a = b$	2, $\wedge$
5.	I	$\models$	$b = c$	2, $\wedge$
6.	I	$\models$	$a = c$	4, 5, (transitivity)
7.	I	$\models$	$f(a) = f(c)$	6, (function)
8.	I	$\models$	$b = a$	4, (symmetry)
9.	I	$\models$	$g(f(a), b) = g(f(c), a)$	7, 8, (function)
10.	I	$\models$	$\bot$	3, 9 contradictory

F is T_{E}-valid.

Motivation

Prove the equivalences of these two programs

In general undecidable, here bounded loops.

```
int power3(int in) {
int i, out_a;
    out_a = in;
    for (i = 0; i < 2; i++)
        out_a = out_a * in;
    return out_a; }
```

(a)

```
int power3_new(int in) {
int out_b;

    out_b = (in * in) * in;
    return out_b; }
```

(b)

Equivalence of programs a and b

- A key observation, only bounded loops,
  - Possible to compute their input/output relations
- Steps for i/o relation.
  - Remove the
  - Unroll the fo
  - Replace the
  - Read (referr
  - Conjoin all p

$$
\begin{aligned}
out0\_a &= in && \land \\
out1\_a &= out0\_a * in && \land \\
out2\_a &= out1\_a * in &&
\end{aligned}
$$

(φ_a)

$$out0\_b = (in*in)*in;$$

(φ_b)

Equivalence check

$$\varphi_a \wedge \varphi_b \implies out2\_a = out0\_b \,.$$

Replace some functions with "Uninterpreted" functions

$$\begin{aligned} out0\_a &= in && \wedge \\ out1\_a &= G(out0\_a, in) && \wedge \\ out2\_a &= G(out1\_a, in) && \end{aligned}$$

$(\varphi_a^{\mathrm{UF}})$

$$out0\_b = G(G(in, in), in)$$

$(\varphi_b^{\mathrm{UF}})$

$$\varphi_a^{\mathrm{UF}} \wedge \varphi_b^{\mathrm{UF}} \implies out2\_a = out0\_b \,.$$

Natural Numbers and Integers

Natural numbers $\mathbb{N} = \{0, 1, 2, \cdots\}$

Integers $\mathbb{Z} = \{\cdots, -2, -1, 0, 1, 2, \cdots\}$

Three variations:

- Peano arithmetic T_{PA}: natural numbers with addition, multiplication, $=$
- Presburger arithmetic $T_{\mathbb{N}}$: natural numbers with addition, $=$
- Theory of integers $T_{\mathbb{Z}}$: integers with $+, -, >, =$, multiplication by constants

1. Peano Arithmetic T_{PA} (first-order arithmetic)

$$\Sigma_{\mathrm{PA}} : \ \{0,\ 1,\ +,\ \cdot,\ =\}$$

Equality Axioms: (reflexivity), (symmetry), (transitivity), (function) for $+$, (function) for $\cdot$.

And the axioms:

1. $\forall x.\ \neg(x + 1 = 0)$ (zero)
2. $\forall x, y.\ x + 1 = y + 1 \ \rightarrow \ x = y$ (successor)
3. $F[0] \ \wedge \ (\forall x.\ F[x] \ \rightarrow \ F[x + 1]) \ \rightarrow \ \forall x.\ F[x]$ (induction)
4. $\forall x.\ x + 0 = x$ (plus zero)
5. $\forall x, y.\ x + (y + 1) = (x + y) + 1$ (plus successor)
6. $\forall x.\ x \cdot 0 = 0$ (times zero)
7. $\forall x, y.\ x \cdot (y + 1) = x \cdot y + x$ (times successor)

Line 3 is an axiom schema.

Example: $3x + 5 = 2y$ can be written using Σ_{PA} as

$$x + x + x + 1 + 1 + 1 + 1 + 1 = y + y$$

Note: we have $>$ and $\geq$ since

$3x + 5 > 2y$ write as $\exists z.\ z \neq 0 \wedge 3x + 5 = 2y + z$

$3x + 5 \geq 2y$ write as $\exists z.\ 3x + 5 = 2y + z$

Example:

Existence of pythagorean triples (F is T_{PA}-valid):

$$F : \exists x, y, z.\ x \neq 0 \wedge y \neq 0 \wedge z \neq 0 \wedge x \cdot x + y \cdot y = z \cdot z$$

Decidability of Peano Arithmetic

T_{PA} is undecidable. (Gödel, Turing, Post, Church)
The quantifier-free fragment of T_{PA} is undecidable.
(Matiyasevich, 1970)

<u>Remark</u>: Gödel's first incompleteness theorem

Peano arithmetic T_{PA} does not capture true arithmetic:

There exist closed Σ_{PA}-formulae representing valid propositions of number theory that are not T_{PA}-valid.

The reason: T_{PA} actually admits *nonstandard interpretations*.

For decidability: no multiplication

2. Presburger Arithmetic $T_{\mathbb{N}}$

Signature $\Sigma_{\mathbb{N}}$: $\{0,\ 1,\ +,\ =\}$ no multiplication!

Axioms of $T_{\mathbb{N}}$ (equality axioms, with 1-5):

1. $\forall x.\ \neg(x + 1 = 0)$ (zero)
2. $\forall x, y.\ x + 1 = y + 1 \ \rightarrow\ x = y$ (successor)
3. $F[0] \wedge (\forall x.\ F[x] \ \rightarrow\ F[x + 1]) \ \rightarrow\ \forall x.\ F[x]$ (induction)
4. $\forall x.\ x + 0 = x$ (plus zero)
5. $\forall x, y.\ x + (y + 1) = (x + y) + 1$ (plus successor)

Line 3 is an axiom schema.

$T_{\mathbb{N}}$-satisfiability (and thus $T_{\mathbb{N}}$-validity) is decidable (Presburger, 1929)

3. Theory of Integers $T_{\mathbb{Z}}$

Signature:

$$\Sigma_{\mathbb{Z}} : \{\ldots, -2, -1, 0, 1, 2, \ldots, -3\cdot, -2\cdot, 2\cdot, 3\cdot, \ldots, +, -, >, =\}$$

where

- $\ldots, -2, -1, 0, 1, 2, \ldots$ are constants
- $\ldots, -3\cdot, -2\cdot, 2\cdot, 3\cdot, \ldots$ are unary functions
  (intended meaning: $2 \cdot x$ is $x + x$, $-3 \cdot x$ is $-x - x - x$)
- $+, -, >, =$ have the usual meanings.

Relation between $T_{\mathbb{Z}}$ and $T_{\mathbb{N}}$:

$T_{\mathbb{Z}}$ and $T_{\mathbb{N}}$ have the same expressiveness:

- For every $\Sigma_{\mathbb{Z}}$-formula there is an equisatisfiable $\Sigma_{\mathbb{N}}$-formula.
- For every $\Sigma_{\mathbb{N}}$-formula there is an equisatisfiable $\Sigma_{\mathbb{Z}}$-formula.

$\Sigma_{\mathbb{Z}}$-formula F and $\Sigma_{\mathbb{N}}$-formula G are *equisatisfiable* iff:

$$F \text{ is } T_{\mathbb{Z}}\text{-satisfiable} \quad \text{iff} \quad G \text{ is } T_{\mathbb{N}}\text{-satisfiable}$$

$\Sigma_{\mathbb{Z}}$-formula to $\Sigma_{\mathbb{N}}$-formula I

Example: consider the $\Sigma_{\mathbb{Z}}$-formula

$$F_0 : \ \forall w, x.\ \exists y, z.\ x + 2y - z - 7 > -3w + 4.$$

Introduce two variables, v_p and v_n (range over the nonnegative integers) for each variable v (range over the integers) of F_0:

$$F_1 : \quad \begin{array}{l} \forall w_p, w_n, x_p, x_n.\ \exists y_p, y_n, z_p, z_n. \\ \quad (x_p - x_n) + 2(y_p - y_n) - (z_p - z_n) - 7 > -3(w_p - w_n) + 4 \end{array}$$

Eliminate $-$ by moving to the other side of $>$:

$$F_2 : \quad \begin{array}{l} \forall w_p, w_n, x_p, x_n.\ \exists y_p, y_n, z_p, z_n. \\ \quad x_p + 2y_p + z_n + 3w_p > x_n + 2y_n + z_p + 7 + 3w_n + 4 \end{array}$$

$\Sigma_{\mathbb{Z}}$-formula to $\Sigma_{\mathbb{N}}$-formula II

Eliminate $>$ and numbers:

$$
\begin{array}{ll}
& \forall w_p, w_n, x_p, x_n.\ \exists y_p, y_n, z_p, z_n.\ \exists u. \\
F_3 : & \quad \neg(u = 0)\ \wedge\ x_p + y_p + y_p + z_n + w_p + w_p + w_p \\
& \qquad\qquad = x_n + y_n + y_n + z_p + w_n + w_n + w_n + u \\
& \qquad\qquad\quad + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
\end{array}
$$

which is a $\Sigma_{\mathbb{N}}$-formula equisatisfiable to F_0.

To decide $T_{\mathbb{Z}}$-validity for a $\Sigma_{\mathbb{Z}}$-formula F:

- transform $\neg F$ to an equisatisfiable $\Sigma_{\mathbb{N}}$-formula $\neg G$,
- decide $T_{\mathbb{N}}$-validity of G.

$\Sigma_{\mathbb{Z}}$-formula to $\Sigma_{\mathbb{N}}$-formula III

Example: The $\Sigma_{\mathbb{N}}$-formula

$$\forall x.\ \exists y.\ x = y + 1$$

is equisatisfiable to the $\Sigma_{\mathbb{Z}}$-formula:

$$\forall x.\ x > -1\ \rightarrow\ \exists y.\ y > -1 \land x = y + 1.$$

Rationals and Reals

Signatures:

$$\begin{aligned}\Sigma_{\mathbb{Q}} &= \{0,\ 1,\ +,\ -,\ =,\ \geq\} \\ \Sigma_{\mathbb{R}} &= \Sigma_{\mathbb{Q}} \cup \{\cdot\}\end{aligned}$$

- Theory of Reals $T_{\mathbb{R}}$ (with multiplication)

$$x \cdot x = 2 \quad \Rightarrow \quad x = \pm\sqrt{2}$$

- Theory of Rationals $T_{\mathbb{Q}}$ (no multiplication)

$$\underbrace{2x}_{x+x} = 7 \quad \Rightarrow \quad x = \frac{7}{2}$$

Note: strict inequality okay; simply rewrite

$$x + y > z$$

as follows:

$$\neg(x + y = z) \ \wedge \ x + y \geq z$$

1. Theory of Reals $T_{\mathbb{R}}$

Signature:

$$\Sigma_{\mathbb{R}} : \{0,\ 1,\ +,\ -,\ \cdot,\ =,\ \geq\}$$

with multiplication. Axioms in text.

Example:

$$\forall a, b, c.\ b^2 - 4ac \geq 0 \ \leftrightarrow\ \exists x.\ ax^2 + bx + c = 0$$

is $T_{\mathbb{R}}$-valid.

$T_{\mathbb{R}}$ is decidable (Tarski, 1930) High time complexity

Recursive Data Structures (RDS) I

Tuples of variables where the elements can be instances of the same structure: e.g., linked lists or trees.

1. Theory T_{cons} (LISP-like lists)

Signature:

$$\Sigma_{\text{cons}} : \{\text{cons}, \text{car}, \text{cdr}, \text{atom}, =\}$$

where

cons(a, b) – list constructed by concatenating a and b
car(x) – left projector of x: $\text{car}(\text{cons}(a, b)) = a$
cdr(x) – right projector of x: $\text{cdr}(\text{cons}(a, b)) = b$
atom(x) – true iff x is a single-element list

Note: an atom is simply something that is not a cons. In this formulation, there is no NIL value.

Recursive Data Structures (RDS) II

Axioms:

1. The axioms of reflexivity, symmetry, and transitivity of $=$
2. Function Congruence axioms

$$\forall x_1, x_2, y_1, y_2.\ x_1 = x_2 \land y_1 = y_2 \ \rightarrow\ \text{cons}(x_1, y_1) = \text{cons}(x_2, y_2)$$
$$\forall x, y.\ x = y \ \rightarrow\ \text{car}(x) = \text{car}(y)$$
$$\forall x, y.\ x = y \ \rightarrow\ \text{cdr}(x) = \text{cdr}(y)$$

3. Predicate Congruence axiom

$$\forall x, y.\ x = y \ \rightarrow\ (\text{atom}(x) \ \leftrightarrow$$

T_{cons} is undecidable
Quantifier-free fragment of T_{cons} is efficiently decidable

4. $\forall x, y.\ \text{car}(\text{cons}(x, y)) = x$ (left projection)
5. $\forall x, y.\ \text{cdr}(\text{cons}(x, y)) = y$ (right projection)
6. $\forall x.\ \neg\text{atom}(x) \ \rightarrow\ \text{cons}(\text{car}(x), \text{cdr}(x)) = x$ (construction)
7. $\forall x, y.\ \neg\text{atom}(\text{cons}(x, y))$ (atom)

Note: the behavior of car and cons on atoms is not specified

Lists with equality

2. Theory T^E_{cons} (lists with equality)

$$T^E_{\text{cons}} \quad = \quad T_{\text{E}} \ \cup \ T_{\text{cons}}$$

Signature:

$$\Sigma_{\text{E}} \ \cup \ \Sigma_{\text{cons}}$$

(this includes uninterpreted constants, functions, and predicates)

Axioms: union of the axioms of T_{E} and T_{cons}

T^E_{cons} is undecidable Quantifier-free fragment of T^E_{cons} is efficiently decidable

Example: The Σ^E_{cons}-formula

is F , T^E_{cons} valid?

$$F: \quad \begin{array}{l} \text{car}(x) = \text{car}(y) \wedge \text{cdr}(x) = \text{cdr}(y) \wedge \neg\text{atom}(x) \wedge \neg\text{atom}(y) \\ \quad \rightarrow \ f(x) = f(y) \end{array}$$

Suppose not; then there exists a T_{cons}^{E}-interpretation I such that $I \not\models F$. Then,

1.	I	$\not\models$	F	assumption
2.	I	$\models$	$\text{car}(x) = \text{car}(y)$	1, $\rightarrow$, $\wedge$
3.	I	$\models$	$\text{cdr}(x) = \text{cdr}(y)$	1, $\rightarrow$, $\wedge$
4.	I	$\models$	$\neg\text{atom}(x)$	1, $\rightarrow$, $\wedge$
5.	I	$\models$	$\neg\text{atom}(y)$	1, $\rightarrow$, $\wedge$
6.	I	$\not\models$	$f(x) = f(y)$	1, $\rightarrow$
7.	I	$\models$	$\text{cons}(\text{car}(x), \text{cdr}(x)) = \text{cons}(\text{car}(y), \text{cdr}(y))$	2, 3, (function)
8.	I	$\models$	$\text{cons}(\text{car}(x), \text{cdr}(x)) = x$	4, (construction)
9.	I	$\models$	$\text{cons}(\text{car}(y), \text{cdr}(y)) = y$	5, (construction)
10.	I	$\models$	$x = y$	7, 8, 9, (transitivity)
11.	I	$\models$	$f(x) = f(y)$	10, (function)

Lines 6 and 11 are contradictory, so our assumption that $I \not\models F$ must be wrong. Therefore, F is T_{cons}^{E}-valid.

First-Order Theories

	Theory	Quantifiers Decidable	QFF Decidable
T_E	Equality	–	✓
T_{PA}	Peano Arithmetic	–	–
$T_{\mathbb{N}}$	Presburger Arithmetic	✓	✓
$T_{\mathbb{Z}}$	Linear Integer Arithmetic	✓	✓
$T_{\mathbb{R}}$	Real Arithmetic	✓	✓
$T_{\mathbb{Q}}$	Linear Rationals	✓	✓
T_{cons}	Lists	–	✓
T_{cons}^E	Lists with Equality	–	✓

Demo CVC5

- https://cvc5.github.io/

Input Format: SMT-LIB 2

- First, directives. E.g., asking models to be reported:

```
(set-option :produce-models true)
```

- Second, set background theory:

```
(set-logic QF_LIA)
```

- Standard theories of interest to us:
  - QF_LRA : quantifier-free linear real arithmetic
  - QF_LIA : quantifier-free linear integer arithmetic
  - QF_RDL : quantifier-free real difference logic
  - QF_IDL : quantifier-free integer difference logic
- SMT-LIB 2 does not allow to have mixed problems (although some solvers support it outside the standard)