Little Engines of Proof: Lecture 9

N. Shankar, L. de Moura, H. Ruess, A. Tiwari

shankar@csl.sri.com

URL: http://www.csl.sri.com/~shankar/LEP.html

Computer Science Laboratory

SRI International

Menlo Park, CA

This lecture . . .

We show how to decide the *uniform word problem* for a number of equality theories including

- Linear arithmetic
- Lists
- Propositional logic
- Sets
- Coproducts
- Finite Sequences, Bitvectors, Arrays, . . .

Recall: Word Problems

The word problem.
Given an equality theory $\mathcal{T}$, the word problem for $\mathcal{T}$ is to decide, for any two Σ terms a and b, whether or not $\mathcal{T} \models a = b$.

The uniform word problem. Given an equality theory $\mathcal{T}$ over signature Σ, the uniform word problem for $\mathcal{T}$ is to decide, for any finite set E of Σ-equations and Σ-equation $a = b$, of whether or not $\mathcal{T} \models E \Rightarrow a = b$.

Exercise. Give an example of a theory with an undecidable WP. Give an example of a theory with decidable WP but undecidable UWP.

In practice. For many theories, word problem and uniform word problem are efficiently decidable.

Application: Compiler Validation

Problem. Prove equivalence of source and target program

Example.

```
1:   y := 1
2:   if z = x*x*x
3:     then y := x*x + y
4:   endif
```

```
1:   y := 1
2:   R1 := x * x
3:   R2 := R1 * x
4:   jmpNE(z, R2, 6)
5:   y := R1 + 1
```

Verification condition.

$y_1 = 1 \wedge z_2 = x_0 * x_0 * x_0 \wedge y_3 = x_0 * x_0 + y_1 \wedge$
$y'_1 = 1 \wedge R1_2 = x'_0 * x'_0 \wedge R2_3 = R1_2 * x'_0 \wedge z'_0 = R2_3 \wedge y'_5 = R1_2 + 1 \wedge$
$x_0 = x'_0 \wedge y_0 = y'_0 \wedge z_0 = z'_0$
$\Rightarrow y_3 = y'_5$

Word problem in machine arithmetic. But wait, $*$, $+$ can be considered to be uninterpreted . . .

Open. Handle large programs, algebraic properties of ops.

Equational Linear Arithmetic

Let $\mathcal{Q}$ ($\mathcal{Z}$) be the structure of the rational (integer) numbers with linear arithmetic and without the inequality predicates.

The signature of $\mathcal{Q}$ includes

- all rational numbers q as constants,
- the binary addition operator $+$,
- for each rational number q, a unary operator $q * \_$ multiplying its argument by q.

$\mathcal{Z}$ is a subsignature of $\mathcal{Q}$ and includes only integer constants and addition.

Examples.

- $\mathcal{Q} \models 1/2 * (x + 1/3 * (y - 1/6)) = 1/6 * y$
- $\mathcal{Z} \models 3 * x = 4 + 7 * m_1 \Rightarrow 3 * x = 4 + 12 * m_2$

Lists

Lists

$$\Sigma_L = \{cons(.,.),\ car(.),\ cdr(.)\}$$

Equational theory $\mathcal{L}$ of lists axiomatized by these (implicitly universally quantified) equations

$$\begin{aligned} car(cons(x,y)) &= x \\ cdr(cons(x,y)) &= y \\ cons(car(x), cdr(x)) &= x \end{aligned}$$

Examples.

- $\mathcal{L} \models cons(car(x), y) = x$
- $\mathcal{L} \models x = cons(u, v) \Rightarrow cons(car(x), y) = x$

Canonizable Theories

A theory $\mathcal{T}$ is canonizable if there is a computable function $\sigma: \ T(\Sigma, X) \rightarrow T(\Sigma, X)$ with

- $\mathcal{T} \models a = b$ iff $\sigma(a) \equiv \sigma(b)$
- $vars(\sigma(a)) \subseteq vars(a)$
- $\sigma(b) \equiv b$ for every subterm b of $\sigma(a)$.

for all $a, b \in T(\Sigma, X)$.

A term $a \in T(\Sigma, X)$ is said to be canonical if $\sigma(b) \equiv b$.

In particular, a canonizer $\sigma_{\mathcal{T}}$ for theory $\mathcal{T}$ solves the word problem for $\mathcal{T}$.

Examples for Canonizable Theories

Canonizer for linear arithmetic as an *ordered sum of monomials*; for example, $\sigma_{\mathcal{Q}}(y + x + x) \equiv 2x + y$.

Exercise. WP for $\mathcal{Q}$ is solvable by $\sigma_{\mathcal{Q}}$.

ROBDDs as canonical forms for propositional logic

$\sigma_{\mathcal{L}}$ is obtained by orienting the list axioms $\mathcal{L}$ as rewrite rules from left to right. Induces canonical list model.

$$\begin{aligned} \mathcal{L} &:= \{a \in T(\Sigma_L, X) | \sigma_{\mathcal{L}}(a) \equiv a\} \\ \mathcal{L}(cons(l_1, l_2)) &:= \sigma_{\mathcal{L}}(cons(l_1, l_2)) \\ \mathcal{L}(car(l)) &:= \sigma_{\mathcal{L}}(car(l)) \\ \mathcal{L}(cdr(l)) &:= \sigma_{\mathcal{L}}(cdr(l)) \end{aligned}$$

Exercise Convince youself that $\mathcal{L}$ satisfies the list axioms.

An equational theory is canonizable if there is a corresponding strongly normalizing rewrite system.

Equality Sets

An *equality set* E is of the form $\{a_1 = b_1, \ldots, a_n = b_n\}$

E is *functional* if $a = b_1, a = b_2 \in E$ implies $b_1 \equiv b_2$

Functional equality sets: read equations left-to-right.

Operations on functional equality sets

Lookup $\quad E(a) := \begin{cases} b & : \quad a = b \in E \\ a & : \quad \text{otherwise} \end{cases}$

Apply:

$$\begin{aligned} E[x] &:= E(x) \\ E[f(a_1, \ldots, a_n)] &:= E(f(E[a_1], \ldots, E[a_n])) \end{aligned}$$

A *solution set* is a functional equality set of the form

$$\{x_1 = b_1, \ldots, x_n = b_n\}$$

with $x_i \notin vars(b_j)$ for $1 \leq i, j \leq n$

Preservation

A variable assignment ρ' extends ρ if

- $dom(\rho) \subseteq dom(\rho')$ and
- $\rho(x) = \rho'(x)$ for all $x \in dom(\rho)$

Let E, E' be equality sets; then: E' *$\mathcal{T}$-preserves* E if

- $vars(E) \subseteq vars(E')$
- For any $\mathcal{T}$-interpretation $\mathcal{M}, \rho$ such that $\mathcal{M}, \rho \models E$ there is a ρ' extending ρ such that $\mathcal{M}, \rho' \models E'$, and conversely whenever $\mathcal{M}, \rho' \models E'$, there is a ρ extending ρ' such that $\mathcal{M}, \rho' \models E$.

In this case: $\mathcal{T} \models E \Rightarrow a = b$ iff $\mathcal{T} \models E' \Rightarrow a = b$

Solvable Theories

A theory $\mathcal{T}$ is called solvable if there is a computable function $solve$ with

1. $solve(a = b) = \perp$ iff $a = b$ is *$\mathcal{T}$-unsatisfiable*
2. Otherwise, $solve(a = b) = S$, where S is a (functional) solution set such that
   - $dom(S) \subseteq vars(a = b)$
   - S *$\mathcal{T}$-preserves* $a = b$

Notice that fresh variables, that is, variables never being used before (`gensym`) might be introduced on right-hand sides of solved forms.

The notion of freshness can be made more precise ...

Integral Solver

Example:

$$solve_{\mathcal{Z}}(3 * x + 5 * y = 1) = \{x = -3 + 5 * k, y = 2 - 3 * k\}$$

where k is a *fresh* integral variable.

In general:
Solving a linear diophantine equation with nonzero, rational coefficients c_i, for $i = 1, \ldots, n$ with $n \geq 1$.

$$c_0 * x_0 + \ldots c_n * x_n \quad = \quad b \qquad (*)$$

Integral Solver: Particular Solutions

The case $n = 1$ is trivial

Let $n \geq 2$. Find, with the Euclidean GCD algorithm c' and integers d, e satisfying

$$c' = (c_0, c_1) = c_0 * d + c_1 * e$$

Now solve (in n variables)

$$c' * x + c_2 * x_2 + \ldots + c_n * x_n \quad = \quad b \qquad (**)$$

If equation has no integral solution, then neither has $(*)$. Otherwise, if $x, x_2, \ldots, x_n$ is an integral solution of $(**)$, then $d * x, e * x, x_2, \ldots, x_n$ gives an integral solution of $(*)$.

Integral Solver: General Solutions

Compute the general solution of a linear Diophantine equation with coefficients $(c_0 \ldots c_n)$, the gcd d of $(c_0 \ldots c_n)$, and a particular solution $(p_0 \ldots p_n)$.

In the case of four coeffients, compute, for example

$$\begin{array}{ll} & (p_0 \ p_1 \ p_2 \ p_3) \\ + & k/d * (c_1 \ \ -c_0 \ 0 \ 0) \\ + & l/d * (0 \ c_2 \ \ -c_1 \ 0) \\ + & m/d * (0 \ 0 \ c_3 \ \ -c_2) \end{array}$$

Here, [k], [l], and [m] are fresh variables.

Exercise 1 *Demonstrate that this yields indeed a solver for $\mathcal{Z}$. Design a solver for $\mathcal{Z}/(3)$.*

Deciding the UWP for Shostak Theories

A *canonizable* and *solvable* theory is also called a *Shostak theory*.

¿From now on, let $\mathcal{T}$ be a *Shostak theory* with canonizer $\sigma_{\mathcal{T}}(.)$ and solver $solve_{\mathcal{T}}$.

We consider the UWP $\mathcal{T} \models E \Rightarrow a = b$ from a solution of the WP $\mathcal{T} \models a = b$.

Template for decision procedure

1. Build a solution set S from E using a finite number of $\mathcal{T}$-preserving transformations.
2. Compute canonical forms a' and b' for a and b in S.
3. If $a' \equiv b'$ then Yes else No.

Deciding a Shostak Theory (Cont.)

Canonization.

$$S\langle\langle a \rangle\rangle := \sigma_{\mathcal{T}}(S[a])$$

Fusion.

$$S \triangleright R := \{a = R\langle\langle b \rangle\rangle \mid a = b \in S\}$$

Composition.

$$\begin{array}{lll} S \circ \bot & := & \bot \\ \bot \circ S & := & \bot \\ S \circ R & := & R \cup (S \triangleright R) \end{array}$$

Fusion can be implemented using so-called *use*-lists, which index occurrences of right-hand side variables.

Exercise. For solved forms, $S \circ S = S$.

Deciding a Shostak Theory (Cont.)

Configuration (S, E) consists of a pair consisting of the unprocessed equalities E and solution sets S.

Building a solution set

$$\frac{\{a = b\} \cup E,\ S}{E,\ S \circ T}\, assert$$

with $T := solve(S\langle\langle a \rangle\rangle = S\langle\langle b \rangle\rangle)$

Termination is immediate.

Starting with $(E, \emptyset)$, let $(\emptyset, S')$ be a corresponding irreducible configuration, then:

$$\mathcal{T} \models E \Rightarrow a = b$$

iff

either $S' = \bot$ or $S'\langle\langle a \rangle\rangle \equiv S'\langle\langle b \rangle\rangle$

Example

$$\mathcal{Z} \models (3 * x = 4 + 7 * m_1 \wedge 3 * x = 4 + 12 * m_2) \Rightarrow 0 = 1$$

$$(\{3 * x = 4 + 7 * m_1, 3 * x = 4 + 12 * m_2\}, \underbrace{\{x = x, m_1 = m_1, m_2 = m_2\}}_{S_0})$$

(assert) $\leadsto$ $(\{3 * x = 4 + 12 * m_2\}, \underbrace{\{x = -8 + 7 * k, m_1 = -4 + 3 * k, m_2 = m_2\}}_{S_1})$

(assert) $\leadsto$ $\bot$

since $S_1\langle\langle 3 * x \rangle\rangle \equiv -24 + 21 * k,\ S_2\langle\langle 4 + 12 * m_2 \rangle\rangle \equiv 4 + 12 * m_2$ and $solve_{\mathcal{Z}}(21 * k - 12 * m = 28)$ yields $\bot$.

Soundness and Completeness

- S' $\mathcal{T}$-preserves E, since each of the steps canonization, solving, composition, and assert is preserving.

  Exercise. Spell out the details.

- *Soundness of canonizer.* If $\sigma_{\mathcal{T}}(S'[a]) \equiv \sigma_{\mathcal{T}}(S'[b])$, then

  $$\mathcal{M}, \rho' \models S' \Rightarrow a = S'[a] = \sigma_{\mathcal{T}}(S'[a]) = \sigma_{\mathcal{T}}(S'[b]) = S'[b] = b$$

  Thus, $\mathcal{M}, \rho \models E \Rightarrow a = b$.

- *Completeness of canonizer.* Construct a model $\mathcal{M}$, θ such that $\mathcal{M}, \theta \models E$ but $\mathcal{M}, \theta \not\models a = b$.

Soundness and Completeness (Cont.)

When $\sigma_{\mathcal{T}}(S'[a]) \not\equiv \sigma_{\mathcal{T}}(S'[b])$

- there is a $\mathcal{T}$-model $\mathcal{M}$, θ s.t $\mathcal{M}, \theta \not\models S'[a] = S'[b]$
- wlog $x \equiv S'(x)$ for variables $x \in dom(\theta)$.
- Extend θ to an assignment θ' s.t

  $\theta'(x) := \mathcal{M}[\![S'(x)]\!]\theta$ if $x \neq S'(x)$

- $$\begin{aligned} \mathcal{M}, \theta' &\models S' \\ \mathcal{M}, \theta' &\models a = S'[a],\ S'[b] = b \end{aligned}$$

- Since S' $\mathcal{T}$-preserves $(E, \emptyset)$, $\mathcal{M}, \theta' \models E$ but $\mathcal{M}, \theta' \not\models a = b$.

Adding Disequalities

Configuration (E, D, S) consists of triples with unprocessed equalities E, disequalities D, and solution sets S.

$\dfrac{\{a = b\} \cup E, D,\ S}{E, D,\ S \circ T}$ assert	with $T := solve(S\langle\langle a \rangle\rangle = S\langle\langle b \rangle\rangle)$
$\dfrac{E, \{a \neq b\} \cup D,\ S}{E,\ S}$ bot	if $S\langle\langle a \rangle\rangle \equiv S\langle\langle b \rangle\rangle$

with $T := solve(S\langle\langle a \rangle\rangle = S\langle\langle b \rangle\rangle)$

Starting with $(E, D, \emptyset)$, let $(\emptyset, S')$ be a corresponding irreducible configuration, then: $\mathcal{T} \models E, D \Rightarrow false$ iff $S' = \bot$.

Normalizing to variable dsiequalities D might be more efficient as cnconsistency test reduces to $S(a) \equiv S(b)$.

Boolean Solver (Cont.)

Signature. $\Sigma_{\mathcal{B}} := \{true, false, ite(., ., .)\}$

Canonizer $\sigma_{\mathcal{B}}$ returns, e.g., a binary decision diagrams (ordering on variables needed)

Solver. process $a \iff b$ instead of $a = b$

$\dfrac{true,\ S}{S}\ Triv$
$\dfrac{false,\ S}{\bot}\ Bot$
$\dfrac{ite(x,\ p,\ n),\ S}{p \vee n,\ S \circ \{x = (p \wedge (n \Rightarrow \delta))\}}\ Slv$

All terms assumed to be in canonical form $\sigma_{\mathcal{B}}$

These rules induce Boolean solver $solve_{\mathcal{B}}$.

Boolean Solver (Cont.)

Termination immediate as the number of variables in processed term is decreasing.

Correctness is based on the equivalence

$$ite(x,\ p,\ n) \quad \iff \quad (p \vee n) \wedge \exists \delta.\ x = (p \wedge (n \Rightarrow \delta))$$

Example for Boolean Solver

Solve $x \wedge y = \neg x$.

This is represented by the ROBDD

$$ite(x,\ ite(y,\ false,\ true),\ false)$$

Derivation.

	$(ite(x,\ ite(y,\ false,\ true),\ false), \{x = x,\ \ y = y\})$
(ite)$\leadsto$	$(ite(y,\ false,\ true),\ \ \{x = true,\ \ y = y\})$
(ite)$\leadsto$	$(true,\ \ \{x = true,\ \ y = false\})$
(true)$\leadsto$	$\{x = true,\ \ y = false\}$