

The Shostak Combination Procedure

Natarajan Shankar, Harald Rueß

ruess@cs.sri.com

<http://www.cs.sri.com/users/ruess>.

Computer Science Laboratory

SRI International

333 Ravenswood

Menlo Park, CA 94025

Motivation

Verification conditions in HW and SW *validation* in combinations of a multitude of theories.

- Arithmetic constraints
- Lists, arrays, finite sequences, bitvectors, strings, characters, enumeration types, records, variant records, abstract datatypes ...

Examples.

- $x + 2 = y \Rightarrow f(a[x := 3][y - 2]) = f(y - x + 1)$
- $f(y - 1) - 1 = y + 1, f(x) + 1 = x - 1, x + 1 = y \Rightarrow \perp$
- $f(f(x) - f(y)) \neq f(z), y \leq x, y \geq x + z, z > 0 \Rightarrow \perp$

Alternatives.

- Encodings in a decidable logic. (e.g. MSO(.))
- Combination of decision procedures. (e.g. NO, Sh)

Shostak Combinations

Shostak at SRI from mid 70s to mid 80s.

- Sh1.** Combination procedure for deciding $SAT(\mathcal{U} + \mathcal{A})$ by reduction to $SAT(\mathcal{A})$ (Ackermann's trick), which is solved by converting to DNF followed by Simplex.
- Sh2.** Combination procedure for $SAT(\mathcal{T})$ by (eager) reduction to SAT (prime implicants for efficiency).
- Sh3.** Use of Shostak theories in a deterministic Nelson-Oppen combination [...; Tiwari; Barrett; Ganzinger; Shankar/R; Krstic/Conchon].
- Sh4.** Canonizer for the combination of Shostak theories $\mathcal{T}_1, \dots, \mathcal{T}_n$ with the free theory \mathcal{U} . (builds on [**Sh3**])

Shostak Combinations

Shostak at SRI from mid 70s to mid 80s.

- Sh1.** Combination procedure for deciding $SAT(\mathcal{U} + \mathcal{A})$ by reduction to $SAT(\mathcal{A})$ (Ackermann's trick), which is solved by converting to DNF followed by Simplex.
- Sh2.** Combination procedure for $SAT(\mathcal{T})$ by (eager) reduction to SAT (prime implicants for efficiency.).
- Sh3.** Use of Shostak theories in a deterministic Nelson-Oppen combination [...; Tiwari; Barrett; Ganzinger; Shankar/R; Krstic/Conchon].
- Sh4.** Canonizer for the combination of Shostak theories $\mathcal{T}_1, \dots, \mathcal{T}_n$ with the free theory \mathcal{U} . (builds on [**Sh3**])

Here: **Sh4.**

Example

$$2 * car(x) - 3 * cdr(x) = f(cdr(x))$$

$$\Rightarrow f(cons(4 * car(x) - 2 * f(cdr(x)), y)) = f(cons(6 * cdr(x), y))$$

Example

$$2 * car(x) - 3 * cdr(x) = f(cdr(x))$$

$$\Rightarrow f(cons(4 * car(x) - 2 * f(cdr(x)), y)) = f(cons(6 * cdr(x), y))$$

- **Process.** ($S_V : \{\};$
 $S_U : \{w = f(v)\};$
 $S_{\mathcal{L}} : \{u = car(x), v = cdr(x)\};$
 $S_{\mathcal{A}} : \{w = 2 * u - 3 * v\})$

Example

$$2 * car(x) - 3 * cdr(x) = f(cdr(x))$$

$$\Rightarrow f(cons(4 * car(x) - 2 * f(cdr(x)), y)) = f(cons(6 * cdr(x), y))$$

- **Process.** ($S_V : \{\};$
 $S_U : \{w = f(v)\};$
 $S_{\mathcal{L}} : \{u = car(x), v = cdr(x)\};$
 $S_{\mathcal{A}} : \{w = 2 * u - 3 * v\})$

- **Canonize lhs.** $4 * car(x) - 2 * f(cdr(x))$
 $\rightsquigarrow 4 * u - 2 * f(v)$
 $\rightsquigarrow 4 * u - 2 * w$
 $\rightsquigarrow 4 * u - 2 * (2 * u - 3 * v)$
 $\rightsquigarrow 6 * u$

Example

$$2 * car(x) - 3 * cdr(x) = f(cdr(x))$$

$$\Rightarrow f(cons(4 * car(x) - 2 * f(cdr(x)), y)) = f(cons(6 * cdr(x), y))$$

- **Process.** ($S_V : \{\};$
 $S_U : \{w = f(v)\};$
 $S_{\mathcal{L}} : \{u = car(x), v = cdr(x)\};$
 $S_{\mathcal{A}} : \{w = 2 * u - 3 * v\})$

- **Canonize lhs.** $4 * car(x) - 2 * f(cdr(x))$
 $\rightsquigarrow 4 * u - 2 * f(v)$
 $\rightsquigarrow 4 * u - 2 * w$
 $\rightsquigarrow 4 * u - 2 * (2 * u - 3 * v)$
 $\rightsquigarrow 6 * u$

- **Canonize rhs.** $6 * cdr(x) \rightsquigarrow 6 * u$

I Can Solve (ICS)

- **Core. Shostak combination** $\mathcal{T} = \mathcal{U} + \mathcal{A} + \mathcal{F} + \mathcal{C} + \mathcal{S} + \mathcal{B}$ of the free theory, arithmetic, arrays, combinatory logic, sets, bitvectors.
- Incomplete extensions: nonlinear arithmetic and integers.
- Decision procedure for $SAT(\mathcal{T})$ through **lazy cooperation** of a non-clausal SAT solver and the core ICS decision procedures.
- Distinguishing features.
 - It's really fast (“30000 theorems a second”), see [CAV'04].
 - API (C, Lisp, Ocaml) for embedded deduction
 - online, multi-threaded
 - suitable for e.g. symbolic simulation or proof search
- Upcoming release: complete for integers, generation of explicit models, extendable, more precise explanations, much faster.

I Can Solve (ICS)

- **Core. Shostak combination** $\mathcal{T} = \mathcal{U} + \mathcal{A} + \mathcal{F} + \mathcal{C} + \mathcal{S} + \mathcal{B}$ of the free theory, arithmetic, arrays, combinatory logic, sets, bitvectors.
- Incomplete extensions: nonlinear arithmetic and integers.
- Decision procedure for $SAT(\mathcal{T})$ through **lazy cooperation** of a non-clausal SAT solver and the core ICS decision procedures.
- Distinguishing features.
 - It's really fast (“30000 theorems a second”), see [CAV'04].
 - API (C, Lisp, Ocaml) for embedded deduction
 - online, multi-threaded
 - suitable for e.g. symbolic simulation or proof search
- Upcoming release: complete for integers, generation of explicit models, extendable, more precise explanations, much faster.
- Download at ics.csl.sri.com!

Outline

- Preliminaries
- Warm-up: Shostak's congruence closure for deciding the uniform word problem for the free theory.
- Shostak theories, definition and examples.
- Uniform word problem for Shostak theories. (“Gaussian elimination”)
- Uniform word problem for combinations of various Shostak theories with the free theory.

Outline

- Preliminaries
- Warm-up: Shostak's congruence closure for deciding the uniform word problem for the free theory.
- Shostak theories, definition and examples.
- Uniform word problem for Shostak theories. (“Gaussian elimination”)
- Uniform word problem for combinations of various Shostak theories with the free theory.

These are the core algorithms of ICS

ics.csl.sri.com

Preliminaries: Decision Problems

Let \mathcal{T} be a given algebraic Σ -theory. Terms $a, b, c, d, \dots \in T(\Sigma, X)$. Variables are implicitly universally quantified.

- **Word Problems.**

$$\models_{\mathcal{T}} a = b$$

- **Uniform Word Problems.**

$$\models_{\mathcal{T}} \bigwedge_{i=1}^n c_i = d_i \Rightarrow a = b$$

- **Clausal Validity Problems.**

$$\models_{\mathcal{T}} \bigwedge_{i=1}^n c_i = d_i \Rightarrow \bigvee_{j=1}^m a_j = b_j$$

Remark. If \mathcal{T} -validity is *convex*, then the clausal validity problem above holds iff there is $j \in \{1, \dots, m\}$ s.t. the uniform word problem $\models_{\mathcal{T}} \bigwedge_{i=1}^n c_i = d_i \Rightarrow a_j = b_j$ holds.

Preliminaries: Equality Sets

- An **equality set** E is of the form $\{a_1 = b_1, \dots, a_n = b_n\}$
- E is **functional** if $a = b_1, a = b_2 \in E$ implies $b_1 \equiv b_2$

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

Apply: $E[x] := E(x)$
 $E[f(a_1, \dots, a_n)] := E(f(E[a_1], \dots, E[a_n]))$

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

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

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

Preliminaries: 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 ψ, ψ' be sets of literals; then: ψ' **\mathcal{T} -preserves** ψ if
 - $vars(\psi) \subseteq vars(\psi')$
 - for all \mathcal{T} -structures M and assignments ρ there is some ρ' extending ρ such that

$$M, \rho \models_{\mathcal{T}} \psi \text{ iff } M, \rho' \models_{\mathcal{T}} \psi'$$

- In this case: $\models_{\mathcal{T}} \psi \Rightarrow \phi$ iff $\models_{\mathcal{T}} \psi' \Rightarrow \phi$
- This restricted notion of preservation is sufficient for our purposes, since none of the algorithms considered here introduces new function symbols.

Outline

- Preliminaries
- Warm-up: Shostak's congruence closure for deciding the uniform word problem for the free theory.
- Shostak theories, definition and examples.
- Uniform word problem for Shostak theories. (“Gaussian elimination”)
- Uniform word problem for combinations of various Shostak theories with the free theory.

Congruence Closure

Σ_F : Signature of uninterpreted function symbols

\mathcal{U}_F : Deductive closure of axioms of equality over Σ_F -terms

- Uniform word problem $\models_{\mathcal{U}_F} E \Rightarrow a = b$
- Configuration of Shostak's *congruence closure* consists of $(S_V; S_U; E)$
 - S_V contains oriented variable equalities $x = y$
 - S_U contains equalities $x = f(x_1, \dots, x_n)$
 - E contains the unprocessed input equalities.
- $(S_V; S_U)$ together form the solution state S
- S_V partitions the variables into equivalence classes
- x, y in same equivalence class if $S_V(x)$ and $S_V(y)$

Shostak's Congruence Closure

- $S := process(id_E; \emptyset; E)$ with

$$process(S_V; S_U; \emptyset) = (S_V; S_U)$$

$$process(S; \{a = b\} \cup E) = process(assert(S; a = b); E)$$

$$assert(S; a = b) = close^*(merge(abst^*(S; can_S(a = b))))$$

- Present treatment a specific strategy of **abstract** CC.
- Canonical forms:

$$can_S(x) = S_V(x)$$

$$can_S(f(a_1, \dots, a_n)) = \begin{cases} S_V(x), & \text{if } (x = f(can_S(a_1), \dots, can_S(a_n))) \in E \\ f(can_S(a_1), \dots, can_S(a_n)), & \text{otherwise.} \end{cases}$$

- For irreducible S :

$$can_S(a) \equiv can_S(b) \text{ iff } \models_{\mathcal{U}} E \Rightarrow a = b.$$

Congruence Closure (Cont.)

1. Variable abstract

$$\frac{(S_V; S_U; a[f(x_1, \dots, x_n)] = b)}{(\{x = x\} \cup S_V; \{x = f(x_1, \dots, x_n)\} \cup S_U; a[x] = b)} \quad x \text{ fresh}$$

Irreducible states of the form $(S_V; S_U; x = y)$.

2. Merge($x = y$)

$$\frac{(S_V; S_U)}{(S_V \circ \{x = y\}; S_U \triangleright \{x = y\})} \quad y \prec x$$

3. Close(S)

$$\frac{(S_V; S_U)}{\text{merge}(S_V; S_U)}$$

if $x, y \in \text{dom}(S_V)$, $S_V(x) \neq S_V(y)$, $S_U(x) \equiv S_U(y)$

Example

- Validity problem

$$\models_{\mathcal{U}} \{f(f(f(x))) = x, x = f(f(x))\} \Rightarrow f(x) = x$$

- Start state

$$S_0 := (\{x = x\}; \emptyset; \{f(f(f(x))) = x, x = f(f(x))\})$$

- Abstraction

$$\begin{aligned} & \text{abst}(\{x = x\}; \emptyset; f(f(f(x))) = x) \\ \rightsquigarrow & \left(\begin{array}{l} \{x = x, v_1 = v_1, v_2 = v_2, v_3 = v_3\} \\ \{v_1 = f(x), v_2 = f(v_1), v_3 = f(v_2)\} \\ v_3 = x \end{array} \right) \end{aligned}$$

Example (Cont.)

$$\text{merge} \rightsquigarrow \begin{pmatrix} \{x = x, v_1 = v_1, v_2 = v_2, v_3 = v_3\} \\ \{v_1 = f(x), v_2 = f(v_1), v_3 = f(v_2)\} \\ v_3 = x \end{pmatrix}$$
$$\begin{pmatrix} \{x = x, v_1 = v_1, v_2 = v_2, v_3 = x\} \\ \{v_1 = f(x), v_2 = f(v_1), v_3 = f(v_2)\} \end{pmatrix}$$

Example (Cont.)

- Variables x, y are **incongruent** in S if
 - $S_V(x) \not\equiv S_V(y)$ and
 - $S_U(\{x\}) \cap S_U(\{y\}) \neq \emptyset$
- There are no incongruences in our running example.

$$\left(\begin{array}{l} \{x = x, v_1 = v_1, v_2 = v_2, v_3 = x\} \\ \{v_1 = f(x), v_2 = f(v_1), v_3 = f(v_2)\} \end{array} \right)$$

Example (Cont.)

$$\left(\begin{array}{l} \{x = x, v_1 = v_1, v_2 = v_2, v_3 = x\} \\ \{v_1 = f(x), v_2 = f(v_1), v_3 = f(v_2)\} \end{array} \right)$$

Processing of $x = f(f(x))$. Canonization and orientation yield $v_2 = x$, which is merged

$$\left(\begin{array}{l} \{x = x, v_1 = v_1, v_2 = x, v_3 = x\} \\ \{v_1 = f(x), v_2 = f(v_1), v_3 = f(x)\} \end{array} \right)$$

The incongruence between v_1, v_3 is fixed by close

$$S^* := \left(\begin{array}{l} \{x = x, v_1 = x, v_2 = x, v_3 = x\} \\ \{v_1 = f(x), v_2 = f(v_1), v_3 = f(x)\} \end{array} \right)$$

Example (Cont.)

- Irreducible configuration S^* .

$$\left(\begin{array}{l} \{x = x, v_1 = x, v_2 = x, v_3 = x\}; \\ \{v_1 = f(x), v_2 = f(v_1), v_3 = f(x)\} \end{array} \right)$$

- Now, $can_{S^*}(f(x)) \equiv x \equiv can_{S^*}(x)$

Soundness and Completeness

For $\text{vars}(a = b) \subseteq \text{vars}(E)$ and $S := \text{process}(id_E; \emptyset; E)$

$$\models_{\mathcal{U}} E \Rightarrow a = b \text{ iff } \text{can}_S(a) \equiv \text{can}_S(b)$$

- Since S \mathcal{U} -preserves $(id_E; \emptyset; E)$,

$$\models_{\mathcal{U}} E \Rightarrow a = b \text{ iff } \models_{\mathcal{U}} S \Rightarrow a = b$$

- To show: $\models_{\mathcal{U}} S \Rightarrow a = b$ iff $\text{can}_S(a) \equiv \text{can}_S(b)$
- (\Leftarrow) $\models_{\mathcal{U}} S \Rightarrow a = \text{can}_S(a) = \text{can}_S(b) = b$.
- (\Rightarrow) assuming $\text{can}_S(a) \not\equiv \text{can}_S(b)$, construct a Σ_F -structure M and a variable assignment ρ s.t. $M, \rho \not\models E \Rightarrow a = b$.

Soundness and Completeness (Cont.)

- $D_S = \{a \in \text{Term}(\Sigma_F, \text{vars}(S)) \mid \text{can}_S(a) \equiv a\}$

- For $a_1, \dots, a_n \in D_S$:

$$M_S(f)(a_1, \dots, a_n) = \begin{cases} S_V(x), & \text{if } x = f(a_1, \dots, a_n) \in S_U \\ f(a_1, \dots, a_n), & \text{otherwise.} \end{cases}$$

- Let $\rho_S(x) = S_V(x)$ for $x \in \text{vars}(S)$

- $M_S[[c]]\rho_S = \text{can}_S(c)$ (induction on structure of c)

- Thus, $M_S, \rho_S \models S$, (*)

- and $M_S, \rho_S \not\models a = b$, (**)

since, by assumption, $\text{can}_S(a) \not\equiv \text{can}_S(b)$.

- From (*), (**), $\not\models_{\mathcal{U}} S \Rightarrow a = b$.

- Finally, since S \mathcal{U} -preserves E , $\not\models_{\mathcal{U}} E \Rightarrow a = b$.

Outline

- Preliminaries
- Warm-up: Shostak's congruence closure for deciding the uniform word problem for the free theory.
- **Shostak theories, definition and examples.**
- Uniform word problem for Shostak theories. (“Gaussian elimination”)
- Uniform word problem for combinations of various Shostak theories with the free theory.

Shostak theories

- A *canonizable* and *solvable* theory is a *Shostak theory*
- A *canonizer* σ maps terms to normal form terms s.t. equal terms in the theory are mapped to same form.
- A *solver* *solve* maps an equation to an “equivalent” solved form.
- e.g., linear arithmetic

- Canonizer returns ordered sum-of-monomials

$$\sigma_{\mathcal{A}}(x - y + 2 + y) \equiv 2 + x$$

- Rational arithmetic solver

$$qsolve(3x + 5y = 1) \equiv \left\{ x = \frac{1}{3} - \frac{1}{5}y \right\}$$

- Integral solver based on Euclid’s algorithm (k fresh)

$$euclid(3x + 5y = 1) = \{x = -3 + 5k, y = 2 - 3k\}$$

Canonizable Theories

- A theory \mathcal{T} is said to be **canonizable** if there is a computable $\sigma(a)$ such that
 - $\models_{\mathcal{T}} a = b$ iff $\sigma(a) \equiv \sigma(b)$
 - $\text{vars}(\sigma(a)) \subseteq \text{vars}(a)$
 - $\sigma(b) \equiv b$ for every subterm b of $\sigma(a)$
 - $\sigma(\sigma(a)) \equiv \sigma(a)$
- Thus σ solves the **word problem** for \mathcal{T} .
- It follows: $\models_{\mathcal{T}} \sigma(a) = a$.
- A term a is said to be **canonical** if $\sigma(a) \equiv a$
- Canonizer for linear arithmetic

$$\sigma_{\mathcal{A}}(y + x + 3 + x) \equiv 2x + y + 3$$

Solvable Theories

- A theory \mathcal{T} is called *solvable* if there is a computable procedure $solve(a = b)$
 - $solve(a = b) \equiv \perp$ iff $a = b$ is *\mathcal{T} -unsatisfiable*
 - Otherwise, $solve(a = b)$, is a (functional) solution set such that
 - $dom(S) \subseteq vars(a = b)$
 - $solve(a = b)$ *\mathcal{T} -preserves* $a = b$
- Notice that *fresh* variables, that is, variables not in $vars(a = b)$ might be introduced on right-hand sides.

Theory of Lists

- **Signature.** $\Sigma_{\mathcal{L}} := \{cons(., .), car(.), cdr(.)\}$
- **Theory \mathcal{L}** of lists axiomatized by (universally closed)

$$car(cons(x, y)) = x$$

$$cdr(cons(x, y)) = y$$

$$cons(car(x), cdr(x)) = x$$

$$car(x) \neq x$$

$$cdr(x) \neq x$$

$$car(cdr(x)) \neq x \quad \dots$$

- **Canonizer.**

$\sigma_{\mathcal{L}}(a)$ is the normal form of the terminating and confluent TRS above.

List Solver

$\frac{K; E; S}{K; \sigma_{\mathcal{L}}(R[E]); S \circ_{\mathcal{L}} R}$	if $car(x) \in \llbracket E \rrbracket$ or $cdr(x) \in \llbracket E \rrbracket$, $R = \{x = cons(y_1, y_2)\}$, y_1, y_2 fresh
$\frac{K; cons(a, b) = cons(a', b'), E; S}{K; a = a', b = b', E; S}$	
$\frac{K; a = a, E; S}{K; E; S}$	
$\frac{K; x = a, E; S}{K; \sigma_{\mathcal{L}}(\{x = a\}[E]); S \circ_{\mathcal{L}} \{x = a\}}$	if $x \in K$, $x \notin vars(a)$, $x \neq a$
$\frac{K; x = a, E; S}{K; \sigma_{\mathcal{L}}(\{x = a\}[E]); S \triangleright_{\mathcal{L}} \{x = a\}}$	if $x \notin K$, ($a \notin K$) or a non-atomic, $x \neq a$
$\frac{K; x = a, \Gamma; S}{\perp}$	if a constructor term $a \neq x, x \in vars(a)$

with

$$S \triangleright_{\mathcal{L}} R := \{x = \sigma_{\mathcal{L}}(R[a]) \mid (x = a) \in R\}$$

(Fuse)

$$S \circ_{\mathcal{L}} R := (S \triangleright_{\mathcal{L}} R) \cup R$$

(Compose)

List Solver (Cont.)

- For list equality $a = b$, let $(vars(a = b), \{\sigma_{\mathcal{L}}(a) = \sigma_{\mathcal{L}}(b)\}, \emptyset)$ be a starting configuration.
- An irreducible configuration is either
 - \perp or
 - of the form (K, \emptyset, S) with S a functional solution set with $dom(S) \subseteq K$.
- List solver inference rules are terminating and \mathcal{L} -preserving.
- In the first case, define $solve_{\mathcal{L}}(a = b)$ to be \perp and otherwise we arbitrarily choose (using Hilbert's ϵ combinator) an irreducible configuration of the form (K, \emptyset, S) and define $solve_{\mathcal{L}}(a = b) := S$.

Example

Solve $x = \mathit{cons}(\mathit{car}(x), y)$ in \mathcal{L} .

$$\{x, y\}; \{x = \mathit{cons}(\mathit{car}(x), y)\}; \emptyset$$

$$\{x, y\}; \{\mathit{cons}(z_1, z_2) = \mathit{cons}(z_1, y)\}; \{x = \mathit{cons}(z_1, z_2)\}$$

$$\{x, y\}; \{z_1 = z_1, z_2 = y\}; \{x = \mathit{cons}(z_1, z_2)\}$$

$$\{x, y\}; \{z_2 = y\}; \{x = \mathit{cons}(z_1, z_2)\}$$

$$\{x, y\}; \emptyset; \{x = \mathit{cons}(z_1, z_2), y = z_2\}$$

Booleans

- **Signature.**

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

- **Canonizer** Given an ordering on variables, $\sigma_{\mathcal{B}}$ returns equivalent reduced ordered binary decision diagrams.

- **Solver.**

$$solve_{\mathcal{B}}(a = b) = solve(a \iff b)$$

$$solve(true) = \{\}$$

$$solve(false) = \perp$$

$$solve(ITE(x, p, n)) = \{x = (p \wedge (n \implies \delta))\} \\ \circ_{\mathcal{B}} solve(p \vee n)$$

where the δ 's are fresh

Example: Boolean Solver

$$\begin{aligned} & \text{solve}_{\mathcal{B}}(x \wedge y = \neg x) \\ = & \text{solve}(\text{ITE}(x, \text{ITE}(y, \text{false}, \text{true}), \text{false})) \\ = & \{x = \text{ITE}(y, \text{false}, \text{true})\} \\ & \circ_{\mathcal{B}} \text{solve}(\text{ITE}(y, \text{false}, \text{true})) \\ = & \{x = \text{true}, y = \text{false}\} \end{aligned}$$

Outline

- Preliminaries
- Warm-up: Shostak's congruence closure for deciding the uniform word problem for the free theory.
- Shostak theories, definition and examples.
- Uniform word problem for Shostak theories.
("Gaussian elimination")
- Uniform word problem for combinations of various Shostak theories with the free theory.

Deciding a Shostak Theory

- Let \mathcal{T} be a *Shostak theory* with canonizer $\sigma_{\mathcal{T}}(\cdot)$ and solver $solve_{\mathcal{T}}(\cdot)$
- We consider the *uniform word problem*

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

- *Template* for decision procedure
 1. Build a solution set $S := process(id(E), E)$ using a finite number of \mathcal{T} -preserving transformations.
 2. Compute canonical forms $a' := can_S(a)$,
 $b' := can_S(b)$
 3. If $a' \equiv b'$ then **Yes** else **No**

Deciding a Shostak Theory (Cont.)

- *Canonization.*

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

- *Fusion.*

$$S \triangleright R := \{a = can_R(b) \mid a = b \in S\}$$

- *Composition.*

$$S \circ \perp := \perp$$

$$\perp \circ S := \perp$$

$$S \circ R := R \cup (S \triangleright R)$$

- For solved forms, $S \circ S = S$

Deciding a Shostak Theory (Cont.)

- Configuration (S, E) with
 - input equalities E , and
 - solution set $S = \{x_1 = a_1, \dots, x_n = a_n\}$
- Building a solution set (“**Gaussian Elimination**”)

$$\text{process}(S, \emptyset) = S$$

$$\text{process}(S, a = b \cup E) = \text{process}(\text{assert}(a = b, S), E)$$

$$\text{process}(\perp, E) = \perp$$

$$\text{assert}(a = b, S) = S \circ \text{solve}(\text{can}_S(a) = \text{can}_S(b))$$

- For $S = \text{process}(\text{id}(E), E)$,

$$\models_{\mathcal{T}} (E \Rightarrow a = b) \text{ iff } (S = \perp \text{ or } \text{can}_S(a) \equiv \text{can}_S(b))$$

Soundness and Completeness

Let $S := process(id(E), E)$;

- S \mathcal{T} -preserves E , that is, for every \mathcal{T} -structure M and assignment ρ

$M, \rho \models E$ iff there is a ρ' extending ρ (to the variables in $vars(S)$) such that $M, \rho' \models S$

- **Soundness.** If $\sigma(S[a]) \equiv \sigma(S[b])$, then

$$\models_{\mathcal{T}} S \Rightarrow a = S[a] = \sigma(S[a]) = \sigma(S[b]) = S[b] = b$$

Thus, $\models_{\mathcal{T}} E \Rightarrow a = b$.

- **Completeness.** Assuming $\sigma_{\mathcal{T}}(S[a]) \not\equiv \sigma_{\mathcal{T}}(S[b])$, construct M, ρ s.t. $M, \rho \models E$ and $M, \rho \not\models a = b$.

Soundness and Completeness (Cont.)

When $\sigma_{\mathcal{T}}(S[a]) \neq \sigma_{\mathcal{T}}(S[b])$

- there is a \mathcal{T} -model M, ρ s.t $M, \rho \models S[a] \neq S[b]$ with $dom(\rho) = vars(S) \setminus dom(S)$.
- Extend ρ to an assignment ρ' with $dom(\rho') = vars(S)$

$$\rho'(x) := \begin{cases} M \llbracket S(x) \rrbracket \rho & \text{if } x \in dom(S) \\ \rho(x) & \text{otherwise} \end{cases}$$

- Now, $M, \rho \models E \Rightarrow a \neq b$, since
 - by construction, $M, \rho' \models S$, and, since S \mathcal{T} -preserves E , $M, \rho \models E$.
 - $M, \rho' \models a = S[a] = \sigma_{\mathcal{T}}(S[a]) \neq \sigma_{\mathcal{T}}(S[b]) = S[b] = b$

Outline

- Preliminaries
- Warm-up: Shostak's congruence closure for deciding the uniform word problem for the free theory.
- Shostak theories, definition and examples.
- Uniform word problem for Shostak theories. (“Gaussian elimination”)
- Uniform word problem for combinations of various Shostak theories with the free theory.

Combining Shostak Theories

- **Problem.** Combination of the theory \mathcal{T}_0 of equality over UIF with several disjoint Shostak theories $\mathcal{T}_1, \dots, \mathcal{T}_n$.
- Let σ_i and $solve_i$ be the canonizer and solver for theory \mathcal{T}_i .
- A term $f(a_1, \dots, a_n)$ is an ***i-term*** if $f \in \Sigma_i$.
- A term a is a **pure *i-term*** if every subterm b of a is an *i-term*.

Composable Shostak Theories

- Resolve possible semantic incompatibilities between Shostak theories.
- **Canonical term model**
 - $D_{\mathcal{T}} := \{a \in T(\Sigma, X) \mid \sigma_{\mathcal{T}}(a) \equiv a\}$
 - For $a_1, \dots, a_n \in D_{\mathcal{T}}$,
$$M_{\mathcal{T}}(f)(a_1, \dots, a_n) := \sigma_{\mathcal{T}}(f(a_1, \dots, a_n))$$
- **Canonical term model $M_{\mathcal{T}}$ not necessarily a \mathcal{T} -model!**
- A Shostak theory \mathcal{T} is **composable** if the canonical model $M_{\mathcal{T}}$ is (isomorphic to) a \mathcal{T} -model.
- **Proposition.**

A composable Shostak theory \mathcal{T} is convex.
- **Exercise.** [Krstic] If \mathcal{T} is convex and universal, then \mathcal{T} is composable.

Term models vs. Canonization

- Let \mathcal{T} be a Shostak theory and E be \mathcal{T} -satisfiable.
- With $S := \text{process}(\text{id}(E), E)$ define, for $x \in \text{vars}(S)$

$$\rho_S(x) := \begin{cases} M_{\mathcal{T}}[S(x)]\text{id} & S(x) \not\equiv x \\ x & S(x) \equiv x \end{cases}$$

- Then, for a with $\text{vars}(a) \subseteq S$, $M_{\mathcal{T}}[a]\rho_S \equiv \sigma_{\mathcal{T}}(S[a])$.
- The proof is by induction on the structure of a .

Term models vs. Canonization

- Let \mathcal{T} be a Shostak theory and E be \mathcal{T} -satisfiable.
- With $S := process(id(E), E)$ define, for $x \in vars(S)$

$$\rho_S(x) := \begin{cases} M_{\mathcal{T}}[S(x)]id & S(x) \not\equiv x \\ x & S(x) \equiv x \end{cases}$$

- Then, for a with $vars(a) \subseteq S$, $M_{\mathcal{T}}[a]\rho_S \equiv \sigma_{\mathcal{T}}(S[a])$.
- The proof is by induction on the structure of a .

1. Case $a \equiv y$ with $S(y) \equiv y$

$$M_{\mathcal{T}}[y]\rho_S \equiv \rho_S(y) \equiv y \equiv \sigma_{\mathcal{T}}(y) \equiv \sigma_{\mathcal{T}}(S(y)) \equiv \sigma_{\mathcal{T}}(S[y])$$

Term models vs. Canonization

- Let \mathcal{T} be a Shostak theory and E be \mathcal{T} -satisfiable.
- With $S := process(id(E), E)$ define, for $x \in vars(S)$

$$\rho_S(x) := \begin{cases} M_{\mathcal{T}}[S(x)]id & S(x) \not\equiv x \\ x & S(x) \equiv x \end{cases}$$

- Then, for a with $vars(a) \subseteq S$, $M_{\mathcal{T}}[a]\rho_S \equiv \sigma_{\mathcal{T}}(S[a])$.
- The proof is by induction on the structure of a .

2. Case $a \equiv y$ with $S(y) \equiv b \not\equiv y$

$$M_{\mathcal{T}}[y]\rho_S \equiv \rho_S(y) \equiv M_{\mathcal{T}}[S(y)]id \equiv \sigma_{\mathcal{T}}(b) \equiv \sigma_{\mathcal{T}}(S(y)) \equiv \sigma_{\mathcal{T}}(S[y])$$

Term models vs. Canonization

- Let \mathcal{T} be a Shostak theory and E be \mathcal{T} -satisfiable.
- With $S := process(id(E), E)$ define, for $x \in vars(S)$

$$\rho_S(x) := \begin{cases} M_{\mathcal{T}}[S(x)]id & S(x) \not\equiv x \\ x & S(x) \equiv x \end{cases}$$

- Then, for a with $vars(a) \subseteq S$, $M_{\mathcal{T}}[a]\rho_S \equiv \sigma_{\mathcal{T}}(S[a])$.
- The proof is by induction on the structure of a .

3. Case $a \equiv f(a_1, \dots, a_n)$

$$\begin{aligned} M_{\mathcal{T}}[f(a_1, \dots, a_n)]\rho_S &\equiv M_{\mathcal{T}}(f)(M_{\mathcal{T}}[a_1]\rho_S, \dots, M_{\mathcal{T}}[a_n]\rho_S) \\ &\equiv M_{\mathcal{T}}(f)(\sigma_{\mathcal{T}}(S[a_1]), \dots, \sigma_{\mathcal{T}}(S[a_n])) \\ &\equiv \sigma_{\mathcal{T}}(f(\sigma_{\mathcal{T}}(S[a_1]), \dots, \sigma_{\mathcal{T}}(S[a_n]))) \\ &\equiv \sigma_{\mathcal{T}}(f(S[a_1], \dots, S[a_n])) \\ &\equiv \sigma_{\mathcal{T}}(S(f(a_1, \dots, a_n))) \end{aligned}$$

Convexity of Composable Theories

For a composable Shostak theory \mathcal{T}

$\models_{\mathcal{T}} E \Rightarrow a_1 = b_1 \vee \dots \vee a_n = b_n$ implies
 $\models_{\mathcal{T}} E \Rightarrow a_k = b_k$ for some k .

- wlog, E is \mathcal{T} -satisfiable. Let $S := process(id(E), E)$
- then $M_{\mathcal{T}}, \rho_S \models S$ and, since S \mathcal{T} -preserves E , $M_{\mathcal{T}}, \rho_S \models E$. (*)
- Proof by contraposition. Thus, for all $k = 1, \dots, n$ assume
 $\not\models_{\mathcal{T}} E \Rightarrow a_k = b_k$ (**)
- From (**) and $M_{\mathcal{T}}[a]\rho_S \equiv \sigma_{\mathcal{T}}(S[a])$, $can_S(a_k) \not\equiv can_S(b_k)$, and therefore, $M_{\mathcal{T}}, \rho_S \models a_k \neq b_k$ for all k . (***)
- From (*), (***), $M_{\mathcal{T}}, \rho_S \not\models E \Rightarrow a_1 = b_1 \vee \dots \vee a_n = b_n$
- Since $M_{\mathcal{T}}$ is composable, $\not\models_{\mathcal{T}} E \Rightarrow a_1 = b_1 \vee \dots \vee a_n = b_n$.

Combining Canonizers

- ...is easy: Treat alien terms as variables and apply σ_i to canonize $f(a)$ when $f \in \mathcal{T}_i$.
- For example
 1. $f(x - y) + car(x) - car(x)$ becomes $u + v - v$.
 2. $\sigma_{\mathcal{A}}(u + v - v) = u$
 3. canonical form is $f(x - y)$.
- **Chosen bijections** between variables and j -terms
$$\pi_i : X \rightarrow \{a \in T(\Sigma, X) \mid a \not\equiv f(a_1, \dots, a_n), f \in \Sigma_i\}$$
- $\pi_i[a]$ replaces **renaming variables** with corresponding alien terms $\pi_i(v)$.
- The inverse $\pi_i^{-1}[a]$ substitutes i -alien terms with corresponding variables v (assume $v \notin vars(a)$).

Combining Canonizers

- *Individual canonizers for impure terms.*

$$\sigma'_i(a) := \pi_i[\sigma_i(\pi_i^{-1}[a])]$$

- *Combined Candidate Canonizer.* $\sigma = \sigma_1 + \dots + \sigma_n$

$$\sigma(x) = x$$

$$\sigma(f_i(a_1, \dots, a_n)) = \sigma'_i(f_i(\sigma(a_1), \dots, \sigma(a_n)))$$

- *Combined Word Problem.* [Krstic/Conchon]
If \mathcal{T}_i (for $i = 1, \dots, n$) are disjoint, convex theories with canonizers σ_i , then $\sigma_1 + \dots + \sigma_n$ as defined above is a canonizer for $\mathcal{T}_1 + \dots + \mathcal{T}_n$.

Combining Canonizers

- *Individual canonizers for impure terms.*

$$\sigma'_i(a) := \pi_i[\sigma_i(\pi_i^{-1}[a])]$$

- *Combined Candidate Canonizer.* $\sigma = \sigma_1 + \dots + \sigma_n$

$$\sigma(x) = x$$

$$\sigma(f_i(a_1, \dots, a_n)) = \sigma'_i(f_i(\sigma(a_1), \dots, \sigma(a_n)))$$

- *Combined Word Problem.* [Krstic/Conchon]
If \mathcal{T}_i (for $i = 1, \dots, n$) are disjoint, convex theories with canonizers σ_i , then $\sigma_1 + \dots + \sigma_n$ as defined above is a canonizer for $\mathcal{T}_1 + \dots + \mathcal{T}_n$.
- **Our Shostak combination does not make use of such a combined canonizer $\sigma_1 + \dots + \sigma_n$.**

Combining Solvers: The Problem

...already shows up when combining Shostak theories

- Consider

$$5 + \text{car}(x + 2) = \text{cdr}(x + 1) + 3$$

in $\mathcal{T}_{\mathcal{A}} + \mathcal{T}_{\mathcal{L}}$.

- The individual theories $\mathcal{T}_{\mathcal{A}}$ (arithmetic) and $\mathcal{T}_{\mathcal{L}}$ (lists) have solvers and canonizers.
- **Assume** a combined solver which treats alien terms as variables and applies component solvers $\text{solve}_{\mathcal{A}}$ or $\text{solve}_{\mathcal{L}}$ according to the top-level symbol.

The Problem (Cont.)

- **Example**
 $5 + \text{car}(x + 2) = \text{cdr}(x + 1) + 3$
 $(\text{solve}_{\mathcal{A}}) \rightsquigarrow \text{car}(x + 2) = \text{cdr}(x + 1) - 2$
 $(\text{solve}_{\mathcal{L}}) \rightsquigarrow x + 2 = \text{cons}(\text{cdr}(x + 1) - 2, k)$
 $(\text{solve}_{\mathcal{A}}) \rightsquigarrow x = \text{cons}(\text{cdr}(x + 1) - 2, k) - 2$
- **Theorem.** [Krstic/Conchon]
Suppose $\mathcal{T}_1, \mathcal{T}_2$ are stably-infinite with non-collapsing function symbols (that is $f_i(x, \dots, x) \neq x$ is \mathcal{T}_i -satisfiable), and suppose σ_1, σ_2 are canonizers for these theories. If $\sigma_1 + \sigma_2$ is a canonizer for $\mathcal{T}_1 + \mathcal{T}_2$, then $\mathcal{T}_1 + \mathcal{T}_2$ does not have a solver.
- **Shostak combination can not use combination of solvers.**

The Solution

- Shostak theories can be combined without combining solvers
- **Key ideas**
 - Maintain theory-wise solution sets
 - Communicate variable equalities as in NO
 - Construct combined canonizer (as required in a Shostak combination)
- For $\mathcal{T}_A + \mathcal{T}_L$ configurations $S := (S_V, S_A, S_L)$ consist of
 - variable equalities S_V in canonical form
 - a solution set S_A for the theory \mathcal{T}_A
 - a solution set S_L for the theory \mathcal{T}_L

Process

$$\text{process}(S; \emptyset) := S$$

$$\text{process}(S; \{a = b\} \cup T) := \text{process}(\text{assert}(S; a = b); T)$$

$$\text{assert}(S; a = b) := \text{close}^*(\text{merge}(\text{abst}^*(S; \text{can}_S(a = b))))$$

1. **Canonize** $a = b$ w.r.t. S to get $a' = b'$.
2. **Variable abstract** $a' = b'$: Replace $f(x_1, \dots, x_n)$ by a fresh x and $x = f(x_1, \dots, x_n)$ to S_i . Iteration yields variable equality $x = y$ from $a' = b'$.
3. **Merge** $x = y$ into S to yield $(y \prec x) S_V \circ \{x = y\}$.
4. **Close** S : When x, y such that
 - $S_i(x) \equiv S_i(y)$ but $S_V(x) \not\equiv S_V(y)$, **merge** $x = y$ into S .
 - $S_V(x) \equiv S_V(y)$ but $S_i(x) \not\equiv S_i(y)$, **merge** $\text{solve}(S_i(x) = S_i(y))$ into S_i

Example

- Variable abstract $5 + \text{car}(x + 2) = \text{cdr}(x + 1) + 3$ to $v_3 = v_6$

$$\left(\begin{array}{l} \{x = x, v_1 = v_1, v_2 = v_2, v_3 = v_6, v_4 = v_4, v_5 = v_5, v_6 = v_6\} \\ \{v_1 = x + 2, v_3 = v_2 + 5, v_4 = x + 1, v_6 = v_5 + 3\} \\ \{v_2 = \text{car}(v_1), v_5 = \text{cdr}(v_4)\} \end{array} \right)$$

- Since v_3 and v_6 are merged in S_V but not in S_A , solve $S_A(v_3) = S_A(v_6)$ in A .

$$\text{solve}_A(v_2 + 5 = v_5 + 3) = \{v_2 = v_5 - 2\}$$

Example (Cont.)

...Result of solve was $\{v_2 = v_5 - 2\}$

- Compose result

$$\left(\begin{array}{l} \{x = x, v_1 = v_1, v_2 = v_2, v_3 = v_6, v_4 = v_4, v_5 = v_5, v_6 = v_6\} \\ \{v_1 = x + 2, v_3 = v_5 + 3, v_4 = x + 1, v_6 = 3 + v_5, v_2 = v_5 - 2\} \\ \{v_2 = \text{car}(v_1), v_5 = \text{cdr}(v_4)\} \end{array} \right)$$

- No new variable equalities to be propagated.
- The different solved forms of both v_2 and v_5 are tolerated, since canonizer picks a solution that is appropriate to the context.

Example (Cont.)

- Canonical state

$$\left(\begin{array}{l} \{x = x, v_1 = v_1, v_2 = v_2, v_3 = v_6, v_4 = v_4, v_5 = v_5, v_6 = v_6\}; \\ \{v_1 = x + 2, v_3 = v_5 + 3, v_4 = x + 1, v_6 = 3 + v_5, v_2 = v_5 - 2\}; \\ \{v_2 = \text{car}(v_1), v_5 = \text{cdr}(v_4)\} \end{array} \right)$$

- $5 + \text{car}(x + 2) \rightsquigarrow 5 + \text{car}(v_1) \rightsquigarrow 5 + v_2 \rightsquigarrow 3 + v_5 \rightsquigarrow v_6$
- $\text{cdr}(x + 1) + 3 \rightsquigarrow \text{cdr}(v_4) + 3 \rightsquigarrow v_5 + 3 \rightsquigarrow v_6$

Combined Canonizer

- **Component Canonizers.** ($i > 0$)

- $\sigma_i : T(\Sigma_i, X) \rightarrow T(\Sigma_i, X)$

- $\sigma'_i : T(\Sigma, X) \rightarrow T(\Sigma, X)$ is extension of σ_i that treats alien terms as variables.

- **Inverse Lookup.** ($i \geq 0$)

$$S_i^{-1}(a) = \begin{cases} S_V(x) & , \text{ if } (x = a) \in S_i \\ S_V(a) & , \text{ otherwise.} \end{cases}$$

- **Combined Canonizer.** ($i > 0$)

$$\text{can}_S(x) = S_V(x)$$

$$\text{can}_S(f_0(a_1, \dots, a_n)) = S_0^{-1}(f_0(\text{can}_S(a_1), \dots, \text{can}_S(a_n)))$$

$$\text{can}_S(f_i(a_1, \dots, a_n)) = S_i^{-1}(\sigma'_i(f_i(S_i(\text{can}_S(a_1)), \dots, S_i(\text{can}_S(a_n))))))$$

Canonical Solution States

- Invariants on $S \equiv (S_V; S_0; S_1; \dots; S_n)$
 - S_V is functional and idempotent
 - S_0 is normalized ($S_0 \triangleright S_V = S_0$)
 - S_i ($i > 0$) are solution sets, idempotent, normalized ($S_i \triangleright S_V = S_i$)
- A solution state S is **confluent** if for all $x, y \in \text{dom}(S_V)$ and $i, 0 \leq i \leq N$:

$$S_V(x) \equiv S_V(y) \iff S_i(\{x\}) \cap S_i(\{y\}) \neq \emptyset$$

- A **canonical** solution state S is confluent and satisfies the invariants above.

Properties of Combined Canonizer

For canonical S

$$\models_{\mathcal{T}} S \Rightarrow \text{can}_S(a) = a$$

$$\text{can}_S(\text{can}_S(a)) \equiv \text{can}_S(a)$$

$$\text{can}_S(f(a_1, \dots, a_n)) \equiv \text{can}_S(f(\text{can}_S(a_1), \dots, \text{can}_S(a_n)))$$

$$\text{can}_S(x) \equiv \text{can}_S(a) \text{ for } (x = a) \in S_i$$

Canonical Term Model

- Let S be a canonical solution state.

- ***Definition.***

- $D_S := \{a \in T(\Sigma, vars(S)) \mid can_S(a) \equiv a\}$
- For $a_1, \dots, a_n \in D_S$,

$$M_S(f)(a_1, \dots, a_n) := can_S(f(a_1, \dots, a_n))$$

- $\rho_S(x) := S_V(x)$

- ***Properties.***

- $M_S[[c]]\rho_S \equiv can_S(c)$
- $M_S, \rho_S \models S$

Canonical Term Model (Cont.)

Lemma. For canonical S ,

$$M_S[[a]]\rho_S \equiv \text{can}_S(a)$$

Proof. by induction on the structure of a

Canonical Term Model (Cont.)

Lemma. For canonical S ,

$$M_S[[a]]\rho_S \equiv \text{can}_S(a)$$

Proof. by induction on the structure of a

Case $a \equiv x$

$$M_S[[x]]\rho_S \equiv \rho_S(x) \equiv S_V(x) \equiv \text{can}_S(x)$$

Canonical Term Model (Cont.)

Lemma. For canonical S ,

$$M_S \llbracket a \rrbracket \rho_S \equiv \text{can}_S(a)$$

Proof. by induction on the structure of a

Case $a \equiv f(a_1, \dots, a_n)$

$$\begin{aligned} & M_S \llbracket f(a_1, \dots, a_n) \rrbracket \rho_S \\ \equiv & M_S(f)(M_S \llbracket a_1 \rrbracket \rho_S, \dots, M_S \llbracket a_n \rrbracket \rho_S) \\ \equiv & M_S(f)(\text{can}_S(a_1), \dots, \text{can}_S(a_n)) \\ \equiv & \text{can}_S(f(\text{can}_S(a_1), \dots, \text{can}_S(a_n))) \\ \equiv & \text{can}_S(f(a_1, \dots, a_n)) \end{aligned}$$

Multi-Shostak

- Consider disjoint union $\mathcal{T} = \mathcal{T}_0 + \mathcal{T}_1 + \dots + \mathcal{T}_n$ of
 - the uninterpreted equality Σ_0 -theory \mathcal{T}_0 ($= \mathcal{U}_0$),
and
 - a finite set of composable Shostak Σ_i -theories \mathcal{T}_i
($i = 1, \dots, n$).
- M is said to be an ***I*-model** if, for every $i = 1, \dots, n$,
the reduct of M to Σ_i is (isomorphic to) a \mathcal{T}_i -model.
- M is an *I*-model iff it is a \mathcal{T} -model.
 - M is a \mathcal{T} -model iff the reduct of M to Σ_i is
(isomorphic to) a \mathcal{T}_i -model for every $i = 0, \dots, n$.
 - Any Σ_0 -structure is a \mathcal{T}_0 -model.
- Uniform word problem

$$\models_I E \Rightarrow a = b \text{ iff } \models_{\mathcal{T}} E \Rightarrow a = b$$

Multi-Shostak: Process

Decision procedure

1. Compute $S := process(id(E); \emptyset)$

$$process(S; \emptyset) = S$$

$$process(S; E) = \perp, \text{ when } i : S_i = \perp$$

$$process(S; \{a = b\} \cup E) = process(assert(S; a = b); E)$$

$$assert(S; a = b) = close^*(merge_V(abst^*(S; a' = b')))$$

$$\text{where } a' = can_S(a), b' = can_S(b)$$

2. If $can_S(a) \equiv can_S(b)$ then **Yes** else **No**

Multi-Shostak: Abstraction

$$\text{abst}(S; x = y) = (S; x = y),$$

$$\text{abst}(S; a = b) = (S'; \{c = x\}[a] = \{c = x\}[b])$$

$$\text{when } S', c, i : c \in \text{max}(\llbracket a = b \rrbracket_i),$$

$$x \notin \text{vars}(S \cup \{a = b\}),$$

$$S'_V = S_V \cup \{x = x\},$$

$$S'_i = S_i \cup \{x = c\},$$

$$S'_j = S_j, \text{ for } i \neq j$$

- $\text{max}(\llbracket a = b \rrbracket_i)$ is a maximal pure i -term
- If $g(x)$ in $f(g(x))$ is replaced with y and $f(y)$ by z then $\{ y = g(x), z = f(y) \}$ is not idempotent.

Multi-Shostak: Close

$$\mathit{close}(S) = \perp, \text{ when } i : S_i = \perp_i$$

$$\mathit{close}(S) = S', \text{ when } S', i, x, y :$$

$$x, y \in \mathit{dom}(S_V),$$

$$(i > 0, S_V(x) \equiv S_V(y), S_i(x) \not\equiv S_i(y), \text{ and}$$

$$S' = \mathit{merge}_i(S; x = y))$$

or

$$(i \geq 0, S_V(x) \not\equiv S_V(y), S_i(\{x\}) \cap S_i(\{y\}) \neq \emptyset, \text{ and}$$

$$S' = \mathit{merge}_V(S; S_V(x) = S_V(y)))$$

$$\mathit{close}(S) = \mathit{normalize}(S), \text{ otherwise.}$$

$$\mathit{normalize}(S) = (S_V; S_0; S_1 \triangleright S_V; \dots; S_N \triangleright S_V).$$

Multi-Shostak: Merge

$$\text{merge}_i(S; x = y) = S', \text{ where } i > 0,$$

$$S'_i = S_i \circ_i \text{solve}(S_i(x) = S_i(y)),$$

$$S'_j = S_j, \text{ for } i \neq j,$$

$$S'_V = S_V.$$

$$\text{merge}_V(S; x = x) = S$$

$$\text{merge}_V(S; x = y) = (S_V \circ R; S_0 \triangleright R; S_1; \dots; S_N)$$

where $R = \text{orient}(x = y)$.

Soundness and Completeness

Theorem. Let $\Sigma = \Sigma_0 + \dots + \Sigma_n$ with $\Sigma_i \cap \Sigma_j = \emptyset$ (for $i \neq j$). Let \mathcal{T} be the union of

- the uninterpreted Σ_0 -theory \mathcal{T}_0
- and \mathcal{T}_i ($i = 1, \dots, n$) be composable Σ_i -Shostak theories.

Furthermore, let $S := process^*(id(E); E)$ and $I = \{1, \dots, n\}$; then:

$$\begin{aligned} & \models_I E \Rightarrow a = b \\ & \text{iff} \\ & \text{either } S = \perp \text{ or } can_S(a) \equiv can_S(b) \end{aligned}$$

Soundness and Completeness

Theorem. Let $\Sigma = \Sigma_0 + \dots + \Sigma_n$ with $\Sigma_i \cap \Sigma_j = \emptyset$ (for $i \neq j$). Let \mathcal{T} be the union of

- the uninterpreted Σ_0 -theory \mathcal{T}_0
- and \mathcal{T}_i ($i = 1, \dots, n$) be composable Σ_i -Shostak theories.

Furthermore, let $S := process^*(id(E); E)$ and $I = \{1, \dots, n\}$; then:

$$\begin{aligned} & \models_I E \Rightarrow a = b \\ & \text{iff} \\ & \text{either } S = \perp \text{ or } can_S(a) \equiv can_S(b) \end{aligned}$$

Proof Outline

- If $S := process(id(E); E)$, then
 - S I -preserves E , and
 - S is canonical.
- **Soundness.** If $can_S(a) \equiv can_S(b)$, then

$$\models_I S \Rightarrow a = can_S(a) = can_S(b) = b$$

Since S I -preserves E ,

$$\models_I E \Rightarrow a = b$$

- **Completeness** by contraposition:
if $can_S(a) \not\equiv can_S(b)$ then $\not\models_I E \Rightarrow a = b$

Proof Outline (Cont.)

Assume. $can_S(a) \not\equiv can_S(b)$

We show. for the canonical term model M_S and ρ_S

- $M_S, \rho_S \models S$,
since $M_S[[c]]\rho_S \equiv can_S(c)$ and $can_S(\cdot)$ validates all equalities in S .
- $M_S, \rho_S \not\models a = b$,
follows from $M_S[[c]]\rho_S \equiv can_S(c)$ and the assumption $can_S(a) \not\equiv can_S(b)$.
- M_S is an I -model,
since the reduct of M_S to Σ_i is isomorphic to M_i for each i ($1 \leq i \leq n$).

Since S I -preserves E , $\not\models_I E \Rightarrow a = b$.

Proof Outline (Cont.)

- The reduct of the canonical term model M_S to Σ_i is isomorphic to the canonical i -term model M_i ($i > 0$).
- **Define.** For $i > 0$ and $a \in D_S$:

$$\mu_i(a) \quad : \quad \pi_i^{-1}[S_i(a)]$$

- μ_i is an isomorphism between M_S and M_i , since
 - μ_i is a bijection between D_S and D_i , and
 - the reduct of μ_i to Σ_i is a homomorphism, since

$$M_i(f)(\mu_i(a_1), \dots, \mu_i(a_n)) \equiv \mu_i(M_S(f)(a_1, \dots, a_n))$$

for $a_1, \dots, a_n \in D_S$ and $f \in \Sigma_i$.

Proof Outline (Cont.)

$$\mu_i(M_S(f)(a_1, \dots, a_n))$$

$$(\text{def } M_S) \equiv \mu_i(\text{can}_S(f(a_1, \dots, a_n)))$$

$$\left(\begin{array}{l} \text{def } \text{can}_S(\cdot), \\ a_i \in M_S \end{array} \right) \equiv \mu_i(S_i^{-1}(\sigma'_i(f(S_i(a_1), \dots, S_i(a_n))))))$$

$$(\text{def } \mu_i) \equiv \pi_i^{-1}[\sigma'_i(f(S_i(a_1), \dots, S_i(a_n)))]$$

$$(\text{def } \sigma'_i) \equiv \pi_i^{-1}[\pi_i[\sigma_i(f(\pi_i^{-1}[S_i(a_1)], \dots, \pi_i^{-1}[S_i(a_n)]))]]$$

$$(\pi_i \text{ bij.}) \equiv \sigma_i(f(\pi_i^{-1}[S_i(a_1)], \dots, \pi_i^{-1}[S_i(a_n)]))$$

$$(\text{def } \mu_i) \equiv \sigma_i(f(\mu_i(a_1), \dots, \mu_i(a_n)))$$

$$(\text{def } M_i) \equiv M_i(f)(\mu_i(a_1), \dots, \mu_i(a_n))$$

Convexity Revisited

- *Property.* Let \mathcal{T} be defined as in Shostak combination. I -validity, and therefore \mathcal{T} -validity is convex.
- Proof is analogous to the one with only a single Shostak theory.
- **clausal \mathcal{T} -validity reduced to uniform word problems.**
- Since every convex Shostak theory is composable:
For pairwise disjoint, convex Shostak theories $\mathcal{T}_1, \dots, \mathcal{T}_n$, the combined theory $\mathcal{T}_1 + \dots + \mathcal{T}_n$ is convex.
- More generally, the Shostak condition might be dropped by doing a similar proof using a NO for convex theories.

Summary

- Decision procedure based on Shostak's ideas for the combination of equality over UIF and disjoint, composable Shostak theories.
- Key idea: separate solution sets for individual theories.
- Variable dependencies can be cyclic across theories.
- Shostak decision procedures are *incremental*.
- Generation of proof objects (similar to LICS'01).
- Adaptations for combining unification and matching algorithms with constraint solving in Shostak theories.
- Shostak combination a *refinement* of NO combination. (Ganzinger, Rueß, Shankar; 2004).
- **Added advantage is a global canonizer.**