Reasoning and Solving Modulo (Intruder) Theories

Christophe Ringeissen

INRIA Nancy - Grand Est & LORIA

Pesto Seminar, January 20, 2023

Outline

Outline

Research Interests

- First-Order Logic wih **Equality**
- Automated reasoning, e.g., **equational** theorem proving
- Satisfiability Modulo Theorie**s** (SMT)
- **Equational** theories, e.g., intruder theories
- Constraint solving, e.g., (dis)unification/matching
- Decision procedures for equational reasoning, e.g., the ones for the deduction and the static equivalence
- Rewriting techniques
- Declarative programming, e.g., rule-based programming and constraint programming
- and last but not least ... **Combination** of reasoners/solvers/procedures for **unions** of theories

Combination Problem

A general issue: Given reasoners/solvers known for single theories T_1 and T_2, how to build a reasoner/solver for the union of theories $T_1 \cup T_2$?

Why? Because a problem is usually expressed using several theories

Theories are usually assumed to be signature-disjoint, the equality being the only shared symbol

Well-known combination methods (disjoint case):

- Unification: Schmidt-Schauss
- Matching: Nipkow
- (Dis)unifiability: Baader-Schulz
- Satisfiability Modulo Theories: Nelson-Oppen
- Deduction and Static Equivalence: Cortier-Delaune

Combination Method: Disjoint Case

Satisfiability Modulo Theories [Nelson and Oppen, 1979]:

Nelson-Oppen combination method is sound but not always complete. To get completeness, assuming *stably infinite* theories is the usual way, but it is restrictive...

Research directions: go beyond stable infiniteness via *politeness*

- A polite theory is combinable with any disjoint theory
- A theory modeling a data structure should be polite [Chocron et al., 2020, Sheng et al., 2021, Sheng et al., 2022]
- Rewrite-based satisfiability procedures to show *politeness*
- Satisfiability procedures based on congruence closure methods (with Laurent Vigneron)

Combination Method: Non-Disjoint Case

Satisfiability Modulo Theories [Ghilardi, 2004]:

It provides a combination method à la Nelson-Oppen for which completeness is based on a model-theoretical framework introducing the notion of T_0-compatibility.

An alternative to non-disjoint combination: consider shared constructor symbols modulo an equational theory E, e.g.,

$$E = AC(+) = \{(x + y) + z = x + (y + z), x + y = y + x\}$$

Remark: $AC(+)$ is an example of a *permutative* theory E, i.e., for any $l = r \in E$ and any (variable/function) symbol s, the number of occurrences of s in l is equal to the one in r

E-Constructed Theories

A theory F is *E-constructed* if there exists a normalizing mapping NF satisfying some properties including

$$s =_{F \cup E} t \text{ iff } NF(s) =_E NF(t)$$

and for any function symbol f in E,

$$NF(f(t_1, \ldots, t_n)) =_E f(NF(t_1), \ldots, NF(t_n))$$

Consequence: $F \cup E$-equality is decidable if NF is computable and E-equality is decidable.

Remark: the definition of an E-constructed theory does not require that NF is computable.

Result [Erbatur et al., 2022]: the class of E-constructed theories is closed by union sharing only the symbols in E.

E-Constructed Theories: Examples

- Pairing

$$R_{\mathcal{P}} = \left\{ \begin{array}{l} fst(p(x,y)) \to x \\ snd(p(x,y)) \to y \end{array} \right\}$$

$(R_{\mathcal{P}}, \emptyset)$ is $\emptyset$-constructed, $\emptyset$ being the empty theory over the binary symbol p

- Key Exchange

$$K = \{keyex(x, pk(u), y, pk(v)) = keyex(u, pk(x), v, pk(y))\}$$

K is $\emptyset$-constructed, $\emptyset$ being the empty theory over the unary symbol pk

- Distributive Exponentiation

$$R_{\mathcal{E}} = \left\{ \begin{array}{l} exp(exp(x,y),z) \to exp(x, y \circledast z) \\ exp(x * y, z) \to exp(x,z) * exp(y,z) \end{array} \right\}$$

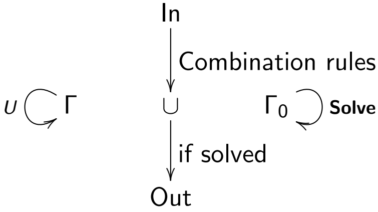
$$R_{\mathcal{F}} = \{enc(enc(x,y),z) \to enc(x, y \circledast z)\}$$

$(R_{\mathcal{E}}, AC)$ and $(R_{\mathcal{F}}, AC)$ are AC-constructed for $AC = AC(\circledast)$

Hierarchical Solvers

A hierarchical solver $H_E(U)$ for $F \cup E$ is given by:

1. some fixed combination rules, to transform the input into a *separate form* $\Gamma \cup \Gamma_0$ such that
   - Γ_0 is built over symbols in E
   - Γ is built over symbols **not** in E
2. a **Solve** algorithm to solve Γ_0 modulo E,
3. an additional inference system U to simplify Γ modulo $F \cup E$.



Syntactic Theories

A class of theories initially studied by Kirchner, Klay, Nipkow, Jouannaud, Comon, ...

In a syntactic theory, there exists a finite set U of mutation rules such that U plus the classical syntactic decomposition rule is sound and complete to simplify equations.

Example: Commutativity $(+)$
$x + y = u + v \vdash (x = u, y = v) \vee (x = v, y = u)$

Other Examples:

- Shallow theories (any variable occurs at depth at most 1 in any axiom)
- Associativity-Commutativity
- Distributive exponentiation
- Theories with the Finite Variant Property, including subterm convergent Term Rewrite Systems (TRSs)

Combined Hierarchical Unification

Individual theory: if $F \cup E$ is syntactic and F is E-constructed, then $F \cup E$ admits a hierarchical unification procedure $H_E(U)$

Union of theories: Given a hierarchical unification procedure $H_E(U_i)$ for $F_i \cup E$ and any $i = 1, 2$, under which conditions do we have that $H_E(U_1 \cup U_2)$ is a hierarchical unification procedure for $F_1 \cup F_2 \cup E$?

Problem considered in several recent papers:

- Terminating hierarchical unification procedures [Erbatur et al., 2020b]
- Hierarchical unification for theories closed by equational paramodulation [Erbatur et al., 2021]
- Hierarchical matching [Erbatur et al., 2022]

Reasoning and Solving Tools

- UNIF: a solver implementing several equational unification algorithms, developed by M. Adi (1989-), with a focus on AC-unification
- ELAN: a rewrite engine for efficient equational rewriting, developed in the Protheo group (1992-), with a focus on AC-rewriting and similar to Maude
- TOM: a matching engine embedded into an imperative programming language (C/Java), developed in the follow-up of Protheo (1999-)
- haRVey: a SMT solver implementing rewrite-based satisfiability procedures, developed by S. Ranise and D. Déharbe (2002-)

Outline

Two Notions of Knowledge

Two decision problems used to express the knowledge modulo an equational theory

1. Deduction: given a sequence of messages S and a message M, can we deduce/compute M from S?

   ➽ Example: a secret m can be deduced from the messages $X = enc(m, k)$ and $Y = k$ by considering $dec(X, Y)$ and the axiom $dec(enc(V, K), K) = V$.

2. Static Equivalence: given two sequences of messages S_1 and S_2, can we distinguish an instance of a protocol running S_1 from one running S_2?

   ➽ important for voting protocols.

Both problems are **static**: only messages are considered, without taking into account the processes that generate them.

Proof System for the Deduction

Remark: The following inference system generates all the terms deducible from ϕ, but it does not provide a decision procedure...

$$\frac{}{\nu\tilde{n}.\sigma \vdash_E M} \text{ if } \exists x \in Dom(\sigma) \text{ s.t. } x\sigma = M$$

$$\frac{}{\nu\tilde{n}.\sigma \vdash_E s} \text{ if } s \notin \tilde{n}$$

$$\frac{\phi \vdash_E M_1, \ \ldots, \phi \vdash_E M_k}{\phi \vdash_E f(M_1, \ \ldots, \ M_k)} \text{ if } f \in \Sigma$$

$$\frac{\phi \vdash_E M}{\phi \vdash_E M'} \text{ if } M =_E M'$$

Figure: Deduction Axioms

Knowledge Decidability

Undecidable in general, but critical to the analysis of security protocols. However, decision procedures are known for particular theories
[Abadi and Cortier, 2006, Comon-Lundh and Treinen, 2003, Ștefan Ciobâcă et al., 2012]

- Subterm convergent theories
- Theories of Homomorphism
- Blind signatures
- Trap-door commitments
- Malleable encryption
- and more

Computing Knowledge in Combined Theories

Decision procedures have already been developed for the two notions of knowledge in combined theories $F \cup E$

- F and E are signature disjoint [Cortier and Delaune, 2010]
- F and E share only constructors modulo the empty theory [Erbatur et al., 2017]
- some particular theories $F \cup E$ where E is the empty theory or AC [Abadi and Cortier, 2006]
- F is given by a subterm E-convergent TRS where E is syntactic permutative [Erbatur et al., 2020a]

Subterm Equational Convergent TRS

Definition: A *subterm E-convergent* TRS is a TRS such that $\rightarrow_{R,E}$ is convergent modulo E and for any $l \rightarrow r$ in R, r is a strict subterm of l or a ground constant.

Example: Abelian Pre Group

$$APG = \left\{ \begin{array}{l} x * e \rightarrow x \\ x * i(x) \rightarrow e \\ i(i(x)) \rightarrow x \\ i(e) \rightarrow e \end{array} \right\} \cup \{x * y = y * x\}$$

APG-unification successfully studied in [Yang et al., 2014] using a variant-based approach.
What about the deduction and static equivalence in APG?

Knowledge in Subterm Modulo Shallow Permutative Theories

Decision procedures for the two notions of knowledge in combined theories $RE = R \cup E$, where

- R is a subterm E-convergent TRS
- E is shallow permutative, e.g., C (Commutativity)

via Reduction Lemmas to the empty theory. These reductions hold since E is shallow.

See [Erbatur et al., 2020a] for more details

Knowledge in Subterm Modulo Syntactic Permutative Theories

Decision procedures for the two notions of knowledge in combined theories $RE = R \cup E$, where

- R is a subterm E-convergent TRS
- E is syntactic permutative and the size of R modulo E is computable

via Reduction Lemmas to E instead of the empty theory used for the shallow permutative case

See [Erbatur et al., 2020a] for more details

Outline

Beyond Subterm

The procedures developed for the knowledge problems have been proven to work for the class of subterm convergent theories.

Many of these same procedures also work for theories that are *beyond subterm*.

However, since these examples don't fit into a known class of theories for which soundness and completeness proofs already exist, they must be proven on an individual basis.

Beyond Subterm: Example

For example, the procedures of [Abadi and Cortier, 2006, Ștefan Ciobâcă et al., 2012] are shown to work on the theory of blind signatures:

Subterm:

$$\begin{aligned} checksign(sign(x,y), pk(y)) &\rightarrow x, \\ unblind(blind(x,y),y) &\rightarrow x, \end{aligned}$$

Non-subterm:

$$unblind(sign(blind(x,y),z),y) \rightarrow sign(x,z)$$

Goal

Can we develop a, hopefully simple, definition that extends the subterm convergent definition and encompasses the "beyond subterm" examples?

Joint work with Saraid Dwyer Satterfield (UMW), Serdar Erbatur (UT Dallas), Andrew Marshall (UMW), presented at the UNIF 2022 workshop

Graph-embedding

We define, $\rightarrow^*_{R_{gemb}}$, to be the reduction relation induced by the set of rewrite rules created after instantiating the following rule schema, R_{gemb}, with Σ:

For any $f \in \Sigma$
(1) $f(x_1, \ldots, x_n) \rightarrow x_i$

(2) $f(x_1, \ldots, \; x_{i-1}, \; x_i, \; x_{i+1} \; \ldots, \; x_n) \rightarrow f(x_1, \ldots, \; x_{i-1}, \; x_{i+1}, \; \ldots, \; x_n)$

For any $f, g \in \Sigma$
(3) $f(x_1, \ldots, x_{i-1}, g(\bar{z}), x_{i+1}, \; \ldots, x_m) \rightarrow g(x_1, \ldots, x_{i-1}, \bar{z}, x_{i+1}, \ldots, x_m)$

(4) $f(x_1, \ldots, x_{i-1}, g(\bar{z}), x_{i+1}, \ldots, x_m) \rightarrow f(x_1, \ldots, x_{i-1}, \bar{z}, x_{i+1}, \ldots, x_m)$

Graph-embedded Systems

We say a term t' is graph embedded in a term t, denoted $t' \succcurlyeq_{gemb} t$, if t' is a well-formed term and $t \rightarrow^*_{R_{gemb}} s \approx t'$ for some well-formed term s.

- $s \approx t'$ represent equivalence modulo an appropriate form of permutation (extending leaf permutation)

A TRS R is *graph-embedded* if for any $l \rightarrow r \in R$, $r \succcurlyeq_{gemb} l$.

Example: Malleable Encryption

Theory of malleable encryption is defined by R_{mal}:

$$\begin{aligned} dec(enc(x, y), y) &\rightarrow x \\ mal(enc(x, y), z) &\rightarrow enc(z, y) \end{aligned}$$

Simple toy example used as a test case for several procedures. For the final rule:

$$\begin{aligned} mal(enc(x, y), z) &\rightarrow_{R_{gemb}} enc(x, y, z) \\ &\rightarrow_{R_{gemb}} enc(y, z) \approx enc(z, y) \end{aligned}$$

Notice that $enc(x, y, z)$ is not well formed since it violates the arity of $enc()$. However, the final term is well formed, as required.

Example: Trap-Door Commitment

Theory of trap-door commitment, R_{tdc},
from [Ștefan Ciobâcă et al., 2012], is also graph-embedded:

$$\begin{aligned} open(td(x,y,z),y) &\rightarrow x \\ open(td(x,y,z),f(x_1,y,z,x_2)) &\rightarrow x_2 \\ td(x_2,f(x_1,y,z,x_2),z) &\rightarrow td(x_1,y,z) \\ f(x_2,f(x_1,y,z,x_2),z,x_3) &\rightarrow f(x_1,y,z,x_3) \end{aligned}$$

Example: Blind Signatures

The theory of blind signatures is also a graph-embedded TRS. All but the final rule are subterm. For the final rule:

$unblind(sign(blind(x, y), z), y) \rightarrow_{R_{gemb}} sign(blind(x, y), z)$ via rule (1)
$sign(blind(x, y), z) \rightarrow_{R_{gemb}} sign(x, y, z)$ via rule (3)
$sign(x, y, z) \rightarrow_{R_{gemb}} sign(x, z) \approx sign(x, z)$ via rule (2)

Local Stability

[Abadi and Cortier, 2006]:

- A convergent TRS, R
- For every frame $\phi = \nu \tilde{n}.\{M_1/x_1, \ldots, M_k/x_k\}$, there exists a finite set $\mathrm{sat}(\phi)$ such that:
  - each M_i is in $\mathrm{sat}(\phi)$,
  - any subterm of ϕ that can be formed from elements of $\mathrm{sat}(\phi)$ by application of function symbols is also in $\mathrm{sat}(\phi)$,
  - and it is closed under the application of small context.

Basically, it represents the intruder's knowledge based on what they can see as the protocol runs

Local Stability: Examples

Subterm Convergent Theories are locally stable [Abadi and Cortier, 2006].

The procedure of [Abadi and Cortier, 2006] also works for many other examples but local stability must be proven individually:

- blind signatures
- theory of addition
- theory of prefix with pairing
- and more

Contracting Convergent Systems

Possibility to identify a "large" subclass of graph-embedded convergent systems, called **contracting** convergent systems, for which any system in that subclass is locally stable.

A (tentative) definition:

- Rule (3) is forbidden.
- When rule (1) $f(\bar{x}) \to x_i$ is applied below the root position, only a variable instance applies, and there exists a rule $l'[f(\bar{x})] \to x_i$ if x_i is not removed later.
- When rule (4) $f(\dots, g(\bar{z}), \dots) \to f(\dots, \bar{z}, \dots)$ is applied, only a variable instance applies, and there exists a rule $l_i'[g(\bar{z})] \to z_i$ for each z_i not removed later.
- $\approx$ corresponds to the permutation of the direct subterms of the root term plus the permutation of leaves. If this is a way to get a rule $l[C[x]] \to r$ where x occurs in r without its cap C, then there exists a rule $l'[C[x]] \to x$.

Main Results (Work in Progress)

Theorem (decidability result): Any contracting convergent TRS R is locally stable. Consequently, both deduction and static equivalence are decidable for R.

Main Results (Work in Progress)

Theorem (decidability result): Any contracting convergent TRS R is locally stable. Consequently, both deduction and static equivalence are decidable for R.

Theorem (undecidability result): There exists a graph-embedded convergent TRS, say PE, for which deduction modulo PE is undecidable.

Proof: an encoding of the (modified) PCP (Post Correspondence Problem) à la [Anantharaman et al., 2012] used initially to get undecidability of unification.
[New] The same TRS as in [Anantharaman et al., 2012] can be applied to deduction as well, considering PCP.

Future Work

- A conference submission on *beyond subterm*
- Constructors defined via normalizing mappings vs. constructors defined via reduction orderings
- A journal submission on hierarchical unification
- Knowledge problems in unions of theories sharing only constructors modulo E
- Hierarchical approach applied to disunification? And to the knowledge problems?
- Congruence closure methods and syntactic theories

References I

Abadi, M. and Cortier, V. (2006).
Deciding knowledge in security protocols under equational theories.
Theor. Comput. Sci., 367(1-2):2–32.

Anantharaman, S., Lin, H., Lynch, C., Narendran, P., and Rusinowitch, M. (2012).
Unification modulo homomorphic encryption.
J. Autom. Reason., 48(2):135–158.

Chocron, P. D., Fontaine, P., and Ringeissen, C. (2020).
Politeness and combination methods for theories with bridging functions.
J. Autom. Reason., 64(1):97–134.

Comon-Lundh, H. and Treinen, R. (2003).
Easy intruder deductions.
In Dershowitz, N., editor, *Verification: Theory and Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, volume 2772 of *Lecture Notes in Computer Science*, pages 225–242. Springer.

Cortier, V. and Delaune, S. (2010).
Decidability and combination results for two notions of knowledge in security protocols.
Journal of Automated Reasoning, 48(4):441–487.

Ştefan Ciobâcă, Delaune, S., and Kremer, S. (2012).
Computing knowledge in security protocols under convergent equational theories.
J. Autom. Reasoning, 48(2):219–262.

References II

Erbatur, S., Marshall, A. M., and Ringeissen, C. (2017).
Notions of knowledge in combinations of theories sharing constructors.
In de Moura, L., editor, *Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction, Gothenburg, Sweden, Proceedings*, volume 10395 of *LNCS*, pages 60–76. Springer.

Erbatur, S., Marshall, A. M., and Ringeissen, C. (2020a).
Computing knowledge in equational extensions of subterm convergent theories.
Math. Struct. Comput. Sci., 30(6):683–709.

Erbatur, S., Marshall, A. M., and Ringeissen, C. (2020b).
Terminating non-disjoint combined unification.
In Fernández, M., editor, *Logic-Based Program Synthesis and Transformation - 30th International Symposium, LOPSTR 2020, Bologna, Italy, September 7-9, 2020, Proceedings*, volume 12561 of *Lecture Notes in Computer Science*, pages 113–130. Springer.

Erbatur, S., Marshall, A. M., and Ringeissen, C. (2021).
Non-disjoint combined unification and closure by equational paramodulation.
In Konev, B. and Reger, G., editors, *Frontiers of Combining Systems - 13th International Symposium, FroCoS 2021, Birmingham, UK, September 8-10, 2021, Proceedings*, volume 12941 of *Lecture Notes in Computer Science*, pages 25–42. Springer.

Erbatur, S., Marshall, A. M., and Ringeissen, C. (2022).
Combined hierarchical matching: the regular case.
In Felty, A. P., editor, *7th International Conference on Formal Structures for Computation and Deduction, FSCD 2022, August 2-5, 2022, Haifa, Israel*, volume 228 of *LIPIcs*, pages 6:1–6:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

References III

Ghilardi, S. (2004).
Model-theoretic methods in combined constraint satisfiability.
J. Autom. Reason., 33(3-4):221–249.

Nelson, G. and Oppen, D. C. (1979).
Simplification by cooperating decision procedures.
ACM Trans. Program. Lang. Syst., 1(2):245–257.

Sheng, Y., Zohar, Y., Ringeissen, C., Lange, J., Fontaine, P., and Barrett, C. W. (2022).
Polite combination of algebraic datatypes.
J. Autom. Reason., 66(3):331–355.

Sheng, Y., Zohar, Y., Ringeissen, C., Reynolds, A., Barrett, C. W., and Tinelli, C. (2021).
Politeness and stable infiniteness: Stronger together.
In Platzer, A. and Sutcliffe, G., editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 148–165. Springer.

Yang, F., Escobar, S., Meadows, C., Meseguer, J., and Narendran, P. (2014).
Theories of homomorphic encryption, unification, and the finite variant property.
In *Proceedings of the 16th International Symposium on Principles and Practice of Declarative Programming*, PPDP '14, pages 123–133, New York, NY, USA. ACM.

UNIF 2023 Call for Papers

Call for Papers
UNIF 2023
The 37th International Workshop on Unification
Rome, Italy, July 2, 2023
A satellite workshop of CADE/FSCD, affiliated with FSCD
`https://project.inria.fr/unif2023`

UNIF 2023 is the 37th event in a series of international meetings devoted to unification theory and its applications.

Submissions on applications of unification to security protocols are very welcome!

Submission deadline: April 21, 2023

Deduction Problem: Reduction Lemma

$\phi \vdash_{RE} t$ iff $\phi_* \vdash t$
where ϕ_* is a new frame defined as the **completion** of ϕ

Fortunately, ϕ_* is **computable** thanks to a fixpoint computation enumerating the finitely many subterms occurring in ϕ

Static Equivalence: Reduction Lemma

$\phi \approx_{RE} \psi$ iff $\psi \models Eq(\phi)$ and $\phi \models Eq(\psi)$
where

- $\psi \models Eq(\phi)$ denotes the fact that for any $s = t \in Eq(\phi), (s =_{RE} t)\psi$
- ζ_ϕ is the *recipe substitution* associated to ϕ_*
- $Eq(\phi)$ contains only finitely many equalities $s\zeta_\phi = t\zeta_\phi$ such that $(s\zeta_\phi =_{RE} t\zeta_\phi)\phi$ and s, t are bounded "public" terms

Lifting of Two Technical Lemmas

Soundness and completeness are proven using two technical lemmas:

Equational step Assume $\psi \models Eq(\phi)$. If $s\phi_* =_E t\phi_*$, then $(s\zeta_\phi)\psi =_{RE} (t\zeta_\phi)\psi$

Rewrite step Assume $\psi \models Eq(\phi)$. If $s\phi_* \rightarrow_R t$, then there exists a term u satisfying the name restriction such that $t = u\phi_*$ and $(s\zeta_\phi)\psi =_{RE} (u\zeta_\phi)\psi$