

International School on Rewriting (ISR 2012) in the Alan Turing Year

MUG: Matching, Unification, Generalizations Part 2

Temur Kutsia
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University Linz, Austria

Overview

Part 1

Syntactic unification and matching

Part 2

Equational unification and matching

Overview

Part 1

Syntactic unification and matching

Part 2

Equational unification and matching

Motivation

- Equational matching and unification algorithms are used in
 - rewriting and completion modulo equalities,
 - automated reasoning,
 - logic programming with equalities,
 - ...

Motivation

- Equational unification is a dual problem for the word problem.
- E : A given set of equalities.
- Word problem:
Does $\forall \bar{x}. s \doteq t$ hold in all models of E ?
- Equational unification:
Does $\exists \bar{x}. s \doteq t$ hold in all nonempty models of E ?

Motivation

- Equational unification generalizes syntactic unification.
- $f(x, y) \doteq f(a, b)$ has only one mgu $\{x \mapsto a, y \mapsto b\}$, if it is a syntactic unification problem.
- If f is commutative, then $\{x \mapsto b, y \mapsto a\}$ is another unifier.

Notation

- First-order language.
- \mathcal{F} : Set of function symbols.
- \mathcal{V} : Set of variables.
- x, y, z : Variables.
- a, b, c : Constants.
- f, g, h : Arbitrary function symbols.
- s, t, r : Terms.
- $\mathcal{T}(\mathcal{F}, \mathcal{V})$: Set of terms over \mathcal{F} and \mathcal{V} .

Notation

- Equation: a pair of terms, written $s \doteq t$.
- $vars(t)$: The set of variables in t . This notation will be used also for sets of terms, equations, and sets of equations.
- $\sigma, \vartheta, \eta, \rho$: Substitutions.
- ε : The identity substitution.

Equational Theory

Equational Theory

- E : a set of equations over $\mathcal{T}(\mathcal{F}, \mathcal{V})$, called identities.
- Equational theory \doteq_E defined by E : The least congruence relation on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ stable under substitution application and containing E .
- That means, \doteq_E is the least binary relation on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ such that:
 - $E \subseteq \doteq_E$.
 - Reflexivity: $s \doteq_E s$ for all s .
 - Symmetry: If $s \doteq_E t$ then $t \doteq_E s$ for all s, t .
 - Transitivity: If $s \doteq_E t$ and $t \doteq_E r$ then $s \doteq_E r$ for all s, t, r .
 - Congruence: If $s_1 \doteq_E t_1, \dots, s_n \doteq_E t_n$ then $f(s_1, \dots, s_n) \doteq_E f(t_1, \dots, t_n)$ for all s, t, n and n -ary f .
 - Stability: If $s \doteq_E t$ then $s\sigma \doteq_E t\sigma$ for all s, t, σ .



Notation, Terminology

- $s \doteq_E t$:
 - The pair (s, t) belongs to the equational theory \doteq_E .
 - The term s is equal modulo E to the term t .
- $s \approx t$: Identities.
- $\text{sig}(E)$: The set of function symbols that occur in E .
- Sometimes E is called an equational theory as well.

Notation, Terminology

Example

- $C := \{f(x, y) \approx f(y, x)\}$: f is commutative.

$$\text{sig}(C) = f.$$

$$f(f(a, b), c) \doteq_C f(c, f(b, a)).$$

- $AU := \{f(f(x, y), z) \approx f(x, f(y, z)), f(x, e) \approx x, f(e, x) \approx x\}$:

f is associative, e is unit.

$$\text{sig}(AU) = \{f, e\}$$

$$f(a, f(x, f(e, a))) \doteq_{AU} f(f(a, x), a).$$

Notation, Terminology

E-Unification Problem, *E*-Unifier, *E*-Unifiability

- *E*: a given set of identities.
- *E*-Unification problem over \mathcal{F} : a finite set of equations

$$\Gamma = \{s_1 \doteq_E^? t_1, \dots, s_n \doteq_E^? t_n\},$$

where $s_i, t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$.

- *E*-Unifier of Γ : a substitution σ such that

$$s_1\sigma \doteq_E t_1\sigma, \dots, s_n\sigma \doteq_E t_n\sigma.$$

- $u_E(\Gamma)$: the set of *E*-unifiers of Γ .
- Γ is *E*-unifiable iff $u_E(\Gamma) \neq \emptyset$.

E -Unification vs Syntactic Unification

- Syntactic unification: a special case of E -unification with $E = \emptyset$.
- Any syntactic unifier of an E -unification problem Γ is also an E -unifier of Γ .
- For $E \neq \emptyset$, $u_E(\Gamma)$ may contain a unifier that is not a syntactic unifier.

E -Unification vs Syntactic Unification

Example

- Terms $f(a, x)$ and $f(b, y)$:
 - Not syntactically unifiable.
 - Unifiable module commutativity of f .
 - C-unifier: $\{x \mapsto b, y \mapsto a\}$
- Terms $f(a, x)$ and $f(y, b)$:
 - Have the most general syntactic unifier $\{x \mapsto b, y \mapsto a\}$.
 - If f is associative, then there are additional unifiers, e.g., $\{x \mapsto f(z, b), y \mapsto f(a, z)\}$.

Notions Adapted

Instantiation Quasi-Ordering (Modified)

- E : equational theory. \mathcal{X} : set of variables.
- A substitution σ is *more general than* ϑ modulo E on \mathcal{X} , written $\sigma \leq_E^{\mathcal{X}} \vartheta$, if there exists η such that $x\sigma\eta \doteq_E x\vartheta$ for all $x \in \mathcal{X}$.
- ϑ is called an *E -instance* of σ modulo E on \mathcal{X} .
- The relation $\leq_E^{\mathcal{X}}$ is quasi-ordering, called *instantiation quasi-ordering*.
- $\equiv_E^{\mathcal{X}}$ is the equivalence relation corresponding to $\leq_E^{\mathcal{X}}$.

No MGU

- When comparing unifiers of Γ , the set \mathcal{X} is $vars(\Gamma)$.
- Unifiable E -unification problems might not have an mgu.

Example

- f is commutative.
- $\Gamma = \{f(x, y) \stackrel{?}{\doteq}_C f(a, b)\}$ has two C-unifiers:

$$\sigma_1 = \{x \mapsto a, y \mapsto b\}$$

$$\sigma_2 = \{x \mapsto b, y \mapsto a\}.$$
- On $vars(\Gamma) = \{x, y\}$, any unifier is equal to either σ_1 or σ_2 .
- σ_1 and σ_2 are not comparable wrt $\leq_C^{\{x, y\}}$.
- Hence, no mgu for Γ .

MCSU vs MGU

In E -unification, the role of mgu is taken on by a complete set of E -unifiers.

Complete and Minimal Complete Sets of E -Unifiers

- Γ : E -unification problem over \mathcal{F} .
- $\mathcal{X} = vars(\Gamma)$.
- \mathcal{C} is a *complete set of E -unifiers* of Γ iff
 1. $\mathcal{C} \subseteq u_E(\Gamma)$: \mathcal{C} 's elements are E -unifiers of Γ , and
 2. For each $\vartheta \in u_E(\Gamma)$ there exists $\sigma \in \mathcal{C}$ such that $\sigma \leq_E^{\mathcal{X}} \vartheta$.
- \mathcal{C} is a *minimal complete set of E -unifiers* ($mcsu_E$) of Γ if it is a complete set of E -unifiers of Γ and
 3. Two distinct elements of \mathcal{C} are not comparable wrt $\leq_E^{\mathcal{X}}$.
- σ is an mgu of Γ iff $mcsu_E(\Gamma) = \{\sigma\}$.

MCSU's

- $mcsu_E(\Gamma) = \emptyset$ if Γ is not E -unifiable.
- Minimal complete sets of unifiers do not always exist.
- When they exist, they may be infinite.
- When they exist, they are unique up to \equiv_E .

Unification Type

Unification Type of a Problem, Theory.

- E : equational theory.
- Γ : E -unification problem over \mathcal{F} .
- Γ has *unification type*
 - *unitary*, if $mcsu(\Gamma)$ has cardinality at most one,
 - *finitary*, if $mcsu(\Gamma)$ has finite cardinality,
 - *infinitary*, if $mcsu(\Gamma)$ has infinite cardinality,
 - *zero*, if $mcsu(\Gamma)$ does not exist.
- Abbreviation: type unitary - 1, finitary - ω , infinitary - ∞ , zero - 0.
- Ordering: $1 < \omega < \infty < 0$.
- *Unification type* of E wrt \mathcal{F} : the maximal type of an E -unification problem over \mathcal{F} .

Unification Type

- The unification type of an E -equational problem over \mathcal{F} depends both
- on E , and
 - on \mathcal{F} (which function symbols are permitted in unification problems).

Unification Type

Example (Type Unitary)

Syntactic unification.

- The empty equational theory \emptyset : Syntactic unification.
- Unitary wrt any \mathcal{F} because any unifiable syntactic unification problem has an mgu.

Unification Type

Example (Type Finitary)

Commutative unification: $\{f(x, y) \approx f(y, x)\}$

- Not unitary.
- $\{f(x, y) \doteq_C^? f(a, b)\}$ has two unifiers $\{x \mapsto a, y \mapsto b\}$ and $\{x \mapsto b, y \mapsto a\}$.
- No mgu.
- C unification is finitary.

Unification Type

Example (Type Finitary)

C unification is finitary for any \mathcal{F} :

- Let $\Gamma = \{s_1 \doteq_C^? t_1, \dots, s_n \doteq_C^? t_n\}$ be a C-unification problem.
- Consider all possible syntactic unification problems $\Gamma' = \{s'_1 \doteq_C^? t'_1, \dots, s'_n \doteq_C^? t'_n\}$, where $s'_i \doteq_C s_i$ and $t'_i \doteq_C t_i$ for each $1 \leq i \leq n$.
- There are only finitely many such Γ' 's, because the C-equivalence class for a given term t is finite.
- It can be shown that collection of all mgu's of Γ' 's is a complete set of C-unifiers of Γ . This set is finite.
- If this set is not minimal (often the case), it can be minimized by removing redundant C-unifiers.

Unification Type

Example (Type Infinitary)

Associative unification: $\{f(f(x, y), z) \approx f(x, f(y, z))\}$.

- $\{f(x, a) \doteq_A f(a, x)\}$ has an infinite *mcsu*:
 $\{\{x \mapsto a\}, \{x \mapsto f(a, a)\}, \{x \mapsto f(a, f(a, a))\}, \dots\}$
- Hence, A-unification can not be unitary or finitary.
- It is not of type zero because any A-unification problem has an *mcsu* that can be enumerated by the procedure from



G. Plotkin.

Building in equational theories.

In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 7, pages 73–90. Edinburgh University Press, 1972.

- A-unification is infinitary for any \mathcal{F} .

Unification Type

Example (Type Zero)

Associative-Idempotent unification:

$$\{f(f(x, y), z) \approx f(x, f(y, z)), f(x, x) \approx x\}.$$

- $\{f(x, f(y, x)) \stackrel{?}{\doteq}_{AI} f(x, f(z, x))\}$ does not have a minimal complete set of unifiers, see



F. Baader.

Unification in idempotent semigroups is of type zero.

J. Automated Reasoning, 2(3):283–286, 1986.

- AI-unification is of type zero.

Unification Type. Signature Matters

Unification Type depends on \mathcal{F} .

Example

Associative-commutative unification with unit (ACU):

- $\{f(f(x, y), z) \approx f(x, f(y, z)), f(x, y) \approx f(y, x), f(x, e) \approx x\}$.
- Any ACU problem built using only f and variables is unitary.
- There are ACU problems containing function symbols other than f and e , which are finitary, not unitary.
- For instance, $mcsu(\{f(x, y) \stackrel{?}{\doteq}_{ACU} f(a, b)\})$ consists of four unifiers (which ones?).

Kinds of E -unification.

Kinds of E -Unification

One may distinguish three kinds of E -unification problems, depending on the function symbols that are allowed to occur in them.

E -Unification Problems: Elementary, with Constants, General.

- E : an equational Theory.
 Γ : an E -unification problem over \mathcal{F} .
- Γ is an elementary E -unification problem iff $\mathcal{F} = \text{sig}(E)$.
- Γ is an E -unification problem with constants iff $\mathcal{F} \setminus \text{sig}(E)$ consists of constants.
- Γ is a general E -unification problem iff $\mathcal{F} \setminus \text{sig}(E)$ may contain arbitrary function symbols.

Unification Types of Theories wrt Kinds

Unification Types Depending on Signature

- Unification type of E wrt elementary unification:
Maximal unification type of E wrt all \mathcal{F} such that $\mathcal{F} = sig(E)$.
- Unification type of E wrt unification with constants:
Maximal unification type of E wrt all \mathcal{F} such that $\mathcal{F} \setminus sig(E)$ is a set of constants.
- Unification type of E wrt general unification:
Maximal unification type of E wrt all \mathcal{F} such that $\mathcal{F} \setminus sig(E)$ is a set of arbitrary function symbols.

Unification Types of Theories wrt Kinds

The same equational theory can have different unification types for different kinds. Examples:

- ACU (Abelian monoids): Unitary wrt elementary unification, finitary wrt unification with constants and general unification.
- AG (Abelian groups): Unitary wrt elementary unification and unification with constants, finitary wrt general unification.

Single Equation vs Systems of Equations

- In syntactic unification, solving systems of equations can be reduced to solving a single equation.
- For equational unification, the same holds only for general unification.
- For elementary unification and for unification with constants it is not the case.

Unification Types wrt of Cardinality of Problems

There exists an equational theory E such that

- all elementary E -unification problems of cardinality 1 (single equations) have minimal complete sets of E -unifiers, but
- E is of type zero wrt to elementary unification: There exists an elementary E -unification problem of cardinality 2 that does not have a minimal complete set of unifiers.



H.-J. Bürckert, A. Herold, and M. Schmidt-Schauß.

On equational theories, unification, and decidability.

J. Symbolic Computation **8**(3,4), 3–49. 1989.

Decision and Unification Procedures

- **Decision procedure** for an equational theory E (wrt \mathcal{F}):
 An algorithm that for each E -unification problem Γ (wrt \mathcal{F}) returns *success* if Γ is E -unifiable, and *failure* otherwise.
- E is **decidable** if it admits a decision procedure.
- (Minimal) **E -unification algorithm** (wrt \mathcal{F}): An algorithm that computes a (minimal) finite complete set of E -unifiers for all E -unification problems over \mathcal{F} .
- E -unification algorithm yields a decision procedure for E .
- (Minimal) **E -unification procedure**: A procedure that enumerates a possible infinite (minimal) complete set of E -unifiers.
- E -unification procedure does not yield a decision procedure for E .

Decidability wrt Kinds

Decidability of an equational theory might depend on the kinds of E -unification.

- There exists an equational theory for which elementary unification is decidable, but unification with constants is undecidable:



H.-J. Bürckert.

Some relationships between unification, restricted unification, and matching.

In J. Siekmann, editor, *Proc. 8th Int. Conference on Automated Deduction*, volume 230 of *LNCS*. Springer, 1986.

Decidability wrt Kinds

Decidability of an equational theory might depend on the kinds of E -unification.

- There exists an equational theory for which unification with constants is decidable, but general unification is undecidable:



J. Otop.

E-unification with constants vs. general E-unification.

Journal of Automated Reasoning, 48(3):363–390, 2012.

Decidability wrt Problem Cardinality

There exists an equational theory E such that

- unifiability of elementary E -unification problems of cardinality 1 (single equations) is decidable, but
- for elementary problems of larger cardinality it is undecidable.



P. Narendran and H. Otto.

Some results on equational unification.

In M. E. Stickel, editor, *Proc. 10th Int. Conference on Automated Deduction*, volume 449 of *LNAI*. Springer, 1990.

Summary

- Unification type depends on
 - equational theory,
 - signature (kinds),
 - cardinality of unification problems.
- Decidability depends on
 - equational theory,
 - signature (kinds),
 - cardinality of unification problems.

Three Main Questions in Unification Theory

Decidability: Is it decidable whether an E -unification problem is solvable?
If yes, what is the complexity of this decision problem?

Unification type: What is the unification type of the theory E ?

Unification algorithm: How can we obtain an (efficient) E -unification algorithm, or a (preferably minimal) E -unification procedure?

Summary of Results for Specific Theories

General unification:

Theory	Decidability	Type	Algorithm/Procedure
\emptyset , BR	Yes	1	Yes
A, AU	Yes	∞	Yes
C, AC, ACU	Yes	ω	Yes
I, CI, ACI	Yes	ω	Yes
AI	Yes	0	?
$D_{\{f,g\}}A_g$	No	∞	?
AG	Yes	ω	Yes
CRU	No	? (∞ or 0)	?

BR - Boolean ring, D - distributivity, CRU - commutative ring with unit.

Commutative Unification and Matching

- C-unification inference system \mathcal{U}_C can be obtained from the \mathcal{U} by adding the C-Decomposition rule:

$$\begin{aligned} \text{C-Decomposition: } \quad & \{f(s_1, s_2) \doteq_C^? f(t_1, t_2)\} \uplus P'; S \implies \\ & \{s_1 \doteq_C^? t_2, s_2 \doteq_C^? t_1\} \cup P'; S, \\ & \text{if } f \text{ is commutative.} \end{aligned}$$

- **C-Decomposition** and **Decomposition** transform the same system in different ways.
- C-matching algorithm \mathcal{M}_C is obtained analogously from \mathcal{M} .

C-Unification

In order to C-unify s and t :

- 1 Create an initial system $\{s \doteq_C^? t\}; \emptyset$.
- 2 Apply successively rules from \mathcal{U}_C , building a complete tree of derivations. **C-Decomposition** and **Decomposition** rules have to be applied concurrently and form branching points in the derivation tree.

Example. C-Unification

C-unify $g(f(x, y), z)$ and $g(f(f(a, b), f(b, a)), c)$, commutative f .

$$\{g(f(x, y), z) \doteq_C^? g(f(f(a, b), f(b, a)), c)\}; \emptyset$$

↓

$$\{f(x, y) \doteq_C^? f(f(a, b), f(b, a)), z \doteq_C^? c\}; \emptyset$$

↙

↘

$$\{x \doteq_C^? f(a, b), y \doteq_C^? f(b, a), z \doteq_C^? c\}; \emptyset$$

$$\{x \doteq_C^? f(b, a), y \doteq_C^? f(a, b), z \doteq_C^? c\}; \emptyset$$

$$\{y \doteq_C^? f(b, a), z \doteq_C^? c\}; \{x \doteq f(a, b)\}$$

$$\{y \doteq_C^? f(a, b), z \doteq_C^? c\}; \{x \doteq f(b, a)\}$$

$$\{z \doteq_C^? c\}; \{x \doteq f(a, b), y \doteq f(b, a)\}$$

$$\{z \doteq_C^? c\}; \{x \doteq f(b, a), y \doteq f(a, b)\}$$

$$\emptyset; \{x \doteq f(a, b), y \doteq f(b, a), z \doteq c\}$$

$$\emptyset; \{x \doteq f(b, a), y \doteq f(a, b), z \doteq c\}$$

Not minimal.

Properties of the C-Unification Algorithm

Theorem

Applied to a C-unification problem P , the C-unification algorithm terminates and computes a complete set of C-unifiers of P .

Proof.

- Termination is proved using the same measure as for syntactic unification.
- Completeness is based on the following two facts:
 - If Γ is transformed by only one rule of \mathcal{U}_C into Γ' , then $u_C(\Gamma) = u_C(\Gamma')$.
 - If Γ is transformed by two rules of \mathcal{U}_C into Γ_1 and Γ_2 , then $u_C(\Gamma) = u_C(\Gamma_1) \cup u_C(\Gamma_2)$.



MCSU for C-Unification/Matching Problems Can Be Large

Example

- Problem: $f(f(x_1, x_2), f(x_3, x_4)) \stackrel{?}{\doteq}_C f(f(a, b), f(c, d))$.
- $mcsu$ contains $4!$ substitutions.

Properties of the C-Unification Algorithm

- The algorithm, in general, does not return a minimal complete set of C-unifiers.
- The obtained complete set can be further minimized, removing redundant unifiers.
- Not clear how to design a C-unification algorithm that computes a minimal complete set of unifiers directly.

Properties of the C-Unification Algorithm

Theorem

The decision problem of C-matching and unification is NP-complete.

Proof.

Exercise.

ACU-Unification

$$\text{ACU} = \{f(f(x, y), z) \approx f(x, f(y, z)), f(x, y) \approx f(y, x), f(x, e) \approx x\}$$

- 1 Associativity, commutativity, unit element.
- 2 f is associative and commutative, e is the unit element.

Example: Elementary ACU-Unification

Elementary ACU-unification problem:

$$\Gamma = \{f(x, f(x, y)) \stackrel{?}{\doteq}_{\text{ACU}} f(z, f(z, z))\}$$

Solving idea:

1. Associate with the equation in Γ a homogeneous linear Diophantine equation $2x + y = 3z$.
2. The equation states that the number of new variables introduced by a unifier σ in both sides of $\Gamma\sigma$ must be the same.

(Continues on the next slide.)

Example. Elementary ACU-Unification (Cont.)

3. Solve $2x + y = 3z$ over nonnegative integers. Three minimal solutions:

$$x = 1, y = 1, z = 1$$

$$x = 0, y = 3, z = 1$$

$$x = 3, y = 0, z = 2$$

Any other solution of the equation can be obtained as a nonnegative linear combination of these three solutions.

(Continues on the next slide.)

Example. Elementary ACU-Unification (Cont.)

4. Introduce new variables v_1, v_2, v_3 for each solution of the Diophantine equation:

	x	y	z
v_1	1	1	1
v_2	0	3	1
v_3	3	0	2

5. Each row corresponds to a unifier of Γ :

$$\sigma_1 = \{x \mapsto v_1, y \mapsto v_1, z \mapsto v_1\}$$

$$\sigma_2 = \{x \mapsto e, y \mapsto f(v_2, f(v_2, v_2)), z \mapsto v_2\}$$

$$\sigma_3 = \{x \mapsto f(v_3, f(v_3, v_3)), y \mapsto e, z \mapsto f(v_3, v_3)\}$$

However, none of them is an mgu.

Example. Elementary ACU-Unification (Cont.)

6. To obtain an mgu, we should combine all three solutions:

	x	y	z
v_1	1	1	1
v_2	0	3	1
v_3	3	0	2

The columns indicate that the mgu we are looking for should have

- in the binding for x one v_1 , zero v_2 , and three v_3 's,
- in the binding for y one v_1 , three v_2 's, and zero v_3 ,
- in the binding for z one v_1 , one v_2 , and two v_3 's

7. Hence, we can construct an mgu:

$$\sigma = \{x \mapsto f(v_1, f(v_3, f(v_3, v_3))), y \mapsto f(v_1, f(v_2, f(v_2, v_2))), \\ z \mapsto f(v_1, f(v_2, f(v_3, v_3)))\}$$

Example: ACU-Unification with constants

- ACU-unification problem with constants

$$\Gamma = \{f(x, f(x, y)) \doteq_{\text{ACU}}^? f(a, f(z, f(z, z)))\}$$

reduces to inhomogeneous linear Diophantine equation

$$S = \{2x + y = 3z + 1\}.$$

- The minimal nontrivial natural solutions of S are $(0, 1, 0)$ and $(2, 0, 1)$.

Example: ACU-Unification with constants

- ACU-unification problem with constants

$$\Gamma = \{f(x, f(x, y)) \doteq_{\text{ACU}}^? f(a, f(z, f(z, z)))\}$$

reduces to inhomogeneous linear Diophantine equation

$$S = \{2x + y = 3z + 1\}.$$

- Every natural solution of S is obtained by as the sum of one of the minimal solution and a solution of the corresponding homogeneous LDE $2x + y = 3z$.
- One element of the minimal complete set of unifiers of Γ is obtained from the combination of one minimal solution of S with the set of all minimal solutions of $2x + y = 3z$.

Example: ACU-Unification with constants

- ACU-unification problem with constants

$$\Gamma = \{f(x, f(x, y)) \doteq_{\text{ACU}}^? f(a, f(z, f(z, z)))\}$$

reduces to inhomogeneous linear Diophantine equation

$$S = \{2x + y = 3z + 1\}.$$

- The minimal complete set of unifiers of Γ is $\{\sigma_1, \sigma_2\}$, where

$$\begin{aligned} \sigma_1 = \{ &x \mapsto f(v_1, f(v_3, f(v_3, v_3))), \\ &y \mapsto f(a, f(v_1, f(v_2, f(v_2, v_2))), \\ &z \mapsto f(v_1, f(v_2, f(v_3, v_3))) \} \end{aligned}$$

$$\begin{aligned} \sigma_2 = \{ &x \mapsto f(a, f(a, f(v_1, f(v_3, f(v_3, v_3))))), \\ &y \mapsto f(v_1, f(v_2, f(v_2, v_2))), \\ &z \mapsto f(a, f(v_1, f(v_2, f(v_3, v_3)))) \} \end{aligned}$$

ACU-Unification with constants

- If an ACU-unification problem contains more than one constant, solve the corresponding inhomogeneous LDE for each constant.
- The unifiers in the minimal complete set correspond to all possible combinations of the minimal solutions of these inhomogeneous equations.

ACU-Unification with constants

Example

$xy \stackrel{?}{\text{ACU}} aabb$:

- Equation for a : $2x + y = 2$. Minimal solutions: $(1, 0)$ and $(0, 2)$.
- Corresponding unifiers: $\{x \mapsto a, y \mapsto e\}$, $\{x \mapsto e, y \mapsto aa\}$
- Equation for b : $2x + y = 3$. Minimal solutions: $(0, 3)$ and $(1, 1)$.
- Corresponding unifiers: $\{x \mapsto e, y \mapsto bbb\}$, $\{x \mapsto b, y \mapsto b\}$
- Unifiers in the minimal complete set: $\{x \mapsto a, y \mapsto bbb\}$,
 $\{x \mapsto ab, y \mapsto b\}$, $\{x \mapsto e, y \mapsto aabb\}$, $\{x \mapsto b, y \mapsto aab\}$.

From ACU to AC

Example

- How to solve $\Gamma_1 = \{f(x, f(x, y)) \doteq_{AC}^? f(z, f(z, z))\}$?
- We “know” how to solve $\Gamma_2 = \{f(x, f(x, y)) \doteq_{ACU}^? f(z, f(z, z))\}$, but its mgu is not an mgu for Γ_1 .
- Mgu of Γ_2 :

$$\sigma = \{x \mapsto f(v_1, f(v_3, f(v_3, v_3))), y \mapsto f(v_1, f(v_2, f(v_2, v_2))), \\ z \mapsto f(v_1, f(v_2, f(v_3, v_3)))\}$$

- Unifier of Γ_1 : $\vartheta = \{x \mapsto v_1, y \mapsto v_1, z \mapsto v_1\}$.
- σ is not more general modulo AC than ϑ .



From ACU to AC

Example

- Idea: Take the mgu of Γ_2 .
- Compose it with all possible erasing substitutions that map a subset of $\{v_1, v_2, v_3\}$ to the unit element.
- Restriction: The result of the composition should not map x , y , and z to the unit element.

From ACU to AC

Example

Minimal complete set of unifiers for Γ_1 :

$$\sigma_1 = \{x \mapsto f(v_1, f(v_3, f(v_3, v_3))), y \mapsto f(v_1, f(v_2, f(v_2, v_2))), \\ z \mapsto f(v_1, f(v_2, f(v_3, v_3)))\}$$

$$\sigma_2 = \{x \mapsto f(v_3, f(v_3, v_3)), y \mapsto f(v_2, f(v_2, v_2)), \\ z \mapsto f(v_2, f(v_3, v_3))\}$$

$$\sigma_3 = \{x \mapsto f(v_1, f(v_3, f(v_3, v_3))), y \mapsto v_1, z \mapsto f(v_1, f(v_3, v_3))\}$$

$$\sigma_4 = \{x \mapsto v_1, y \mapsto f(v_1, f(v_2, f(v_2, v_2))), z \mapsto f(v_1, v_2)\}$$

$$\sigma_5 = \{x \mapsto v_1, y \mapsto v_1, z \mapsto v_1\}$$

How to Solve Systems of LDEs over Naturals?

Contejean-Devie Algorithm:



Evelyne Contejean and Hervé Devie.

An Efficient Incremental Algorithm for Solving Systems of Linear Diophantine Equations.

Information and Computation 113(1): 143–172 (1994).

Generalizes Fortenbacher's Algorithm for solving a single equation:



Michael Clausen and Albrecht Fortenbacher.

Efficient Solution of Linear Diophantine Equations.

J. Symbolic Computation 8(1,2): 201–216 (1989).

Homogeneous Case

Homogeneous linear Diophantine system with m equations and n variables:

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = 0 \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n = 0 \end{cases}$$

- a_{ij} 's are integers.
- Looking for nontrivial natural solutions.

Homogeneous Case

Example

$$\begin{cases} -x_1 + x_2 + 2x_3 - 3x_4 = 0 \\ -x_1 + 3x_2 - 2x_3 - x_4 = 0 \end{cases}$$

Nontrivial solutions:

- $s_1 = (0, 1, 1, 1)$
- $s_2 = (4, 2, 1, 0)$
- $s_3 = (0, 2, 2, 2) = 2s_1$
- $s_4 = (8, 4, 2, 0) = 2s_2$
- $s_5 = (4, 3, 2, 1) = s_1 + s_2$
- $s_6 = (8, 5, 3, 1) = s_1 + 2s_2$
- ...

Homogeneous Case

Homogeneous linear Diophantine system with m equations and n variables:

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = 0 \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n = 0 \end{cases}$$

- a_{ij} 's are integers.
- Looking for a **basis** in the set of nontrivial natural solutions.
- Does it exist?

Homogeneous Case

The basis in the set S of nontrivial natural solutions of a homogeneous LDS is the set of \gg -minimal elements S .

\gg is the ordering on tuples of natural numbers:

$$(x_1, \dots, x_n) \gg (y_1, \dots, y_n)$$

if and only if

- $x_i \geq y_i$ for all $1 \leq i \leq n$ and
- $x_i > y_i$ for some $1 \leq i \leq n$.

Matrix Form

Homogeneous linear Diophantine system with m equations and n variables:

$$Ax \downarrow = 0 \downarrow,$$

where

$$A := \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \quad x \downarrow := \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad 0 \downarrow := \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

Matrix Form

- Canonical basis in \mathbb{N}^n : $(e_1\downarrow, \dots, e_n\downarrow)$.

- $e_j\downarrow = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$, with 1 in j 's row.

- Then $Ax\downarrow = x_1 Ae_1\downarrow + \dots + x_n Ae_n\downarrow$.

Matrix Form

- a : The linear mapping associated to A .

$$a(x\downarrow) = \begin{pmatrix} a_{11}x_1 & +\cdots+ & a_{1n}x_n \\ \vdots & & \vdots \\ a_{m1}x_1 & +\cdots+ & a_{mn}x_n \end{pmatrix} = x_1a(e_1\downarrow) + \cdots + x_na(e_n\downarrow).$$

Single Equation: Idea

Case $m = 1$: Single homogeneous LDE $a_1x_1 + \dots + a_nx_n = 0$.

Fortenbacher's idea:

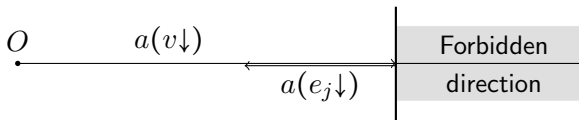
- Search minimal solutions starting from the elements in the canonical basis of \mathbb{N}^n .
- Suppose the current vector $v \downarrow$ is not a solution.
- It can be nondeterministically increased, component by component, until it becomes a solution or greater than a solution.
- To decrease the search space, the following restrictions can be imposed:
 - If $a(v \downarrow) > 0$, then increase by one some v_j with $a_j < 0$.
 - If $a(v \downarrow) < 0$, then increase by one some v_j with $a_j > 0$.
 - (If $a(v \downarrow)a(e_j \downarrow) < 0$ for some j , increase v_j by one.)

Single Equation: Geometric Interpretation of the Idea

- Fortenbacher's condition

If $a(v \downarrow)a(e_j \downarrow) < 0$ for some j , increase v_j by one.

- Increasing v_j by one: $a(v \downarrow + e_j \downarrow) = a(v \downarrow) + a(e_j \downarrow)$.
- Going to the "right direction", towards the origin.



Single Equation: Algorithm

Case $m = 1$: Single homogeneous LDE $a_1x_1 + \dots + a_nx_n = 0$.

Fortenbacher's algorithm:

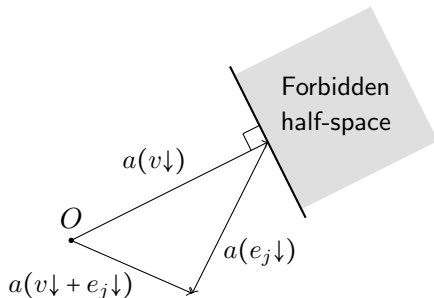
- Start with the pair P, M of the set of potential solutions $P = \{e_1\downarrow, \dots, e_n\downarrow\}$ and the set of minimal nontrivial solutions $M = \emptyset$.
- Apply repeatedly the rules:
 - ① $\{v\downarrow\} \cup P', M \implies P', M$,
if $v\downarrow \gg u\downarrow$ for some $u\downarrow \in M$.
 - ② $\{v\downarrow\} \cup P', M \implies P', \{v\downarrow\} \cup M$,
if $a(v\downarrow) = 0$ and rule 1 is not applicable.
 - ③ $P, M \implies \{v\downarrow + e_j\downarrow \mid v\downarrow \in P, a(v\downarrow)a(e_j\downarrow) < 0, j \in 1..n\}, M$,
if rules 1 and 2 are not applicable.
- If \emptyset, M is reached, return M .

System of Equations: Idea

- General case: System of homogeneous LDEs.
- $a(x\downarrow) = 0\downarrow$.
- Generalizing Fortenbacher's idea:
 - Search minimal solutions starting from the elements in the canonical basis of \mathbb{N}^n .
 - Suppose the current vector $v\downarrow$ is not a solution.
 - It can be nondeterministically increased, component by component, until it becomes a solution or greater than a solution.
 - To decrease the search space, increase only those components that lead to the "right direction".

System of Equations: How to Restrict

- “Right direction”: Towards the origin.
- If $a(v\downarrow) \neq 0\downarrow$, then do $a(v\downarrow + e_j\downarrow) = a(v\downarrow) + a(e_j\downarrow)$.
- $a(v\downarrow) + a(e_j\downarrow)$ should lie in the half-space containing O .
- **Contejean-Devie condition:** If $a(v\downarrow) \cdot a(e_j\downarrow) < 0$ for some j , increase v_j by one. (\cdot is the scalar product.)



How to Restrict: Comparison

- Fortenbacher's condition

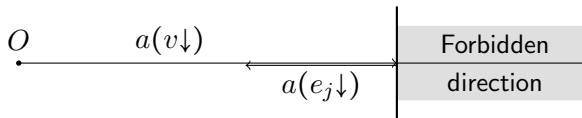
If $a(v \downarrow) a(e_j \downarrow) < 0$ for some j , increase v_j by one.

- Contejean-Devie condition

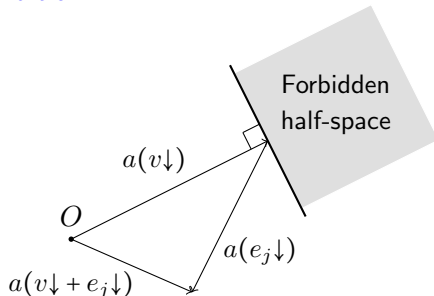
If $a(v \downarrow) \cdot a(e_j \downarrow) < 0$ for some j , increase v_j by one.

How to Restrict: Comparison

Fortenbacher's condition



Contejean-Devie condition



System of Equations: Algorithm

System of homogeneous LDEs: $a(x\downarrow) = 0\downarrow$.

Contejean-Devie algorithm:

- Start with the pair P, M where
 - $P = \{e_1\downarrow, \dots, e_n\downarrow\}$ is the set of potential solutions,
 - $M = \emptyset$ is the set of minimal nontrivial solutions.
- Apply repeatedly the rules:
 - 1 $\{v\downarrow\} \cup P', M \implies P', M,$
if $v\downarrow \gg u\downarrow$ for some $u\downarrow \in M$.
 - 2 $\{v\downarrow\} \cup P', M \implies P', \{v\downarrow\} \cup M,$
if $a(v\downarrow) = 0\downarrow$ and rule 1 is not applicable.
 - 3 $P, M \implies \{v\downarrow + e_j\downarrow \mid v\downarrow \in P, a(v\downarrow) \cdot a(e_j\downarrow) < 0, j \in 1..n\}, M,$
if rules 1 and 2 are not applicable.
- If \emptyset, M is reached, return M .

Contejean-Devie Algorithm on an Example

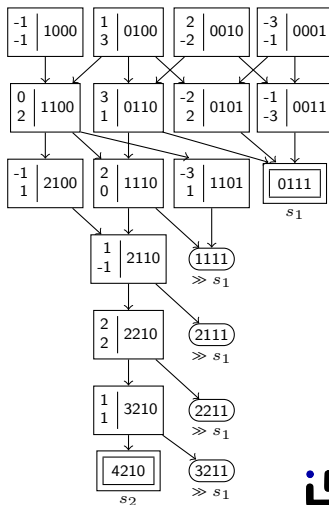
$$\begin{cases} -x_1 + x_2 + 2x_3 - 3x_4 = 0 \\ -x_1 + 3x_2 - 2x_3 - x_4 = 0 \end{cases}$$

$$e_1 \downarrow = (1, 0, 0, 0)^T \quad e_2 \downarrow = (0, 1, 0, 0)^T$$

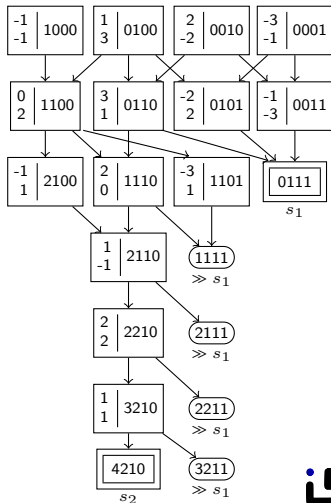
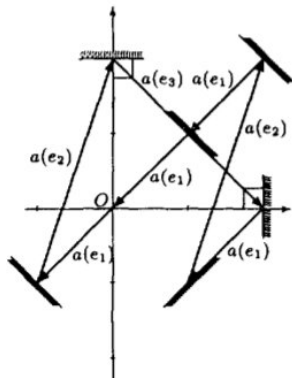
$$e_3 \downarrow = (0, 0, 1, 0)^T \quad e_4 \downarrow = (0, 0, 0, 1)^T$$

Start: $\{e_1 \downarrow, \dots, e_4 \downarrow\}, \emptyset$.

- 1 $\{v \downarrow\} \cup P', M \implies P', M$,
if $v \downarrow \gg u \downarrow$ for some $u \downarrow \in M$.
- 2 $\{v \downarrow\} \cup P', M \implies P', \{v \downarrow\} \cup M$,
if $a(v \downarrow) = 0 \downarrow$ and rule 1 is not applicable.
- 3 $P, M \implies \{v \downarrow + e_j \downarrow \mid v \downarrow \in P, a(v \downarrow) \cdot a(e_j \downarrow) < 0, j \in 1..n\}, M$,
if rules 1 and 2 are not applicable.



Contejean-Devie Algorithm on an Example



Properties of the Algorithm

$a(x\downarrow) = 0\downarrow$: An n -variate system of homogeneous LDEs.

$(e_1\downarrow, \dots, e_n\downarrow)$: The canonical basis of \mathbb{N}^n .

$\mathcal{B}(a(x\downarrow) = 0\downarrow)$: Basis in the set of nontrivial natural solutions of $a(x\downarrow) = 0\downarrow$.

Theorem

- *The Contejean-Devie algorithm terminates on any input.*
- *Let $(e_1\downarrow, \dots, e_n\downarrow), \emptyset \implies^* \emptyset, M$ be the sequence of transformations performed by the Contejean-Devie algorithm for $a(x\downarrow) = 0\downarrow$. Then*

$$\mathcal{B}(a(x\downarrow) = 0\downarrow) = M.$$

Notation

- $\|x\downarrow\| = \sqrt{x_1^2 + \dots + x_n^2}$.
- $|(s_1, \dots, s_n)| = s_1 + \dots + s_n$.

Completeness

Theorem

Let $P_0, M_0 \implies^* \emptyset, M$ be the sequence of transformations performed by the Contejean-Devie algorithm for $a(x\downarrow) = 0\downarrow$ with $P_0 = (e_1\downarrow, \dots, e_n\downarrow)$ and $M_0 = \emptyset$. Then $\mathcal{B}(a(x\downarrow) = 0\downarrow) \subseteq M$.

Proof.

Assume $s\downarrow \in \mathcal{B}(a(x\downarrow) = 0\downarrow)$ and show that there exists a sequence of vectors

$$v_1\downarrow = e_{j_0}\downarrow \ll \dots \ll v_k\downarrow \ll v_{k+1}\downarrow = v_k\downarrow + e_{j_k}\downarrow \ll \dots \ll v_{|s\downarrow|}\downarrow = s\downarrow$$

such that $v_i\downarrow \in P_{l_i}$, where P_{l_i} is from the given sequence of transformations and $l_i < l_j$ for $i < j$.

Completeness

Theorem

Let $P_0, M_0 \implies^* \emptyset, M$ be the sequence of transformations performed by the Contejean-Devie algorithm for $a(x\downarrow) = 0\downarrow$ with $P_0 = (e_1\downarrow, \dots, e_n\downarrow)$ and $M_0 = \emptyset$. Then $\mathcal{B}(a(x\downarrow) = 0\downarrow) \subseteq M$.

Proof (cont.)

For $e_{j_0}\downarrow$, any basic vector $\ll s\downarrow$ can be chosen. Such basic vectors do exist (since $s\downarrow \neq 0\downarrow$) and are in P_0 . Assume now we have $v_1\downarrow \ll \dots \ll v_k\downarrow \ll s\downarrow$ with $v_k\downarrow \in P_{l_k}$. Then there exists $s_k\downarrow$ with $s\downarrow = v_k\downarrow + s_k\downarrow$ and $0 = \|a(s\downarrow)\|^2 = \|a(v_k\downarrow)\|^2 + \|a(s_k\downarrow)\|^2 + 2a(v_k\downarrow) \cdot a(s_k\downarrow)$, which implies $a(v_k\downarrow) \cdot a(s_k\downarrow) < 0$.

Completeness

Theorem

Let $P_0, M_0 \Longrightarrow^* \emptyset, M$ be the sequence of transformations performed by the Contejean-Devie algorithm for $a(x\downarrow) = 0\downarrow$ with $P_0 = (e_1\downarrow, \dots, e_n\downarrow)$ and $M_0 = \emptyset$. Then $\mathcal{B}(a(x\downarrow) = 0\downarrow) \subseteq M$.

Proof (cont.)

Hence, there exists $e_{j_k}\downarrow$ with $s_k\downarrow \gg e_{j_k}\downarrow$ such that $a(v_k\downarrow) \cdot a(e_{j_k}\downarrow) < 0$. We take $v_{k+1}\downarrow = v_k\downarrow + e_{j_k}\downarrow$. Then $s\downarrow \gg v_{k+1}\downarrow$ and by rule 3, $v_{k+1}\downarrow \in P_{l_{k+1}}$. After $|s\downarrow|$ steps, we reach s . Hence, $s\downarrow \in P_{l_{|s|}}$. Since $a(s\downarrow) = 0$, application of rule 2 moves $s\downarrow$ to M . □

Soundness

Theorem

Let $P_0, M_0 \implies^* \emptyset, M$ be the sequence of transformations performed by the Contejean-Devie algorithm for $a(x\downarrow) = 0\downarrow$ with $P_0 = (e_1\downarrow, \dots, e_n\downarrow)$ and $M_0 = \emptyset$. Then $M \subseteq \mathcal{B}(a(x\downarrow) = 0\downarrow)$.

Proof.

Any $s\downarrow \in M$ is a solution. Show that it is minimal. Assume it is not: $s\downarrow = s_1\downarrow + s_2\downarrow$, where $s_1\downarrow$ and $s_2\downarrow$ are non-null solutions smaller than s . Assume $s\downarrow$ was obtained during the transformations as $s\downarrow = v_i\downarrow + e_{j_i}\downarrow$, where $v_i\downarrow \in P_i$. But then $v_i\downarrow \gg s_1\downarrow$ or $v_i\downarrow = s_1\downarrow$ or $v_i\downarrow \gg s_2\downarrow$ or $v_i\downarrow = s_1\downarrow$ and $v_i\downarrow$ is greater than an already computed minimal solution. Therefore, it should have been removed from P_i . A contradiction. \square

Termination

Theorem

Let $v_1\downarrow, v_2\downarrow, \dots$ be an infinite sequence satisfying the Contejean-Devie condition for $a(x\downarrow) = 0\downarrow$:

- u_1 is a basic vector and for each $i \geq 1$ there exists $1 \leq j \leq n$ such that $a(v_i\downarrow) \cdot a(e_j\downarrow) < 0$ and $v_{i+1}\downarrow = v_i\downarrow + e_j\downarrow$.

Then there exist $v\downarrow$ and k such that

- $v\downarrow$ is a solution of $a(x\downarrow) = 0\downarrow$, and
- $v\downarrow \ll v_k\downarrow$.

Non-Homogeneous Case

Non-homogeneous linear Diophantine system with m equations and n variables:

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ \vdots \\ a_{m1}x_1 + \dots + a_{mn}x_n = b_m \end{cases}$$

- a 's and b 's are integers.
- Matrix form: $a(x\downarrow) = b\downarrow$.

Non-Homogeneous Case. Solving Idea

Turn the system into a homogeneous one, denoted S_0 :

$$\begin{cases} -b_1x_0 + a_{11}x_1 + \dots + a_{1n}x_n = 0 \\ \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ -b_mx_0 + a_{m1}x_1 + \dots + a_{mn}x_n = 0 \end{cases}$$

- Solve S_0 and keep only the solutions with $x_0 \leq 1$.
- $x_0 = 1$: a minimal solution for $a(x\downarrow) = b\downarrow$.
- $x_0 = 0$: a minimal solution for $a(x\downarrow) = 0\downarrow$.
- Any solution of the non-homogeneous system $a(x\downarrow) = b\downarrow$ has the form $x\downarrow + y\downarrow$ where:
 - $x\downarrow$ is a minimal solution of $a(x\downarrow) = b\downarrow$.
 - $y\downarrow$ is a linear combination (with natural coefficients) of minimal solutions of $a(x\downarrow) = 0\downarrow$.

Back to ACU-Unification

Theorem

The decision problem for ACU-Matching and ACU-unification is NP-complete.

Specific vs General Results

For each specific equational theory separately studying

- decidability,
- unification type,
- unification algorithm/procedure.

Can one study these problems for bigger classes of equational theories?

General Results

In general, unification modulo equational theories

- is undecidable,
- unification type of a given theory is undecidable,
- admits a complete unification procedure
(Gallier & Snyder, called an universal E -unification procedure).

General Results

Universal E -unification procedure \mathcal{U}_E .

Rules:

- **Trivial, Orient, Decomposition, Variable Elimination** from \mathcal{U} , plus
- **Lazy Paramodulation:**

$$\{e[u]\} \cup P'; S \Longrightarrow \{l \doteq^? u, e[r]\} \cup P'; S,$$

for a fresh variant of the identity $l \approx r$ from $E \cup E^{-1}$, where

- $e[u]$ is an equation where the term u occurs,
- u is not a variable,
- if l is not a variable, then the top symbol of l and u are the same.

General Results

Universal E -unification procedure. Control.

In order to solve a unification problem Γ modulo a given E :

- Create an initial system $\Gamma; \emptyset$.
- Apply successively rules from \mathcal{U}_E , building a complete tree of derivations.
- No other inference rule may be applied to the equation $l \doteq^? u$ that is generated by the Lazy Paramodulation rule before it is subjected to a Decomposition step.

General Results

Example

$$E = \{f(a, b) \approx a, a \approx b\}.$$

Unification problem: $\{f(x, x) \doteq_E^? x\}.$

Computing a unifier $\{x \mapsto a\}$ by the universal procedure:

$$\begin{aligned} \{f(x, x) \doteq_E^? x\}; \emptyset &\Longrightarrow_{LP} \{f(a, b) \doteq_E^? f(x, x), a \doteq_E^? x\}; \emptyset \\ &\Longrightarrow_D \{a \doteq_E^? x, b \doteq_E^? x, a \doteq_E^? x\}; \emptyset \\ &\Longrightarrow_O \{x \doteq_E^? a, b \doteq_E^? x, a \doteq_E^? x\}; \emptyset \\ &\Longrightarrow_S \{b \doteq_E^? a, a \doteq_E^? a\}; \{x \doteq a\} \\ &\Longrightarrow_{LP} \{a \doteq_E^? a, b \doteq_E^? b, a \doteq_E^? a\}; \{x \doteq a\} \\ &\Longrightarrow_T^+ \emptyset; \{x \doteq a\} \end{aligned}$$

General Results

Pros and cons of the universal procedure:

- Pros: Is sound and complete. Can be used for any E .
- Cons: Very inefficient. Usually does not yield a decision procedure or a (minimal) E -unification algorithm even for unitary or finitary theories with decidable unification.

General Results

More useful results can be obtained by imposing additional restrictions on equational theories:

- Syntactic approaches: Restricting syntactic form of the identities defining equational theories.
- Semantic approaches: Depend on properties of the free algebras defined by the equational theory.

Summary

- Syntactic unification and matching.
 - Unification and matching algorithms.
 - Unification on term graphs, algorithms with improved complexity.
- Equational unification and matching
 - Classification with respect to unification type.
 - Algorithms for commutative and ACU-unification, including solving systems of linear Diophantine equations.
 - Universal E -unification procedure.



ISR 2012 sponsors



ALAN TURING YEAR