Asymmetric Unification and Disunification

Veena Ravishankar[1], Kimberly A. Gero[2], and Paliath Narendran[1]

[1] University at Albany–SUNY
{vravishankar,pnarendran}@albany.edu
[2] The College of Saint Rose
gerok@strose.edu

Abstract

We compare two kinds of unification problems: Asymmetric Unification and Disunification, which are variants of Equational Unification. Asymmetric Unification is a type of Equational Unification where the right-hand sides of the equations are in normal form with respect to the given term rewriting system. In Disunification we solve equations and disequations with respect to an equational theory. We contrast the time complexities of both for the case with *free constants* and show that the two problems are incomparable: there are theories where one can be solved in polynomial time while the other is NP-hard. This goes both ways. The time complexity also varies based on the termination ordering used in the term rewriting system.

1 Introduction and Motivation

This is a short introductory survey on two variants of unification, namely asymmetric unification [9] and disunification [2, 7]. We contrast the two in terms of their time complexities for different equational theories, for the case where terms in the input can also have free constant symbols. Asymmetric unification is a new paradigm comparatively, which requires one side of the equation to be irreducible [9], while disunification [7] deals with solving equations and disequations. Complexity analysis has been performed separately on asymmetric unification [4, 10] and disunification [2, 6], but not much work has been done on contrasting the two paradigms. In [9], it was shown that there are theories which are decidable for symmetric unification but are undecidable for asymmetric unification, so here we investigate this further. Initially, it was thought that the two are reducible to one another [10], but our results indicate that they are not at least where time complexity is concerned. In our last section we show that the time complexity of asymmetric unification varies depending on the symbol ordering chosen for the theory. Due to lack of space we have shortened some of the proof details. They can be found in our tech report [18].

2 Notations and preliminaries

We assume the reader is accustomed with the terminologies of term rewriting systems (TRS), equational rewriting [1], unification and equational unification [3]. A term rewriting system [1] is a set of rewrite rules, where a rewrite rule is an identity $l \approx r$ such that l is not a variable and $Var(l) \supseteq Var(r)$. We denote this by $l \to r$. These oriented equations are commonly called *rewrite rules*. The equational theory $\mathscr{E}(R)$ associated with a term rewriting system R is the set of equations obtained from R by treating every rule as a (bidirectional) equation. An equational term rewriting system consists of a set of identities E (which often contains identities such as Commutativity and Associativity) and a set of rewrite rules R.

Definition 1. *Given a* decomposition (Σ, E, R) *of an equational theory, a substitution* σ *is an asymmetric* R,E*-unifier of a set* Q *of asymmetric equations* $\{s_1 \approx_{\downarrow}^{?} t_1, \ldots, s_n \approx_{\downarrow}^{?} t_n\}$ *iff for each asymmetric equation* $s_i \approx_{\downarrow}^{?} t_i$*,* σ *is an* $(E \cup R)$*-unifier of the equation* $s_i \approx^{?} t_i$*, and* $\sigma(t_i)$ *is in* R,E*-normal form. In other words,* $\sigma(s_i) \rightarrow_{R,E}^{!} \sigma(t_i)$*.*

(Note that symmetric unification can be reduced to asymmetric unification. Thus we could also include symmetric equations in a problem instance.)

Example: Let $R = \{x + a \rightarrow x\}$ be a rewrite system. An asymmetric unifier θ for $\{u + v =_{\downarrow}^{?} v + w\}$ modulo this system is $\theta = \{u \mapsto v, w \mapsto v\}$. However, another unifier $\rho = \{u \mapsto a, v \mapsto a, w \mapsto a\}$ is not an asymmetric unifier. But note that $\theta \preceq_E \rho$, i.e., ρ is an *instance* of θ, or, alternatively, θ is more general than ρ. This shows that instances of asymmetric unifiers need not be asymmetric unifiers.

Definition 2. *A disunification problem deals with solving a set of equations and disequations, with respect to an equational theory E,* $\mathscr{L} = \{s_1 \approx_E^? t_1, \ldots, s_n \approx_E^? t_n\} \bigcup \{s_{n+1} \not\approx_E^? t_{n+1}, \ldots, s_{n+m} \not\approx_E^? t_{n+m}\}$*. A solution to this problem is a substitution* σ *such that:* $\sigma(s_i) \approx_E \sigma(t_i)$ *and* $\sigma(s_{n+j}) \not\approx_E \sigma(t_{n+j})$ *where* $i = 1, \ldots, n$ *and* $j = 1, \ldots, m$*.*

Example: Given $E = \{x + a \approx x\}$, a disunifier θ for $\{u + v \not\approx_E v + u\}$ is $\theta = \{u \mapsto a, v \mapsto b\}$.

If $a + x \approx x$ is added to the identities E, then $\theta = \{u \mapsto a, v \mapsto b\}$ is clearly no longer a disunifier modulo this equational theory.

3 A theory for which asymmetric unification is in P whereas disunification is NP-complete

Let R_1 be the following term rewriting system: $h(a) \rightarrow f(a,c)$ $h(b) \rightarrow f(b,c)$. We show that asymmetric unifiability modulo this theory can be solved in polynomial time. The algorithm is outlined in our tech report [18]. However, disunification modulo R_1 is NP-hard. The proof is by a polynomial-time reduction from the three-satisfiability (3SAT) problem. Let $U = \{x_1, x_2, \ldots, x_n\}$ be the set of variables, and $B = \{C_1, C_2, \ldots, C_m\}$ be the set of clauses. Each clause C_k, where $1 \leq k \leq m$, has 3 literals.

We construct an instance of a disunification problem from 3SAT. There are 8 different combinations of T and F assignments to the variables in a clause in 3SAT, out of which there is exactly one truth-assignment to the variables in the clause that makes the clause evaluate to false. For the 7 other combinations of T and F assignments to the literals, the clause is rendered true. We represent T by a and F by b. Hence for each clause C_i we create a disequation DEQ_i of the form

$$f(x_p, f(x_q, x_r)) \not\approx_{R_1} f(d_1, f(d_2, d_3))$$

where x_p, x_q, x_r are variables, $d_1, d_2, d_3 \in \{a, b\}$, and (d_1, d_2, d_3) corresponds to the falsifying truth assignment. For example, given a clause $C_k = x_p \vee \overline{x_q} \vee x_r$, we create the corresponding disequation $DEQ_k = f(x_p, f(x_q, x_r)) \not\approx_{R_1} f(b, f(a, b))$. We also create the equation $h(x_j) \approx_{R_1} f(x_j, c)$ for each variable x_j. These make sure that each x_j is mapped to either a or b. Thus for B, the instance of disunification constructed is

$$S = \Big\{ h(x_1) \approx f(x_1, c), h(x_2) \approx f(x_2, c), \ldots, h(x_n) \approx f(x_n, c) \Big\} \cup \Big\{ DEQ_1, DEQ_2, \ldots, DEQ_m \Big\}$$

Example: Given $U = \{x_1, x_2, x_3\}$ and $B = \{x_1 \vee \overline{x_2} \vee x_3, \ \overline{x_1} \vee \overline{x_2} \vee x_3\}$, the constructed instance of disunification is

$$\{h(x_1) \approx f(x_1,c),\ h(x_2) \approx f(x_2,c),\ h(x_3) \approx f(x_3,c),\ f(x_1, f(x_2,x_3)) \not\approx f(b, f(a,b)),\\ f(x_1, f(x_2,x_3)) \not\approx f(a, f(a,b))\}$$

Note that membership in NP is not hard to show since R_1 is saturated by paramodulation [17].

4 A theory for which disunification is in P whereas asymmetric unification is NP-hard

The theory we consider consists of the following term rewriting system R_2 :

$x + x \to 0$ $\qquad$ $x + 0 \to x$ $\qquad$ $x + (y + x) \to y$

and the equational theory AC:

$(x + y) + z \approx x + (y + z)$ $\qquad$ $x + y \approx y + x$

This theory is called **ACUN** because it consists of *associativity, commutativity, unit* and *nilpotence*. This is the theory of the boolean XOR operator. An algorithm for *general* **ACUN** unification is provided by Zhiqiang Liu [16] in his Ph.D. dissertation [16]. (See also [9, Section 4].)

Disunification modulo this theory can be solved in polynomial time by what is essentially Gaussian Elimination over $\mathbb{Z}_2$. Suppose we have m variables $x_1, x_2, \ldots, x_m$, and n constant symbols $c_1, c_2, \ldots, c_n$, and q such equations and disequations to be unified. We can assume an ordering on the variables and constants $x_1 > x_2 > \ldots > x_m > c_1 > c_2 > \ldots > c_n$. We first pick an equation with leading variable x_1 and eliminate x_1 from all *other* equations and disequations. We continue this process with the next equation consisting of leading variable x_2, followed by an equation containing leading variable x_3 and so on, until no more variables can be eliminated. The problem has a solution if and only if (i) there are no equations that contain only constants, such as $c_3 + c_4 \approx c_5$, and (ii) there are no disequations of the form $0 \not\approx 0$ at this point. This way we can solve the disunification problem in polynomial time using Gaussian Elimination over $\mathbb{Z}_2$.

Example: Suppose we have two equations $x_1 + x_2 + x_3 + c_1 + c_2 \approx^?_{R_2,AC} 0$ and $x_1 + x_3 + c_2 + c_3 \approx^?_{R_2,AC} 0$, and a disequation $x_2 \not\approx^?_{R_2,AC} 0$.

Eliminating x_1 from the second equation, results in the equation $x_2 + c_1 + c_3 \approx_{R_2,AC} 0$. We can now eliminate x_2 from the first equation, resulting in $x_1 + x_3 + c_2 + c_3 \approx_{R_2,AC} 0$. x_2 can also be eliminated from the disequation $x_2 \not\approx_{R_2,AC} 0$, which gives us $c_1 + c_3 \not\approx_{R_2,AC} 0$. Thus the procedure terminates with $x_1 + x_3 + c_2 + c_3 \approx_{R_2,AC} 0$, $x_2 + c_1 + c_3 \approx_{R_2,AC} 0$, $c_1 + c_3 \not\approx_{R_2,AC} 0$. Thus we get $x_2 \approx_{R_2,AC} c_1 + c_3$, $x_1 + x_3 \approx_{R_2,AC} c_2 + c_3$ and the following substitution is clearly a solution:

$$\left\{x_1 \mapsto c_2,\ x_2 \mapsto c_1 + c_3,\ x_3 \mapsto c_3\right\}$$

However, asymmetric unification is NP-hard. The proof is by a polynomial-time reduction from the graph 3-colorability problem. Let $G = (V, E)$ be a graph where $V = \{v_1, v_2, v_3, \ldots, v_n\}$ are the vertices, $E = \{e_1, e_2, e_3, \ldots, e_m\}$ the edges and $C = \{c_1, c_2, c_3\}$ the color set with $n \geq 3$. G is 3-colorable if none of the adjacent vertices $\{v_i, v_j\} \in E$ have the same color assigned from C. We construct an instance of asymmetric unification as follows. We create variables for vertices and edges in G: for each vertex v_i we assign a variable y_i and for each edge e_k we assign a variable z_k. Now for every edge $e_k = \{v_i, v_j\}$ we create an equation $EQ_k = c_1 + c_2 + c_3 \approx_{\downarrow}^{?} y_i + y_j + z_k$. Note that each z_k appears in only one equation. Thus for E, the instance of asymmetric unification problem constructed is

$$S = \Big\{EQ_1, EQ_2, \ldots, EQ_m\Big\}$$

Example: Given $G = (V,E), V = \{v_1, v_2, v_3, v_4\}, E = \{e_1, e_2, e_3, e_4\}$, where $e_1 = \{v_1, v_3\}, e_2 = \{v_1, v_2\}$, $e_3 = \{v_2, v_3\}, e_4 = \{v_3, v_4\}$ and $C = \{c_1, c_2, c_3\}$, the constructed instance of asymmetric unification is

$$\begin{array}{lll} EQ_1 = c_1 + c_2 + c_3 \approx_{\downarrow}^{?} y_1 + y_3 + z_1, & EQ_2 = c_1 + c_2 + c_3 \approx_{\downarrow}^{?} y_1 + y_2 + z_2, \\ EQ_3 = c_1 + c_2 + c_3 \approx_{\downarrow}^{?} y_2 + y_3 + z_3, & EQ_4 = c_1 + c_2 + c_3 \approx_{\downarrow}^{?} y_3 + y_4 + z_4. \end{array}$$

Now suppose the vertices in the graph G are given this color assignment: $\theta = \{v_1 \mapsto c_1, v_2 \mapsto c_2, v_3 \mapsto c_3, v_4 \mapsto c_1\}$. The asymmetric unifier is

$$\Big\{y_1 \mapsto c_1,\ y_2 \mapsto c_3,\ y_3 \mapsto c_2,\ z_1 \mapsto c_3,\ z_2 \mapsto c_2,\ z_3 \mapsto c_1,\ z_4 \mapsto c_3\Big\}.$$

We have not yet looked into whether the problem is in NP, but we expect it to be so.

5 A theory for which ground disunifiability is in P whereas asymmetric unification is NP-hard

This theory is the same as the one mentioned in previous section, **ACUN**, but with a homomorphism added. It has an AC-convergent term rewriting system, which we call R_3:

$$\begin{array}{lll} x + x \to 0 & x + 0 \to x & x + (y + x) \to y \\ h(x+y) \to h(x) + h(y) & h(0) \to 0 & \end{array}$$

5.1 Ground disunification

Ground disunifiability [2] problem refers to checking for ground solutions for a set of disequations and equations. The restriction is that only the set of constants provided in the input, i.e., the equational theory and the equations and disequations, can be used; no new constants can be introduced.

We show that ground disunifiability modulo this theory can be solved in polynomial time, by reducing the problem to that of solving systems of linear equations. This involves finding the Smith Normal Form [11, 14, 13]. This gives us a general solution to all the variables or unknowns.

Suppose we have m equations in our ground disunifiability problem. We can assume without loss of generality that the disequations are of the form $z \neq 0$. For example, if we have disequations of the form $e_1 \neq e_2$, we introduce a new variable z and set $z = e_1 + e_2$ and $z \neq 0$. Let n be the number of variables or unknowns for which we have to find a solution.

For each constant in our ground disunifiability problem, we follow the approach similar to [12], of forming a set of linear equations and solving them to find ground solutions. We use $h^k x$ to represent the term $h(h(\ldots h(x)\ldots))$ and $H^k = h^{k_1}x + h^{k_2}x + \cdots + h^{k_n}x$ is a polynomial over $\mathbb{Z}_2[h]$. We have $s_i = H_{i1}x_1 + H_{i2}x_2 + \ldots + H_{im}x_n,\ H_{ij} \in \mathbb{Z}_2[h]$ and $t_i = H'_{i1}c_1 + H'_{i2}c_2 + \ldots + H'_{im}c_l,\ H'_{ij} \in \mathbb{Z}_2[h]$, where, $\{c_1, \ldots c_l\}$ is the set of constants and $\{x_1, \ldots x_n\}$ is the set of variables. For each constant $c_i, 1 \leq i \leq l$, and each variable x, we create a variable x^{c_i}. We then generate, for each constant c_i, a set of linear equations S^{c_i} of the form $AX =^? B$ with coefficients from the polynomial ring $\mathbb{Z}_2[h]$. The solutions are found by computing the Smith Normal Form of A. The procedure is provided in our tech report [18].

5.2 Ground Asymmetric Unification

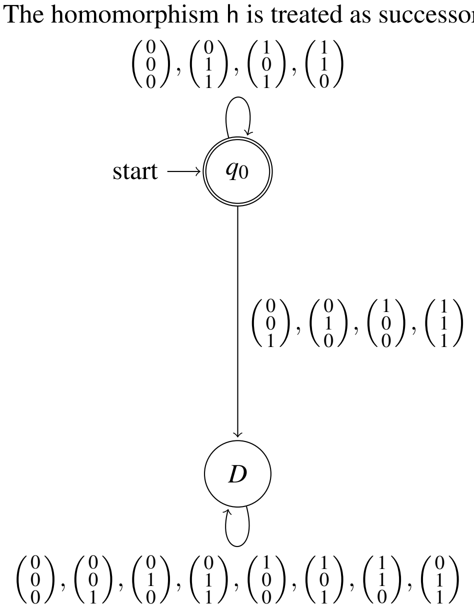
However, asymmetric unification modulo R_3 is NP-hard. Decidability can be shown by automata-theoretic methods as for Weak Second Order Theory of One successor (WS1S) [8, 5].

In WS1S we consider quantification over finite sets of natural numbers, along with one successor function. All equations or formulas are transformed into finite-state automata which accepts the strings that correspond to a model of the formula [15, 19]. This automata-based approach is key to showing decidability of WS1S, since the satisfiability of WS1S formulas reduces to the automata intersection-emptiness problem. We follow the same approach here.

For ease of exposition, let us consider the case where there is only one constant a. Thus every ground term can be viewed as a set of natural numbers. The homomorphism h is treated as a successor function. Just as in WS1S, the input to the automata are column vectors of bits. The length of each column vector is the number of variables in the problem.

$$\Sigma = \left\{ \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \right\}$$

Note that the $+$ operator behaves like the *symmetric set difference* operator. We illustrate how automata is constructed for one equation or formula $\mathsf{P} = \mathsf{Q} + \mathsf{R}$ in standard form, with the case of one constant a. The homomorphism h is treated as successor function.



Let $\mathsf{P_i}, \mathsf{Q_i}$ and $\mathsf{R_i}$ denote the i^{th} bits of P, Q and R *respectively*. $\mathsf{P_i}$ has a value 1, when either $\mathsf{Q_i}$ or $\mathsf{R_i}$ has a value 1. We need 3-bit alphabet symbols for this equation. For example, if $\mathsf{R_2} = 0$, $\mathsf{Q_2} = 1$, then $\mathsf{P_2} = 1$. The corresponding alphabet symbol is $\begin{pmatrix} P_2 \\ Q_2 \\ R_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$.

Hence, only strings with the alphabet symbols $\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \}$ are accepted by this automaton. Rest of the input symbols like $\{ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \}$ go to the dead state D as they violate the XOR property. Note that the string $\binom{1}{\substack{0\\1}}\binom{1}{\substack{1\\0}}$ is accepted by automaton. This corresponds to $\mathsf{P} = \mathsf{a} + \mathsf{h}(\mathsf{a})$. $\mathsf{Q} = \mathsf{h}(\mathsf{a})$ and $\mathsf{R} = \mathsf{a}$.

Once we have automata constructed for all the formulas, we take the intersection and check if there exists a string accepted by corresponding automata. If the intersection is not empty, then we have a solution or an asymmetric unifier for set of formulas. This technique can be extended to the case where we have more than one constant. Refer to our tech report [18] for more automata construction for single constant and details on more than one constant.

The exact complexity of this problem is open.

6 A theory for which time complexity of Asymmetric Unification varies based on ordering of function symbols

Let E_4 be the following equational theory:

$$g(a) \approx f(a,a,a) \qquad\qquad g(b) \approx f(b,b,b)$$

and let R_4 denote the following term rewriting system:

$$f(a,a,a) \to g(a) \qquad\qquad f(b,b,b) \to g(b).$$

This is clearly terminating, as can be easily shown by the *lexicographic path ordering (lpo)* [1] using the symbol ordering $f > g > a > b$. We show that asymmetric unification modulo the rewriting system R_4 is NP-complete. The proof is by a polynomial-time reduction from the Not-All-Equal Three-Satisfiability (NAE-3SAT) problem [4]. Let $U = \{x_1, x_2, \ldots, x_n\}$ be the set of variables, and $C = \{C_1, C_2, \ldots, C_m\}$ be the set of clauses. Each clause C_k, has to have at least one *true* literal and at least one *false* literal.

We create an instance of asymmetric unification as follows. We represent T by a and F by b. For each variable x_i we create the equation $f(x_i, x_i, x_i) \approx_{R_4} g(x_i)$. These make sure that each x_i is mapped to either a or b. For each clause $C_j = x_p \vee x_q \vee x_r$, we introduce a new variable z_j and create an asymmetric equation $EQ_j : z_j \approx_{\downarrow}^{?} f(x_p, x_q, x_r)$. Thus for any C, the instance of asymmetric unification problem constructed is

$$\mathscr{S} = \Big\{ f(x_1,x_1,x_1) \approx g(x_1), \ldots, f(x_n,x_n,x_n) \approx g(x_n) \Big\} \cup \Big\{ EQ_1, EQ_2, \ldots, EQ_m \Big\}$$

Example: Given $U = \{x_1, x_2, x_3, x_4\}$ and $C = \{x_1 \vee x_2 \vee x_3,\ x_1 \vee x_2 \vee x_4,\ x_1 \vee x_3 \vee x_4,\ x_2 \vee x_3 \vee x_4\}$ the constructed instance of asymmetric unification $\mathscr{S}$ is

$$\Big\{ f(x_1,x_1,x_1) \approx g(x_1),\ f(x_2,x_2,x_2) \approx g(x_2),\ f(x_3,x_3,x_3) \approx g(x_3),\ f(x_4,x_4,x_4) \approx g(x_4),$$
$$z_1 \ \approx_{\downarrow}^{?}\ f(x_1,x_2,x_3),\ z_2 \ \approx_{\downarrow}^{?}\ f(x_1,x_2,x_4),\ z_3 \ \approx_{\downarrow}^{?}\ f(x_1,x_3,x_4),\ z_4 \ \approx_{\downarrow}^{?}\ f(x_2,x_3,x_4) \Big\}$$

Again, membership in NP can be shown using the fact that R_4 is saturated by paramodulation [17].

However, if we orient the rules the other way, i.e., when $g > f > a > b$, we can show that asymmetric unifiability modulo this theory can be solved in polynomial time, where our term rewriting system R_5 is

$$g(a) \to f(a,a,a) \qquad\qquad g(b) \to f(b,b,b)$$

The algorithm is outlined in our tech report [18].

Acknowledgements: We appreciate and thank the referees for their comments which were very useful in improving the paper.

References

[1] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1999.

[2] Franz Baader and Klaus U. Schulz. Combination techniques and decision problems for disunification. *Theor. Comput. Sci.*, 142(2):229–255, 1995. Available at http://dx.doi.org/10.1016/0304-3975(94)00277-0.

[3] Franz Baader and Wayne Snyder. Unification theory. *Handbook of automated reasoning*, 1:445–532, 2001.

[4] Shreyaben Brahmakshatriya, Sushma Danturi, Kimberly A. Gero, and Paliath Narendran. Unification problems modulo a theory of *Until*. In Konstantin Korovin and Barbara Morawska, editors, *27th International Workshop on Unification, UNIF 2013, Eindhoven, Netherlands, June 26, 2013*, volume 19 of *EPiC Series in Computing*, pages 22–29. EasyChair, 2013. Available at http://www.easychair.org/publications/?page=723757558.

[5] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

[6] Wray L. Buntine and Hans-Jürgen Bürckert. On solving equations and disequations. *J. ACM*, 41(4):591–629, 1994. Available at http://doi.acm.org/10.1145/179812.179813.

[7] Hubert Comon. Disunification: A survey. In Jean-Louis Lassez and Gordon D. Plotkin, editors, *Computational Logic - Essays in Honor of Alan Robinson*, pages 322–359. The MIT Press, 1991.

[8] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961. Available at http://www.jstor.org/stable/1993511.

[9] Serdar Erbatur, Santiago Escobar, Deepak Kapur, Zhiqiang Liu, Christopher Lynch, Catherine A. Meadows, José Meseguer, Paliath Narendran, Sonia Santiago, and Ralf Sasse. Asymmetric unification: A new unification paradigm for cryptographic protocol analysis. In Maria Paola Bonacina, editor, *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, volume 7898 of *Lecture Notes in Computer Science*, pages 231–248. Springer, 2013. Available at http://dx.doi.org/10.1007/978-3-642-38574-2.

[10] Serdar Erbatur, Deepak Kapur, Andrew M. Marshall, Catherine A. Meadows, Paliath Narendran, and Christophe Ringeissen. On asymmetric unification and the combination problem in disjoint theories. In Anca Muscholl, editor, *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8412 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014. Available at http://dx.doi.org/10.1007/978-3-642-54830-7.

[11] Raymond N. Greenwell and Stanley Kertzner. Solving linear diophantine matrix equations using the Smith normal form (more or less), 2009.

[12] Qing Guo, Paliath Narendran, and David A. Wolfram. Complexity of nilpotent unification and matching problems. *Information and Computation*, 162(1-2):3–23, 2000.

[13] Erich Kaltofen, M.S. Krishnamoorthy, and B. David Saunders. Fast parallel computation of Hermite and Smith forms of polynomial matrices. *SIAM Journal on Algebraic Discrete Methods*, 8(4):683–690, 1987.

[14] R. Kannan. Solving systems of linear equations over polynomials. *Theoretical Computer Science*, 39:69 – 88, 1985. Available at http://www.sciencedirect.com/science/article/pii/0304397585901318.

[15] Felix Klaedtke and Harald Ruess. Parikh automata and monadic second-order logics with linear cardinality constraints. 2002.

[16] Zhiqiang Liu. *Dealing Efficiently with Exclusive-OR, Abelian Groups and Homomorphism in Cryptographic Protocol Analysis*. PhD thesis, Clarkson University, 2012.

[17] Christopher Lynch and Barbara Morawska. Basic syntactic mutation. In Andrei Voronkov, editor, *Automated Deduction - CADE-18, 18th International Conference on Automated Deduction, Copenhagen, Denmark, July 27-30, 2002, Proceedings*, volume 2392 of *Lecture Notes in Computer Science*, pages 471–485. Springer, 2002. Available at https://doi.org/10.1007/3-540-45620-1_37.

[18] Veena Ravishankar, Kimberly A. Gero, and Paliath Narendran. Asymmetric unification and disunification. Technical report, University at Albany–SUNY and The College of Saint Rose, Departments of Computer Science, 2017. Available at arXiv:1706.05066[cs.LO].

[19] Moshe Y Vardi and Thomas Wilke. Automata: from logics to algorithms. *Logic and automata*, 2:629–736, 2008.