Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Approximately Solving Set Equations

Franz Baader Pavlos Marantidis Alexander Okhotin

LTCS-Report 16-03

Submitted to *30th International Workshop on Unification*

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Approximately Solving Set Equations

Franz Baader
Department of Computer Scince
Technische Universität Dresden

Pavlos Marantidis
Department of Computer Scince
Technische Universität Dresden

Alexander Okhotin
Department of Mathematics and Statistics
University of Turku

April 18, 2017

Abstract

Unification with constants modulo the theory ACUI of an associative (A), commutative (C) and idempotent (I) binary function symbol with a unit (U) corresponds to solving a very simple type of set equations. It is well-known that solvability of systems of such equations can be decided in polynomial time by reducing it to satisfiability of propositional Horn formulae. Here we introduce a modified version of this problem by no longer requiring all equations to be completely solved, but allowing for a certain number of violations of the equations. We introduce three different ways of counting the number of violations, and investigate the complexity of the respective decision problem, i.e., the problem of deciding whether there is an assignment that solves the system with at most ℓ violations for a given threshold value ℓ.

1 Unification modulo ACUI and set equations

The complexity of testing solvability of unification problems modulo the theory

$$\mathsf{ACUI} \ := \ \{x+0=x, x+(y+z)=(x+y)+z, x+y=y+x, x+x=x\}$$

of an associative, commutative and idempotent function symbol "+" with a unit "0" was investigated in detail by Kapur and Narendran [**?**], who show that elementary ACUI-unification and ACUI-unification with constants are polynomial whereas general ACUI-unification is NP-complete. Here we concentrate on ACUI-unification with constants, but formally introduce the problem in its disguise of testing solvability of set equations.

Given a *finite* base set B and a set of variables $\mathbf{X} = \{Z_1, \ldots, Z_N\}$ that can assume as values subsets of B, consider a *system* Σ *of set equations*, which consists of finitely many equations of the following form:

$$K \cup X_1 \cup \ldots \cup X_m = L \cup Y_1 \cup \ldots \cup Y_n, \tag{1}$$

where K, L are subsets of B and $X_1, \ldots, X_m, Y_1, \ldots, Y_n \in \mathbf{X}$.

A B*-assignment* is a mapping of subsets of B to the variables, i.e., it is of the form $\sigma\colon \mathbf{X} \to \mathfrak{P}(B)$. If there is no confusion, we will omit the prefix B- from B-assignment. Such an assignment σ is a *solution* of the system of set equations Σ if

$$K \cup \sigma(X_1) \cup \ldots \cup \sigma(X_m) = L \cup \sigma(Y_1) \cup \ldots \cup \sigma(Y_n)$$

holds for all equations of the form (1) in Σ.

The correspondence between unification and set equations can be understood more easily with an example.

Example 1. Consider the following ACUI-unification problem:

$$\begin{aligned} x_1 + x_2 &= a + b + c \\ a + x_2 &= b + x_3 \\ x_1 + x_3 &= a + c \end{aligned}$$

Setting $B = \{a, b, c\}$ to be the set of all constants and introducing set variables X_i for every variable x_i, the above problem can be transformed into the system of set equations:

$$\begin{aligned} X_1 \cup X_2 &= \{a, b, c\} \\ \{a\} \cup X_2 &= \{b\} \cup X_3 \\ X_1 \cup X_3 &= \{a, c\} \end{aligned}$$

A solution to the latter is the assignment σ, such that $\sigma(X_1) = \{a, c\}$, $\sigma(X_2) = \{b\}$, $\sigma(X_3) = \{a\}$. This corresponds to the substitution τ, such that $\tau(x_1) = a+c$, $\tau(x_2) = b$, $\tau(x_3) = a$, which is a solution to the initial unification problem.

Solvability of a system of set equations can be reduced in polynomial time (see below) to satisfiability of propositional Horn formulae [?], which can be tested in linear time [?].

To introduce this reduction, we define Boolean variables $p(a, X)$ for every $a \in B$ and $X \in \mathbf{X}$. The intuitive semantics of these variables is that $p(a, X)$ is true iff a is *not* in X for the given assignment.

Now, for each equation of the form (1) and each $a \in K \setminus L$ we generate the Horn clauses

$$p(a, Y_1) \wedge \ldots \wedge p(a, Y_n) \rightarrow \perp.$$

Indeed, whenever an element $a \in B$ is in K but not in L, for the equation to hold true, a must be in some Y_j. The symmetric Horn clauses are also produced, i.e., for each $a \in L \setminus K$

$$p(a, X_1) \wedge \ldots p(a, X_m) \rightarrow \perp.$$

It remains to deal with the elements $a \notin K \cup L$. First, if a belongs to none of the variables on the right-hand side, then it should not belong to any of the variables on the left-hand side, which is expressed by the Horn clauses

$$p(a, Y_1) \wedge \ldots \wedge p(a, Y_n) \rightarrow p(a, X_j) \qquad \text{for all } j = 1, \ldots, m.$$

Symmetrically, if a is not on the left-hand side, it cannot be on the right-hand side, which yields

$$p(a, X_1) \wedge \ldots \wedge p(a, X_m) \rightarrow p(a, Y_j) \qquad \text{for all } j = 1, \ldots, n.$$

The number of derived Horn clauses and their sizes are polynomial in the size of the given system Σ of set equations, where the size of Σ is the sum of the cardinality of B, the number of variables in $\mathbf{X}$, and the number of equations in Σ. The size of a Horn clause is just the number of literals occurring in it.

It is easy to see that the Horn formula obtained by conjoining all the Horn clauses derived from a system of set equations is satisfiable iff the original system of set equations has a solution (see [?] for details). Consequently, solvability of systems of set equations can be decided in polynomial time.

2 Minimizing the number of violated equations

We say that the B-assignment σ *violates* a set equation of the form (1) if

$$K \cup \sigma(X_1) \cup \ldots \cup \sigma(X_m) \neq L \cup \sigma(Y_1) \cup \ldots \cup \sigma(Y_n).$$

Given a base set B, a set of variables $\mathbf{X} = \{Z_1, \ldots, Z_N\}$, a system Σ of k set equations of the form (1), and a nonnegative integer ℓ, we now ask whether there exists a B-assignment σ such that at most ℓ of the equations of the system are violated by σ. We call this decision problem MinVEq-SetEq. For a given ℓ, MinVEq-SetEq(ℓ) consists of all systems of set equations for which there is a B-assignment that violates at most ℓ equations of the system.

We will show that MinVEq-SetEq is NP-complete using reductions to and from Max-HSAT. Given a Horn formula φ that is a conjunction of k Horn clauses and a nonnegative integer ℓ, Max-HSAT asks whether there is a propositional assignment τ that satisfies at least ℓ of the Horn clauses of φ. For a given ℓ, Max-HSAT(ℓ) consists of those Horn formulae for which there is a propositional assignment that satisfies at least ℓ of its Horn clauses. It is well-known that Max-HSAT is NP-complete [?].

Reducing MinVEq-SetEq to Max-HSAT For this purpose, we introduce new Boolean variables $good(i)$, whose rôle is to determine whether the ith equation is to be satisfied or not. We conjoin $good(i)$ to the left-hand side of each of the Horn clauses derived from the ith equation, i.e., if the ith equation is of the form (1), then we generate the following Horn clauses:

- For each $a \in K \setminus L$: $good(i) \wedge p(a, Y_1) \wedge \ldots \wedge p(a, Y_n) \rightarrow \bot$;
- For each $a \in L \setminus K$: $good(i) \wedge p(a, X_1) \wedge \ldots \wedge p(a, X_m) \rightarrow \bot$;
- For each $a \notin K \cup L$:

$$\begin{aligned} good(i) \wedge p(a, Y_1) \wedge \ldots \wedge p(a, Y_n) &\rightarrow p(a, X_j) \qquad \text{for all } j = 1, \ldots, m; \\ good(i) \wedge p(a, X_1) \wedge \ldots \wedge p(a, X_m) &\rightarrow p(a, Y_j) \qquad \text{for all } j = 1, \ldots, n. \end{aligned}$$

- Furthermore, we add the Horn clause $\top \rightarrow good(i)$.

If k' is the number of clauses generated by the original reduction (see Section 1) and k is the number of set equations in the system Σ, then we obtain $k' + k$ Horn clauses in this modified reduction. Let $\varphi_\Sigma = C_1 \wedge \cdots \wedge C_{k'+k}$ denote the Horn formula obtained by conjoining these Horn clauses.

Let us illustrate the above construction with a small example.

Example 2. Consider the follwing system of set equations, that is not solvable:

$$\begin{aligned} X_1 \cup X_2 &= \{a\} \\ \{b, c\} \cup X_2 &= \{b\} \cup X_3 \\ X_1 \cup X_3 &= \{c\} \end{aligned}$$

For the first equation, the reduction in Section 1 introduces the following clauses. Since $K = \emptyset$, $L = \{a\}$:

- $a \in L \setminus K$, thus we get

$$p(a, X_1) \wedge p(a, X_2) \rightarrow \bot$$

- $b, c \notin K \cup L$, thus we get

$$\top \rightarrow p(b, X_1) \qquad \top \rightarrow p(b, X_2) \qquad \top \rightarrow p(c, X_1) \qquad \top \rightarrow p(c, X_2)$$

Proceeding likewise for the other two equations, conjoining the $good(i)$ literals and introducing the $\top \rightarrow good(i)$ clauses, we finally get the Horn clauses:

$$\begin{array}{ll}
good(1) \wedge p(a, X_1) \wedge p(a, X_2) \rightarrow \bot & good(3) \rightarrow p(a, X_1) \\
good(1) \rightarrow p(b, X_1) & good(3) \rightarrow p(a, X_3) \\
good(1) \rightarrow p(b, X_2) & good(3) \rightarrow p(b, X_1) \\
good(1) \rightarrow p(c, X_1) & good(3) \rightarrow p(b, X_3) \\
good(1) \rightarrow p(c, X_2) & good(3) \wedge p(c, X_1) \wedge p(c, X_3) \rightarrow \bot \\
good(2) \wedge p(a, X_2) \rightarrow p(a, X_3) & \top \rightarrow good(1) \\
good(2) \wedge p(a, X_3) \rightarrow p(a, X_2) & \top \rightarrow good(2) \\
good(2) \wedge p(c, X_3) \rightarrow \bot & \top \rightarrow good(3)
\end{array}$$

The system contains $k = 3$ equations and the original reduction would produce $k' = 13$ clauses. Thus, φ_Σ contains $k' + k = 16$ Horn clauses.

The truth assignment that sets $p(a, X_1)$, $p(a, X_2)$, $p(a, X_3)$, $p(c, X_3)$ and $good(3)$ to false and all other variables to true, manages to satisfy all but the last clause. This corresponds to the B-assignment σ such that $\sigma(X_1) = \sigma(X_2) = \{a\}$ and $\sigma(X_3) = \{a, c\}$, that satisfies the first two equations and violates the third one.

Intuitively, setting the Boolean variable $good(i)$ to false "switches off" the Horn clauses induced by the ith equation in the original reduction. Consequently, the satisfaction of these clauses is no longer enforced, which means that the ith equation may be violated. By maximizing satisfaction of the clauses $\top \rightarrow good(i)$, we thus minimize the number of violated set equations. More precisely, we can show the following lemma.

Lemma 1. *Let Σ be a system of set equations consisting of k equations and generating k' clauses in the reduction introduced in Section 1. Then we have*

$$\Sigma \in \textit{MinVEq-SetEq}(\ell) \quad \textit{iff} \quad \varphi_\Sigma \in \textit{Max-HSAT}((k' + k) - \ell).$$

Proof. Let $\Sigma \in$ MinVEq-SetEq(ℓ). This means that there exists an assignment σ of sets to variables such that at most ℓ of the equations are violated. Suppose that the ith equation is not violated. Then, set $good(i)$ to true, and for all elements $a \in B$ and every variable Z appearing in the ith equation, set $p(a, Z)$ to true iff

$a \notin \sigma(Z)$. This assignment makes all clauses corresponding to the ith equation valuate to true, plus the $\top \to \mathit{good}(i)$ clause.

If the jth equation is violated, set $\mathit{good}(j)$ to false. Then all clauses corresponding to the jth equation valuate to true. Thus all original k' equations valuate to true, plus at least $k - \ell$ of the $\top \to \mathit{good}(i)$ clauses. We conclude that $\varphi_\Sigma \in$ Max-HSAT$((k' + k) - \ell)$.

For the opposite direction, let $\varphi_\Sigma \in$ Max-HSAT$((k' + k) - \ell)$. This means that there is a truth assignment v and a set of indices $I \subseteq \{1, \ldots, k'+k\}$, $|I| \geqslant k'+k-\ell$ such that $\bigwedge_{i \in I} C_i(v) = 1$. Initially, observe that since there are k equations, $k \geqslant \ell$. Thus, if at least $k' + k - \ell$ clauses valuate to true, this implies that h of the $\top \to \mathit{good}(i)$ clauses (and thus $\mathit{good}(i)$s) valuate to true, with $k \geqslant h \geqslant k - \ell$.

Suppose that $\mathit{good}(i)$ is set to true. If all clauses corresponding to the ith equation valuate to true, we can derive an assignment that does not violate the ith equation by setting $a \in \sigma(Z)$ iff $p(a, Z)$ is set to false. If not all clauses corresponding to the ith equation valuate to true, consider a new valuation, where $\mathit{good}(i)$ is set to false. This makes all clauses corresponding to the ith equation valuate to true. Thus, at least as many clauses as before valuate to true. We can continue this procedure until all original k' clauses valuate to true, while at least $k - \ell$ of the $\mathit{good}(i)$s are set to true. Thus $\Sigma \in$ MinVEq-SetEq(ℓ). □

Since Max-HSAT is in NP, this lemma implies that MinVEq-SetEq also belongs to NP.

Reducing Max-HSAT to MinVEq-SetEq Consider the Horn formula $\varphi = C_1 \wedge \ldots \wedge C_k$, where C_i is a Horn clause for $i = 1, \ldots, k$. To construct a corresponding system of set equations, we use the singleton base set $B = \{a\}$. For every Boolean variable p appearing in φ, we introduce a set variable X_p. Intuitively, a belongs to X_p iff p is set to false. Now, each Horn clause in φ yields the following set equations:

- If C_i is of the form $p_1 \wedge \ldots \wedge p_n \to p$, then the corresponding set equation is
$$X_{p_1} \cup \ldots \cup X_{p_n} \cup X_p = X_{p_1} \cup \ldots \cup X_{p_n}.$$
Obviously, this equation enforces that a cannot belong to X_p if it does not belong to any of the variables X_{p_i}.

- If C_i is of the form $p_1 \wedge \ldots \wedge p_n \to \perp$, then the corresponding set equation is
$$X_{p_1} \cup \ldots \cup X_{p_n} = \{a\}.$$
This equation enforces that a must belong to one of the variables X_{p_i}.

- If C_i is of the form $\top \to p$, then the corresponding set equation is

$$\emptyset = X_p.$$

  This equation ensures that a cannot belong to X_p.

The following example exhibits the construction of Σ_φ.

Example 3. Consider the Horn formula

$$\varphi = (p_1 \wedge p_2 \to p_3) \wedge (p_1 \wedge p_3 \to \bot) \wedge (p_2 \wedge p_3 \to \bot) \wedge (\top \to p_1) \wedge (\top \to p_2).$$

The corresponding set equations are

$$\begin{aligned} X_{p_1} \cup X_{p_2} \cup X_{p_3} &= X_{p_1} \cup X_{p_2} \\ X_{p_1} \cup X_{p_3} &= \{a\} \\ X_{p_2} \cup X_{p_3} &= \{a\} \\ X_{p_1} &= \emptyset \\ X_{p_2} &= \emptyset \end{aligned}$$

Given the intuition underlying the variables X_p (a belongs to X_p iff p is set to false), it is easy to prove the following lemma.

Lemma 2. *Let $\varphi = C_1 \wedge \ldots \wedge C_k$ be a Horn formula and Σ_φ the corresponding system of set equations. Then $\varphi \in$ Max-HSAT(ℓ) iff $\Sigma_\varphi \in$ MinVEq-SetEq$(k - \ell)$.*

Proof. Suppose that $\varphi \in$ Max-HSAT(ℓ). This means that there exists a truth assignment v and a set of indices $I \subseteq \{1, \ldots, k\}$, $|I| = \ell$ such that $\bigwedge_{i \in I} C_i(v) = 1$.

It suffices to show that there is a mapping $\sigma\colon \{X_{p_1}, \ldots, X_{p_n}\} \to \{\emptyset, \{a\}\}$ such that ℓ equations of Σ_φ are not violated. Indeed, given the assignment v mentioned before, define $\sigma(X_{p_i}) = \{a\}$ iff $v(p_i) = 0$. Then the claim is that, for every $i \in I$, the ith equation is not violated for this mapping. Indeed, if the ith equation is of the form:

- $X_{p_1} \cup \ldots \cup X_{p_n} \cup X_p = X_{p_1} \cup \ldots \cup X_{p_n}$, then $C_i = p_1 \vee \cdots \vee p_n \to p$ evaluated to true under v, thus meaning that either one of the p_is are evaluated to false, or p is evaluated to true. In the first case, a belongs to both sides, thus the equation is not violated. In the second case, $\sigma(X_p) = \emptyset$, and again the equation is not violated.

- $X_{p_1} \cup \cdots \cup X_{p_n} = \{a\}$, then $C_i = p_1 \vee \cdots \vee p_n \to \bot$ evaluated to true under v, thus meaning that one of the p_is are evaluated to false. By the definition of σ, this means that $\sigma(X_{p_i}) = \{a\}$ for some $i = 1, \ldots, n$, and thus $\sigma(X_{p_1} \cup \cdots \cup X_{p_n}) = \{a\}$, i.e., the equation is not violated.

- $\emptyset = X_p$, then $C_i = \top \to p$ evaluated to true under v, thus meaning that p is evaluated to true. By the definition of σ, this means that $\sigma(X_p) = \emptyset$, and thus the equation is not violated.

Thus, an assignment was derived w.r.t. which at least ℓ equations of Σ_φ are not violated. Thus $\Sigma_\varphi \in$ MinVEq-SetEq$(k - \ell)$.

For the opposite direction, if $\Sigma_\varphi \in$ MinVEq-SetEq$(k - \ell)$, there is a mapping σ such that ℓ equations of Σ_φ are not violated. Define $v(p) = 0$ iff $\sigma(X_p) = \{a\}$ and proceed in the same way to prove that $\varphi \in$ Max-HSAT(ℓ). $\square$

Since Max-HSAT is NP-hard, this lemma implies that MinVEq-SetEq is also NP-hard. Put together, the two lemmas yield the exact complexity of the MinVEq-SetEq problem.

Theorem 1. *MinVEq-SetEq is NP-complete. NP-hardness holds even if we restrict the cardinality of the base set B to* 1.

3 Minimizing the number of violating elements

Instead of minimizing the number of violated equations, we can also minimize the number of violating elements of B.

Given an assignment σ, we say that $a \in B$ *violates an equation* of the form (1) *w.r.t.* σ if $a \in (K \cup \sigma(X_1) \cup \ldots \cup \sigma(X_m)) \,\Delta (L \cup \sigma(Y_1) \cup \ldots \cup \sigma(Y_n))$, where Δ denotes the symmetric difference of two sets. We say that $a \in B$ *violates the system of set equations* Σ *w.r.t.* σ if it violates some equation in Σ w.r.t. σ. Given a base set B, a set of variables $\mathbf{X} = \{Z_1, \ldots, Z_N\}$, a system Σ of k set equations and a nonnegative integer ℓ, we now ask whether there exists a B-assignment σ such that at most ℓ of the elements of B violate Σ w.r.t. σ. We call this decision problem MinVEl-SetEq. For a given ℓ, MinVEl-SetEq(ℓ) consists of all systems of set equations for which there is a B-assignment σ such that at most ℓ of the elements of B violate Σ w.r.t. σ.

In contrast to the problem MinVEq-SetEq considered in the previous section, MinVEl-SetEq can be solved in polynomial time. In order to show this, we introduce the notion of projection. Given an element $a \in B$, the *projection of an equation* of the form (1) to a is the equation

$$(K \cap \{a\}) \cup X_1 \cup \ldots \cup X_m = (L \cap \{a\}) \cup Y_1 \cup \ldots \cup Y_n. \qquad (2)$$

The *projection of a system* of set equations Σ to a, Σ^a, is the system of the projections of all equations in Σ to a. Note that, for Σ^a, we use the base set $\{a\}$. Finally, the *projection of a B-assignment* σ to a is the $\{a\}$-assignment $\sigma^a \colon \mathbf{X} \to \mathfrak{P}(\{a\})$ defined as $\sigma^a(X) = \sigma(X) \cap \{a\}$.

Continuing Example 1 from above, note that a violates Σ w.r.t. σ, while b, c do not. Furthermore, for each element of $B = \{a, b, c\}$, we can get a projection of Σ:

$$\begin{array}{ccc}
\underline{\text{Projection to } a} & \underline{\text{Projection to } b} & \underline{\text{Projection to } c} \\[1em]
\begin{array}{rcl} X_1 \cup X_2 & = & \{a\} \\ X_2 & = & X_3 \\ X_1 \cup X_3 & = & \emptyset \end{array} &
\begin{array}{rcl} X_1 \cup X_2 & = & \emptyset \\ \{b\} \cup X_2 & = & \{b\} \cup X_3 \\ X_1 \cup X_3 & = & \emptyset \end{array} &
\begin{array}{rcl} X_1 \cup X_2 & = & \emptyset \\ \{c\} \cup X_2 & = & X_3 \\ X_1 \cup X_3 & = & \{c\} \end{array}
\end{array}$$

Likewise, for the assignment σ we get the projections:

$$\begin{array}{ccc}
\underline{\sigma^a} & \underline{\sigma^b} & \underline{\sigma^c} \\
\begin{array}{rcl} \sigma^a(X_1) & = & \{a\} \\ \sigma^a(X_2) & = & \{a\} \\ \sigma^a(X_3) & = & \{a\} \end{array} &
\begin{array}{rcl} \sigma^b(X_1) & = & \emptyset \\ \sigma^b(X_2) & = & \emptyset \\ \sigma^b(X_3) & = & \emptyset \end{array} &
\begin{array}{rcl} \sigma^c(X_1) & = & \emptyset \\ \sigma^c(X_2) & = & \emptyset \\ \sigma^c(X_3) & = & \{c\} \end{array}
\end{array}$$

One can easily check that σ^b and σ^c solve Σ^b and Σ^c respectively, while this is not the case for σ^a. This is not unrelated to earlier note that a violates Σ w.r.t. σ, as shown in the next lemma.

Lemma 3. *The following facts hold:*

1. *The element $a \in B$ violates Σ w.r.t. σ iff σ^a does not solve Σ^a.*
2. *Given $\{a\}$-assignments σ_a for all $a \in B$, define the B-assignment σ as*
$$\sigma(X) = \bigcup_{a \in B} \sigma_a(X) \quad \textit{for all } X \in \mathbf{X}.$$
Then we have $\sigma^a = \sigma_a$ for all $a \in B$.
3. *There is a B-assignment σ such that at most ℓ of the elements of B violate Σ w.r.t. σ iff at most ℓ of the systems of set equations Σ^a $(a \in B)$ are not solvable.*

Proof. 1. Assume that a violates Σ w.r.t. σ. Then, a violates at least one equation w.r.t. σ, i.e.,

$$a \in (K \cup \sigma(X_1) \cup \ldots \cup \sigma(X_m)) \,\Delta (L \cup \sigma(Y_1) \cup \ldots \cup \sigma(Y_n)).$$

Thus,

$$a \in ((K \cap \{a\}) \cup \sigma(X_1) \cap \{a\} \cup \ldots \cup \sigma(X_m) \cap \{a\}) \,\Delta ((L \cap \{a\}) \cup \sigma(Y_1) \cap \{a\} \cup \ldots \cup \sigma(Y_n) \cap \{a\}),$$

which is exactly

$$a \in ((K \cap \{a\}) \cup \sigma^a(X_1) \cup \ldots \cup \sigma^a(X_m)) \,\Delta ((L \cap \{a\}) \cup \sigma^a(Y_1) \cup \ldots \cup \sigma^a(Y_n)).$$

Thus, σ^a does not solve Σ^a.

For the opposite direction, assume that σ^a does not solve Σ^a. Then, there exists an equation of the form (1), such that

$$a \in ((K \cap \{a\}) \cup \sigma^a(X_1) \cup \ldots \cup \sigma^a(X_m)) \,\Delta((L \cap \{a\}) \cup \sigma^a(Y_1) \cup \ldots \cup \sigma^a(Y_n)).$$

The definition of σ^a implies that

$$a \in (K \cup \sigma(X_1) \cup \ldots \cup \sigma(X_m)) \,\Delta(L \cup \sigma(Y_1) \cup \ldots \cup \sigma(Y_n)),$$

and thus a violates Σ w.r.t. σ.

2. $\sigma^a(X) = \left(\bigcup_{a' \in B} \sigma_{a'}(X)\right) \cap \{a\} = \bigcup_{a' \in B} (\sigma_{a'}(X) \cap \{a\}) = \sigma_a(X) \cap \{a\} = \sigma_a(X)$.

3. Suppose that there is a B-assignment σ such that at most ℓ of the elements of B violate Σ w.r.t. σ. This means that at least $k - \ell$ of B do not violate Σ w.r.t. σ, and thus, by (1), the corresponding $k - \ell$ systems are solvable. Equivalently, at most ℓ systems are not solvable.

   For the opposite direction, suppose that at most ℓ of the systems Σ^a are not solvable. Solvability of systems Σ^a is independant for different elements $a \in B$. For every $a \in B$, if Σ^a is solvable, consider the $\{a\}$-assignment σ_a that solves it, or any $\{a\}$-assignment otherwise. Then $\sigma(X) = \bigcup_{a \in B} \sigma_a(X)$ for all $X \in \mathbf{X}$ is a B-assignment such that at most ℓ of the elements of B violate Σ w.r.t. σ.

□

Thus, to check whether $\Sigma \in$ MinVEl-SetEq(ℓ), it is sufficient to check which of the systems of set equations Σ^a for $a \in B$ are solvable. This can obviously be done in polynomial time.

Theorem 2. *MinVEl-SetEq is in P.*

4 Minimizing the number of violations

A disadvantage of the measure used in the previous section is that it does not distinguish between elements that violate only one equation and those violating many equations. To overcome this problem, we count for each violating element how many equations it actually violates. We say that $a \in B$ *violates the system of set equations* Σ p *times w.r.t.* σ if it violates p equations in Σ w.r.t. σ. Further, we say that σ *violates* Σ q *times* if $q = \sum_{a \in B} p_a$ where, for each $a \in B$, the element a violates Σ p_a times w.r.t. σ.

Given a base set B, a set of variables $\mathbf{X} = \{Z_1, \ldots, Z_N\}$, a system Σ of k equations, and a positive integer ℓ, we now ask whether there is an assignment σ that violates Σ at most ℓ times. We call this decision problem MinV-SetEq. For a given ℓ, MinV-SetEq(ℓ) consists of all systems of set equations for which there is a B-assignment σ such that σ violates Σ at most ℓ times.

It is easy to adapt the approach used in Section 2 to solve MinVEq-SetEq to this new problem. Basically, we now introduce Boolean variables $\mathit{good}(i, a)$ (instead of simply $\mathit{good}(i)$) to characterize whether the element $a \in B$ violates the ith equation. We conjoin $\mathit{good}(i, a)$ to the left-hand side of each of the Horn clauses derived from the ith equation for a. Furthermore, we add the Horn clauses $\top \to \mathit{good}(i, a)$.

Following the earlier notation, we obtain $k' + k|B|$ Horn clauses in this modified reduction, and again use φ_Σ to denote the obtained Horn formula.

Continuing Example 1, completely similarly to the MinVEq-SetEq case, from Σ we can derive the following Horn clauses:

$$\begin{array}{ll}
\mathit{good}(1) \wedge p(a, X_1) \wedge p(a, X_2) \to \bot & \top \to \mathit{good}(1, a) \\
\mathit{good}(1) \to p(b, X_1) & \top \to \mathit{good}(1, b) \\
\mathit{good}(1) \to p(b, X_2) & \top \to \mathit{good}(1, c) \\
\mathit{good}(1) \to p(c, X_1) & \top \to \mathit{good}(2, a) \\
\mathit{good}(1) \to p(c, X_2) & \top \to \mathit{good}(2, b) \\
\mathit{good}(2) \wedge p(a, X_2) \to p(a, X_3) & \top \to \mathit{good}(2, c) \\
\mathit{good}(2) \wedge p(a, X_3) \to p(a, X_2) & \top \to \mathit{good}(3, a) \\
\mathit{good}(2) \wedge p(c, X_3) \to \bot & \top \to \mathit{good}(3, b) \\
\mathit{good}(3) \to p(a, X_1) & \top \to \mathit{good}(3, c) \\
\mathit{good}(3) \to p(a, X_3) & \\
\mathit{good}(3) \to p(b, X_1) & \\
\mathit{good}(3) \to p(b, X_3) & \\
\mathit{good}(3) \wedge p(c, X_1) \wedge p(c, X_3) \to \bot &
\end{array}$$

The following lemma implies that MinV-SetEq is in NP.

Lemma 4. *Let Σ be a system of set equations over the base set B, consisting of k equations and generating k' clauses in the reduction introduced in Section 1. Denote with $\varphi_\Sigma = C_1 \wedge \cdots \wedge C_{k'+k|B|}$ the Horn formula derived by the modified reduction. Then we have*

$$\Sigma \in \mathit{MinV\text{-}SetEq}(\ell) \quad \mathit{iff} \quad \varphi_\Sigma \in \mathit{Max\text{-}HSAT}((k' + k|B|) - \ell).$$

Proof. Let $\Sigma \in$ MinV-SetEq(ℓ). This means that there exists an assignment σ of sets to variables such that σ violates Σ at most ℓ times. Suppose that a does not violate the ith equation w.r.t σ. Then, set $good(i, a)$ to true, and for every variable Z appearing in the ith equation, set $p(a, Z)$ to true iff $a \notin \sigma(Z)$. This assignment makes all clauses corresponding to the ith equation and the element a valuate to true, plus the $\top \rightarrow good(i, a)$ clause.

If a violates the jth equation w.r.t. σ, set $good(j, a)$ to false. Then all clauses corresponding to the jth equation valuate to true. Thus all original k' equations valuate to true, plus at least $k|B| - \ell$ of the $\top \rightarrow good(i, a)$ clauses. We conclude that $\varphi_\Sigma \in$ Max-HSAT$((k' + k|B|) - \ell)$.

For the opposite direction, let $\varphi_\Sigma \in$ Max-HSAT$((k' + k|B|) - \ell)$. This means that there is a truth assignment v and a set of indices $I \subseteq \{1, \ldots, k' + k|B|\}$, $|I| \geqslant k' + k|B| - \ell$ such that $\bigwedge_{i \in I} C_i = 1$. Observe that, since every element of B can violate Σ at most k times, $k|B| \geqslant \ell$. Thus, if at least $k' + k|B| - \ell$ clauses valuate to true, this implies that h of the $\top \rightarrow good(i, a)$ clauses (and thus $good(i, a)$s) valuate to true, with $k|B| \geqslant h \geqslant k|B| - \ell$.

Suppose that $good(i, a)$ is set to true. If all clauses corresponding to the ith equation and $a \in B$ valuate to true, we can derive an assignment that does not violate the ith equation by setting $a \in \sigma(Z)$ iff $p(a, Z)$ is set to false. If not all clauses corresponding to the ith equation and a valuate to true, consider a new valuation, where $good(i, a)$ is set to false. This makes all clauses corresponding to the ith equation and a valuate to true. Thus, at least as many clauses as before valuate to true. We can continue this procedure until all original k' clauses valuate to true, while at least $k|B| - \ell$ of the $good(i, a)$s are set to true. Thus $\Sigma \in$ MinV-SetEq(ℓ). □

For base sets of cardinality 1, MinV-SetEq coincides with MinVEq-SetEq, which we have shown to be NP-hard even in this restricted setting. This shows that the complexity upper bound of NP is optimal.

Theorem 3. *MinV-SetEq is NP-complete.*

5 Conclusion

Our investigation of how to approximately solve set equations was motivated by unification modulo the equational theory ACUI. We have shown that, depending on how we measure violations, the complexity of the problem may stay in P or increase to NP. As further work, we have started to look at approximate unification modulo the equational theory ACUIh. Since ACUIh-unification can be reduced to solving certain language equations [?], we thus need to investigate approximately solving language equations. In this setting, the elements of the sets are words, i.e., structured objects, and measures for violations should take this structure into account.

References