Approximately Solving Set Equations

Franz Baader[1], Pavlos Marantidis[1], and Alexander Okhotin[2]

[1] TU Dresden, Germany, firstname.lastname@tu-dresden.de
[2] Dept. of Mathematics and Statistics, University of Turku, Finland, alexander.okhotin@utu.fi

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

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

of an associative, commutative and idempotent function symbol "+" with a unit "0" was investigated in detail by Kapur and Narendran [KN92], 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.

Solvability of a system of set equations can be reduced in polynomial time (see below) to satisfiability of propositional Horn formulae [KN92], which can be tested in linear time [DG84].

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

$$p(a, Y_1) \wedge \ldots \wedge p(a, Y_n) \to \bot.$$

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 $a \in L \setminus K$

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

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 [KN92] 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 [JS87].

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 $a \notin K \cup L$:

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

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

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.

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 $\top \to good(i)$, 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*

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

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 $p_1 \wedge \ldots \wedge p_n \to \bot$, 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.

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 $\varphi \in \textit{Max-HSAT}(\ell)$ iff $\Sigma_\varphi \in \textit{MinVEq-SetEq}(k - \ell)$.*

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

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

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 $\sigma^a(X) = \sigma(X) \cap \{a\}$.

The following facts are easy to show:

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

$$\sigma(X) = \bigcup_{a \in B} \sigma_a(X) \quad \text{for all } X \in \mathbf{X}.$$

   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.

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 $good(i, a)$ (instead of simply $good(i)$) to characterize whether the element $a \in B$ violates the ith equation. We conjoin $good(i, a)$ to the left-hand side of each of the Horn clauses derived from the ith equation for a. Furthermore, we add the Horn clauses $\top \rightarrow good(i, a)$.

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

Lemma 3. *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*

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

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. The idea is that, even if there is no unifier, there may be substitutions that almost are unifiers, i.e., that almost solve the unification problem. In some applications it may be interesting to find such approximate solutions, which violate some of the equations, but in a minimal way. 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, which corresponds to unification in the description logic $\mathcal{FL}_0$ [BN01]. This sort of unification can be used to detect redundancies in ontologies, and approximate unification may allow to detect more potential cases of redundancy. Since ACUIh-unification can be reduced to solving certain language equations [BN01], 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. Our investigation of unification modulo ACUI can be seen as a warm-up exercise for this more challenging task.

References

[BN01] Franz Baader and Paliath Narendran. Unification of concept terms in description logics. *J. of Symbolic Computation*, 31(3):277–305, 2001.

[DG84] W. F. Dowling and J. Gallier. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *Journal of Logic Programmming*, 1(3):267–284, 1984.

[JS87] Brigitte Jaumard and Bruno Simeone. On the complexity of the maximum satisfiability problem for Horn formulas. *Inf. Process. Lett.*, 26(1):1–4, 1987.

[KN92] D. Kapur and P. Narendran. Complexity of unification problems with associative-commutative operators. *J. Automated Reasoning*, 9:261–288, 1992.