Systems of Set Constraints with Negative Constraints are *NEXPTIME*-Complete

Kjartan Stefansson *
stefan@cs.cornell.edu

August 30, 1993

Abstract

A system of set constraints is a system of expressions $E \subseteq F$ where E and F describe sets of ground terms over a ranked alphabet. Aiken *et al.* [AKVW93] classified the complexity of such systems. In [AKW93] it was shown that if negative constraints $E \not\subseteq F$ were allowed, then the problem is decidable. This was done by reduction to a Diophantine problem, the Nonlinear Reachability Problem, which was shown to be decidable.

We show that nonlinear reachability is *NP*-complete. By bounding the reduction of [AKW93] we conclude that systems of set constraints, allowing negative constraints, is *NEXPTIME*-complete.

1 Introduction

In [AKW93] Aiken *et al.* show that it is decidable whether a system of set constraints, including negative constraints, has a solution. The same result was achieved independently by Gilleron *el at.* [GTT93]. The result of Aiken *et al.* is achieved by reducing the set constraint problem to a hypergraph reachability problem, which in turn is reduced to the Nonlinear Reachability Problem (*NRP*). They proceed to show that *NRP* is decidable, thus proving the set constraint problem decidable.

*Computer Science Department, Cornell University, Ithaca, NY 14853

In this paper we show that if *NRP* has a solution, then it must have one of polynomial size. In particular, this shows $NRP \in NP$, giving the first elementary bound on the Nonlinear Reachability Problem. It is easy to show that *NRP* is *NP*-hard, so *NRP* is *NP*-complete. This also gives the first elementary bound on the system of set constraints where negative constraints are allowed. We bound the reduction of [AKW93] and conclude that the general set constraint problem can be solved in nondeterministic exponential time. Combining this with the fact that the simpler problem, solving systems of set constraints with positive constraints only is *NEXPTIME*-hard [AKVW93], we have shown the general problem to be *NEXPTIME*-complete too.

We proceed by defining the *NRP* and an associated graph, whose properties characterize how close the *NRP* instance is to being solved.

2 The Nonlinear Reachability Problem

In [AKW93] Aiken *et al.* define the Nonlinear Reachability Problem (*NRP*). We will define *NRP* again, but we refer the reader to [AKW93] for more details and proofs of some of the lemmas we will use.

Let X be a set of variables ranging over $\mathbb{N}$. Consider a system C of inequalities of the form $p \leq q$ with $p, q \in \mathbb{N}[X]$ where each p is a sum of variables over X, and each $x \in X$ occurs in at most one left hand side p.

If x does occur in a left hand side of a constraint, x is said to be *constrained*, and we denote the constraint by $p_x \leq q_x$.

A *valuation* is a map $u : X \to \mathbb{N}$. The map u extends uniquely to a semiring morphism $\mathbb{N}[X] \to \mathbb{N}$.

For a system C and $x \in X$, x is *enabled* under u if either

- x does not occur on any left hand side of C; or
- $u(p_x) < u(q_x)$.

The operation of incrementing the value of x under a valuation u will be denoted by appending x to u, *i.e.* define ux by

$$ux(y) \quad = \quad \begin{cases} u(y) + 1 & \text{if } x = y, \\ u(y) & \text{otherwise.} \end{cases}$$

The definition of ux extends to $u\sigma$ for any $\sigma \in X^*$ by defining

$$\begin{aligned} u\varepsilon &= u \\ u(\sigma x) &= (u\sigma)x \end{aligned}$$

The valuation u is said to *satisfy* C if for all $p \leq q \in C$, $u(p) \leq u(q)$. Denote by V_C be the set of all valuations satisfying C.

For $u, v \in V_C$ and $x \in X$, we write $u \xrightarrow[C]{x} v$ if $v = ux$ and x is u-enabled. Define the graph $G_C = (V_C, E)$ with a directed edge (u, v) labelled x if $u \xrightarrow[C]{x} v$.

Let $X^*(C, u)$ denote the set of paths $\sigma \in X^*$ in the graph G_C starting at u. A path $\sigma \in X^*(C, u)$ gives rise to a new valuation $u\sigma : X \to \mathbb{N}$ with $u\sigma \in V_C$. A string $\sigma \in X^*$ is said to be valid (for (C, u)) if $\sigma \in X^*(C, u)$. We will use Greek letters for valuation strings $\sigma \in X^*(C, u)$, but Roman letters for valuations $u \in V_C$.

For $\alpha, \beta \in X^*$, we write $\alpha \equiv_X \beta$ if $\alpha(x) = \beta(x)$ for all $x \in X$. Note that $\alpha \equiv_X \beta$ iff α and β are permutations of each other.

Definition 1 The *Nonlinear Reachability Problem* (*NRP*) is the following problem:

> Given a system C of constraints in variables X, a valuation $s \in V_C$ and a special variable $x_0 \in X$, decide if there exists a $\sigma \in X^*(C, s)$ such that $s\sigma(x_0) > 0$.

□

3 The Exposure Graph

As in [AKW93] we observe that for $q \in \mathbb{N}[X]$ and $x \in X$ there is a unique way to write $q = \sum_{i=0}^{n} q_i x^i$ with $q_i \in \mathbb{N}[X - \{x\}]$, and we say that x is *u-exposed* in q if $u(q_i) > 0$ for some $1 \leq i \leq n$.

For an instance (X, C, u) of *NRP*, we define the *exposure graph* to be the directed graph $G(u) = (V, E)$, where $V = X$ and

$$E = \{(x, y) \in V \times V \ : \ \ x \text{ is } u\text{-exposed in } q_y\}.$$

It follows from the definition of exposure that $G(u)$ is contained in $G(u\sigma)$, *i.e.* the graph is monotonic under the operation of incrementing the valuation.

Let $\sigma \in X^*(C,u)$. We say that σ is *u-orderly* if for all $\alpha, \beta, \gamma \in X^*$ and $x, y \in X$, if there is a path from x to y in $G(u\sigma)$, then all occurrences of x in σ must occur before all occurrences of y in σ.

In this definition x and y may be the same, in which case x occurring in the u-orderly σ implies there is no loop through x in $G(u\sigma)$, self-loops included.

Lemma 2 *Let $\sigma \in X^*(C,u)$. If σ satisfies*

$$\forall x \in X \text{ if } x \text{ is on a cycle of } G(u\sigma) \text{ then } \sigma(x) = 0 \tag{1}$$

then there exists $\sigma' \in X^*(C,u)$ with $\sigma' \equiv_X \sigma$ such that σ' is orderly.

Proof. If $\sigma = \varepsilon$ then σ itself is orderly. Assume $|\sigma| \geq 1$, and write $\sigma = \alpha x \beta$, where for all $y \in \alpha\beta$, there is no path in $G(u\sigma)$ from y to x. Such a partition of σ exists since otherwise for every element x_i of σ there would be another element x_{i+1} of σ with a path from x_{i+1} to x_i in $G(u\sigma)$. Since there are finitely many variables x_i, we would eventually get a repeated occurrence of some variable of σ, contradicting the property (1).

Since $\sigma = \alpha x \beta$ and for no $y \in \sigma$ is (y, x) an edge of $G(u\sigma)$, we have that x is u-enabled. We claim that $x\alpha \in X^*(C,u)$.

If not, say $\alpha = \alpha_1 z \alpha_2$, where $x\alpha_1$ is valid but $x\alpha_1 z$ is not. Since $\alpha_1 z$ is valid, x and z must have the same defining constraint, $p \leq q$.

Since $x\alpha_1 \in X^*(C,u)$ and $\alpha_1 z \in X^*(C,u)$ we have $ux\alpha_1(q-p) = 0$ and $u\alpha_1(q-p) = 1$. We note that z cannot be $u\alpha_1$-exposed in q (then z would have a self-loop in $G(u\sigma)$, contradicting (1) above), so $u\alpha_1 z(q-p) = 0$. Since x is $u\alpha_1 z \alpha_2$-enabled but not $u\alpha_1 z$-enabled, there is some $y \in \alpha_2$ where (y, x) is an edge of $G(u\sigma)$, contradicting the choice of x. Hence, $x\alpha$ must be valid, and so is $x\alpha\beta$.

By induction, $\alpha\beta$ can be rearranged to be ux-orderly, say $\alpha\beta \equiv_X \gamma$. Then $x\gamma \equiv_X \sigma$, and $x\gamma$ is orderly. □

Lemma 3 *Let $\alpha y \in X^*(C,u)$ with $\alpha \in X^*$ and $y \in X$. If α is u-orderly and y is u-enabled, then $y\alpha \in X^*(C,u)$.*

Proof. We use induction, the case $\alpha = \varepsilon$ being trivial. Assume $\alpha = \gamma x$. We will show that $\gamma y x \in X^*(C,u)$. Then by induction $y\gamma \in X^*(C,u)$ and hence $y\gamma x \in X^*(C,u)$. If y is not constrained, the lemma holds trivially, so assume $p \leq q$ is the constraint of y.

In an orderly string a variable cannot change from enabled to nonenabled and back to enabled. Thus y is enabled throughout α.

If x is also constrained by $p \leq q$ we cannot have $u\gamma(q-p) = 1$, because that would imply (x,x) were an edge of $G(u\alpha)$, contradicting that σ is orderly. Hence $u\gamma(q-p) > 1$ and we clearly have $\gamma y x \in X^*(C,u)$.

If x is not constrained by $p \leq q$ then since $\gamma y \in X^*(C,u)$ we clearly have $\gamma y x \in X^*(C,u)$. □

Lemma 4 *Let $\alpha \in X^*(C,u), x \in X, n = |X|$ with $\alpha(x) > 0$. If α is orderly and $G(u) = G(u\alpha)$, then there exists $\alpha' = \alpha_1\alpha_2 \in X^*(C,u)$ with $\alpha' \equiv_X \alpha$, $\alpha_1(x) > 0$, $|\alpha_1| \leq n$ and α_2 orderly.*

Proof. Write $\alpha = \beta x \gamma$. Either x is u-enabled or there is some $x_1 \in \beta$ such that (x_1, x) is an edge of $G(u)$. Continuing this way we get a path $x_k, \ldots, x_1, x$ in $G(u)$, where x_k is u-enabled. Since α is orderly, $k < n$. Then we can write $\beta = \beta_1 x_k \beta_2$. By applying Lemma 3 on $\beta_1 x_k$ we see that $x_k\beta_1\beta_2$ is valid. Then $\beta_1\beta_2$ is orderly since $\beta = \beta_1 x_k \beta_2$ is. Now note that x_{k-1} is ux_k-enabled and $x_{k-1}, \ldots, x_1, x$ is a path in $G(ux_k) = G(u\alpha)$. By induction, we get that $\beta x \equiv_X x_k x_{k-1} \cdots x_1 x \overline{\beta}$ where $x_k x_{k-1} \cdots x_1 x \overline{\beta} \in X^*(C,u)$. □

For a valuation u, let **sign** u denote the valuation

$$\mathbf{sign}\ u(y) = \begin{cases} 1, & \text{if } u(y) > 0, \\ 0, & \text{if } u(y) = 0. \end{cases}$$

Lemma 5 *Let $x \in X$, $p \leq q \in C$ and $u, v \in V_C$. If* **sign** $u(y) \leq$ **sign** $v(y)$ *for all $y \in X - \{x\}$ and x is u-exposed in q, then x is v-exposed in q. In particular if* **sign** $u =$ **sign** v *then $G(u) = G(v)$.*

Proof. This is Lemma 6.5(i) of [AKW93]. □

Lemma 6 *Assume $\alpha x \in X^*(C,u)$ where α is orderly and $G(u) = G(u\alpha) \subsetneq G(u\alpha x)$. Then there exist $\beta\gamma \in X^*(C,u)$ such that*

1. *$\beta\gamma \equiv_X \alpha x$,*

2. *$|\beta| \leq n^2$,*

3. $G(u\alpha x) = G(u\beta)$.

Proof. By Lemma 4, any variable of α can be assumed to be fired within n steps of α, with the remainder of α orderly. Thus it can be assumed that if α fires $k \leq n$ variables, they are all fired within kn steps. So we can assume $\alpha = \alpha_1\alpha_2$ with $|\alpha_1| \leq kn$ and α_2 orderly.

If x is $u\alpha_1$-enabled, by Lemma 3 $\alpha_1 x \alpha_2 \in X^*(C, u)$. If x is not $u\alpha_1$-enabled, there is $y \in \alpha_2$ such that y is u-exposed in q_x. By Lemma 3 we can assume y is fired within n steps of α_1. Since **sign** $u\alpha_1 =$ **sign** $u\alpha$, y will still be exposed in q_x, by Lemma 5.

We have shown that $\alpha x \equiv_X \beta\gamma$ with $|\beta| \leq n^2$ and **sign** $\alpha x =$ **sign** β. Along with Lemma 5 this proves conditions 1-3 of this lemma. □

Lemma 7 *For a system (X, u, C) with $|X| = n$, let $\sigma \in X^*(C, u)$ with $\sigma(x_0) > 0$. Then there exists a $\sigma' \in X^*(C, u)$ with $\sigma' = \alpha\beta$, $\sigma' \equiv_X \sigma$ and $|\alpha| \leq n^4$, where α satisfies either*

1. *$G(u\alpha)$ has a self-loop based on an exposed variable; or*

2. *$G(u\alpha)$ has a cycle whose vertices are in at least two different constraints; or*

3. *$\alpha(x_0) > 0$.*

Proof. It is clear that every σ that is a solution for the system satisfies the third condition of the lemma. Let α be the smallest prefix of σ satisfying one (or more) of the conditions of the lemma. If $\alpha = \varepsilon$ we are done, otherwise write $\alpha = \alpha' y$. By Lemma 2 we can assume α' is orderly.

If $G(u\alpha)$ has a cycle, then every edge of the cycle can be added within n^2 steps, by Lemma 6. Hence we can assume α creates the cycle within n^3 steps.

If $G(u\alpha)$ contains a self-loop based on an exposed variable x, we can again assume α creates the loop within n^3 steps. By a proof similar to that of Lemma 6, it takes at most n more steps to enable the variable x. □

4 Reducing Solutions

We restate three important lemmas from [AKW93].

Lemma 8 *Let (X, u, C) be an instance of NRP, $p \leq q \in C$ and $C' = C - \{p \leq q\}$. If $x \in X$ is unconstrained and exposed in q, then (X, u, C) has a solution if and only if (X, u, C') does. Furthermore, any solution to (X, u, C) is also solution to (X, u, C').*

Proof. This is just a restatement of Lemma 6.14 in [AKW93]. □

Lemma 9 *Let (X, u, C) be an instance of NRP. Let $x \in X$ and assume $p \leq q \in C$ constrains x. Let*

$$C' = \begin{cases} (C - \{p \leq q\}) \cup \{p - x \leq q - x\}, & \text{if } q - x \in \mathbb{N}[X] \\ C - \{p \leq q\} & \text{otherwise.} \end{cases}$$

If x is u-enabled then (X, u, C) has a solution if and only if (X, u, C') does. Furthermore, any solution to (X, u, C) is also solution to (X, u, C').

Proof. This is just a restatement of Lemma 6.15 in [AKW93], except for the last statement, which is implicit in the proof given in [AKW93]. □

Lemma 10 *Let (X, u, C) be an instance of NRP. Assume $x_1, \ldots, x_k \in X$ have pairwise different constraints $p_i \leq q_i \in C$, $1 \leq i \leq k$. Define*

$$\begin{aligned} p' &= \sum_{i=1}^{k} p_i \\ q' &= \sum_{i=1}^{k} q_i, \\ C' &= (C - D) \cup \{p' \leq q'\} \end{aligned}$$

If $x_1 \cdots x_k$ is a loop in $G(u)$, then (X, u, C) has a solution if and only if (X, u, C') does. Furthermore, any solution to (X, u, C) is a solution to (X, u, C').

Proof. This is just a rewording of Lemma 6.16 in [AKW93]. □

Remark We note that if $G(u)$ contains a simple loop $x_1, \ldots, x_k$, where x_i and x_j have the same constraint, $0 \leq i < j \leq k$, then $x_i, \ldots, x_{j-1}$ is also a loop. So whenever $G(u)$ contains a loop, it contains either a self-loop or a cycle, where all the constraints of the vertices differ. □

We now have a simple nondeterministic algorithm to compute a solution to an instance of *NRP*, if such a solution exists. Consider the initial system (X, u, C). If $u(x_0) > 0$, we are done.

1. If for some $p \leq q \in C$ there is an unconstrained variable $x \in X$ exposed in q, we can remove $p \leq q$ from C by Lemma 8.

2. If $G(u)$ contains a self-loop based on $x \in X$ and x is u-exposed, either replace $p \leq q$ by $p - x \leq q - x$ or remove $p \leq q$ from C, as determined by Lemma 9.

3. If $G(u)$ contains a cycle through variables with different constraints, $p_1 \leq q_1, \ldots, p_t \leq q_t$, collapse these into one constraint, $p_1 + \cdots + p_t \leq q_1 + \cdots + q_t$, by Lemma 10.

By the lemmas mentioned, the reduced system has a solution iff the original system does.

Guess a string $\sigma \in X^*$, where σ is the smallest string invoking one of the events of Lemma 7. By that lemma, $|\sigma| \leq n^4$. We can recursively solve $(X, u\sigma, C)$.

Each time we invoke one of the events of Lemma 7, it leads either to a solution, or it removes a constraint from C or it makes a variable unconstrained. Each of the events can happen at most n times, so this is a nondeterministic $\mathcal{O}(n^5)$ algorithm to find a solution.

Theorem 11 *NRP is NP-complete.*

Proof. By the above argument, *NRP* $\in$ *NP*. It remains to show that *NRP* is *NP*-hard. We give a reduction from CNFSAT.

Let $\mathcal{B} = \bigwedge_{i=1}^{m} C_i$ be a Boolean formula in conjunctive normal form over the Boolean variables $x_1, \ldots, x_n$. Let $x_i^{\text{pos}}, x_i^{\text{neg}}$ and c_j be integer variables, $1 \leq i \leq n$, $1 \leq j \leq m$. For every clause $C_i = \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} \overline{x}_j$ we create an inequality

$$c_i \leq \sum_{i \in I} x_i^{\text{pos}} + \sum_{j \in J} x_j^{\text{neg}}.$$

Adding the inequalities

$$\begin{aligned} x_i^{\text{pos}} + x_i^{\text{neg}} &\leq 1 \\ b \leq \prod_{i=1}^{m} c_i, & \end{aligned}$$

we have an instance of *NRP* where the special variable b can be fired iff $\mathcal{B}$ is satisfiable. $\Box$

5 Application to Systems of Set Constraints

In [AKVW93] the complexity of solving systems $\mathcal{S}$ of set constraints is considered. Assuming only positive set constraints (of the form $E \subseteq F$), the problem is shown to be *NEXPTIME*-complete in general.

When negative set constraints are also allowed ($E \not\subseteq F$), the problem becomes significantly harder. In [AKW93] this problem is reduced to a hypergraph problem. An instance $\mathcal{S}$ of set constraints is reduced nondeterministically to a hypergraph $(U, E_f | f \in \Sigma)$, where $|U| = 2^{\mathcal{O}(|\mathcal{S}|)}$. The hypergraph is then reduced to a disjunction of *NRP* instances $\bigvee_{v \in V} N(v)$, where each problem instance has size $|N(v)| = |U|^{\mathcal{O}(1)}$ and $|V| = \mathcal{O}(|U|)$.

In particular, we have an overall nondeterministic reduction of a general system $\mathcal{S}$ of set constraints to an *NRP* instance of size $2^{n^{\mathcal{O}(1)}}$. Since *NRP* is in *NP*, we have a *NEXPTIME* algorithm to solve systems of set constraints, where negative constraints are allowed.

Since the simpler problem of solving set constraints with positive constraints only is *NEXPTIME*-hard we have shown:

Corollary 12 *A system of set constraints, including negative constraints, is NEXPTIME-complete.* $\Box$

6 Acknowledgments

Many thanks are due to Dexter Kozen for valuable discussions and suggestions. This work was supported in part by the National Science Foundation and in part by the United States Army Research Office through the Army Center of Excellence for Symbolic Methods in Algorithmic Mathematics (ACSyAM), Mathematical Sciences Institute of Cornell University under contract DAAL03-91-C-0027.

References

[AKW93] A. Aiken, D. Kozen, and E. Wimmers. Decidability of systems of set constraints with negative constraints. Technical Report 93-1362, Computer Science Department, Cornell University, June 1993

[AKVW93] A. Aiken, D. Kozen, M. Vardi and E. Wimmers. The complexity of set constraints. Technical Report 93-1352, Computer Science Department, Cornell University, May 1993

[GTT93] R. Gilleron, S. Tison, and M. Tommasi. Solving systems of set constraints with negated subset relationships. Technical Report LIFL IT 247, Lille 1 University, March 1993