Decidability of Systems of Set Constraints With Negative Constraints

Alexander Aiken*
Dexter Kozen**
Ed Wimmers*

TR 93-1362
June 1993

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120.
**Computer Science Department, Cornell University, Ithaca, NY 14853.

Decidability of Systems of Set Constraints with Negative Constraints

Alexander Aiken*
aiken@almaden.ibm.com

Dexter Kozen†
kozen@cs.cornell.edu

Ed Wimmers*
wimmers@almaden.ibm.com

June 19, 1993

Abstract

Set constraints are relations between sets of terms. They have been used extensively in various applications in program analysis and type inference. Recently, several algorithms for solving general systems of positive set constraints have appeared. In this paper we consider systems of mixed positive and negative constraints, which are considerably more expressive than positive constraints alone. We show that it is decidable whether a given such system has a solution. The proof involves a reduction to a number-theoretic decision problem that may be of independent interest.

1 Introduction

Set constraints are formal inclusions or negated inclusions between expressions representing subsets of T_Σ, the set of ground terms over a finite ranked alphabet Σ. Formally, a *positive set constraint* is of the form $E \subseteq F$ and a

*IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120
†Computer Science Department, Cornell University, Ithaca, NY 14853

negative set constraint is of the form $E \not\subseteq F$, where E and F are expressions built from a set $X = \{x, y, \ldots\}$ of variables ranging over subsets of T_Σ, the usual set-theoretic operators 0, 1, $\cup$, $\cap$, and $\sim$, and an n-ary set operator f for each n-ary symbol $f \in \Sigma$ with semantics

$$f(A_1, \ldots, A_n) \quad = \quad \{ft_1 \ldots t_n \mid t_i \in A_i,\ 1 \le i \le n\}\ .$$

A system $\mathcal{S}$ of constraints is *satisfiable* if there is an assignment of subsets of T_Σ to the variables satisfying all the constraints in $\mathcal{S}$.

Set constraints have numerous applications in program analysis and type inference [16, 12, 14, 15, 17, 11, 2, 3, 6]. Most of these systems deal with positive constraints only. Several algorithms for determining the satisfiability of general systems of positive constraints have appeared [4, 10, 8, 5, 1]. In [1], the satisfiability problem for a system $\mathcal{S}$ of positive constraints is shown to be equivalent to deciding whether a certain finite hypergraph constructed from $\mathcal{S}$ has an induced subhypergraph that is *closed* (see Section 4). This characterization is used to obtain an exhaustive hierarchy of complexity results depending on the number of elements of Σ of each arity.

In this paper we consider systems with both positive and negative constraints. Negative constraints considerably increase the power of the constraint language and have important applications in program analysis. For example, in [2, 3], opportunities for program optimization are identified by an *ad hoc* technique for checking the satisfiability of systems of negative constraints. Set constraints with only nullary symbols correspond to Boolean algebras over a finite set of atoms; in [13] general results on solving negative constraints in arbitrary Boolean algebras are given.

In this paper we give a general decision procedure for determining whether a given system of positive and negative constraints over an arbitrary signature is satisfiable. The proof reduces the satisfiability problem to a reachability problem involving diophantine inequalities which may be of independent interest. We reduce the satisfiability problem to the diophantine problem and then show that the diophantine problem is decidable. The proof has a nonconstructive step involving Dickson's Lemma and does not give any complexity bounds.

The decidability result for systems of positive and negative set constraints has recently been obtained independently by Gilleron, Tison, and Tommasi [9] using automata-theoretic techniques.

2 Set Expressions and Set Constraints

Let Σ be a finite ranked alphabet consisting of symbols f, each with an associated arity $\mathbf{arity}\, f \in \mathbb{N}$. Symbols in Σ of arity 0, 1, 2, 3, 4, and n are called *nullary, unary, binary, ternary, quaternary*, and *n-ary*, respectively. Nullary elements of Σ are often called *constants*. The set of elements of Σ of arity n is denoted Σ_n.

The set of ground terms over Σ is denoted T_Σ. This is the smallest set such that if $t_1, \ldots, t_n \in T_\Sigma$ and $f \in \Sigma_n$, then $ft_1 \ldots t_n \in T_\Sigma$. If $X = \{x, y, \ldots\}$ is a set of variables, then $T_\Sigma(X)$ denotes the set of terms over Σ and X, considering the elements of X as symbols of arity 0.

Let $\mathrm{B} = (\cup,\ \cap,\ \sim,\ 0,\ 1)$ be the usual signature of Boolean algebra. Let $\Sigma + \mathrm{B}$ denote the signature consisting of the disjoint union of Σ and B. A *set expression* over X is any element of $T_{\Sigma+\mathrm{B}}(X)$. The following is a typical set expression:

$$f(g(x \cup y), {\sim} g(x \cap y)) \quad \cup \quad a$$

where $f \in \Sigma_2$, $g \in \Sigma_1$, $a \in \Sigma_0$, and $x, y \in X$. Set expressions are denoted $E, F, \ldots$. A *Boolean expression* over X is any element of $T_\mathrm{B}(X)$.

A *positive set constraint* is a formal inclusion $E \subseteq F$, where E and F are set expressions. We also allow equational constraints $E = F$, although inclusions and equations are interdefinable. A *negative set constraint* is the negation of a positive set constraint: $E \not\subseteq F$.

We interpret set expressions over the powerset 2^{T_Σ} of T_Σ. This forms an algebra of signature $\Sigma + \mathrm{B}$ where the Boolean operators have their usual set-theoretic interpretations and elements $f \in \Sigma_n$ are interpreted as functions

$$f : (2^{T_\Sigma})^n \quad \rightarrow \quad 2^{T_\Sigma}$$

$$f(A_1, \ldots, A_n) \quad = \quad \{ft_1 \ldots t_n \mid t_i \in A_i,\ 1 \leq i \leq n\}\ .$$

A *set assignment* is a map

$$\sigma : X \quad \rightarrow \quad 2^{T_\Sigma}$$

assigning a subset of T_Σ to each variable in X. Any set assignment σ extends uniquely to a $(\Sigma + \mathrm{B})$-homomorphism

$$\sigma : T_{\Sigma+\mathrm{B}}(X) \quad \rightarrow \quad 2^{T_\Sigma}$$

by induction on the structure of the set expression in the usual way. The set assignment σ satisfies the positive constraint $E \subseteq F$ if $\sigma(E) \subseteq \sigma(F)$, and satisfies the negative constraint $E \not\subseteq F$ if $\sigma(E) \not\subseteq \sigma(F)$. We write $\sigma \models \varphi$ if the set assignment σ satisfies the constraint φ. A family $\mathcal{S}$ of set constraints is *satisfiable* if there is a set assignment that satisfies all constraints in $\mathcal{S}$ simultaneously. If φ is a constraint, we write $\mathcal{S} \models \varphi$ if all set assignments that satisfy Σ also satisfy φ. The *satisfiability problem* is to determine whether a given finite system $\mathcal{S}$ of set constraints over Σ is satisfiable.

A *truth assignment* is a map

$$u : X \;\rightarrow\; \mathbf{2}$$

where $\mathbf{2} = \{0, 1\}$ is the two-element Boolean algebra. Any truth assignment u extends uniquely to a B-homomorphism

$$u : T_{\mathrm{B}}(X) \;\rightarrow\; \mathbf{2}$$

inductively according to the rules of Boolean algebra. If $X = \{x_1, \ldots, x_m\}$, we use the notation

$$B[x_i := a_i]$$

to denote the truth value of the Boolean formula B under the truth assignment $x_i \mapsto a_i$, $1 \le i \le m$.

3 Expressibility

Positive and negative constraints together are strictly more expressive than positive constraints alone. We will prove this as a corollary of a general compactness theorem for positive constraints.

Theorem 3.1 (Compactness) *A system $\mathcal{S}$ of positive set constraints is satisfiable if and only if all finite subsets of $\mathcal{S}$ are satisfiable.*

Proof. The implication $(\Rightarrow)$ is straightforward. Now suppose $\mathcal{S}$ is finitely satisfiable. We wish to construct a satisfying set assignment for $\mathcal{S}$. By Zorn's Lemma, there exists a maximal finitely satisfiable set $\widehat{\mathcal{S}}$ of positive constraints containing $\mathcal{S}$. One can show that for all ground terms t and set expressions E, exactly one of the constraints $t \subseteq E$, $t \subseteq {\sim}E$ is in $\widehat{\mathcal{S}}$; if neither are in

$\widehat{\mathcal{S}}$, then $\widehat{\mathcal{S}}$ is not maximal, and if both are, then $\widehat{\mathcal{S}}$ is not finitely satisfiable. Now define a map

$$\sigma(E) \quad = \quad \{t \mid t \subseteq E \in \widehat{\mathcal{S}}\} \; .$$

One can show by induction on the structure of set expressions that σ is a valid set assignment and satisfies $\widehat{\mathcal{S}}$. For example, to show that

$$\sigma(fE_1 \ldots E_n) \quad = \quad \{ft_1 \ldots t_n \mid t_i \in \sigma(E_i),\ 1 \leq i \leq n\} \; ,$$

note

$$t \in \sigma(fE_1 \ldots E_n) \iff t \subseteq fE_1 \ldots E_n \in \widehat{\mathcal{S}} \; .$$

Then t must be of the form $ft_1 \ldots t_n$, otherwise $\widehat{\mathcal{S}}$ would not be finitely satisfiable. Now we use the fact that

$$\begin{array}{rcl} ft_1 \ldots t_n \subseteq fE_1 \ldots E_n & \models & t_i \subseteq E_i \; , \quad 1 \leq i \leq n \\ \{t_i \subseteq E_i \mid 1 \leq i \leq n\} & \models & ft_1 \ldots t_n \subseteq fE_1 \ldots E_n \end{array}$$

to argue that $t \subseteq fE_1 \ldots E_n \in \widehat{\mathcal{S}}$ iff $t_i \subseteq E_i \in \widehat{\mathcal{S}}$, $1 \leq i \leq n$, otherwise $\widehat{\mathcal{S}}$ would not be finitely satisfiable. Thus

$$\begin{array}{rcl} t \in \sigma(fE_1 \ldots E_n) & \iff & t \subseteq fE_1 \ldots E_n \in \widehat{\mathcal{S}} \\ & \iff & t_i \subseteq E_i \in \widehat{\mathcal{S}} \; , \quad 1 \leq i \leq n \\ & \iff & t_i \in \sigma(E_i) \; , \quad 1 \leq i \leq n \; . \end{array}$$

To show that σ satisfies all constraints of $\widehat{\mathcal{S}}$, let $E \subseteq F$ be any constraint in $\widehat{\mathcal{S}}$. For any term t,

$$\begin{array}{rcl} t \in \sigma(E) & \Rightarrow & t \subseteq E \in \widehat{\mathcal{S}} \\ & \Rightarrow & t \subseteq F \in \widehat{\mathcal{S}} \\ & \Rightarrow & t \in \sigma(F) \; ; \end{array} \tag{1}$$

the reason for the implication (1) is that

$$\{t \subseteq E,\ E \subseteq F\} \models t \subseteq F \; ,$$

and if $t \subseteq F$ were not in $\widehat{\mathcal{S}}$, then $t \subseteq \sim F$ would be, and $\widehat{\mathcal{S}}$ would not be finitely satisfiable. □

Corollary 3.2 *Positive and negative constraints are strictly more expressive than positive constraints only.*

Proof. Consider the negative constraint $x \neq 0$ over any ranked alphabet Σ with at least one constant and at least one symbol of higher arity. Solutions are $\sigma : \{x\} \to T_\Sigma$ with $\sigma(x)$ nonempty. Let $\mathcal{S}$ be any set of positive constraints over any set of variables X containing x. We claim that it is *not* the case that the set

$$\{\sigma(x) \mid \sigma : X \to T_\Sigma,\ \sigma \models \mathcal{S}\}$$

is exactly the set of nonempty subsets of T_Σ.

Consider the infinite set of positive constraints

$$\mathcal{S} \ \cup \ \{t \subseteq {\sim}x \mid t \in T_\Sigma\} \ .$$

Either this is satisfiable or not. If so, then there is a satisfying assignment σ. But $t \in \sigma({\sim}x)$ for all terms t, so $\sigma(x) = \emptyset$ and the claim is verified. If not, then by compactness there is a finite subset $F \subseteq T_\Sigma$ such that

$$\mathcal{S} \ \cup \ \{t \subseteq {\sim}x \mid t \in F\}$$

is not satisfiable. Therefore there is no solution σ of $\mathcal{S}$ with $\sigma(x) = \{t\}$, where t is any term not in F. □

4 Set Constraints and Hypergraph Closure

In [1] it is shown how to transform a given system of positive set constraints into an equivalent system in a special normal form. The transformation does not significantly increase the size of the system. Applying this transformation to a system containing negative constraints, we obtain the following normal form. Let X be a set of variables, and for each $f \in \Sigma$, let

$$Z_f \quad = \quad \{z^f_{ix} \mid 0 \leq i \leq \textbf{arity}\, f,\ x \in X\}$$

be a set of variables such that the sets X and Z_f, $f \in \Sigma$ are pairwise disjoint. A system of set constraints in normal form (with respect to X and the Z_f) consists of

- a positive constraint $B = 1$, where $B \in T_{\mathrm{B}}(X)$
- for each $f \in \Sigma$, a positive constraint $C_f = 1$, where $C_f \in T_{\mathrm{B}}(Z_f)$
- positive constraints

$$\begin{aligned} z^f_{0x} &= f\underbrace{1\ldots1}_{n} \cap x \\ z^f_{ix} &= f\underbrace{1\ldots1}_{i-1}x\underbrace{1\ldots1}_{n-i} \end{aligned}$$

  for each $f \in \Sigma_n$ and each $1 \le i \le n$ and $x \in X$

- a finite set of negative constraints $D \neq 0$, where $D \in T_{\mathrm{B}}(X)$, $D \in \mathcal{D}$.

The last component is absent with positive constraints only.

As described in [1], a system of set constraints $\mathcal{S}$ in normal form determines a hypergraph

$$H = (U,\ E_f \mid f \in \Sigma)$$

as follows. The vertex set U is the set of all truth assignments $u : X \to \mathbf{2}$ satisfying B. Each such truth assignment corresponds to a conjunction of literals (also denoted u) in which each variable in X occurs exactly once, either positively or negatively, such that $u \subseteq B$ tautologically. The variable x occurs positively iff $u(x) = 1$. We often call the elements of U *atoms* because they represent atoms (minimal nonzero elements) of the free Boolean algebra on generators X modulo $B = 1$. Each Boolean expression over X is equivalent modulo $B = 1$ to a disjunction of atoms.

For each $f \in \Sigma_n$, the hyperedge relation E_f of H is defined to be the set of all $(n+1)$-tuples $(u_0, \ldots, u_n) \in U^n$ such that

$$C_f[z^f_{ix} := u_i(x)] = 1 . \tag{2}$$

Intuitively, we think of the formula C_f as a Boolean-valued mapping on $(n+1)$-tuples of truth assignments to X. To emphasize this intuition, we abbreviate the left hand side of (2) by

$$C_f[u_0, \ldots, u_n] .$$

Thus

$$(u_0, \ldots, u_n) \in E_f \quad \text{iff} \quad C_f[u_0, \ldots, u_n] = 1 \ .$$

In general, the size of H can be exponential in the size of $\mathcal{S}$.

An $(n+1)$-ary hyperedge relation E_f of the hypergraph H is said to be *closed* if for each n-tuple $u_1, \ldots, u_n \in U^n$, there exists a $u_0 \in U$ such that $(u_0, u_1, \ldots, u_n) \in E_f$. In the case $n = 0$, this definition just says $E_f \cap U \neq \emptyset$. Abusing notation, we can think of E_f as a function

$$E_f : U^n \quad \rightarrow \quad 2^U$$

where

$$E_f(u_1, \ldots, u_n) \quad = \quad \{u_0 \mid (u_0, u_1, \ldots, u_n) \in E_f\} \ .$$

In this view, E_f is *closed* iff $E_f(u_1, \ldots, u_n) \neq \emptyset$ for each n-tuple $u_1, \ldots, u_n \in U^n$. The hypergraph H is said to be *closed* if all its hyperedge relations are closed.

The *induced subhypergraph* of H on vertices $U' \subseteq U$ is the hypergraph

$$H' \quad = \quad (U',\ E'_f \mid f \in \Sigma)$$

such that $E'_f = E_f \cap (U')^{n+1}$ for $f \in \Sigma_n$.

The *hypergraph closure problem* is the problem of determining whether a given hypergraph H has a closed induced subhypergraph.

The following theorem was proved in [1].

Theorem 4.1 *The hypergraph H corresponding to a system $\mathcal{S}$ of positive set constraints has a closed induced subhypergraph if and only if $\mathcal{S}$ is satisfiable.*

In brief, the proof of [1] establishes a one-to-one correspondence between set assignments σ satisfying $\mathcal{S}$ and maps

$$\theta : T_\Sigma \quad \rightarrow \quad U$$

such that for all $f \in \Sigma$ and for all terms $ft_1 \ldots t_n$,

$$\theta(ft_1 \ldots t_n) \quad \in \quad E_f(\theta(t_1), \ldots, \theta(t_n)) \ . \tag{3}$$

The set assignment corresponding to θ is

$$\begin{aligned}
\sigma(x) &= \{t \mid \theta(t)(x) = 1\} \\
\sigma(z^f_{ix}) &= \sigma(f\underbrace{1\ldots1}_{i-1}x\underbrace{1\ldots1}_{n-i}) \\
\sigma(z^f_{0x}) &= \sigma(f\underbrace{1\ldots1}_{n} \cap x) \,.
\end{aligned}$$

Thus deciding the satisfiability of $\mathcal{S}$ is tantamount to determining the existence of a map θ satisfying (3). In turn, this is equivalent to the hypergraph closure problem: if such a θ exists, then the induced subhypergraph of H on the image of θ is closed, and conversely, if there exists a closed induced subhypergraph on vertices $U' \subseteq U$, then one can inductively define $\theta(ft_1 \ldots t_n)$ to be the lexicographically first element of $U' \cap E_f(\theta(t_1), \ldots, \theta(t_n))$.

In the presence of negative constraints $D \neq 0$, $D \in \mathcal{D}$, the map θ must not only satisfy (3), but must also take on some value u such that $u(D) = 1$ for each $D \in \mathcal{D}$. Thus in the presence of negative constraints, the satisfiability problem becomes:

> Given a finite set $\mathcal{D}$ of Boolean formulas $D \in T_{\mathrm{B}}(X)$ and a hypergraph
>
> $$H = (U,\ E_f \mid f \in \Sigma)$$
>
> specified by $B \in T_{\mathrm{B}}(X)$ and $C_f \in T_{\mathrm{B}}(Z_f)$, $f \in \Sigma$, determine whether there exists a map
>
> $$\theta : T_\Sigma \ \rightarrow\ U$$
>
> satisfying (3) such that for each $D \in \mathcal{D}$ there exists an atom u in $\theta(T_\Sigma)$ satisfying D, where $\theta(T_\Sigma)$ denotes the image of T_Σ under the map θ.

5 A Reachability Problem

Our decision procedure first reduces the satisfiability problem for mixed systems of set constraints to the following reachability problem involving Diophantine inequalities.

Let X be a set of variables ranging over $\mathbb{N}$, the natural numbers. Suppose we are given a finite system C of formal inequalities $p \leq q$, where p and q are polynomials in the variables X with coefficients in $\mathbb{N}$ such that

- each left hand side p is a sum of variables in X
- each variable occurs in at most one left hand side.

A *valuation* is a map $u : X \to \mathbb{N}$. Each valuation u extends uniquely to a semiring morphism $u : \mathbb{N}[X] \to \mathbb{N}$. A variable x is said to be *enabled* under a valuation u if either

- the variable u does not occur on any left hand side of an inequality in C; or
- the unique inequality in C in which x appears on the left hand side is a strict inequality under the valuation u.

Consider the following nondeterministic procedure. Starting with the zero valuation, repeatedly choose a variable that is enabled and "fire" it by adding 1 to it. The problem is to decide whether there exists a sequence of firings that allows a particular distinguished variable to be fired.

We will give a more rigorous presentation of this problem below, then reduce the satisfiability problem to this problem, then finally show that this problem is decidable.

5.1 Polynomials and Valuations

We use the term *semiring* to mean commutative semiring with unit.

Let X be a finite set of variables and let $\mathbb{N}[X]$ denote the semiring of polynomials in the indeterminates X with coefficients in $\mathbb{N}$. This is the free semiring on generators X.

Any map $u : X \to R$ to a semiring R extends uniquely to a semiring morphism $u : \mathbb{N}[X] \to R$. Such a map is called a *valuation* if $R = \mathbb{N}$. *Composition* of polynomials is effected by taking $R = \mathbb{N}[X]$.

Let v be any valuation, and let $\mathbf{inc}_v : \mathbb{N}[X] \to \mathbb{N}[X]$ be the unique semiring homomorphism such that

$$\mathbf{inc}_v(x) \quad = \quad x + v(x) \ , \quad x \in X \ .$$

For $q \in \mathbb{N}[X]$, the constant coefficient of $\mathbf{inc}_v(q)$ is

$$\mathbf{0}(\mathbf{inc}_v(q)) = v(q)$$

where $\mathbf{0}$ is the zero valuation. Let $\mathbf{inc}'_v$ be the map on valuations defined by

$$\mathbf{inc}'_v(u) = u \circ \mathbf{inc}_v \ . \tag{4}$$

Then for $x \in X$ and $q \in \mathbb{N}[X]$,

$$\begin{aligned} \mathbf{inc}'_v(u)(x) &= u(x) + v(x) \\ \mathbf{inc}'_v(\mathbf{0}) &= \mathbf{0} \circ \mathbf{inc}_v \\ &= v \ . \end{aligned}$$

One example of particular importance will be incrementing the value of a variable x under a valuation u by 1. This is obtained by taking $v = \delta_x$ in the above construction, where $\delta_x(x) = 1$ and $\delta_x(y) = 0$ for $y \neq x$. This case gets a special concise notation:

$$\begin{aligned} ux &= \mathbf{inc}'_{\delta_x}(u) \\ &= u \circ \mathbf{inc}_{\delta_x} \\ q[x/x+1] &= \mathbf{inc}_{\delta_x}(q) \ . \end{aligned} \tag{5}$$

In this case (4) takes the form

$$ux(q) = u(q[x/x+1]) \ . \tag{6}$$

We extend the definition of ux to $u\sigma$ for $\sigma \in X^*$ by taking

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

5.2 Systems of Diophantine Inequalities

We consider finite systems C of Diophantine inequalities of the form $p \leq q$ where $p, q \in \mathbb{N}[X]$ such that

- each left hand side p is a sum of distinct variables; and

- each variable in X occurs in at most one left hand side.

There is no restriction on the form of the right hand sides q except that they be in $\mathbb{N}[X]$. The inequalities in C are called *(Diophantine) constraints.* A variable $x \in X$ is said to be *constrained* in C if x occurs in some left hand side of a constraint in C. In this case we denote the unique such constraint by $\mathbf{con}\,(x, C) = p_x \leq q_x$. If x does not occur in any left hand side of an inequality in C, then x is said to be *unconstrained* in C, and we write $\mathbf{con}\,(x, C) = *$.

We say that the valuation u *satisfies* the inequality $p \leq q$ if

$$u(p) \;\leq\; u(q)\;.$$

We say that u *satisfies* C if u satisfies all the inequalities in C. The set of all valuations satisfying C is denoted V_C.

5.3 The Nonlinear Reachability Problem

Definition 5.1 Let $u \in V_C$. The inequality $p \leq q \in C$ is said to be *u-enabled* if $u(p) < u(q)$; *i.e.*, the inequality is strict under the valuation u. The variable $x \in X$ is said to be (u, C)*-enabled* if either

- x is unconstrained in C, or
- x is constrained in C and $\mathbf{con}\,(x, C)$ is u-enabled.

We write

$$u \;\xrightarrow[C]{x}\; v$$

if x is (u, C)-enabled and $v = ux$. For $\sigma \in X^*$, we write

$$u \;\xrightarrow[C]{\sigma}\; v$$

if $\sigma = x_1 x_2 \ldots x_n$, $n \geq 0$, and there exist valuations $u_0, \ldots, u_n$, such that $u = u_0$, $v = u_n$, and

$$u_{i-1} \;\xrightarrow[C]{x_i}\; u_i\;, \quad 1 \leq i \leq n\;.$$

$\square$

Note that if $u \xrightarrow[C]{\sigma} v$ then $v = u\sigma$. Note also that if $u \in V_C$ and $u \xrightarrow[C]{x} v$, then $v \in V_C$. In other words, if u satisfies C and x is (u, C)-enabled, then we can increment the value of x by 1 and the resulting valuation still satisfies C. Informally we refer to this action as *firing* x. The converse is false in general; *i.e.*, it is possible that u and ux both satisfy C but x is not (u, C)-enabled: consider the constraint $x \leq x$.

Any set C of constraints gives rise to a (possibly infinite) labeled directed acyclic graph G_C with vertices V_C and labeled directed edges $\xrightarrow[C]{x}$, $x \in X$.

Definition 5.2 The *Nonlinear Reachability Problem* (NRP) is to determine, given C and $s \in V_C$, whether there is a vertex t reachable from s in the graph G_C such that $t(x_0) > 0$, where x_0 is some distinguished variable of X. In other words, determine whether there exists a $\sigma \in X^*$ such that $s \xrightarrow[C]{\sigma} s\sigma$ and $s\sigma(x_0) > 0$. Such a σ is called a *solution* of the instance (s, C) of the NRP. □

Note that if $s(x_0) > 0$ already, then the null string ϵ is a solution of (s, C); otherwise σ is a solution iff x_0 occurs in σ.

If $s \in V_C$, we define the graph $G(s, C)$ to be the induced subgraph of G_C on all vertices reachable from s. In other words, the vertices of $G(s, C)$ are

$$\{t \mid s \xrightarrow[C]{\sigma} t \text{ for some } \sigma \in X^*\} .$$

The graph $G(s, C)$ is rooted at s. A *path* in $G(s, C)$, otherwise unspecified, will always mean a path starting at the root. We also abuse notation and identify paths with their labels; thus we may use the term *path* to refer to an element $\sigma \in X^*$ such that $s \xrightarrow[C]{\sigma} s\sigma$.

5.4 Order

Definition 5.3 For C a system of contraints and valuations u, v, define

- $u \leq_X v$ if $u(x) \leq v(x)$ for all $x \in X$
- $u \leq_C v$ if $u(q - p) \leq v(q - p)$ for all $p \leq q \in C$
- $u \leq_{X,C} v$ if both $u \leq_X v$ and $u \leq_C v$.

We write $u \equiv_C v$ if both $u \leq_C v$ and $v \leq_C u$. □

Note that if $u \xrightarrow[C]{\sigma} v$ then $u \leq_X v$. The same statement is not true in general for $\leq_C$.

Lemma 5.4 *Let $x \in X$, u, v valuations such that $u \leq_X v$, and $p \leq q \in C$. Then*

$$ux(q-p) - u(q-p) \quad \leq \quad vx(q-p) - v(q-p) \ .$$

Proof. Since p is a sum of distinct variables,

$$vx(p) - v(p) \quad = \quad ux(p) - u(p) \quad = \quad \begin{cases} 1\ , & \text{if } x \text{ occurs in } p \\ 0\ , & \text{otherwise.} \end{cases}$$

Then

$$\begin{aligned}
& vx(q-p) - v(q-p) - ux(q-p) + u(q-p) \\
= \quad & vx(q) - vx(p) - v(q) + v(p) - ux(q) + ux(p) + u(q) - u(p) \\
= \quad & vx(q) - v(q) - ux(q) + u(q) \\
= \quad & v(q[x/x+1]) - v(q) - u(q[x/x+1]) + u(q) \quad \text{by (6)} \\
= \quad & v(q[x/x+1] - q) - u(q[x/x+1] - q) \\
\geq \quad & 0\ ,
\end{aligned}$$

since $u \leq_X v$ and $q[x/x+1] - q \in \mathbb{N}[X]$. □

Lemma 5.5 *Let $u, v \in V_C$ and $x \in X$.*

(i) If x is (u, C)-enabled and $u \leq_C v$, then x is (v, C)-enabled.

(ii) If $u \leq_X v$ then $ux \leq_X vx$.

(iii) If $u \leq_{X,C} v$, then $ux \leq_{X,C} vx$.

Proof. The assertions (i) and (ii) are straightforward consequences of the definitions. The assertion (iii) follows from (ii) and Lemma 5.4. □

5.5 Reduction of Set Constraint Satisfiability to Nonlinear Reachability

Theorem 5.6 *The satisfiability problem for systems of mixed positive and negative set constraints reduces effectively to a finite disjunction of instances of the Nonlinear Reachability Problem.*

Proof. As argued in Section 4, the satisfiability problem for systems of mixed positive and negative constraints is equivalent to the following problem:

> Given a finite set $\mathcal{D}$ of Boolean formulas $D \in T_{\mathrm{B}}(X)$ and a hypergraph
>
> $$H = (U,\ E_f \mid f \in \Sigma)$$
>
> specified by Boolean formulas $B \in T_{\mathrm{B}}(X)$ and $C_f \in T_{\mathrm{B}}(Z_f)$, $f \in \Sigma$, determine whether there exists a map
>
> $$\theta : T_\Sigma \rightarrow U$$
>
> such that
>
> (i) for all $f \in \Sigma_n$ and for all terms $ft_1 \ldots t_n$,
>
> $$\theta(ft_1 \ldots t_n) \in E_f(\theta(t_1), \ldots, \theta(t_n)) \qquad (7)$$
>
> (ii) for each $D \in \mathcal{D}$ there exists an atom u in $\theta(T_\Sigma)$ with $u(D) = 1$.

Let $\mathcal{V}$ be the set of all subsets $V \subseteq U$ such that for all $D \in \mathcal{D}$ there exists a $v \in V$ with $v(D) = 1$. The problem above is equivalent to the disjunction over all $V \in \mathcal{V}$ of instances of the same problem with property (ii) replaced by the property

(ii′) $V \subseteq \theta(T_\Sigma)$.

Furthermore, we will only need to construct a finite partial approximation θ' to θ satisfying (i) and (ii′), provided

- the domain of θ' is closed downward under the subterm relation

- there is a closed induced subhypergraph of H containing the image of θ'.

The second property will allow θ' to be completed to a total function θ.

Thus the problem now becomes:

> Given a hypergraph
>
> $$H = (U,\ E_f \mid f \in \Sigma)$$
>
> specified by B and C_f, $f \in \Sigma$, and a subset $V \subseteq U$, determine whether there exist $U' \subseteq U$ and a partial map $\theta : T_\Sigma \to U'$ with finite domain such that
>
> - the induced subhypergraph on U' is closed
> - **domain** θ is closed downward under the subterm relation
> - θ satisfies (7) on all terms in its domain
> - $V \subseteq \theta(T_\Sigma) \subseteq U'$.

Consider the following nondeterministic procedure for constructing θ. We first guess the subset U' containing the target set V and check that it is closed. We start with θ totally undefined. At any point, say we have a partial θ with finite domain closed downward under the subterm relation. We nondeterministically pick some term $ft_1 \ldots t_n$ such that the $\theta(t_i)$ are defined but $\theta(ft_1 \ldots t_n)$ is not yet defined, nondeterministically choose some u in $E_f(\theta(t_1), \ldots, \theta(t_n)) \cap U'$, and assign $\theta(ft_1 \ldots t_n) := u$. We are always able to continue, since U' is closed. We halt successfully when and if all elements of V have been chosen as $\theta(t)$ for some t.

During this process, we use an integer variable

$$x_{u,f,u_1,\ldots,u_n}$$

where $n = \textbf{arity}\, f$, to count the number of terms of the form $ft_1 \ldots t_n$ such that

- $\theta(t_i)$ exists and equals u_i, $1 \leq i \leq n$, and
- $\theta(ft_1 \ldots t_n)$ exists and equals u.

There is one such variable for each choice of f in Σ, $u_1, \ldots, u_n \in U'$ where $n = \textbf{arity}\, f$, and $u \in U' \cap E_f(u_1, \ldots, u_n)$.

Now for each $f \in \Sigma_n$ and $u_1, \ldots, u_n \in U'$, consider the formal inequality

$$\sum_{u \in U' \cap E_f(u_1,\ldots,u_n)} x_{u,f,u_1,\ldots,u_n} \;\leq\; \prod_{i=1}^{n} \sum_{m=0}^{M} \sum_{\substack{v_1,\ldots,v_m \in U' \\ g \in \Sigma_m}} x_{u_i,g,v_1,\ldots,v_m} \tag{8}$$

where M is the maximum arity of symbols in Σ. This inequality has the following significance. Given a partial map θ, let

$$A_{f,u_1,\ldots,u_n} \;=\; \{ft_1 \ldots t_n \mid \theta(t_i) \text{ exists and equals } u_i,\, 1 \leq i \leq n\}\ .$$

The value of the right hand side of (8) is the size of $A_{f,u_1,\ldots,u_n}$, and the value of the left hand side of (8) is the size of the subset of $A_{f,u_1,\ldots,u_n}$ consisting of all elements t for which $\theta(t)$ is defined. The inequality expresses the fact that θ is defined on the subterms of t before being defined on t.

Consider the collection C of all such inequalities (8). To say that a variable $x_{u,f,u_1,\ldots,u_n}$ is enabled says that there exists a term t with head symbol f such that θ is defined on the n immediate subterms and takes values $u_1, \ldots, u_n$ on those subterms respectively, but $\theta(t)$ is not yet defined. To fire $x_{u,f,u_1,\ldots,u_n}$ says that we choose one such t and define $\theta(t) := u$.

The process of defining θ from the bottom up as described above corresponds to a sequence of legal firings. Conversely, any legal sequence of firings gives a corresponding sequence of definitions of θ starting with the totally undefined map.

We have thus reduced the satisfiability problem for systems of mixed positive and negative set constraints to a disjunction of instances of the problem of determining, given C and V, whether there is a finite sequence of legal firings after which for all $v \in V$ there are f and $u_1, \ldots, u_n$ such that the value of $x_{v,f,u_1,\ldots,u_n}$ is nonzero.

We reduce this problem to a finite disjunction of instances of the NRP as follows. For each $v \in V$, choose f and $u_1, \ldots, u_n$ and let $y_v = x_{v,f,u_1,\ldots,u_n}$. Add the constraint

$$x_0 \;\leq\; \prod_{v \in V} y_v$$

where x_0 is a new variable, and make x_0 the distinguished variable of the NRP so obtained. The variable x_0 can be fired only after all the y_v have been fired. The problem above is equivalent to the disjunction of all such instances of the NRP over all possible choices of the y_v. $\square$

6 Decidability of the Nonlinear Reachability Problem

6.1 Well Partial Orders and Dickson's Lemma

A *well partial order* is a partially ordered set in which every infinite sequence has an infinite monotone nondecreasing subsequence. That is, for every infinite sequence $d_0, d_1, \ldots$, there exist indices $i_0 < i_1 < \cdots$ such that $d_{i_0} \leq d_{i_1} \leq \cdots$.

Lemma 6.1 (Dickson's Lemma) *The set $\mathbb{N}^k$ of k-tuples of natural numbers under the componentwise order is a well partial order.*

For a proof of Dickson's Lemma, see [7].

6.2 Reset

Let C be a system of inequalities. Let v be any valuation in V_C, and let $\mathbf{inc}'_v$ and $\mathbf{inc}_v$ be as in Section 5.1. Let $\mathbf{inc}_v(C)$ be the system of inequalities

$$\mathbf{inc}_v(C) \quad = \quad \{p \leq \mathbf{inc}_v(q) - v(p) \mid p \leq q \in C\} \ .$$

The right hand sides $\mathbf{inc}_v(q) - v(p)$ are in $\mathbb{N}[X]$, since the constant coefficient of $\mathbf{inc}_v(q)$ is at least $v(p)$. This is a consequence of the fact that v satisfies C:

$$v(p) \quad \leq \quad v(q) \quad = \quad \mathbf{0}(\mathbf{inc}_v(q)) \ ,$$

where $\mathbf{0}$ is the zero valuation. Moreover, x is constrained in C iff it is constrained in $\mathbf{inc}_v(C)$, since the left hand sides are the same.

Lemma 6.2 *The graphs $G(t, \mathbf{inc}_v(C))$ and $G(\mathbf{inc}'_v(t), C)$ are isomorphic under the map $\mathbf{inc}'_v$.*

Proof. It follows easily from the definitions that the map $\mathbf{inc}'_v$ is one-to-one. We need to show

(i) u satisfies $\mathbf{inc}_v(C)$ iff $\mathbf{inc}'_v(u)$ satisfies C

(ii) x is $(u, \mathbf{inc}_v(C))$-enabled iff x is $(\mathbf{inc}'_v(u), C)$-enabled

(iii) $(\mathbf{inc}'_v(u))x = \mathbf{inc}'_v(ux)$.

For any inequality $p \leq q \in C$, by (4) and the fact that p is linear we have

$$\begin{aligned} \mathbf{inc}'_v(u)(q) &= u(\mathbf{inc}_v(q)) \\ \mathbf{inc}'_v(u)(p) &= u(p) + v(p) \end{aligned}$$

from which it follows that

$$\begin{aligned} \mathbf{inc}'_v(u)(q - p) &= u(\mathbf{inc}_v(q)) - u(p + v(p)) \\ &= u(\mathbf{inc}_v(q) - v(p) - p) \\ &= u(\mathbf{inc}_v(q - p)) \ . \end{aligned}$$

The assertions (i) and (ii) follow. For (iii), we use the fact that $\mathbf{inc}'_v$ and $\mathbf{inc}'_{\delta_x}$ commute:

$$\begin{aligned} (\mathbf{inc}'_v(u))x &= \mathbf{inc}'_{\delta_x}(\mathbf{inc}'_v(u)) \\ &= \mathbf{inc}'_v(\mathbf{inc}'_{\delta_x}(u)) \\ &= \mathbf{inc}'_v(ux) \ . \end{aligned}$$

□

We will apply this lemma as follows. If (s, C) is any instance of the NRP, take $t = \mathbf{0}$ and $v = s$ in Lemma 6.2, and we obtain an isomorphism

$$\mathbf{inc}'_s : G(\mathbf{0}, \mathbf{inc}_s(C)) \quad \rightarrow \quad G(s, C) \ .$$

Informally, we refer to the operation of passing from $G(s, C)$ to the isomorphic graph $G(\mathbf{0}, \mathbf{inc}_s(C))$ as a *reset*. This will allow us to restrict our attention to instances of the NRP of the form $(\mathbf{0}, C)$ without loss of generality.

6.3 Exposed Variables

Definition 6.3 Let $x \in X$, $q \in \mathbb{N}[X]$, and $u \in V_C$. We say that x is *u-exposed* in q if $u(q_i) > 0$ for some $1 \le i \le n$, where $q_0, \ldots, q_n$ are the unique polynomials in $\mathbb{N}[X - \{x\}]$ such that

$$q = \sum_{i=0}^{n} q_i x^i .$$

We say that x is *(u, C)-exposed* if x is u-exposed in q for some $p \le q \in C$. □

Lemma 6.4 *Let $x \in X$, $q \in \mathbb{N}[X]$, and $u \in V_C$. Then x is u-exposed in q iff $u(q) < ux(q)$.*

Proof. Let the q_i be as in Definition 6.3. Since $q_i \in \mathbb{N}[X - \{x\}]$, $u(q_i) = ux(q_i)$. Then

$$\begin{aligned}
ux(q) - u(q) &= ux(\sum_{i=0}^{n} q_i x^i) - u(\sum_{i=0}^{n} q_i x^i) \\
&= \sum_{i=0}^{n} (ux(q_i)ux(x)^i - u(q_i)u(x)^i) \\
&= \sum_{i=0}^{n} u(q_i)((u(x)+1)^i - u(x)^i) \\
&= \sum_{i=1}^{n} u(q_i)((u(x)+1)^i - u(x)^i) \\
&\ge \sum_{i=1}^{n} u(q_i) \\
&\ge 0 ,
\end{aligned}$$

with equality holding iff $u(q_i) = 0$, $1 \le i \le n$. □

For u a valuation, let $\mathbf{sign}\, u$ be the valuation

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

Lemma 6.5 *Let $x \in X$, $p \le q \in C$, and $u \in V_C$.*

(i) If $\mathbf{sign}\, u(y) \leq \mathbf{sign}\, v(y)$ *for all* $y \in X - \{x\}$ *and* x *is* u*-exposed in* q*, then* x *is* v*-exposed in* q*. In particular, if* $\mathbf{sign}\, u(y) = \mathbf{sign}\, v(y)$ *for all* $y \in X - \{x\}$*, then* x *is* u*-exposed in* q *iff* x *is* v*-exposed in* q*.*

(ii) If x *is* u*-exposed in* q *and* $u \leq_X v$*, then* x *is* v*-exposed in* q *(once exposed, always exposed).*

(iii) If $p \leq q$ *is* u*-enabled and* x *is* u*-exposed in* q*, then* $p \leq q$ *is* ux*-enabled.*

(iv) If x *is* u*-exposed in* q *and* x *does not appear in* p*, then* $p \leq q$ *is* ux*-enabled.*

(v) If x *is not* (u, C)*-exposed, then* $ux \leq_C u$*.*

(vi) The property of exposure in the right hand side of a constraint $p \leq q \in C$ *is preserved under the isomorphism of Lemma 6.2. In other words,* x *is* $\mathbf{inc}'_v(u)$*-exposed in* q *iff* x *is* u*-exposed in* $\mathbf{inc}_v(q) - v(p)$*.*

(vii) If $u(x) > 0$*,* x *is not* u*-exposed in* q*, and* x *is* uy*-exposed in* q*, then* y *is* u*-exposed in* q*.*

Proof. Except for (i), (vi), and (vii), all statements are direct consequences of Definition 6.3 and Lemma 6.4.

To prove (i), we observe first that for any $r \in \mathbb{N}[X]$, $\mathbf{sign}\, u(r) = 0$ iff $u(r) = 0$, since any term of r is nonzero under the valuation u iff it is nonzero under the valuation $\mathbf{sign}\, u$. Thus if $\mathbf{sign}\, u(y) \leq \mathbf{sign}\, v(y)$ for all $y \in X - \{x\}$, then $u(q) > 0$ implies $v(q) > 0$ for any $q \in \mathbb{N}[X - \{x\}]$. We obtain the desired conclusion by applying this to the q_i of Definition 6.3.

To prove (vi), we use (4):

$$\begin{aligned}
&ux(\mathbf{inc}_v(q) - v(p)) - u(\mathbf{inc}_v(q) - v(p)) \\
&\quad = \; ux(\mathbf{inc}_v(q)) - u(\mathbf{inc}_v(q)) \\
&\quad = \; \mathbf{inc}'_v(ux)(q) - \mathbf{inc}'_v(u)(q) \\
&\quad = \; \mathbf{inc}'_v(u)x(q) - \mathbf{inc}'_v(u)(q) \; .
\end{aligned}$$

For (vii), by (i), $y \neq x$. Let the q_i be as in Definition 6.3. There must be a q_i, $i > 0$, such that $u(q_i) = 0$ and $uy(q_i) > 0$. Since $u(x) = uy(x) > 0$, we have $u(q_i x^i) = 0$ and $uy(q_i x^i) > 0$, thus $uy(q) > u(q)$. By Lemma 6.4, y is u-exposed in q. □

6.4 Inhibited Variables and Admissible Paths

Definition 6.6 Let C be a system of constraints and $u \in V_C$. We say $x \in X$ is (u, C)-*inhibited* if

- x is unconstrained in C,
- x is not (u, C)-exposed, and
- $u(x) > 0$.

We say that a path in $G(u, C)$ is (u, C)-*admissible* if no inhibited variable is ever fired. In other words, $\sigma \in X^*$ is (u, C)-*admissible* if $u \xrightarrow[C]{\sigma} u\sigma$, and for all prefixes τy of σ, y is not $(u\tau, C)$-inhibited. □

Lemma 6.7 *(i) If y is (u, C)-inhibited, then $u(p) = uy(p)$ and $u(q) = uy(q)$ for all constraints $p \leq q \in C$. In particular, $uy \equiv_C u$.*

(ii) If y, z are (u, C)-inhibited, then z is (uy, C)-inhibited.

Proof.

(i) Since y is unconstrained, it does not appear in p, therefore $u(p) = uy(p)$. Since y is not u-exposed in q, we have $u(q) = uy(q)$ by Lemma 6.4.

(ii) Surely $uy(z) \geq u(z) > 0$ and z is still unconstrained in C, so it remains to show that z is not (uy, C)-exposed. This follows from Lemma 6.5(i) and the facts that z is not (u, C)-exposed and $\mathbf{sign}\, u = \mathbf{sign}\, uy$.

□

The following two lemmas imply that we can restrict our attention to admissible paths when looking for solutions.

Lemma 6.8 *For every path σ of $G(u, C)$, there exists a (u, C)-admissible path τ in $G(u, C)$ such that $u\sigma \leq_C u\tau$.*

Proof. Let us call a prefix $\sigma_1 y$ of σ *bad* if y is $(u\sigma_1, C)$-inhibited. The proof is by lexicographical induction on the length of σ; among paths of the same length, the number of bad prefixes; and among paths of the same length and same number of bad prefixes, the length of the longest bad prefix.

If σ is null or has no bad prefix, there is nothing to prove. If the longest bad prefix $\sigma_1 y$ is σ itself, then since y is not $(u\sigma_1, C)$-exposed, we have by Lemma 6.5(v) that $u\sigma_1 y \leq_C u\sigma_1$, and we are done by the induction hypothesis. Otherwise, there exists a z and σ_2 such that $\sigma = \sigma_1 yz\sigma_2$. Now z is not $(u\sigma_1 y, C)$-inhibited, by the maximality of $\sigma_1 y$. Neither is it $(u\sigma_1, C)$-inhibited, by Lemma 6.7(ii). Moreover, z is $(u\sigma_1, C)$-enabled, by Lemma 6.7(i) and the fact that it is $(u\sigma_1 y, C)$-enabled, and y is $(u\sigma_1 z, C)$-enabled since it is unconstrained. Therefore $\sigma_1 zy\sigma_2$ is a path in $G(u, C)$ of the same length as σ, but with either strictly fewer bad prefixes (if $\sigma_1 zy$ is not a bad prefix) or the same number of bad prefixes and a strictly longer maximal one (if it is). □

Lemma 6.9 *If (u, C) has a solution, then it has a (u, C)-admissible solution.*

Proof. Let σ be a solution of minimal length. If σ is null, *i.e.* if $u(x_0) > 0$ already, then we are done. Otherwise σ is of the form τx_0 and $u\tau(x_0) = 0$. By Lemma 6.8, there exists a (u, C)-admissible path ρ such that $u\tau \leq_C u\rho$. If $u\rho(x_0) > 0$, then ρ is the desired admissible solution. Otherwise, x_0 is $(u\rho, C)$-enabled (since $u\tau \leq_C u\rho$ and x_0 is $(u\tau, C)$-enabled) and not $(u\rho, C)$-inhibited (since $u\rho(x_0) = 0$), therefore ρx_0 is the desired admissible solution. □

A valuation u is said to be *useless* (with respect to C) if all (u, C)-enabled variables are (u, C)-inhibited. It follows from Lemma 6.7(ii) that if u is useless, then so is ux for any (u, C)-enabled x; moreover, by Lemma 6.7(i), $ux \equiv_C u$. Thus if u is useless, then either

- $u(x_0) > 0$, in which case the null path is a solution of the instance (u, C) of the NRP; or
- x_0 is (u, C)-enabled, in which case the path x_0 is a solution; or
- neither of the above, in which case (u, C) has no solution.

6.5 The Graph $H(u, C)$

For a system C of inequalities and $u \in V_C$, construct a finite labeled directed graph $H(u, C)$ as follows. The vertices of $H(u, C)$ are $C \cup \{*\}$. For each

$p \leq q \in C$ and $x \in X$ such that x is u-exposed in q, there is an edge labeled x from $\mathbf{con}\,(x, C)$ to $p \leq q$. (Recall that $\mathbf{con}\,(x, C)$ is $*$ if x is unconstrained in C, otherwise $\mathbf{con}\,(x, C)$ is $p_x \leq q_x$.) Self-loops are allowed in this definition: if x is constrained in C and x is u-exposed in q_x, $H(u, C)$ has a self-loop labeled x on vertex $\mathbf{con}\,(x, C)$.

Lemma 6.10 *(i) If $u \in V_C$, x is (u, C)-enabled, and $H(u, C)$ contains an edge labeled x into $p \leq q$, then $p \leq q$ is ux-enabled.*

(ii) If $u \leq_X v$ then $H(u, C)$ is a subgraph of $H(v, C)$.

Proof.

(i) Either x does not occur in p, in which case

$$ux(p) \;=\; u(p) \;\leq\; u(q) \;<\; ux(q)\ ,$$

or x does occur in p, in which case

$$ux(p) \;=\; u(p) + 1 \;\leq\; u(q) \;<\; ux(q)\ .$$

(ii) This follows immediately from Lemma 6.5(ii).

□

Lemma 6.11 *Let $u \in V_C$ and $\sigma \in X^+$ such that $u \xrightarrow[C]{\sigma} u\sigma$ and $u \leq_C u\sigma$. Assume further that σ contains at least one variable constrained in C. Then $H(u\sigma, C)$ contains either a cycle all of whose labels are in σ or an edge out of $*$ whose label is in σ.*

Proof. Let x be constrained in C such that x occurs in σ at least once. Then

$$u(p_x) \;<\; u\sigma(p_x)\ .$$

Also,

$$u(q_x - p_x) \;\leq\; u\sigma(q_x - p_x)\ ,$$

since $u \leq_C u\sigma$. Combining these inequalities, we obtain

$$u(q_x) \quad < \quad u\sigma(q_x) \ .$$

By Lemma 6.4, there must be a $y \in X$ and a prefix τy of σ such that y is $u\tau$-exposed in q_x. Then $H(u\tau, C)$ contains an edge labeled y from $\mathbf{con}\,(y, C)$ to $\mathbf{con}\,(x, C)$. By Lemma 6.10(ii), this edge also exists in $H(u\sigma, C)$.

Now either y is unconstrained in C, in which case $\mathbf{con}\,(y, C) = *$ and we are done, or we can continue in the same fashion with y. Following these edges backwards, we must eventually either arrive at $*$ or cycle. □

6.6 Equivalence of Problem Instances

Definition 6.12 Let C, D be systems of inequalities. We write $C \leq D$ if for every path σ of $G(\mathbf{0}, C)$ there is a path τ of $G(\mathbf{0}, D)$ such that $\mathbf{0}\sigma \leq_X \mathbf{0}\tau$. We write $C \equiv D$ and say that C and D are *equivalent* if both $C \leq D$ and $D \leq C$. □

The following lemma is immediate.

Lemma 6.13 *If $C \equiv D$, then $(\mathbf{0}, C)$ has a solution iff $(\mathbf{0}, D)$ does.*

6.7 Proof of Decidability

Let C be a system of inequalities.

Lemma 6.14 *Let $p \leq q \in C$. If C has an unconstrained variable $\mathbf{0}$-exposed in q, then*

$$C \quad \equiv \quad C - \{p \leq q\} \ .$$

Proof. Let $C' = C - \{p \leq q\}$. The easier direction is $C \leq C'$. If $u \in V_C$ then $u \in V_{C'}$, since $C' \subseteq C$; and if y is (u, C)-enabled then y is also (u, C')-enabled, since y is either constrained by the same inequality in C and C' or unconstrained in C'. It follows that if σ is a path in $G(\mathbf{0}, C)$ then it is also a path in $G(\mathbf{0}, C')$.

For the other direction, suppose σ is a path in $G(\mathbf{0}, C')$. Let x be a C-unconstrained variable $\mathbf{0}$-exposed in q. Let $n = |\sigma|$ and let

$$\tau \quad = \quad \underbrace{xx\cdots x}_{n}\sigma \quad = \quad x^n\sigma \; .$$

Then $\sigma \leq_X \tau$. We show that τ is a path in $G(\mathbf{0}, C)$. Certainly x^n is a path in $G(\mathbf{0}, C)$, since x is unconstrained. It remains to show that σ is a path in $G(\mathbf{0}x^n, C)$, or equivalently by Lemma 6.2 in the isomorphic graph $G(\mathbf{0}, \mathbf{inc}_{n\delta_x}(C))$. Thus we need to show that for any prefix ρy of σ, y is $(\mathbf{0}\rho, \mathbf{inc}_{n\delta_x}(C))$-enabled. This follows from the fact that y is $(\mathbf{0}\rho, C')$-enabled: for any $f \leq g \in C'$,

$$\begin{array}{rcll} \mathbf{0}\rho(\mathbf{inc}_{n\delta_x}(g-f)) & = & \mathbf{0}\rho x^n(g-f) & \text{by (5)} \\ & = & \mathbf{0}\rho x^n(g) - \mathbf{0}\rho(f) & \text{since } x \text{ does not occur in } f \\ & \geq & \mathbf{0}\rho(g-f) \; , & \end{array}$$

and for the constraint $p \leq q$,

$$\begin{array}{cll} \multicolumn{3}{l}{\mathbf{0}\rho(\mathbf{inc}_{n\delta_x}(q-p))} \\ = & \mathbf{0}\rho x^n(q-p) & \text{by (5)} \\ = & \mathbf{0}\rho x^n(q) - \mathbf{0}\rho(p) & \text{since } x \text{ does not occur in } p \\ \geq & \mathbf{0}\rho(q) + n - \mathbf{0}\rho(p) & \text{by Lemmas 6.4 and 6.5(ii)} \\ \geq & \mathbf{0}\rho(q) + n - |\rho| & \text{since } p \text{ is linear} \\ > & 0 & \text{since } |\rho| < n. \end{array}$$

□

Lemma 6.15 *If $H(\mathbf{0}, C)$ has a self-loop labeled x on vertex $p \leq q$, and if x is $(\mathbf{0}, C)$-enabled, let*

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

Then $C \equiv C'$.

Proof. Since x is $\mathbf{0}$-exposed in q, it follows from Definition 6.3 that q has a term of the form ax^k where $a, k \in \mathbb{N}$ and $a, k \geq 1$; *i.e.*, q can be written

$q'+x^k$ with $q' \in \mathbb{N}[X]$. If the first alternative in the definition of C' holds, *i.e.* if q has a linear term ax, then we can take $k = 1$. If the second alternative holds, we can take $k > 1$. Let us call these two cases (i) and (ii), respectively. Either way, since $\mathbf{con}\,(x, C)$ is $p \leq q$, x also occurs in p, and since p is linear, $p = p' + x$ for some $p' \in \mathbb{N}[X]$.

First we show $C \leq C'$. This is immediate for case (ii) as in Lemma 6.14. For case (i), note that $V_C = V_{C'}$, since $u(q-p) = u(q'-p')$ for all valuations u. Now for any variable $y \in X - \{x\}$ and any valuation u, y is (u, C)-enabled iff it is (u, C')-enabled: either $\mathbf{con}\,(y, C)$ is $p \leq q$, in which case $\mathbf{con}\,(y, C')$ is $p' \leq q'$ and $u(q-p) = u(q'-p')$; or not, in which case $\mathbf{con}\,(y, C) = \mathbf{con}\,(y, C')$. Since x is unconstrained in C', x is always (u, C')-enabled. Thus any variable that is (u, C)-enabled is also (u, C')-enabled. It follows that any path σ of $G(\mathbf{0}, C)$ is also a path of $G(\mathbf{0}, C')$, thus $C \leq C'$.

Now we show $C' \leq C$ for both cases. Let σ be any path of $G(\mathbf{0}, C')$, and let $n = \max 2, |\sigma|$. Let σ' be obtained by deleting all occurrences of x from σ, and let $\tau = x^n\sigma'$. Then $\sigma \leq_X \tau$. We claim that τ is a path of $G(\mathbf{0}, C)$. Since x is $\mathbf{0}$-exposed in q and $(\mathbf{0}, C)$-enabled, by Lemmas 6.4 and 6.5(ii), x^n is a path in $G(\mathbf{0}, C)$, so we need only prove that σ' is a path of $G(\mathbf{0}x^n, C)$. Using Lemma 6.2 to reset, it suffices to prove that σ' is a path of $G(\mathbf{0}, \mathbf{inc}_{n\delta_x}(C))$. We need to show that for any prefix $\rho'y$ of σ', y is $(\mathbf{0}\rho', \mathbf{inc}_{n\delta_x}(C))$-enabled. This will follow from the fact that y is $(\mathbf{0}\rho, C'))$-enabled, where ρy is the unique prefix of σ such that $\rho'y$ is ρy with all occurrences of x removed (note $y \neq x$, since it occurs in σ').

Suppose ρ has m occurrences of x. For any $f \leq g \in C - \{p \leq q\}$,

$$\begin{array}{rcll}
\mathbf{0}\rho'(\mathbf{inc}_{n\delta_x}(g-f)) & = & \mathbf{0}\rho'x^n(g-f) & \text{by (5)} \\
& = & \mathbf{0}\rho'x^n(g) - \mathbf{0}\rho(f) & \text{since } x \text{ does not occur in } f \\
& = & \mathbf{0}\rho x^{n-m}(g) - \mathbf{0}\rho(f) & \\
& \geq & \mathbf{0}\rho(g) - \mathbf{0}\rho(f) & \\
& = & \mathbf{0}\rho(g-f)\ . &
\end{array}$$

For the final argument involving constraint $p \leq q$, we split on cases. In case (i),

$$\begin{array}{rcll}
\mathbf{0}\rho'(\mathbf{inc}_{n\delta_x}(q-p)) & = & \mathbf{0}\rho'x^n(q-p) & \text{by (5)} \\
& = & \mathbf{0}\rho'x^n(q'-p') & \\
& = & \mathbf{0}\rho'x^n(q') - \mathbf{0}\rho(p') & \text{since } x \text{ does not occur in } p'
\end{array}$$

$$
\begin{aligned}
&= \mathbf{0}\rho x^{n-m}(q') - \mathbf{0}\rho(p') \\
&\geq \mathbf{0}\rho(q') - \mathbf{0}\rho(p') \\
&= \mathbf{0}\rho(q' - p') \ .
\end{aligned}
$$

In case (ii),

$$
\begin{aligned}
\mathbf{0}\rho'(\mathbf{inc}_{n\delta_x}(q-p)) &= \mathbf{0}\rho' x^n(q-p) \\
&= \mathbf{0}\rho' x^n(q') + \mathbf{0}\rho' x^n(x^k) - \mathbf{0}\rho' x^n(p') - \mathbf{0}\rho' x^n(x) \\
&\geq \mathbf{0}x^n(x^k) - \mathbf{0}\rho'(p') - \mathbf{0}x^n(x) \\
&\geq n^k - (n-1) - n \\
&\geq (n-1)^2 \\
&> 0 \ .
\end{aligned}
$$

□

Lemma 6.16 *If there is a cycle in $H(\mathbf{0}, C)$ on vertices*

$$
D = \{p_0 \leq q_0, \ldots, p_{n-1} \leq q_{n-1}\} \ ,
$$

then $C \equiv C'$, where

$$
\begin{aligned}
p' &= \sum_{i=0}^{n-1} p_i \\
q' &= \sum_{i=0}^{n-1} q_i \\
C' &= (C - D) \cup \{p' \leq q'\} \ .
\end{aligned}
$$

Proof. First we show $C \leq C'$. As above, it suffices to show that for any valuation $u \in V_C$ and variable y, if y is (u, C)-enabled then y is (u, C')-enabled. If $\mathbf{con}\,(y, C) \notin D$, then $\mathbf{con}\,(y, C') = \mathbf{con}\,(y, C)$, thus y is (u, C)-enabled iff it is (u, C')- enabled. Otherwise, if $\mathbf{con}\,(y, C) \in D$, say $p_k \leq q_k$, then $\mathbf{con}\,(y, C')$ is $p' \leq q'$. Since $u \in V_C$, we have

$$
u(p_i) \leq u(q_i) \ , \quad 0 \leq i \leq n-1 \ .
$$

Moreover, since y is (u, C)-enabled, we have $u(p_k) < u(q_k)$. Thus $u(p') < u(q')$, so y is (u, C')-enabled.

Now we show $C' \leq C$. Assume that the vertices in D occur on the cycle of $H(\mathbf{0}, C)$ in the order $p_0 \leq q_0, \ldots, p_{n-1} \leq q_{n-1}$ and that y_i is the label on the edge from $p_i \leq q_i$ to $p_{i+1} \leq q_{i+1}$, $0 \leq i \leq n-1$ (arithmetic on subscripts is modulo n).

Let σ be any path of $G(\mathbf{0}, C')$. We construct by induction on the length of σ a path σ' of $G(\mathbf{0}, C)$ such that $\mathbf{0}\sigma \leq_{X,C'} \mathbf{0}\sigma'$. Define $\epsilon' = \epsilon$. Now suppose σ' has been defined. By the induction hypothesis,

(i) $\mathbf{0}\sigma \leq_{X,C'} \mathbf{0}\sigma'$

(ii) $\mathbf{0}\sigma' \in V_C$.

Since y is $(\mathbf{0}\sigma, C')$-enabled, by (i) we have that y is $(\mathbf{0}\sigma', C')$-enabled.

If $\mathbf{con}\,(y, C)$ is in $C - D$, let $(\sigma y)' = \sigma' y$. Then $\mathbf{0}\sigma y \leq_X \mathbf{0}(\sigma y)'$, and since $\mathbf{con}\,(y, C) = \mathbf{con}\,(y, C')$, y is $(\mathbf{0}\sigma', C)$-enabled. Moreover, $\mathbf{0}\sigma y \leq_{C'} \mathbf{0}\sigma' y$ by Lemma 5.4.

If $\mathbf{con}\,(y, C)$ is in D, say $p_k \leq q_k$, then $\mathbf{con}\,(y, C')$ is $p' \leq q'$. By (i) and (ii),

$$\begin{aligned} \mathbf{0}\sigma'(p') &< \mathbf{0}\sigma'(q')\ , \\ \mathbf{0}\sigma'(p_i) &\leq \mathbf{0}\sigma'(q_i)\ , \quad 0 \leq i \leq n-1\ . \end{aligned}$$

It follows that there must exist an i, $0 \leq i \leq n-1$, such that

$$\mathbf{0}\sigma'(p_i) \ < \ \mathbf{0}\sigma'(q_i)\ . \tag{9}$$

Define

$$(\sigma y)' \ = \ \sigma' y_i y_{i+1} y_{i+2} \cdots y_{k-1} y$$

(the sequence $i, i+1, \ldots, k-1$ wraps modulo n if necessary). Then $\mathbf{0}\sigma y \leq_X \mathbf{0}(\sigma y)'$. By (9), y_i is $(\mathbf{0}\sigma', C)$-enabled. Since each y_j is $\mathbf{0}$-exposed in q_{j+1}, $0 \leq j \leq n-1$, it follows inductively that each y_j is $(\mathbf{0}\sigma' y_i y_{i+1} \cdots y_{j-1}, C)$-enabled, and y is $(\mathbf{0}\sigma' y_i y_{i+1} \cdots y_{k-1}, C)$-enabled. Thus $\sigma' y_i y_{i+1} \cdots y_{k-1} y$ is a path of $G(\mathbf{0}, C)$.

It remains to show that $\mathbf{0}\sigma y \leq_{C'} \mathbf{0}(\sigma y)'$. For $p \leq q$ in $C - D$,

$$\begin{aligned} \mathbf{0}(\sigma y)'(q-p) &= \mathbf{0}(\sigma y)'(q) - \mathbf{0}(\sigma y)'(p) \\ &\geq \mathbf{0}\sigma y(q) - \mathbf{0}(\sigma y)'(p) \\ &= \mathbf{0}\sigma y(q) - \mathbf{0}\sigma y(p) \qquad \text{since the } y_i \text{ do not appear in } p \\ &= \mathbf{0}\sigma y(q-p)\ . \end{aligned}$$

For $p' \leq q''$, since each y_j is $\mathbf{0}$-exposed in q_{j+1} and hence also in q', we have

$$\begin{aligned}\mathbf{0}(\sigma y)'(q'-p') &= \mathbf{0}\sigma' y_i y_{i+1} y_{i+2} \cdots y_{k-1} y(q'-p') \\ &\geq \mathbf{0}\sigma' y_{i+1} y_{i+2} \cdots y_{k-1} y(q'-p') \\ &\geq \mathbf{0}\sigma' y_{i+2} \cdots y_{k-1} y(q'-p') \\ &\geq \cdots \\ &\geq \mathbf{0}\sigma' y(q'-p') \qquad (10)\end{aligned}$$

By Lemma 5.4 and the induction hypothesis, (10) is bounded below by $\mathbf{0}\sigma y(q'-p')$. □

Theorem 6.17 *It is decidable whether a given instance (u, C) of the NRP has a solution.*

Proof. Assume that $u(x_0) = 0$, otherwise there is nothing to prove. By Lemma 6.2, we may also assume without loss of generality that $u = \mathbf{0}$. We proceed by induction. If $C = \emptyset$, then all variables are unconstrained and therefore enabled, thus we can increment x_0 immediately. Otherwise assume C is nonempty.

We identify a number of cases below, each of which allows us to reduce the size of the system in some respect (either fewer inequalities or fewer constrained variables). In each case, the induction hypothesis gives a procedure for deciding whether the smaller system has a solution, and this will determine whether $(\mathbf{0}, C)$ has a solution.

Case 1 C contains an unconstrained $(\mathbf{0}, C)$-exposed variable. By Lemma 6.14, C is equivalent to a system with fewer inequalities.

Case 2 $H(\mathbf{0}, C)$ has a self-loop labeled x, and x is $(\mathbf{0}, C)$-enabled. By Lemma 6.15, C is equivalent to a system with either fewer constrained variables or fewer inequalities.

Case 3 $H(\mathbf{0}, C)$ has a cycle on a set of at least two vertices. By Lemma 6.16, C is equivalent to a system with fewer inequalities.

Case 4 None of Cases 1,2, or 3 occur. In this case, consider the tree $T \subseteq X^*$ consisting of all $(\mathbf{0}, C)$-admissible paths $\sigma \in X^*$. The tree T contains the empty string ϵ and is closed under the prefix relation, and for any $\sigma \in T$, $\sigma x \in T$ iff x is $(\mathbf{0}\sigma, C)$-enabled but not $(\mathbf{0}\sigma, C)$-inhibited. By Lemma 6.9, $(\mathbf{0}, C)$ has a solution if and only if it has one in T.

Now let T' be the subtree of T obtained by deleting all strings containing a proper prefix of the form $\sigma\tau$, where $|\tau| > |X|$ and $\mathbf{0}\sigma \leq_C \mathbf{0}\sigma\tau$. By Dickson's Lemma, T' has no infinite paths, and by König's Lemma, T' is finite. T' can be constructed effectively since the conditions for extending a branch and for halting along any path are effective.

Now $(\mathbf{0}, C)$ has a solution iff $(\mathbf{0}\sigma, C)$ has a solution for some leaf σ of T'. Thus we can consider each leaf separately. The leaves are of two types:

(i) The leaf is σ and the valuation $\mathbf{0}\sigma$ is useless; *i.e.*, all $(\mathbf{0}\sigma, C)$-enabled variables are $(\mathbf{0}\sigma, C)$-inhibited. This is a "natural" leaf, since it has no $(\mathbf{0}, C)$-admissible extensions.

(ii) The leaf is of the form $\sigma\tau$ where $\mathbf{0}\sigma \leq_C \mathbf{0}\sigma\tau$. This is an "artificial" leaf, since it was obtained by pruning T.

If $\mathbf{0}\sigma(x_0) > 0$ or x_0 is $(\mathbf{0}\sigma, C)$-enabled for some leaf σ, we are done. Otherwise, by Lemma 6.9, leaves of the form (i) have no solution. Thus we are left with leaves of the form (ii). For each such leaf $\sigma\tau$, we have

$$\begin{array}{ccc} \mathbf{0}\sigma & \xrightarrow[C]{\tau} & \mathbf{0}\sigma\tau \\ \mathbf{0}\sigma & \leq_C & \mathbf{0}\sigma\tau \, . \end{array}$$

Since $\sigma\tau$ is $(\mathbf{0}, C)$-admissible, for every prefix ρx of τ, either

- x is constrained in C,
- x is $(\mathbf{0}\sigma\rho, C)$-exposed, or
- $\mathbf{0}\sigma\rho(x) = 0$.

Suppose τ contains a variable constrained in C. By Lemma 6.11, $H(\mathbf{0}\sigma\tau, C)$ contains either an edge out of $*$ or a cycle whose labels are in τ. If the former, we revert to Case 1. If the latter and the cycle is of length at least two, we revert to Case 3. Otherwise there is a self-loop in $H(\mathbf{0}\sigma\tau, C)$ with label x, where ρx is a prefix of τ. If that self-loop already exists in $H(\mathbf{0}\sigma\rho, C)$, then since x is $\mathbf{0}\sigma\rho$-enabled, we revert to Case 2. Otherwise, let vy be the shortest prefix of τ such that $H(\mathbf{0}\sigma vy, C)$ contains that self-loop. By Lemma 6.5(vii), x is $\mathbf{0}\sigma vy$-enabled, and we revert to Case 2.

If all variables occurring in τ are unconstrained in C and at least one is $(\mathbf{0}\sigma\rho, C)$-exposed for some prefix ρ of τ, then by Lemma 6.10(ii), $H(\mathbf{0}\sigma\tau, C)$ has an edge out of $*$, and we revert to Case 1.

Finally, if all variables occurring in τ are unconstrained in C and not $(\mathbf{0}\sigma\tau, C)$-exposed, we must have $\mathbf{0}\sigma\rho(x) = 0$ for every prefix ρx of τ, otherwise the path would not be admissible. But since $|\tau| > |X|$, this is impossible. $\square$

Acknowledgements

We are indebted to Moshe Vardi for many valuable ideas.

Dexter Kozen was supported by the Danish Research Academy, the National Science Foundation, the John Simon Guggenheim Foundation, and the U.S. Army Research Office through the ACSyAM branch of the Mathematical Sciences Institute of Cornell University under contract DAAL03-91-C-0027.

References

[1] 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.

[2] A. Aiken and B. Murphy. Implementing regular tree expressions. In *Proc. 1991 Conf. Functional Programming Languages and Computer Architecture*, pages 427–447, August 1991.

[3] A. Aiken and B. Murphy. Static type inference in a dynamically typed language. In *Proc. 18th Symp. Principles of Programming Languages*, pages 279–290. ACM, January 1991.

[4] A. Aiken and E. Wimmers. Solving systems of set constraints. In *Proc. 7th Symp. Logic in Computer Science*, pages 329–340. IEEE, June 1992.

[5] Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Set constraints are the monadic class. In *Proc. 8th Symp. Logic in Computer Science*. IEEE, June 1993. to appear.

[6] J. A. Brzozowski and E. Leiss. On equations for regular languages, finite automata, and sequential networks. *Theor. Comput. Sci.*, 10:19–35, 1980.

[7] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag, 1992.

[8] R. Gilleron, S. Tison, and M. Tommasi. Solving systems of set constraints using tree automata. In *Proc. Symp. Theor. Aspects of Comput. Sci.*, volume 665, pages 505–514. Springer-Verlag Lect. Notes in Comput. Sci., February 1993.

[9] 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.

[10] N. Heintze and J. Jaffar. A decision procedure for a class of set constraints. In *Proc. 5th Symp. Logic in Computer Science*, pages 42–51. IEEE, June 1990.

[11] N. Heintze and J. Jaffar. A finite presentation theorem for approximating logic programs. In *Proc. 17th Symp. Principles of Programming Languages*, pages 197–209. ACM, January 1990.

[12] N. D. Jones and S. S. Muchnick. Flow analysis and optimization of LISP-like structures. In *Proc. 6th Symp. Principles of Programming Languages*, pages 244–256. ACM, January 1979.

[13] K. Marriott and M. Odersky. Systems of negative boolean constraints. Technical Report YALEU/DCS/RR-900, Computer Science Department, Yale University, April 1992.

[14] P. Mishra. Towards a theory of types in PROLOG. In *Proc. 1st Symp. Logic Programming*, pages 289–298. IEEE, 1984.

[15] P. Mishra and U. Reddy. Declaration-free type checking. In *Proc. 12th Symp. Principles of Programming Languages*, pages 7–21. ACM, 1985.

[16] J. C. Reynolds. Automatic computation of data set definitions. In *Information Processing 68*, pages 456–461. North-Holland, 1969.

[17] J. Young and P. O'Keefe. Experience with a type evaluator. In D. Bjørner, A. P. Ershov, and N. D. Jones, editors, *Partial Evaluation and Mixed Computation*, pages 573–581. North-Holland, 1988.