Decidability of affine solution problems

STEFAN HETZL, *Institute of Discrete Mathematics and Geometry, Faculty of Mathematics and Geoinformation, TU Wien, 1040 Vienna, Austria.*
E-mail: stefan.hetzl@tuwien.ac.at

SEBASTIAN ZIVOTA, *Institute of Discrete Mathematics and Geometry, Faculty of Mathematics and Geoinformation, TU Wien, 1040 Vienna, Austria.*

Abstract

We present formula equations—first-order formulas with unknowns standing for predicates—as a general formalism for treating certain questions in logic and computer science, like the *Auflösungsproblem* and loop invariant generation. In the case of the language of affine terms over $\mathbb{Q}$, we translate a quantifier-free formula equation into an equivalent statement about affine spaces over $\mathbb{Q}$, which can then be decided by an iteration procedure.

1 Introduction

The question of how to solve formulas containing predicate variables, i.e. substitute formulas for the predicate variables such that the entire formula becomes valid, has a history going back to the 19th century (see [22, 23]). It was considered by Schröder in the context of propositional logic under the name *Auflösungsproblem* [20]. The *Auflösungsproblem* is related to the notion of Boolean unification: given propositional formulas φ, ψ with Boolean variables, find a substitution σ such that $\varphi\sigma = \psi\sigma$ modulo the theory of Boolean algebra. We can view this as an instance of the *Auflösungsproblem* because every substitution σ that makes $(\varphi \leftrightarrow \psi)\sigma$ true modulo Boolean algebra serves as a unifier for φ and ψ. On the other hand, solving a formula φ is tantamount to unifying φ with $\top$. Boolean unification has been studied extensively; see [17] and [2, 16]. It plays a role in several application areas such as database systems [13, 14] and logic programming [4].

Boolean unification has been extended to first-order logic under the name Boolean unification with predicates (BUPs) in [6]. There, the authors consider the following problem: given a formula $\varphi(X_1, \ldots, X_n)$ in first-order logic with equality, where $X_1, \ldots, X_n$ are predicate variables, are there quantifier-free first-order formulas $G_1, \ldots, G_n$ such that $\varphi(G_1, \ldots, G_n)$ is valid modulo equality? They show that the problem is undecidable if φ is of the form $\forall \bar{x} \varphi'$ or $\exists \bar{x} \varphi'$ with φ' quantifier-free by reduction from the Post correspondence problem and the validity problem of first-order logic, respectively. On the other hand, they prove that for quantifier-free φ, the problem is $\prod^p_2$-complete. The solvability of BUP problems is relevant for the automated introduction of cuts [8, 9] and for automatically finding induction invariants [5].

There is a parallel between Boolean unification problems and equations over, e.g. the rational numbers: solving the equation $t(x_1, \ldots, x_n) = 0$ amounts to finding terms $g_1, \ldots, g_n$ such that $t(g_1, \ldots, g_n)$ evaluates to 0 modulo the theory of $\mathbb{Q}$; similarly, solving the Boolean unification problem $\varphi(X_1, \ldots, X_n)$ amounts to finding formulas $G_1, \ldots, G_n$ such that $\varphi(G_1, \ldots, G_n)$ is equivalent to $\top$ modulo equality. In other words, the problem is to solve the 'equation' $\varphi(x_1, \ldots, x_n) \leftrightarrow \top$. For this reason, we call formulas of the form $\varphi(X_1, \ldots, X_n)$ *formula equations*.

Formula equations are hardly, if at all, considered in the literature. We argue that they ought to be, as they allow one to represent a variety of problems in a uniform way. This paper is intended to serve

Vol. 30, No. 3, © The Author(s) 2020. Published by Oxford University Press. All rights reserved.
For permissions, please e-mail: journals.permission@oup.com.
Advance Access Publication on 2 January 2020 doi:10.1093/logcom/exz033

as an example of how a previous result can be expressed in a more general manner with formula equations.

The problem of solving formula equations is related to that of second-order quantifier elimination: we say that a theory $\mathcal{T}$ has quantifier elimination if every formula is equivalent to a quantifier-free formula modulo $\mathcal{T}$. Consider the second-order formula $\psi \equiv \exists X \varphi(X)$. If we can find a solution G to the formula equation $\varphi(X)$ modulo $\mathcal{T}$, then ψ is equivalent to the quantifier-free formula $\varphi(G)$. On the other hand, a theory having second-order quantifier elimination does not necessarily allow us to solve formula equations, as it does not guarantee the existence of a witness. Ackermann investigated second-order quantifier elimination in [1].

Formula equations naturally lend themselves to expressing problems of *loop invariant generation*. The conditions necessary for a formula I to be a correct loop invariant of some program p can typically be expressed as a formula $\varphi(I)$. If the invariant is unknown, then I is a variable and φ is a formula equation whose solutions are exactly the invariants of p. We will say more about this topic in Section 2.3.

The authors of [3] advocate the use of sets of constrained Horn clauses as a target language for problems of program verification. They give a procedure for extracting verification conditions from a program formalism with assertions and subroutine calls that results in nonlinear constrained Horn clauses, i.e. those with more than one formula variable in the antecedent. Formula equations are more general in not restricting the number of positive formula variables that can occur in a clause, resulting in a structure that has no obvious correlate in programs. We follow the line of reasoning of [3] in considering a logical formalism useful for the representation and solution of problems of program verification and we extend it from solving sets of constrained Horn clauses to solving formula equations. The interesting question then becomes whether methods for finding loop invariants generalize to (certain classes of) formula equations. In this paper, we answer this question in the affirmative for one particular setting: that of affine invariants for affine programs.

Algorithms for computing affine invariants of affine programs have been given in [19] and its antecedent [15]. When translated to formula equations, their methods provide solutions for formula equations which are conjunctions of *affine Horn clauses*. We significantly generalize this result to arbitrary quantifier-free formula equations. The gist is that an affine formula equation φ can be clausified and the clauses translated into statements about affine spaces over $\mathbb{Q}$ called *affine conditions*; during this process, the unknown formulas in φ (the predicate variables) become unknown affine subspaces of $\mathbb{Q}^k$ for some k. At this point, we use the fact that affine subspaces of $\mathbb{Q}^k$ have a certain disjunction property: if a subspace $\mathcal{Y}$ is covered by $\bigcup_{i=1}^{n} \mathcal{X}_i$, then it is already covered by some $\mathcal{X}_{i_0}$. This allows us to break the affine conditions of φ apart into *affine Horn conditions*. If a solution to some set of affine Horn conditions exists, we find it with an iteration procedure that is guaranteed to terminate. In fact, the procedure returns the smallest subspaces satisfying all affine Horn conditions.

In Section 2, we define formula equations and investigate their relationship to problems of loop invariant generation. We also state the affine solution problem and its decidability. In Section 3, we prove the decidability result. In Section 3.5, we describe how a solution in terms of affine spaces can be translated back to a solution in terms of linear equation systems.

2 Formula equations and invariant generation

2.1 Logical preliminaries

We review some notions of first-order logic that will be essential to this paper. A more in-depth treatment can be found in [11]. A *first-order language L* is a collection of constant, function and

predicate symbols. Terms over $\mathscr{L}$ are constructed from variables and constant and function symbols of $\mathscr{L}$. First-order formulas over $\mathscr{L}$, or $\mathscr{L}$-formulas, are in turn constructed from predicate symbols of $\mathscr{L}$, terms over $\mathscr{L}$ and logical connectives.

For a logical language $\mathscr{L}$, an $\mathscr{L}$-structure $\mathscr{M}$ is a set M (the domain) together with a function $\cdot^{\mathscr{M}}$ that interprets constants, functions and predicates of $\mathscr{L}$ as elements, functions and relations of M, respectively. We stipulate that if $\mathscr{L}$ contains the binary predicate symbol $=$, it is always interpreted as the actual equality relation. On this basis, the truth of a closed formula φ in a structure $\mathscr{M}$, written as $\mathscr{M} \models \varphi$, is defined inductively.

We call a set of closed $\mathscr{L}$-formulas a *theory*. A theory $\mathscr{T}$ is true in a structure $\mathscr{M}$, written as $\mathscr{M} \models \mathscr{T}$, if all formulas in $\mathscr{T}$ are true in $\mathscr{M}$. In this case, $\mathscr{M}$ is called a *model* of $\mathscr{T}$. If $\mathscr{M}$ is an $\mathscr{L}$-structure, we call the set $\mathrm{Th}(\mathscr{M})$ of all $\mathscr{L}$-formulas that are true in $\mathscr{M}$ the theory of $\mathscr{M}$.

Let $\mathscr{L}$ be a first-order language, $\mathscr{T}$ an $\mathscr{L}$-theory and φ an $\mathscr{L}$-formula. We say that $\mathscr{T}$ entails φ, written as $\mathscr{T} \models \varphi$, if $\mathscr{M} \models \varphi$ for every model $\mathscr{M}$ of $\mathscr{T}$.

We give some examples to illustrate these concepts.

EXAMPLE

Let $\mathscr{L}^{+}_{\mathbb{N}}$ be the language consisting of the constant symbols 0 and 1, the binary function symbol $+$ and the binary predicate symbol $=$. Let PB be the theory consisting of the following formulas:

1. $\forall x.\, x + 1 \neq 0$
2. $\forall x, y.\, x + 1 = y + 1 \rightarrow x = y$
3. $\forall x.\, x + 0 = x$
4. $\forall x, y.\, x + (y + 1) = (x + y) + 1$
5. For any $\mathscr{L}^{+}_{\mathbb{N}}$-formula φ with one free variable:

$$\varphi(0) \wedge \forall y(\varphi(y) \rightarrow \varphi(y + 1)) \rightarrow \forall x \varphi(x).$$

PBis called *Presburger arithmetic*. Let $\mathscr{N}^{+}$ be the $\mathscr{L}^{+}_{\mathbb{N}}$-structure with domain $\mathbb{N}$ and 0, 1 and $+$ interpreted as actual 0, 1 and addition. It is easy to see that $\mathscr{N}^{+}$ is a model of PB.

$\forall x.\, 0 + x = x$ is an example of a formula that is entailed by PB.

EXAMPLE

Let $\mathscr{L}^{+\times}_{\mathbb{N}}$ be the language $\mathscr{L}^{+}_{\mathbb{N}}$ augmented by the binary function symbol $\times$. Clearly, every $\mathscr{L}^{+}_{\mathbb{N}}$-formula is also a $\mathscr{L}^{+\times}_{\mathbb{N}}$-formula. Let PA be the theory containing all formulas of PB plus the following formulas:

1. $\forall x.\, x \times 0 = 0$
2. $\forall x, y.\, x \times (y + 1) = x \times y + x$
3. For any $\mathscr{L}^{+\times}_{\mathbb{N}}$-formula φ with one free variable:

$$\varphi(0) \wedge \forall y(\varphi(y) \rightarrow \varphi(y + 1)) \rightarrow \forall x \varphi(x)$$

PAis called *Peano arithmetic*. Let $\mathscr{N}^{+\times}$ be the $\mathscr{L}^{+\times}_{\mathbb{N}}$-structure that interprets $\mathscr{L}^{+}_{\mathbb{N}}$ as $\mathscr{N}^{+}$ does and $\times$ as actual multiplication. $\mathscr{N}^{+\times}$ is a model of PA.

2.2 Formula equations

DEFINITION 2.1 (Formula equation).

Let $\mathscr{L}$ be a first-order language. A *formula equation* over $\mathscr{L}$ is an $\mathscr{L}$-formula φ that may additionally contain formula variables, i.e. variables standing for m-ary predicates for some $m \in \mathbb{N}$. To distinguish

them from the formula variables, we call the first-order variables in φ *individual variables*. The formula variables contained in φ are called the *unknowns* of φ.

EXAMPLE 2.2
$\varphi \equiv \forall n.\, X(2{\times}n) \wedge (X(n) \rightarrow \neg X(n+1))$ is a formula equation over the language of PA. It is intuitively clear that φ should have a solution modulo PA; we clarify this intuition in the next definition.

DEFINITION 2.3 (Solution problem).
Let $\mathscr{L}$ be a first-order language. A solution problem over $\mathscr{L}$ has three components:

1. A theory $\mathscr{T}$ (the background theory)
2. A class $\mathscr{C}$ of formulas (the candidate solutions)
3. A class Φ of formula equations.

Given $\mathscr{T}, \mathscr{C}, \Phi$ as above, the *solution problem* $\langle \mathscr{T}, \mathscr{C}, \Phi \rangle$ is the set of formula equations in Φ that have solutions in $\mathscr{C}$ modulo the theory $\mathscr{T}$, i.e. the set

$$\left\{ \varphi \in \Phi \;\middle|\; \exists \bar{F} \in \mathscr{C} \text{ s.t. } \mathscr{T} \models \varphi[X_1 \backslash F_1, \ldots, X_n \backslash F_n] \right\}.$$

EXAMPLE 2.4
Let $\mathscr{L}_{\mathbb{N}}^{+\times}$ be the language of PA. Consider the solution problem $\mathscr{P} = \langle \mathrm{PA}, \mathscr{C}, \Phi \rangle$, where $\mathscr{C}$ is the set of purely existential formulas of $\mathscr{L}_{\mathbb{N}}^{+\times}$ and Φ is the set of all formula equations over $\mathscr{L}_{\mathbb{N}}^{+\times}$. Then φ from Example 2.2 is an instance of $\mathscr{P}$, since $E(n) \equiv \exists k\; n = 2{\times}k \in \mathscr{C}$ and

$$\mathrm{PA} \models \forall n.\, \exists k\; 2{\times}n = 2{\times}k \wedge (\exists k\; n = 2{\times}k \rightarrow \neg \exists k\; n + 1 = 2{\times}k).$$

Let us give some more general examples of the expressive power of formula equations.

1. Let $\mathscr{L}$ be any first-order language and Φ the set of variable-free formula equations—in other words, first-order formulas—over $\mathscr{L}$. The solution problem $\langle \emptyset, \emptyset, \Phi \rangle$ is the problem of first-order validity.
2. Let $\mathscr{C} = \{\top, \bot\}$ and Φ the set of formula equations that contain only nullary formula variables and propositional connectives. Then $\langle \emptyset, \mathscr{C}, \Phi \rangle$ is SAT.
3. Provability in PA can be obtained as a solution problem as follows. A formula φ is provable in PA iff φ is provable from a single induction axiom and the axioms of a simple base theory such as Robinson arithmetic Q in first-order logic; see, e.g. [10, Corollary 2.1]. Let $\mathscr{L}$ be the language of arithmetic and $\mathscr{C}$ the class of all first-order formulas over $\mathscr{L}$. Furthermore, let IND be the formula equation $X(0) \wedge \forall n(X(n) \rightarrow X(n+1)) \rightarrow \forall n X(n)$, i.e. the induction principle for an unknown formula X and Φ the set of all formula equations of the form $Q \wedge \mathrm{IND} \rightarrow \varphi$, where Q is the conjunction of the axioms of Robinson arithmetic and φ is a closed formula over $\mathscr{L}$. Then $\langle \emptyset, \mathscr{C}, \Phi \rangle$ is the problem of provability in PA.

These examples show that a large variety of questions can be formulated as solution problems, among them also some whose algorithmic solution poses considerable difficulties in practice. We are particularly interested in the decidability of loop invariant generation.

2.3 Invariant generation

Formula equations can be used as a formalism for investigating questions of loop invariant generation: as soon as a program formalism contains some notion of loops or recursion, verifying

a program entails finding invariants that are generally non-analytic. While there are several ways to represent programs and reason about them, such as Hoare calculus, state transition diagrams and control flow graphs, they all have the problem of finding invariants in common. Formula equations are of interest here because irrespective of the formalism used to represent it, the correctness of a program with respect to some pre- and postcondition can be put in the form of a formula equation in which the unknowns are the loop invariants.

For instance, in Hoare calculus, one investigates *Hoare triples* $\{A\}p\{B\}$, where A and B are formulas and p is a program, expressing that if A holds before running p, then B must hold afterwards. The triple $\{A\}p\{B\}$ can be transformed into a set of first-order formulas $\mathrm{vc}(A,p,B)$ (the *verification conditions*) such that $\bigwedge \mathrm{vc}(A,p,B)$ implies $\{A\}p\{B\}$. However, the Hoare calculus rule for loops involves a loop invariant that cannot trivially be derived from the program but has to be supplied by a human. Thus, the formulas in $\mathrm{vc}(A,p,B)$ will contain formula variables for the unknown invariants, making $\bigwedge \mathrm{vc}(A,p,B)$ a formula equation. Each element of $\mathrm{vc}(A,p,B)$ has the form $X(\bar{x}), A_1(\bar{x}), \ldots, A_m(\bar{x}) \vdash Y(\bar{t}(\bar{x}))$, where X and Y are formula variables and the A_i are first-order formulas. Such clauses are called *linear constrained Horn clauses* and the A_i are called *constraint formulas*.

Another popular formalism are state transition systems. Before showing how to translate the state transition systems of [19] to formula equations in Section 2.5 we introduce affine formula equations that will be useful for that purpose in Section 2.4.

2.4 Affine formula equations

DEFINITION 2.5 ($\mathscr{L}_{\mathrm{aff}}$).
Let $\mathscr{L}_{\mathrm{aff}}$ be the first-order language consisting of

- the constants 0 and 1;
- the binary function symbol $+$;
- the unary function symbols $\{c \mid c \in \mathbb{Q}\}$;
- the binary predicate symbol $=$.

In the sequel, let $\mathrm{Th}(\mathbb{Q})$ be the theory of $\mathbb{Q}$ over $\mathscr{L}_{\mathrm{aff}}$, with $0, 1, +, =$ interpreted in the natural manner and the unary function symbol c interpreted as multiplication with c for each $c \in \mathbb{Q}$. For $c \in \mathbb{Q}$, we write the term $c1$ as c. Moreover, for any term t, we write the term $-1t$ as $-t$.

We write $t(x_1, \ldots, x_n)$ for a term that contains exactly the variables $x_1, \ldots, x_n$. Similarly, $A(x_1, \ldots, x_n)$ denotes a formula whose free variables are exactly $x_1, \ldots, x_n$. We call t and A a term and a formula over $\{x_1, \ldots, x_n\}$, respectively.

For the rest of the paper, we will tacitly interpret terms and formulas over $\mathscr{L}_{\mathrm{aff}}$ modulo $\mathrm{Th}(\mathbb{Q})$. Consequently, we can assume without loss of generality that every term $t(x_1, \ldots, x_n)$ is of the form $c_0 + \sum_{i=1}^{n} c_i x_i$ and every atomic formula $A(x_1, \ldots, x_n)$ is of the form $t(x_1, \ldots, x_m) = 0$. We call such atomic formulas *linear equations* and conjunctions of linear equations *linear equation systems*.

Since $\mathrm{Th}(\mathbb{Q}) \models c_0 + \sum_{i=1}^{m} c_i x_i = c_0 + \sum_{i=1}^{m} c_i x_i + 0y$, a term over the set of variables S is also a term over any $S' \supseteq S$. Similarly, a linear equation (system) over S is also a linear equation (system) over any $S' \supseteq S$.

Thus, if φ is a formula over $\mathscr{L}_{\mathrm{aff}}$ containing exactly the individual variables $x_1, \ldots, x_n$, we may assume without loss of generality that each term in φ is an affine term over $x_1, \ldots, x_n$ and each atomic formula in φ is a linear equation over $x_1, \ldots, x_n$. Note that the latter stipulation does not

include formula variables: a formula variable in an affine formula equation may contain any number of terms.

DEFINITION 2.6 (Affine formula equation).
An *affine formula equation* is a quantifier-free formula equation over $\mathscr{L}_{\text{aff}}$.

Let $X_1, \ldots, X_m$ be the formula variables in φ with respective arities $k_1, \ldots, k_m$. Then the tuple $\langle n; k_1, \ldots, k_m \rangle$ is called the *dimensions* of φ.

DEFINITION 2.7 (Affine solution problem).
The *affine solution problem* is the solution problem $\langle \text{Th}(\mathbb{Q}), \mathscr{C}, \Phi \rangle$ where $\mathscr{C}$ is the class of linear equation systems and Φ is the class of affine formula equations.

Throughout the rest of this paper, we will illustrate the parts of our procedure with the following running example.

EXAMPLE 2.8
Let φ be the affine formula equation $X(1,0) \wedge (X(x,y) \to X(-y,x) \vee X(x,-y)) \wedge (X(x,y) \to x = y \vee y = 0)$. The dimensions of φ are $\langle 2; 2 \rangle$.

THEOREM 2.9 (Main result).
The affine solution problem is decidable.

Restricting the affine solution problem to quantifier-free formulas allows to obtain this decidability result. In Section 3, we develop the means necessary to prove Theorem 3.27. Before we start work on the proof, let us discuss the connection between affine solution problems and invariant generation in affine programs.

2.5 Invariant generation in affine programs

In [19] the authors consider a formalism for the verification of affine programs based on state transition diagrams: a program consists of points connected by directed edges, each of which is labeled with either an affine variable assignment or a subroutine call. There are no guards of any kind; loops continue iterating or terminate nondeterministically. Operations other than affine assignments are abstracted as *nondeterministic assignments* $x :=?$ that may set a variable to an arbitrary value. The authors present an algorithm that calculates, for each program point, the set of all linear equations that hold at that point whenever execution reaches it. This algorithm is based on iteratively computing (an abstract representation of) the set of runs that reach each program point. This idea has no obvious analogue in formula equations, since there is no sense of 'flow' from one location to the next. Instead, we take the position that an affine formula equation describes a relationship between certain affine spaces and their images and preimages under some affine transformations. It is then straightforward to iteratively calculate minimal affine spaces that can possibly satisfy this relationship. If they do, they are a (maximally precise) solution; if they do not, then no solution exists.

Let us present the transformation from the graph formalism of [19] to formula equations in a bit more detail. Each procedure p in the program has an entry point e_p and a return point r_p. Moreover, there is a special procedure **Main** whose entry and return points serve as the entry and return points of the whole program. For each point s, let X_s be a formula variable over all individual variables of the program. We give the resulting formula equation as a set of constrained Horn clauses:

- $\vdash X_{e_{\mathbf{Main}}}(\bar{x})$ asserts that at the start of the program, no nontrivial linear equations are valid;

- If $s \xrightarrow{x_j:=t(\bar{x})} s'$ is an edge in the program, then we add the clause $X_s(\bar{x}) \vdash X_{s'}(x_1, \ldots, t(\bar{x}), \ldots, x_n)$.
- If $s \xrightarrow{x_j:=?} s'$ is an edge in the program, we add the clauses $X_s(\bar{x}) \vdash X_{s'}(x_1, \ldots, 0, \ldots, x_n)$ and $X_s(\bar{x}) \vdash X_{s'}(x_1, \ldots, 1, \ldots, x_n)$. The idea here is that if $(x_1, \ldots, 0, \ldots, x_n)$ and $(x_1, \ldots, 1, \ldots, x_n)$ both solve the linear equation system $X_{s'}$, then so does $(x_1, \ldots, y, \ldots, x_n)$ for any $y \in \mathbb{Q}$.
- If $s \xrightarrow{p} s'$ is an edge in the program and p is a procedure, we add the clauses $X_s(\bar{x}) \vdash X_{e_p}(\bar{x})$ and $X_{r_p}(\bar{x}) \vdash X_{s'}(\bar{x})$.

These formula equations are even simpler than in the case of Hoare calculus: there is only at most one negative formula variable in each clause; constrained Horn clauses with this property are called *linear constrained Horn clauses*. Moreover, due to the absence of guards, there are no constant formulas in these clauses.

Note that [19] is not concerned with whether a solution exists; in fact, the existence of a solution of φ is not in question, since we can let $X_p \equiv \top$ for all program points p of P. Rather, they compute a *strongest* solution for each unknown in φ. In this sense, our approach diverges from theirs. Moreover, it is easily seen that in general, a formula equation need not have a unique strongest solution. It turns out, however, that our algorithm MINIMALSOLUTION produces a *maximally strong* solution. We will elaborate on this in Sections 3.4 and 3.5.

3 Deciding affine solution problems

This section is devoted to proving Theorem 3.27. We proceeds as follows. We first translate φ into a set of clauses $\mathrm{Cl}(\varphi)$. Each clause C induces a statement $\mathscr{C}$ that asserts the inclusion of an affine space in a union of affine spaces. The statements we obtain in this way from clauses in $\mathrm{Cl}(\varphi)$ form the set $\mathrm{AC}(\varphi)$ of affine conditions of φ. Notably, solutions of φ and $\mathrm{AC}(\varphi)$ correspond to each other. We then apply Algorithm 1, which iteratively computes a solution of $\mathrm{AC}(\varphi)$ if a solution exists and reports failure otherwise. The resulting tuple of affine subspaces can be translated back into a tuple of linear equation systems.

3.1 Clausification

Let φ be an affine formula equation. As mentioned before, we may assume that terms and atomic formulas in φ are respectively affine functions of their variables and linear equation systems. It follows that φ is a Boolean combination of linear equations and formula variables. We go a step further and regard linear equation systems—i.e. conjunctions of linear equations—as the building blocks of affine formula equations. Therefore, we use the word ‘clause’ to refer to a disjunction of possibly negated linear equation systems (and formula variables, which range over linear equation systems).

The goal is to transform φ into a conjunction of clauses, i.e. formulas of the form

$$\neg A(\bar{x}) \vee \neg X_1(\bar{s}_1(\bar{x})) \vee \ldots \vee \neg X_m(\bar{s}_m(\bar{x}))$$
$$\vee B_1(\bar{x}) \vee \ldots \vee B_r(\bar{x}) \vee Y_1(\bar{t}_1(\bar{x})) \vee \ldots \vee Y_n(\bar{t}_n(\bar{x})),$$

where s_i, t_i are affine terms, A, B_i are linear equation systems and X_i, Y_i are formula variables ranging over linear equation systems. As per the usual convention, such a clause is implicitly universally closed over its individual variables. Note that for $i \neq j$, X_i and X_j are not necessarily distinct formula variables; for instance, a clause might contain the same formula variable multiple times with different terms.

We write clauses in sequent form, i.e. the list of negative atoms, then the dividing symbol '$\vdash$', then the list of positive atoms. The negative atoms are implicitly joined by conjunction, the positive atoms by disjunction. Thus, the clause above is written as

$$A(\bar{x}), X_1(\bar{s}_1(\bar{x})), \ldots, X_m(\bar{s}_m(\bar{x})) \vdash B_1(\bar{x}), \ldots, B_r(\bar{x}), Y_1(\bar{t}_1(\bar{x})), \ldots, Y_n(\bar{t}_n(\bar{x})).$$

In principle, any kind of clause form transformation that preserves satisfiability will serve here if we are only interested in the existence of solutions, but for transformations that do not preserve logical equivalence, it may be unclear how the solutions of the original formula equation and the clause form relate to each other. Thus, for the sake of simplicity, we choose the naive method of computing a clause form by distributivity.

DEFINITION 3.1 (Cl(φ)).
Let φ be an affine formula equation. Then Cl(φ) is the set of clauses obtained from φ by distributing out its logical connectives.

EXAMPLE 3.2
Let φ be the affine formula equation defined in Example 2.8. φ is already in clause form; writing the clauses of φ as sequents yields

$$\vdash X(1,0)$$
$$X(-x,y) \vdash X(-y,x), X(x,-y)$$
$$X(x,y) \vdash x = y, y = 0.$$

3.2 Affine spaces

$\mathbb{Q}$ interprets terms and formulas over $\mathscr{L}_{\text{aff}}$ as functions and sets, respectively: let $t_1(x_1,\ldots,x_m),\ldots,t_n(x_1,\ldots,x_m)$ be terms over $\mathscr{L}_{\text{aff}}$. We write $\bar{t}^{\mathbb{Q}}\colon \mathbb{Q}^m \to \mathbb{Q}^n$ for the function denoted by the tuple of terms $\bar{t}$. Likewise, if $A(x_1,\ldots,x_n)$ is a formula over $\mathscr{L}_{\text{aff}}$, we write $A^{\mathbb{Q}}$ for the n-ary relation on $\mathbb{Q}$ denoted by A.

We present the following facts about the interpretations of terms and formulas.

PROPOSITION 3.3
Let $A(\bar{x}), B(\bar{x}), A_1(\bar{x}),\ldots,A_m(\bar{x})$ be formulas over $\mathscr{L}_{\text{aff}}$ and $t_1(\bar{x}),\ldots,t_n(\bar{x})$ terms over $\mathscr{L}_{\text{aff}}$.

1. $(\bigvee_{i=1}^m A_i)^{\mathbb{Q}} = \bigcup_{i=1}^m A_i^{\mathbb{Q}}$
2. $(\bigwedge_{i=1}^m A_i)^{\mathbb{Q}} = \bigcap_{i=1}^m A_i^{\mathbb{Q}}$
3. $A[\bar{x}\backslash\bar{t}]^{\mathbb{Q}} = (\bar{t}^{\mathbb{Q}})^{-1}(A^{\mathbb{Q}})$
4. $\text{Th}(\mathbb{Q}) \models A(\bar{x}) \to B(\bar{x})$ iff $A^{\mathbb{Q}} \subseteq B^{\mathbb{Q}}$

Affine geometry (in the context of vector spaces) plays an important role in this paper. We therefore briefly review some important concepts. For a more rigorous treatment, see [21].

An *affine subspace* $\mathscr{A}$ of $\mathbb{Q}^n$ is a subset of $\mathbb{Q}^n$ that can be written as $\mathscr{a} + \vec{\mathscr{A}}$ for a vector $\mathscr{a} \in \mathbb{Q}^n$ and a linear subspace $\vec{\mathscr{A}} \subseteq \mathbb{Q}^n$. The *dimension* of $\mathscr{A}$ is the dimension of $\vec{\mathscr{A}}$.

An affine transformation from $\mathbb{Q}^m$ to $\mathbb{Q}^n$ is a map that can be written as a linear map followed by a translation. The images and preimages of affine subspaces under affine transformations are themselves affine subspaces.

DEFINITION 3.4 (Aff).
Let V be a vector space over $\mathbb{Q}$. Then $\operatorname{Aff} V$ is the set of all subsets of V that are either empty or affine subspaces of V.

$\operatorname{Aff} V$ is a complete lattice: its least element is $\emptyset$, its greatest element is V and, since the intersection of elements of $\operatorname{Aff} V$ is always either an affine space or empty, it is closed under infima. We write the supremum of $\mathscr{A}, \mathscr{B} \in \operatorname{Aff} V$ as $\mathscr{A} \vee \mathscr{B}$. Moreover, if $\dim V = n$, the height of $\operatorname{Aff} V$ is equal to $n + 2$: for affine subspaces $\mathscr{A}, \mathscr{B}$, $\mathscr{A} \subsetneq \mathscr{B}$ implies $\dim \mathscr{A} < \dim \mathscr{B}$. Consequently, for any chain $\emptyset \subsetneq \mathscr{A}_1 \subsetneq \mathscr{A}_2 \subsetneq \ldots$, $\dim \mathscr{A}_1 < \dim \mathscr{A}_2 < \ldots$, is a chain in $[0, n]$ and can therefore be at most of length $n + 1$.

The following proposition shows the connection between formulas and terms of $\mathscr{L}_{\text{aff}}$ and affine spaces and transformations, respectively.

PROPOSITION 3.5

1. Let $A(x_1, \ldots, x_n)$ be a linear equation system. Then $A^{\mathbb{Q}} \in \operatorname{Aff} \mathbb{Q}^n$.
2. Let $t_1(x_1, \ldots, x_m), \ldots, t_n(x_1, \ldots, x_m)$ be affine terms, i.e. $t_i = c_{i,0} + \sum_{j=1}^{m} c_{i,j} x_j$. Then $\bar{t}^{\mathbb{Q}}$ is an affine transformation from $\mathbb{Q}^m$ to $\mathbb{Q}^n$.

We write elements, subspaces and transformations of affine spaces in calligraphic typeface, e.g. $\mathscr{x}, \mathscr{A}, \mathscr{T}$.

DEFINITION 3.6 (Affine condition).
Let $m, n, \ell, r \in \mathbb{N}$. An *affine condition* $\mathscr{C}$ is a statement of the form

$$\mathscr{A} \cap \bigcap_{i=1}^{\ell} \mathscr{T}_i^{-1}(\mathscr{X}_i) \subseteq \bigcup_{i=1}^{m} \mathscr{B}_i \cup \bigcup_{j=\ell+1}^{\ell+r} \mathscr{T}_j^{-1}(\mathscr{X}_j),$$

where $\mathscr{A}, \mathscr{B}_1, \ldots, \mathscr{B}_m \in \operatorname{Aff} \mathbb{Q}^n$ and for $i = 1, \ldots, \ell + r$, $\mathscr{T}_i : \mathbb{Q}^n \to \mathbb{Q}^{k_i}$ is an affine transformation and $\mathscr{X}_i$ is a variable ranging over $\operatorname{Aff} \mathbb{Q}^{k_i}$. The tuple $\langle n; k_1, \ldots, k_{\ell+r} \rangle$ is called the dimensions of $\mathscr{C}$.

A tuple $\bar{\mathscr{F}} \in \operatorname{Aff} \mathbb{Q}^{k_1} \times \ldots \times \operatorname{Aff} \mathbb{Q}^{k_{l+r}}$ is a *solution* of $\mathscr{C}$ if $\mathscr{C}[\bar{\mathscr{X}} \backslash \bar{\mathscr{F}}]$ is true.

DEFINITION 3.7 (Affine conditions of an affine formula equation).
Let φ be an affine formula equation and

$$C \equiv A(\bar{x}), X_1(\bar{t}_1(\bar{x})), \ldots, X_\ell(\bar{t}_\ell(\bar{x}))$$
$$\vdash$$
$$B_1(\bar{x}), \ldots, B_m(\bar{x}), X_{\ell+1}(\bar{t}_{\ell+1}(\bar{x})), \ldots, X_{\ell+r}(\bar{t}_{\ell+r}(\bar{x})),$$

a clause with dimensions $\langle n; k_1, \ldots, k_{\ell+r} \rangle$ in $\operatorname{Cl}(\varphi)$. Then we call

$$\mathscr{C} :\equiv A^{\mathbb{Q}} \cap \bigcap_{i=1}^{\ell} \left(\bar{t}_i^{\mathbb{Q}}\right)^{-1}(\mathscr{X}_i) \subseteq \bigcup_{i=1}^{m} B_i^{\mathbb{Q}} \cup \bigcup_{j=\ell+1}^{\ell+r} \left(\bar{t}_j^{\mathbb{Q}}\right)^{-1}(\mathscr{X}_j)$$

the *affine condition* corresponding to C. The dimensions of $\mathscr{C}$ are equal to those of C.

We say that $\mathscr{C}$ is an affine condition of φ iff it corresponds to some clause in $\mathrm{Cl}(\varphi)$. We write $\mathrm{AC}(\varphi)$ for the set of affine conditions of φ.

EXAMPLE 3.8
Consider the affine formula equation φ from Example 3.2. We obtain the affine conditions of φ by translating each of the clauses:

$$\mathbb{Q}^2 \subseteq \mathscr{T}_1^{-1}(\mathscr{X}),$$
$$\mathscr{S}^{-1}(\mathscr{X}) \subseteq \mathscr{T}_2^{-1}(\mathscr{X}) \cup \mathscr{T}_3^{-1}(\mathscr{X}),$$
$$\mathscr{X} \subseteq \mathscr{B}_1 \cup \mathscr{B}_2,$$

where $\mathscr{X}$ is a variable ranging over $\mathrm{Aff}\,\mathbb{Q}^2$ and

$$\mathscr{T}_1 : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \mathscr{B}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} \right]$$
$$\mathscr{T}_2 : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} -y \\ x \end{pmatrix} \qquad \mathscr{B}_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} \right]$$
$$\mathscr{T}_3 : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ -y \end{pmatrix} \qquad \mathscr{S} : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} -x \\ y \end{pmatrix}$$

$\mathscr{B}_1$ and $\mathscr{B}_2$ are straightforwardly obtained as solution spaces of the linear equations $x = y$ and $y = 0$, respectively.

The following theorem shows that the translation from clauses to affine conditions is justified, because solutions of one translate to solutions of the other.

THEOREM 3.9
Let φ be an affine formula equation, C a clause of φ containing the formula variables $\bar{X}$, $\mathscr{C}$ the corresponding affine condition and $\bar{F}$ a tuple of linear equation systems of appropriate arities. Then $\mathbb{Q} \models C[\bar{X} \backslash \bar{F}]$ iff $\bar{F}^{\mathbb{Q}}$ is a solution of $\mathscr{C}$.

PROOF. Let W, Z be such that $C[\bar{X} \backslash \bar{F}]$ is syntactically equal to $W \vdash Z$ and $\mathscr{W}, \mathscr{Z}$ such that $\mathscr{C}[\bar{\mathscr{X}} \backslash \bar{F}^{\mathbb{Q}}]$ is syntactically equal to $\mathscr{W} \subseteq \mathscr{Z}$. Then $\mathscr{W} = W^{\mathbb{Q}}$ and $\mathscr{Z} = Z^{\mathbb{Q}}$ by Proposition 3.3 1–3. Moreover, by Proposition 3.3 (4), $\mathscr{W} \subseteq \mathscr{Z}$ iff $\mathbb{Q} \models W \to Z$, from which the claim obtains immediately. □

COROLLARY 3.10
Let φ be an affine formula equation. A tuple $\bar{F}$ of linear equation systems is a solution of φ iff $\bar{F}^{\mathbb{Q}}$ is a solution of $\mathrm{AC}(\varphi)$.

3.3 Projections

DEFINITION 3.11 (Affine Horn condition).
We call affine conditions of the form

$$\mathscr{A} \cap \bigcap_{i=1}^{\ell} \mathscr{S}_i^{-1}(\mathscr{X}_i) \subseteq \mathscr{B} \quad \text{or} \quad \mathscr{A} \cap \bigcap_{i=1}^{\ell} \mathscr{S}_i^{-1}(\mathscr{X}_i) \subseteq \mathscr{S}^{-1}(\mathscr{Y})$$

affine Horn conditions.

DEFINITION 3.12 (Projections).

1. Let
$$\mathscr{C} : \mathscr{A} \cap \bigcap_{i=1}^{\ell} \mathscr{T}_i^{-1}(\mathscr{X}_i) \subseteq \bigcup_{i=1}^{m} \mathscr{B}_i \cup \bigcup_{j=\ell+1}^{\ell+r} \mathscr{T}_j^{-1}(\mathscr{X}_j)$$
be an affine condition. We call the affine Horn conditions
$$\mathscr{A} \cap \bigcap_{i=1}^{\ell} \mathscr{T}_i^{-1}(\mathscr{X}_i) \subseteq \mathscr{B}_{i_0}, i_0 \in \{1, \ldots, m\},$$
$$\mathscr{A} \cap \bigcap_{i=1}^{\ell} \mathscr{T}_i^{-1}(\mathscr{X}_i) \subseteq \mathscr{T}_{j_0}^{-1}(\mathscr{X}_{j_0}), j_0 \in \{\ell+1, \ldots, \ell+r\}$$
projections of $\mathscr{C}$.
2. Let S be a set of affine conditions. We call a set of affine Horn conditions a projection of S if it consists of exactly one projection of each element of S. We write $\Pr(S)$ for the set of projections of S.
3. Let φ be an affine formula equation. We abbreviate $\Pr(\mathrm{AC}(\varphi))$ as $\Pr(\varphi)$ and call its elements *affine projections* of φ.

EXAMPLE 3.13
Let $\mathrm{AC}(\varphi)$ be the affine conditions of φ from Example 3.8. Since the latter two affine conditions each have 2 projections, there are four affine projections of φ:

$$\begin{aligned} \mathscr{T}_1(\mathbb{Q}^2) &\subseteq \mathscr{X} \\ \mathscr{T}_2(\mathscr{S}^{-1}(\mathscr{X})) &\subseteq \mathscr{X} \\ \mathscr{X} &\subseteq \mathscr{B}_1 \end{aligned} \tag{P1}$$

$$\begin{aligned} \mathscr{T}_1(\mathbb{Q}^2) &\subseteq \mathscr{X} \\ \mathscr{T}_2(\mathscr{S}^{-1}(\mathscr{X})) &\subseteq \mathscr{X} \\ \mathscr{X} &\subseteq \mathscr{B}_2 \end{aligned} \tag{P2}$$

$$\begin{aligned} \mathscr{T}_1(\mathbb{Q}^2) &\subseteq \mathscr{X} \\ \mathscr{T}_3(\mathscr{S}^{-1}(\mathscr{X})) &\subseteq \mathscr{X} \\ \mathscr{X} &\subseteq \mathscr{B}_1 \end{aligned} \tag{P3}$$

$$\begin{aligned} \mathscr{T}_1(\mathbb{Q}^2) &\subseteq \mathscr{X} \\ \mathscr{T}_3(\mathscr{S}^{-1}(\mathscr{X})) &\subseteq \mathscr{X} \\ \mathscr{X} &\subseteq \mathscr{B}_2 \end{aligned} \tag{P4}$$

The following lemma lets us make the step from affine conditions to projections. Its proof is elementary.

LEMMA 3.14
Let V be a vector space over $\mathbb{Q}$ and let $\mathscr{A}_1, \ldots, \mathscr{A}_n$ be proper affine subspaces of V. Then $\bigcup_{i=1}^{n} \mathscr{A}_i \subsetneq V$.

COROLLARY 3.15
Let V be a vector space over $\mathbb{Q}$ and $\mathscr{A}_1, \ldots, \mathscr{A}_n, \mathscr{B}$ affine subspaces of V. If $\mathscr{B} \subseteq \bigcup_{i=1}^{n} \mathscr{A}_i$, then $\mathscr{B} \subseteq \mathscr{A}_{i_0}$ for some i_0.

PROOF. Apply Lemma 3.14 to $\mathscr{B}$ and $\mathscr{A}_1 \cap \mathscr{B}, \ldots, \mathscr{A}_n \cap \mathscr{B}$. □

THEOREM 3.16
Let $\mathscr{C}$ be an affine condition with dimensions $\langle n; k_1, \ldots, k_{\ell+r} \rangle$ and $\bar{\mathscr{F}} \in \operatorname{Aff}\mathbb{Q}^{k_1} \times \ldots \times \operatorname{Aff}\mathbb{Q}^{k_\ell+r}$. Then $\bar{\mathscr{F}}$ is a solution of $\mathscr{C}$ iff it is a solution of some projection of $\mathscr{C}$.

PROOF. Let $\mathscr{W}, \mathscr{Z}_1, \ldots, \mathscr{Z}_m$ be such that $\mathscr{C}[\bar{\mathscr{X}} \backslash \bar{\mathscr{F}}] = \mathscr{W} \subseteq \bigcup_{i=1}^{m} \mathscr{Z}_i$. If $\mathscr{F}$ satisfies $\mathscr{C}$, then $\mathscr{W} \subseteq \bigcup_{i=1}^{m} \mathscr{Z}_i$. By Corollary 3.15, this implies that $\mathscr{W} \subseteq \mathscr{Z}_{i_0}$ for some i_0; hence, $\bar{\mathscr{F}}$ satisfies a projection of $\mathscr{C}$. The other direction is obvious. □

COROLLARY 3.17
Let S be a set of affine conditions with dimensions $\langle n; k_1, \ldots, k_{\ell+r} \rangle$ and $\bar{\mathscr{F}} \in \operatorname{Aff}\mathbb{Q}^{k_1} \times \ldots \times \operatorname{Aff}\mathbb{Q}^{k_\ell+r}$. Then $\bar{\mathscr{F}}$ is a solution of $\mathcal{S}$ iff it is a solution of some element of $\Pr(S)$.

Theorem 3.16 shows that the solvability of affine formula equations reduces to the solvability of sets of affine Horn conditions. In the next sections, we will show that the latter question is decidable.

3.4 Finding a fixed point by iteration

DEFINITION 3.18 (Upper and lower bound conditions).
Let ψ be a set of affine Horn conditions. As per Definition 3.11, ψ contains elements of the forms

$$\mathscr{A} \cap \bigcap_{i=1}^{m} \mathcal{S}_i^{-1}(\mathscr{X}_i) \subseteq \mathscr{B}$$

and

$$\mathscr{A} \cap \bigcap_{i=1}^{m} \mathcal{S}_i^{-1}(\mathscr{X}_i) \subseteq \mathcal{T}^{-1}(\mathscr{Y}) \equiv \mathcal{T}\left(\mathscr{A} \cap \bigcap_{i=1}^{m} \mathcal{S}_i^{-1}(\mathscr{X}_i)\right) \subseteq \mathscr{Y}.$$

We call the former *upper bound* conditions and the latter *lower bound* conditions of ψ.

Let ψ be a set of affine Horn conditions with dimensions $\langle k_0; k_1, \ldots, k_n \rangle$. Recall that this means that the unknown $\mathscr{X}_i$ ranges over $\operatorname{Aff}\mathbb{Q}^{k_i}$. It follows that the candidate solutions of ψ are elements of $\operatorname{Aff}\mathbb{Q}^{k_1} \times \ldots \times \operatorname{Aff}\mathbb{Q}^{k_n}$.

DEFINITION 3.19 (Lat(ψ)).
Let ψ be a set of affine Horn conditions with dimensions $\langle k_0; k_1, \ldots, k_n \rangle$. We write $\operatorname{Lat}(\psi)$ for the lattice $\operatorname{Aff}\mathbb{Q}^{k_1} \times \ldots \times \operatorname{Aff}\mathbb{Q}^{k_n}$ of candidate solutions of ψ.

The order on $\mathrm{Lat}(\psi)$ is defined by

$$(\mathscr{U}_1, \ldots, \mathscr{U}_n) \leq (\mathscr{V}_1, \ldots, \mathscr{V}_n) \text{ if } \mathscr{U}_1 \subseteq \mathscr{V}_1 \wedge \ldots \wedge \mathscr{U}_n \subseteq \mathscr{V}_n.$$

Its least element is $0 := (\emptyset, \ldots, \emptyset)$.

As the finite product of lattices of finite height, $\mathrm{Lat}(\psi)$ has finite height as well.

DEFINITION 3.20 (Φ).
Let ψ be a set of affine Horn conditions. Let $\mathscr{X}_1, \ldots, \mathscr{X}_n$ be the variables in ψ and $k_1, \ldots, k_n$ their respective dimensions. We define an operator Φ_ψ on $\mathrm{Lat}(\psi)$ in the following manner: let $1 \leq \alpha \leq n$ and

$$\mathscr{T}_1\left(\mathscr{A}_1 \cap \bigcap_{i=1}^{m_1} \mathscr{S}_{1,i}^{-1}(\mathscr{X}_{j_{1,i}})\right) \subseteq \mathscr{X}_\alpha$$

$$\vdots$$

$$\mathscr{T}_\ell\left(\mathscr{A}_\ell \cap \bigcap_{i=1}^{m_\ell} \mathscr{S}_{\ell,i}^{-1}(\mathscr{X}_{j_{\ell,i}})\right) \subseteq \mathscr{X}_\alpha$$

be the lower bound conditions in ψ whose right-hand side is $\mathscr{X}_\alpha$. Then

$$\Phi_\psi(\mathscr{F})_\alpha := \mathscr{F}_\alpha \vee \bigvee_{i'=1}^{\ell} \mathscr{T}_{i'}\left(\mathscr{A}_{i'} \cap \bigcap_{i=1}^{m_{i'}} \mathscr{S}_{i',i}^{-1}(\mathscr{F}_{j_{i',i}})\right).$$

It is immediate from the definition that Φ_ψ is monotone. Φ_ψ can be viewed as a procedure for improving an approximate solution of the lower bound conditions in ψ. Indeed, the fixed points of Φ_ψ are precisely the solutions of the lower bound conditions.

LEMMA 3.21
Let ψ be a set of affine Horn conditions with dimensions $\langle m; k_1, \ldots, k_n\rangle$. Then a tuple $\bar{\mathscr{F}} \in \mathrm{Lat}(\psi)$ solves ψ iff $\Phi_\psi(\bar{\mathscr{F}}) = \bar{\mathscr{F}}$.

PROOF. Assume $\bar{\mathscr{F}}$ solves all lower bound conditions in ψ. Let

$$\mathscr{T}_1\left(\mathscr{A}_1 \cap \bigcap_{i=1}^{m_1} \mathscr{S}_{1,i}^{-1}(\mathscr{X}_{j_{1,i}})\right) \subseteq \mathscr{X}_\alpha$$

$$\vdots$$

$$\mathscr{T}_\ell\left(\mathscr{A}_\ell \cap \bigcap_{i=1}^{m_\ell} \mathscr{S}_{\ell,i}^{-1}(\mathscr{X}_{j_{\ell,i}})\right) \subseteq \mathscr{X}_\alpha$$

be the lower bound conditions in ψ whose right-hand side is $\mathcal{X}_\alpha$. Then by assumption,

$$\mathcal{T}_1\left(\mathcal{A}_1 \cap \bigcap_{i=1}^{m_1} \mathcal{S}_{1,i}^{-1}(\mathcal{F}_{j_{1,i}})\right) \subseteq \mathcal{F}_\alpha$$

$$\vdots$$

$$\mathcal{T}_\ell\left(\mathcal{A}_\ell \cap \bigcap_{i=1}^{m_\ell} \mathcal{S}_{\ell,i}^{-1}(\mathcal{F}_{j_{\ell,i}})\right) \subseteq \mathcal{F}_\alpha$$

are all true. This implies that

$$\Phi_\psi(\mathcal{F})_\alpha = \mathcal{F}_\alpha \vee \underbrace{\bigvee_{i'=1}^{\ell} \mathcal{T}_{i'}(\mathcal{A}_{i'} \cap \bigcap_{i=1}^{m_{i'}} \mathcal{S}_{i',i}^{-1}(\mathcal{F}_{j_{i',i}}))}_{\subseteq \mathcal{F}_\alpha} = \mathcal{F}_\alpha,$$

so $\bar{\mathcal{F}}$ is a fixed point of Φ_ψ.

Conversely, assume that $\bar{\mathcal{F}}$ is a fixed point of Φ_ψ, i.e. for every j,

$$\mathcal{F}_\alpha \vee \bigvee_{i'=1}^{\ell} \mathcal{T}_{i'}\left(\mathcal{A}_{i'} \cap \bigcap_{i=1}^{m_{i'}} \mathcal{S}_{i',i}^{-1}(\mathcal{F}_{j_{i',i}})\right) = \Phi_\psi(\mathcal{F})_\alpha = \mathcal{F}_\alpha.$$

This implies that $\bigcup_{i'=1}^{\ell} \mathcal{T}_{i'}\left(\mathcal{A}_{i'} \cap \bigcap_{i=1}^{m_{i'}} \mathcal{S}_{i',i}^{-1}(\mathcal{F}_{j_{i',i}})\right) \subseteq \mathcal{F}_\alpha$, so $\bar{\mathcal{F}}$ solves all lower bound conditions whose right-hand side is $\mathcal{X}_\alpha$. Since this works for all α, $\bar{\mathcal{F}}$ solves all lower bound conditions. □

THEOREM 3.22 (Kleene's fixed point theorem).
Let L be a lattice with least element 0 in which all chains have suprema. Let $f: L \to L$ be continuous on chains, i.e. for any chain M, $f(\sup M) = \sup f(M)$. Then f has a least fixed point m and $m = \sup\left\{f^i(0) \;\middle|\; i \in \mathbb{N}\right\}$.

DEFINITION 3.23 ($\bar{\mathcal{F}}^\psi$).
Let ψ be a set of affine Horn conditions with dimensions $\langle k_0; k_1, \ldots, k_n\rangle$. Then $\bar{\mathcal{F}}^\psi \in \mathrm{Lat}(\psi)$ denotes the least fixed point of Φ_ψ.

Definition 3.23 is justified because $\mathrm{Lat}(\psi)$ and Φ_ψ satisfy the prerequisites of Theorem 3.22: since $\mathrm{Lat}(\psi)$ has finite height, every chain must have a supremum (in fact, a maximum). From the monotonicity of f, we obtain $\Phi_\psi(\max\{\bar{\mathcal{F}}_1 < \ldots < \bar{\mathcal{F}}_m\}) = \Phi_\psi(\bar{\mathcal{F}}_m) = \max \Phi_\psi(\{\bar{\mathcal{F}}_1 < \ldots < \bar{\mathcal{F}}_m\})$. Moreover, there is some $k \in \mathbb{N}$ such that $\bar{\mathcal{F}}^\psi = \Phi_\psi^k(0)$.

THEOREM 3.24
Let ψ be a set of affine Horn conditions. If there is a solution of ψ, then $\bar{\mathcal{F}}^\psi$ is its least solution. Conversely, if $\bar{\mathcal{F}}^\psi$ is not a solution of ψ, then ψ has no solution.

PROOF. Let $\bar{\mathcal{G}}$ be a solution of ψ. In particular, $\bar{\mathcal{G}}$ must satisfy all lower bound conditions of ψ, so $\bar{\mathcal{F}}^\psi \leq \bar{\mathcal{G}}$ by Lemma 3.21. As a solution, $\bar{\mathcal{G}}$ must also satisfy all upper bound conditions, and it follows that $\bar{\mathcal{F}}^\psi$ does as well. □

THEOREM 3.25
The function $\psi \longmapsto \bar{\mathcal{F}}^{\psi}$ is computable. Moreover, it is decidable whether $\bar{\mathcal{F}}^{\psi}$ solves ψ.

PROOF. Computability of $\psi \longmapsto \bar{\mathcal{F}}^{\psi}$: let ψ be a set of affine Horn conditions. As remarked previously, there is a $k \in \mathbb{N}$ such that $\bar{\mathcal{F}}^{\psi} = \Phi_{\psi}^{k}(0)$. Thus, we have to establish that we can compute Φ_{ψ}. The proof hinges on the fact that given finite representations of affine spaces $\mathcal{A}, \mathcal{B}$ and an affine transformation $\mathcal{T}$ it is possible to compute finite representations of $\mathcal{A} \cap \mathcal{B}, \mathcal{A} \vee \mathcal{B}, \mathcal{T}(\mathcal{A})$ and $\mathcal{T}^{-1}(\mathcal{A})$. This is an elementary result of affine geometry and linear algebra, cf. [21].

Decidability: by Theorem 3.24, it is sufficient to check whether $\bar{\mathcal{F}}^{\psi}$ is a solution of all upper bound conditions in ψ. To do this, we need to decide whether an affine subspace is contained within another. The fact that this can be done is, again, a basic result. □

We are now able to define the algorithm MINIMALSOLUTION and prove it is correct.

Algorithm 1 MinimalSolution

Require: A set S of affine conditions
Ensure: An element of $\mathrm{Lat}\, S$ or **failure** .

```
1:  Sol := Ø
2:  for ψ ∈ Pr(S) do
3:      compute F̄^ψ
4:      if F̄^ψ satisfies the upper bound conditions in ψ
5:          add F̄^ψ to Sol
6:      end if
7:  end for
8:  if Sol ≠ Ø then
9:      return a minimal element of Sol
10: else
11:     return failure
12: end if
```

THEOREM 3.26
Let S be a set of affine conditions. If S is solvable, then MINIMALSOLUTION(S) is a minimal solution of S. If S is unsolvable, then MINIMALSOLUTION(S) = **failure**.

PROOF. By Theorems 3.24 and 3.25, after the end of the **for** loop in line 7, Sol contains exactly the minimal solutions of the solvable $\psi \in \mathrm{Pr}(S)$. By Corollary 3.17, Sol is nonempty iff S is solvable, and if Sol $\neq \emptyset$, then each of its elements is a solution of S. It immediately follows that MINIMALSOLUTION outputs a solution of S if S is solvable and **failure** otherwise. It only remains to show that in the positive case, the returned solution is minimal.

Assume Sol $\neq \emptyset$ and let $\bar{\mathcal{F}} \in$ Sol be minimal. If $\bar{\mathcal{G}}$ is any solution of S, then $\bar{\mathcal{G}}$ is a solution of some projection of S (again by Corollary 3.17) and hence $\bar{\mathcal{G}}$ is above some element of Sol. Since $\bar{\mathcal{F}}$ is minimal in Sol, this shows that $\bar{\mathcal{G}} \not< \bar{\mathcal{F}}$. So $\bar{\mathcal{F}}$ is in fact a minimal solution of S. □

We are now able to prove the following.

THEOREM 3.27 (Main result).
The affine solution problem is decidable.

PROOF. Let φ be an affine formula equation. By Corollary 3.10, we have that φ is solvable iff $\mathrm{AC}(\varphi)$ is solvable. The latter is decidable by Theorem 3.26, provided that we can supply a coordinate system of each subspace occurring in the affine conditions of φ. Every such subspace $\mathscr{A}$ is the interpretation of some linear equation system A in φ, so solving A results in a basis of $\mathscr{A}$. □

EXAMPLE 3.28
Let P1, P2, P3 and P4 be the affine projections defined in Example 3.13. Each projection Pi induces an operator Φ_i on $\mathrm{Aff}\,\mathbb{Q}^3$:

$$\Phi_{1,2}\colon \mathscr{F} \mapsto \mathscr{F} \vee \mathscr{T}_1(\mathbb{Q}^2) \vee \mathscr{T}_2(\mathscr{S}^{-1}(\mathscr{F}))$$

$$\Phi_{3,4}\colon \mathscr{F} \mapsto \mathscr{F} \vee \mathscr{T}_1(\mathbb{Q}^2) \vee \mathscr{T}_3(\mathscr{S}^{-1}(\mathscr{F})).$$

P1 and P2 induce the same operator because they contain the same lower bound conditions, and so do P3 and P4.

Iterating $\Phi_{1,2}$ on $\emptyset \in \mathrm{Aff}\,\mathbb{Q}^2$ results in the sequence

$$\emptyset, \left\{\begin{pmatrix}1\\0\end{pmatrix}\right\}, \begin{pmatrix}1\\0\end{pmatrix} + \left[\begin{pmatrix}1\\1\end{pmatrix}\right] =: \mathscr{F}_1^*.$$

Since $\begin{pmatrix}1\\0\end{pmatrix} \notin \mathscr{B}_1$ and $\begin{pmatrix}1\\1\end{pmatrix} \notin \vec{\mathscr{B}}_2$, $\mathscr{F}_1^*$ solves neither upper bound conditions, so P1 and P2 are unsolvable.

On the other hand, $\Phi_{3,4}$ results in the iteration sequence

$$\emptyset, \left\{\begin{pmatrix}1\\0\end{pmatrix}\right\}, \begin{pmatrix}1\\0\end{pmatrix} + \left[\begin{pmatrix}1\\0\end{pmatrix}\right] =: \mathscr{F}_2^*.$$

$\mathscr{F}_2^* \subseteq \mathscr{B}_1$ is false and $\mathscr{F}_2^* \subseteq \mathscr{B}_2$ is true, so P3 is unsolvable and P4 has the solution $\mathscr{F}_2^*$.

3.5 Backwards translation

Let φ be an affine formula equation. In the previous sections, we have shown that it is decidable whether φ has a solution in linear equation systems. In fact, we can effectively compute a maximally strong solution $F_1, \ldots, F_n$ of φ, in the following sense: if $G_1, \ldots, G_n$ is another solution of φ such that for all $i = 1, \ldots, n$, $G_i \to F_i$, then for all $i = 1, \ldots, n$, $G_i \leftrightarrow F_i$. First, we show that any solution of $\mathrm{AC}(\varphi)$ translates back to a solution of φ.

Let $\bar{\mathscr{F}}$ be a solution of $\mathrm{AC}(\varphi)$. For each $\mathscr{F}_i \subseteq \mathbb{Q}^{m_i}$, we need a linear equation system F_i such that $\mathscr{F}_i = F_i^{\mathbb{Q}}$. If $\mathscr{F}_i = \emptyset$, then we let $F_i \equiv \bot$. Otherwise, let $\langle a_0; a_1, \ldots, a_{k_i}\rangle$ be a coordinate system of $\mathscr{F}_i$. We obtain $m_i - k_i + 1$ independent linear equations by solving the linear equation system

$$\begin{pmatrix} 1 & a_0^T \\ 0 & a_1^T \\ \vdots & \vdots \\ 0 & a_{k_i}^T \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_{m_i} \end{pmatrix} = \bar{0}.$$

By Corollary 3.10, $\bar{F}$ is a solution of φ.

Note that if $\mathcal{F} \subseteq \mathcal{G}$, then $F \to G$. In other words, smaller spaces translate to stronger formulas. Since we can compute a minimal solution of $\mathrm{AC}(\varphi)$ by Theorem 3.26, we also obtain a maximally strong solution of φ.

EXAMPLE 3.29
Let $\mathcal{F}_2^*$ be the solution of the projection P4 computed in Example 3.28. We can translate $\mathcal{F}_2^*$ back to the linear equation system $F(x, y) \equiv y = 0$. This gives us the solved formula equation $\varphi[X \backslash F]$:

$$0 = 0 \wedge (y = 0 \to x = 0 \vee -y = 0) \wedge (y = 0 \to x = y \vee y = 0).$$

4 Conclusion and future work

On the logical level, formula equations capture classic problems like the *Auflösungsproblem*. On the level of formal verification, they are a significant generalization of the established concept of constrained Horn clauses. We have shown that a specific loop invariant generation method that operates on the equivalent of linear constrained Horn clauses can be generalized to arbitrary quantifier-free formula equations. To this end, we have exploited specific properties of linear subspaces of $\mathbb{Q}^k$.

The two most obvious avenues for generalizing our result are nonlinearity and inequalities. Extending our language with inequalities means that the sets under investigation are polyhedra, and consequently one cannot rely on a disjunction property like the one expressed in Corollary 3.15 for affine spaces. This raises the question of how one would deal with non-Horn clauses.

One does not have to contend with this problem in the case of nonlinearity: the solution sets of polynomial equalities are already closed under disjunction, due to the fact that $p(\bar{x}) = 0 \vee q(\bar{x}) = 0$ iff $pq(\bar{x}) = 0$. Therefore, it may be possible to extend the result of [12] with relatively little difficulty.

We can already say that in the presence of both nonlinearity and inequalities, the solution problem becomes undecidable. This follows from [18]. Moreover, [7] implies that allowing arbitrary first-order quantifiers and adding a single unary predicate to the language results in undecidability as well.

Funding

This work was supported by the Vienna Science and Technology Fund (WWTF) as part of the Vienna Research Group 12-004.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] W. Ackermann. Untersuchungen über das Eliminationsproblem der mathematischen Logik. *Mathematische Annalen*, **110**, 390–413, 1935.
[2] F. Baader. On the complexity of Boolean unification. In *LTCS-Report LTCS-97-03*. LuFg Theoretical Computer Science, RWTH Aachen, Germany, 1997.
[3] N. Bjørner, A. Gurfinkel, K. McMillan and A. Rybalchenko. *Horn Clause Solvers for Program Verification*, pp. 24–51. Springer International Publishing, Cham, 2015.

[4] W. Büttner and H. Simonis. Embedding Boolean expressions into logic programming. *Journal of Symbolic Computation*, **4**, 191–205, 1987.
[5] S. Eberhard and S. Hetzl. Inductive theorem proving based on tree grammars. *Annals of Pure and Applied Logic*, **166**, 665–700, 2015.
[6] S. Eberhard, S. Hetzl and D. Weller. Boolean unification with predicates. *Journal of Logic and Computation*, **27**, 109–128, 2017.
[7] S. Garfunkel and J. H. Schmerl. The undecidability of theories of groupoids with an extra predicate. *Proceedings of the American Mathematical Society*, **42**, 286–289, 1974.
[8] S. Hetzl, A. Leitsch, G. Reis, J. Tapolczai and D. Weller. Introducing quantified cuts in logic with equality. In *Automated Reasoning—7th International Joint Conference, IJCAR*, S. Demri, D. Kapur and C. Weidenbach, eds, pp. 240–254. Vol. 8562 of Lecture Notes in Computer Science. Springer, 2014.
[9] S. Hetzl, A. Leitsch, G. Reis and D. Weller. Algorithmic introduction of quantified cuts. *Theoretical Computer Science*, **549**, 1–16, 2014.
[10] S. Hetzl and T. L. Wong. Some observations on the logical foundations of inductive theorem proving. *Logical Methods in Computer Science*, **13**, 2018.
[11] W. Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997.
[12] E. Hrushovski, J. Ouaknine, A. Pouly and J. Worrell. Polynomial invariants for affine programs. In *33rd Annual Symposium on Logic in Computer Science (LICS)*, A. Dawar and E. Grädel, eds, pp. 530–539. ACM, 2018.
[13] P. C. Kanellakis, G. M. Kuper and P. Z. Revesz. Constraint query languages. In *Proceedings of the Ninth Symposium on Principles of Database Systems (PODS)*, D. J. Rosenkrantz and Y. Sagiv, eds, pp. 299–313. ACM Press, 1990.
[14] P. C. Kanellakis, G. M. Kuper and P. Z. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, **51**, 26–52, 1995.
[15] M. Karr. Affine relationships among variables of a program. *Acta Informatica*, **6**, 133–151, 1976.
[16] U. Martin and T. Nipkow. Unification in Boolean rings. *Journal of Automated Reasoning*, **4**, 381–396, 1988.
[17] U. Martin and T. Nipkow. Boolean unification—the story so far. *Journal of Symbolic Computation*, **7**, 275–293, 1989.
[18] D. Monniaux. On the decidability of the existence of polyhedral invariants in transition systems. *Acta Informatica*, **56**, 385–389, Jun 2019.
[19] M. Müller-Olm and H. Seidl. Precise interprocedural analysis through linear algebra. In *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '04*, pp. 330–341. ACM, New York, NY, 2004.
[20] E. Schröder. *Vorlesungen über Die Algebra der Logik* . Teubner, vol. 1, 1890; vol. 2, pt. 1, 1891; vol.2, pt. 2, 1905; vol. 3, 1895.
[21] E. Snapper and R. J. Troyer. *Metric Affine Geometry*. Academic Press, 1971.
[22] C. Wernhard. The Boolean solution problem from the perspective of predicate logic. In *11th International Symposium on Frontiers of Combining Systems, FroCoS 2017*, C. Dixon and M. Finger, eds, pp. 333–350. Vol. 10483 of LNCS (LNAI). Springer, 2017.
[23] C. Wernhard. The Boolean solution problem from the perspective of predicate logic—extended version. *Clinical Orthopaedics and Related Research*, abs/1706.08329, 2017.

Received 30 October 2019