Linear Arithmetic with Stars

Ruzica Piskac and Viktor Kuncak

School of Computer and Communication Sciences, EPFL, Switzerland

Abstract. We consider an extension of integer linear arithmetic with a "star" operator takes closure under vector addition of the solution set of a linear arithmetic subformula. We show that the satisfiability problem for this extended language remains in NP (and therefore NP-complete). Our proof uses semilinear set characterization of solutions of integer linear arithmetic formulas, as well as a generalization of a recent result on sparse solutions of integer linear programming problems. As a consequence of our result, we present worst-case optimal decision procedures for two NP-hard problems that were previously not known to be in NP. The first is the satisfiability problem for a logic of sets, multisets (bags), and cardinality constraints, which has applications in verification, interactive theorem proving, and description logics. The second is the reachability problem for a class of transition systems whose transitions increment the state vector by solutions of integer linear arithmetic formulas.

1 Introduction

Decision procedures [5, 1, 15, 10, 7] are among key techniques that enable automated verification of infinite state systems, as, for example, in software model checkers [2, 6, 12]. These techniques are also increasingly used to raise the level of automation in interactive theorem provers [24, 8, 17]. We believe that an important step towards making such theorem provers even more effective is the development of decision procedures for new classes of formulas that go beyond the traditionally considered uninterpreted function symbols, arrays, free data structures, and linear arithmetic. In this paper we present a decision procedure for one such class, which introduces certain *unbounded* sums into linear arithmetic. Specifically, our decision procedure solves the satisfiability problem

$$F_0(\vec{u}) \wedge \exists N \geq 0. \exists \vec{x}_1, \ldots, \vec{x}_N. \vec{u} = \sum_{i=1}^{N} \vec{x}_i \wedge \bigwedge_{i=1}^{N} F(\vec{x}_i) \tag{1}$$

where F_0 and $F(\vec{x})$ are any quantifier-free Presburger arithmetic (QFPA) formulas, all variables of F are among $\vec{x}$, and where $\vec{u}, \vec{x}_i$ are integer vectors. Because N is not known, (1) is not immediately a QFPA formula. Using notation $A^* = \{\vec{u} \mid \exists N \geq 0. \exists \vec{x}_1, \ldots, \vec{x}_N \in A. \vec{u} = \sum_{i=1}^{N} \vec{x}_i\}$ for closure of a set of vectors under addition, we denote (1) by $F_0(\vec{u}) \wedge \vec{u} \in \{\vec{x} \mid F(\vec{x})\}^*$. This paper shows that the satisfiability for the class QFPA* of such formulas with $*$ operator (Figure 1) is in NP, generalizing the well known NP-completeness for QFPA satisfiability.

QFPA^* formulas: $F_0 \wedge \vec{u} \in \{\vec{x} \mid F\}^*$ (free variables of F are among $\vec{x}$)
QFPA formulas:
$F ::= A \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \neg F_1$
$A ::= T_1 \leq T_2 \mid T_1 = T_2$
$T ::= k \mid C \mid T_1 + T_2 \mid C \cdot T_1 \mid \mathsf{ite}(F, T_1, T_2)$
terminals: k - integer variable; C - integer constant

Fig. 1. Quantifer-free Presburger arithmetic and our QFPA^* extension

Previous results. We consider the satisfiability problem for QFPA^* (Figure 1) where variables are interpreted over non-negative integers (the version with arbitrary integers reduces to this case by representing integers and their sums as differences of non-negative integers). The satisfiability problem for QFPA^* is decidable because the set of solutions of a QFPA formula is a *semilinear set* [11], and a closure of a semilinear set under star can be expressed in QFPA [16]. However, the constructions behind these decidability results do not provide good complexity bounds because semilinear set representation can be exponentially large. First algorithm for QFPA^* satisfiability that avoids explicit construction of semilinear set representation is the PSPACE algorithm in [21]. The present paper is the first to establish the exact complexity of QFPA^* satisfiability, namely NP-completeness. To show this result, we will use bounds on solutions of integer linear programming problems with exponentially many variables [20], bounds on seminilinear set generators [22], and Carathéodory bounds for integer cones [9]. Our proof builds on some of the ideas previously introduced in [14, 21].
Application to reasoning about collections. Our decision procedure enables reasoning about collections of objects (sets and multisets) and their cardinalities, which was our original motivation for introducing it in [21]. Previously, Zarba [25] considered decision procedures for multiset constraints but without the cardinality operator, presenting a direct reduction to QFPA. The cardinality operator makes the reduction in [25] inapplicable. Section 2 reviews the approach [21] to reduce multiset constraints with cardinalities to QFPA^*.
Application to reasoning about transition systems. In addition to reasoning about multisets, we identify another application of constraints with stars. We consider infinite-state transitions systems whose state consists of finite control and a finite number of integer counters, and whose transitions increment counters by a solution vector of a linear arithmetic formula given by the finite control. We show that the reachability problem for a class of such systems reduces to a generalization of QFPA^* with multiple nested star operators. We show that our proof techniques and the NP-membership result extend to this more general case, which gives NP-completeness of the reachability problem.
Contributions. We summarize the contributions of our paper as follows:

- We present a polynomial-time algorithm (Section 3) for reducing QFPA^* satisfiability to QFPA satisfiability, showing NP-completeness of QFPA^* satisfiability. This yields an algorithm for reasoning about constraints on multisets in the presence of cardinality operators (see Section 2);

– We generalize our reduction to multiple nested star operators. We show that the generalized constraints enable reasoning about reachability in a class of symbolically represented transition systems (Section 4).

2 From Multisets to Linear Arithmetic with Star

The satisfiability problem for QFPA^* arises as the key step in checking the satisfiability of constraints on multisets in the presence of a cardinality operator. The reduction from multiset constraints to QFPA^* is presented in [21]. We here motivate multiset constraints and use examples to illustrate the reduction.

Uses of set and multiset constraints. Sets and multisets directly arise in verification conditions for proving properties of programs in languages and paradigms such as SETL [23] and Gamma [4, Page 103]. In programming languages such as Java, data abstraction can be used to show that data structures satisfy set specifications, and then techniques based on sets become applicable for verifying data structure clients [13, 18]. Multisets and sets are also present in libraries of interactive provers Isabelle [19] and KIV [3]. Our results yield decision procedures that can increase the automation within such systems. As a simple running example, consider a verification condition for insertion of an element represented by a singleton multiset s into a container represented by multiset L. To prove that an integer field correctly maintains the size of a container, we need to prove validity of the constraint $|s| = 1 \rightarrow |L \uplus s| = |L| + 1$, that is, unsatisfiability of the corresponding negation $|s| = 1 \wedge |L \uplus s| \neq |L| + 1$.

Representation of multisets and sets. We represent sets as well as multisets (*bags*) with their characteristic functions. A multiset m is a function $E \rightarrow \mathbb{N}$, where E is the universe and $\mathbb{N}$ is the set of non-negative integers. The value $m(e)$ is multiplicity (number of occurrences) of element e in multiset m. We assume that the domain E is fixed and finite but of unknown size. We represent sets within our formulas as special multisets m for which $m(e) = 0 \vee m(e) = 1$ for all elements e.

Operations on multisets. We consider a natural class of operations and relations on multisets that are given pointwise by linear arithmetic formulas. For a relation given by QFPA formula $F(x_1, \ldots, x_k)$, we define the corresponding relation on multisets $m_1, \ldots, m_k$ by $\forall e.\, F(m_1(e), \ldots, m_k(e))$. For example, we define subset $m_1 \subseteq m_2$ by $\forall e. m_1(e) \leq m_2(e)$, multiset sum $m_1 = m_2 \uplus m_3$ by $\forall e. m_1(e) = m_2(e) + m_3(e)$, and union $m_1 = m_2 \cup m_3$ by $\forall e. m_1(e) = \max(m_2(e), m_3(e))$. To define max and other operations we use if-then-else operator $\mathsf{ite}(F, t_1, t_2)$ in QFPA, which denotes t_1 when F holds and t_2 otherwise. We define multiset difference $m_1 = m_2 \setminus m_3$ by $\forall e. m_1(e) = \mathsf{ite}(m_2(e) \leq m_3(e), 0, m_2(e) - m_3(e))$. In our example verification condition, we introduce a new multiset variable y such that $y = L \uplus s$ and we express this condition by $\forall e.\, y(e) = L(e) + s(e)$.

Cardinality operator and sums. We also permit the cardinality operator $|m|$ on multisets, given by $|m| = \sum_{e \in E} m(e)$. This operator turns a multiset expression into an integer expression, and we allow arbitrary QFPA operators on cardinalities. In our example verification condition, in addition to $\forall e.\, y(e) =$

$L(e) + s(e)$ we have constraint $|s| = 1$, which becomes $\sum_{e \in E} s(e) = 1$, and the constraint $|y| \neq |L| + 1$, which becomes $(\sum_{e \in E} y(e)) \neq (\sum_{e \in E} L(e)) + 1$.

Without changing expressive power, we generalize the sum notation and introduce expressions of the form $(u_1, \ldots, u_d) = \sum_{e \in E}(m_1(e), \ldots, m_d(e))$ where $u_1, \ldots, u_d$ are integer variables and $m_1, \ldots, m_d$ are multiset variables. [1] By introducing non-negative integer variables u_i for the results of sums and grouping multiple sums into sum of vectors, we reduce any multiset formula to form

$$F_0 \wedge \bigwedge_{i=1}^{c} (\forall e. F_i(\vec{m}(e))) \wedge \vec{u} = \sum_{e \in E} \vec{m}(e) \tag{2}$$

where $F_0, F_1, \ldots, F_c$ are QFPA formulas, $\vec{m}(e)$ denotes $(m_1(e), \ldots, m_d(e))$, and $\vec{u} = (u_1, \ldots, u_d)$. The negation of our example verification condition becomes

$$\begin{array}{l} u_s = 1 \wedge u_y \neq u_L + 1 \wedge \\ (\forall e. y(e) = L(e) + s(e)) \ \wedge \ (u_s, u_y, u_L) = \sum_{e \in E}(s(e), y(e), L(e)) \end{array} \tag{3}$$

Reduction to QFPA*. The satisfiability of the example constraint (3) is equivalent to the satisfiability of the QFPA* constraint

$$u_s = 1 \wedge u_y \neq u_L + 1 \ \wedge \ (u_s, u_y, u_L) \in \{(s', y', L') \mid y' = L' + s'\}^* \tag{4}$$

We prove equisatisfiability of (3) and (4); see [21, Theorem 2] for the analogous proof for an arbitrary formulas on multisets. Suppose that (3) has a solution $E = \{e_1, \ldots, e_N\}$. Because functions $x, y, L : E \to \mathbb{N}$ satisfy $\forall e. y(e) = L(e) + s(e)$, the vectors $(s(e_i), y(e_i), L(e_i))$ for $1 \leq i \leq N$ represent N solutions of QFPA formula $y' = L' + s'$. Consequently, the assignment to u_s, u_y, u_L is a sum of N solutions of $y' = L' + s'$, so (4) is satisfiable with the same values of u_s, u_y, u_L. Conversely, given a solution to (4) we know that u_s, u_y, u_L is a sum of a finite number, say N, of solutions of $y' = L' + s'$, that is, there are N vectors (y'_i, L'_i, s'_i) such that $y'_i = L'_i + s'_i$. We then introduce a distinct element e_i for each of these N solutions, let $E = \{e_1, \ldots, e_N\}$ and let $s(e_i) = s'_i$, $y(e_i) = y'_i$, $L(e_i) = L'_i$. We obtain a solution of (4), as desired. Note that this proof did not depend on the structure of QFPA formulas. In general, we obtain equisatisfiability of (2) and the QFPA* formula $F_0 \wedge \vec{u} \in \{\vec{x} \mid \bigwedge_{i=1}^{c} F_i(\vec{x})\}^*$. Therefore, to check satisfiability of an expressive class of constraints on sets and multisets, we construct (in polynomial time) an equisatisfiable QFPA* formula. We next present an algorithm for checking QFPA* satisfiability.

3 Linear Arithmetic with Star Operator is in NP

We show how to reduce, in polynomial time, QFPA* satisfiability to QFPA satisfiability. This will show that QFPA* satisfiability is in NP. QFPA and therefore

[1] We assume that the summands are multiset variables because we introduce fresh multiset variables for subterms. However, it is easy to see that we can allow arbitrary QFPA terms as summands without changing the expressive power, see [21].

QFPA* subsume propositional logic, their satisfiability is therefore NP-hard, so our results establish NP-completeness of QFPA* satisfiability.

Consider satisfiability of a QFPA* formula $F_0 \wedge \vec{u} \in \{\vec{v} \mid F(\vec{v})\}^*$. Because F is a QFPA formula, its solution set $\{\vec{v} \mid F(\vec{v})\}$ is a semilinear set [11]. Therefore, there exist finitely many *generating vectors* $\vec{a}_i$, $\vec{b}_{ij}$ whose non-negative integer linear combination spans $\{\vec{v} \mid F(\vec{v})\}$. The number of generating vectors can be exponential, so we avoid explicitly constructing them. We instead apply [22] to compute an upper bound on the size of generating vectors. This gives us bounds on coefficients in an *exponentially large* QFPA formula equisatisfiable with (1). We combine the following two constructions to find a *polynomially large* equisatisfiable formula.

1. We apply a small model theorem for QFPA that follows from [20]. Because the exponential QFPA formula has only polynomially many atomic formulas, we obtain a polynomial bound on the number of bits needed for $\vec{u}$ in the smallest solution of (1).
2. We apply twice a theorem on the size of minimal generator of integer cone [9] to prove that only polynomially many vectors suffice to generate $\vec{u}$.

Finally, we show that we can group linear combinations of generating vectors into linear combination of polynomially many variables denoting solution vectors of F. Despite the multiplication of variables, we can express such linear combination as a QFPA formula because coefficients in linear combination are bounded by the bound on $\vec{u}$.

Our proof builds on several non-trivial previous results, but its algorithmic consequences are simple: we can replace (1) with a problem where N is bounded by a polynomial function of F_0 and F, and where sum over solutions of F is replaced by an integer linear combination of solutions of F, with coefficients of the linear combination polynomially bounded. We proceed to describe our construction in more detail, including concrete bounds needed to implement our algorithm.

3.1 Estimating Coefficient Bounds of Disjunctive Form

The results on which we rely are usually expressed for integer linear programming problems, so we compute dimensions and coefficient bounds for integer linear programming problems arising from QFPA formula.

Let F be a QFPA formula. We can convert F into an equivalent disjunction of integer linear programming problems $\bigvee_{i=1}^{l} A_i\vec{x} = \vec{b}_i$. Let m_i be a number of rows in A_i and let n_i be a number of columns in A_i and let a_i be a maximal absolute value of all coefficient occurring in A_i and b_i. For a given F, define $m_F = \max_{i=1}^{l} m_i$, $n_F = \max_{i=1}^{l} n_i$ and $a_F = \max_{i=1}^{l} a_i$.

Lemma 1 (Values of m_F, n_F and a_F). *Let F be a QFPA formula. If a subformula does not occur within any ite expression we say that it has positive polarity if it occurs under an even number of negations and say it has negative polarity if it occurs under an odd number of negations. If a subformula occurs within an ite* *expression we say that it has no polarity. Let g be the number of atomic formula occurences of the form $t_1 = t_2$ that have positive polarity in F, and let h be the number of remaining atomic formulas. Let v be the number of variables in F and a the maximum of absolute values of integer constants. Then $m_F \leq g + h$, $n_F \leq v + h$, and $a_F \leq a + 1$.*

Proof. We can transform $F[\mathsf{ite}(C, t_1, t_2)]$ into a disjunction of $C \wedge F[t_1]$ and $\neg C \wedge F[t_2]$. Repeating this transformation we eliminate all ite expressions and obtain disjuncts whose size is polynomial in the size of F. Let D be one of the disjuncts after such ite elimination. The polarity of all g atomic formulas $t_1 = t_2$ that occur positively in F remains positive in each D. Each of the remaining h atomic formulas becomes of the form $t_1 \leq t_2$, $t_1 = t_2$ or disjunction $t_1 \leq t_2 \vee t_1' \leq t_2'$. In disjunctive normal form of D, each of the h atomic formulas $t_1 \leq t_2$ may require addition of at most one fresh variable to be converted into equality $t_1 + x \leq t_2$. The resulting number of variables is therefore bounded by $v + h$ whereas the total number of atomic formulas is bounded by $g + h$. When transforming $t_1 < t_2$ into $t_1 + 1 \leq t_2$ we change the constants part of $t_2 - t_1$ by one, so $a_F \leq a + 1$. ∎

3.2 Existence and Size of Solution Set Generators

This section describes the solutions of a QFPA formula F using semilinear sets, provides bounds on the norms of vectors that represent these semilinear sets, and uses this characterization to describe the set $\{\vec{v} \mid F(\vec{v})\}^*$. Define vector set addition by $A + B = \{\vec{a} + \vec{b} \mid \vec{a} \in A, \vec{b} \in B\}$.

Definition 1. *Given a finite set $S \subseteq \mathbb{N}^n$ and $\vec{a} \in \mathbb{N}^n$, we define the* linear set *$L(a; S)$ as $\{a\} + S^*$. We call $\vec{a}$ the* base *vector, and call elements of S the* step *vectors. A* semilinear set *is a union of finitely many linear sets.*

If $Z = \cup_{i=1}^{q} L(a_i; S_i)$ is a representation of a semilinear set, we call the base vectors a_i and step vectors b_{ij} the *generators* for semilinear set. [11] showed that the set of solutions of a QFPA formula is a semilinear, so it is given by some finite set of generators. Morover, [22] shows that for formula $F(\vec{v})$ of form $A\vec{v} \leq b$ each generator $\vec{g}$ satisfies $||\vec{g}||_1 \leq (2 + ||A||_{1,\infty} + ||\vec{b}||_\infty)^m$ where A is a $m \times n$ matrix. Combining this result with Lemma 1, we obtain the following Lemma 2:

Lemma 2. *For each QFPA formula F, there exist q base vectors $\vec{a}_i$, $1 \leq i \leq q$, and for each i the corresponding q_i step vectors $\vec{b}_{ij}$ for $1 \leq j \leq q_i$, all with norms bounded by $(2+2(n_F+1)a_F)^{2m_F}$ where n_F, m_F, a_F are from Lemma 1 such that*

$$F(\vec{u}) \quad \Leftrightarrow \quad \exists \nu_{ij}. \bigvee_{i=1}^{q} (\vec{u} = \vec{a}_i + \sum_{j=1}^{q_i} \nu_{ij} \vec{b}_{ij}) \tag{5}$$

We can now express membership in the set $\{\vec{v} \mid F(\vec{v})\}^*$ using QFPA formula, using the following lemma that follows from Lemma 2 and the definition of the star operator.

Lemma 3. *Let F by a* QFPA *formula and $\vec{a}_i$, $\vec{b}_{ij}$ be from Lemma 2. Then $\vec{u} \in \{\vec{v} \mid F(\vec{v})\}^*$ is equivalent to*

$$\exists (\mu_i)_i, (\nu_{ij})_{ij}.\ \vec{u} = \sum_{i=1}^{q} (\mu_i \vec{a}_i + \sum_{j=1}^{q_i} \nu_{ij} \vec{b}_{ij}) \ \wedge\ \bigwedge_{i=1}^{q} (\mu_i = 0 \rightarrow \sum_{j=1}^{q_i} \nu_{ij} = 0) \tag{6}$$

Because the number of $\vec{a}_i$ and $\vec{b}_{ij}$ vectors can be exponential, Lemma 3 shows that QFPA* satisfiability reduces to satisfiability of an exponentially larger QFPA formula. Our goal is to improve the reduction and obtain a polynomial QFPA formula.

3.3 Selecting Polynomially Many Generators

In this section we establish bounds on the number of generators needed to generate any particular solution vector $\vec{u}$: if $\vec{u}$ is a linear combination of generators, then it is also a linear combination of a polynomial subset of generators that form a smaller semilinear set. We prove this fact using a theorem about sparse solutions of integer linear programming problems. Given a set of vectors X and a vector $\vec{b} \in X^*$, the following fact determines the bound on the number of vectors sufficient for representing $\vec{b}$ as a linear combination of vectors from X.

Fact 1 (Theorem 1 (ii) in [9]) *Let $X \subseteq \mathbb{Z}^d$ be a finite set of integer vectors and let $\vec{b} \in X^*$. Then there exists a subset $\tilde{X}$ such that $\vec{b} \in \tilde{X}^*$ and $|\tilde{X}| \leq 2d \log(4dM)$, where $M = \max_{x \in X} ||x||_\infty$.*

Fact 1 has been applied in [14] in order to establishing membership in NP for constraints on sets with cardinality operators. However, in the case of multisets and QFPA* we need to generalize this idea because of dependencies between the base vectors and the corresponding step vectors.

Theorem 1. *Let F be* QFPA *formula and $\vec{a}_i$, $\vec{b}_{ij}$, $\vec{u}$, q, q_i be from Lemma 3. Then there exists sets $I_0, I_1 \subseteq \{1, \ldots, q\}$ and $J \subseteq \cup_{i=1}^{q} \{(i,1), \ldots, (i, q_i)\}$ such that*

$$\exists (\mu_i)_i, (\nu_{ij})_{ij}.\ \vec{u} = \sum_{i \in I_0} (\vec{a}_i + \sum_{(i,j) \in J} \nu'_{ij} \vec{b}_{ij}) + \sum_{i \in I_1} \mu'_i \vec{a}_i \tag{7}$$

and $|I_0| \leq |J| \leq B$, and $|I_1| \leq B$, where $B = 2n_F(\log 4n_F + 2m_F \log(2 + 2(n_F + 1)a_F))$.

Proof. By assumption, $\vec{u} = \sum_{i=1}^{q} (\mu_i \vec{a}_i + \sum_{j=1}^{q_i} \nu_{ij} \vec{b}_{ij})$ and $\bigwedge_{i=1}^{q} (\mu_i = 0 \rightarrow \sum_{j=1}^{q_i} \nu_{ij} = 0)$. Removing zero indices, assume that μ_i and ν_{ij} are strictly positive. Define $\vec{a} = \sum_i \mu_i \vec{a}_i$ and $\vec{b} = \sum_{ij} \nu_{ij} \vec{b}_{ij}$, so $\vec{u} = \vec{a} + \vec{b}$. From $\vec{b} = \sum_i \nu_{ij} \vec{b}_{ij}$ and Fact 1 we conclude that there exists a set J of indices (i, j) and coefficients ν'_{ij} such that $\vec{b} = \sum_{(i,j) \in J} \nu'_{ij} \vec{b}_{ij}$ and $|J| \leq B = 2n_F \log(4n_F M)$ where M is the bound on generators. To satisfy the dependencies between $\vec{b}_{ij}$ and $\vec{a}_i$, let $I_0 = \{i \mid \exists j.(i,j) \in J\}$. Note $|I_0| \leq |J|$. Let $\vec{a}_0 = \sum_{i \in I_0} \vec{a}_i$. Then $\vec{a}_0 + \vec{b}$ is generated by vectors whose indices are I_0 and J. It remains to generate $\vec{a} - \vec{a}_0$. Note that $\vec{a} - \vec{a}_0 = \sum_{i \in I_0} (\mu_i - 1)\vec{a}_i + \sum_{i \in \{1,\ldots,q\} \setminus I_0} \mu_i \vec{a}_i$. Applying once again Fact 1 we conclude that there exists $I_1 \subseteq \{1, \ldots, q\}$ with $|I_1| \leq B$ such that $\vec{a} - \vec{a}_0 = \sum_{i \in I_1} \mu'_i a_i$. Using the bound $M = (2 + 2(n_F + 1)a_F)^{2m_F}$ from Lemma 2 we obtain the desired value of B. ■

3.4 Grouping Generators into Solutions

In previous two sections we have shown that if $\vec{u} \in \{\vec{v} \mid F(\vec{v})\}^*$, then $\vec{u}$ is a particular linear combination of polynomially many generating vectors $\vec{a}_i$, $\vec{b}_{ij}$ that are themselves polynomially bounded. This suggests the idea of guessing polynomially many bounded vectors, checking whether they are generators, and then checking whether $\vec{u}$ is their linear combination. We next show that we can avoid the problem of checking whether a vector is a generator and reduce the problem to checking whether a vector is a solution of F. The way we stated Theorem 1 already suggests this approach.

Lemma 4. *Let F be a QFPA formula and $\vec{u} \in \{\vec{v} \mid F(\vec{v})\}^*$. Then there exist k vectors $\vec{c}_1, \ldots, \vec{c}_k$ for $k \leq 4n_F(\log 4n_F + 2m_F \log(2 + 2(n_F + 1)a_F))$ such that $\wedge_{i=1}^{k} F(\vec{c}_i) \ \wedge u = \sum_{i=1}^{k} \lambda_i \vec{c}_i$ for some non-negative integers λ_i.*

Proof. In Theorem 1 simply note that $\vec{a}_i + \sum_{(i,j) \in J} \nu'_{ij} \vec{b}_{ij}$ are solutions of F and that their number is bounded by B. Similarly, $\vec{a}_i$ are solutions of F and their number is bounded by B. The total number of solutions is bounded by $2B$ where B is from Theorem 1. ■

3.5 NP-Algorithm

Our NP-algorithm for checking satisfiability of QFPA* formula $F_0 \wedge \vec{u} \in \{\vec{v} \mid F(\vec{v})\}^*$ uses previously introduced bounds. First, using Lemma 1 we calculate the values of m_F, n_F and a_F. Using those values and Lemma 4 we estimate an upper bound $k = 4n_F(\log 4n_F + 2m_F \log(2 + 2(n_F + 1)a_F))$ on the number of solution vectors $\vec{x}_i$. We obtain equisatisfiable formula

$$F_0 \wedge \vec{u} = \lambda_1 \vec{x}_1 + \ldots + \lambda_k \vec{x}_k \wedge \bigwedge_{i=1}^{k} F(\vec{x}_i) \tag{8}$$

Note, however, that, although it is polynomial in size, (8) is not a QFPA formula because it contains multiplication of variables $\lambda_i \cdot \vec{x}_i$. We address this problem by showing that the values of λ_i in smallest solutions have a polynomial number of bits, which allows us to express multiplication using bitwise expansion.

3.6 Multiplication by Bounded Bit Vectors

To express terms $\lambda_i \vec{c}_i$ from Lemma 4 as a QFPA term, we show that the smallest solution $\vec{u}$, if exists, is bounded [20]. Suppose that r' is a bound on $\vec{u}$ of formula $F_0 \wedge \vec{u} \in \{\vec{v} \mid F(\vec{v})\}^*$. Because λ_i in formula (8) must be a non-negative integer, $\lambda_i \leq ||\vec{u}||_\infty \leq r'$, so each λ_i is also bounded by r' and can be represented as a bit-vector of size r for $r = \lceil \log r' \rceil$. Let $\lambda_i = \overline{\lambda_{ir} \ldots \lambda_{i1} \lambda_{i0}} = \sum_{j=0}^{r} \lambda_{ij} 2^j$. Then

$$\lambda_i \vec{c}_i = (\sum_{j=0}^{r} \lambda_{ij} 2^j) \vec{c}_i = \sum_{j=0}^{r} 2^j (\lambda_{ij} \vec{c}_i) = \sum_{j=0}^{r} 2^j \mathsf{ite}(\lambda_{ij}, \vec{c}_i, 0) =$$
$$\mathsf{ite}(\lambda_{i0}, \vec{c}_i, 0) + 2(\mathsf{ite}(\lambda_{i1}, \vec{c}_i, 0) + 2(\mathsf{ite}(\lambda_{i2}, \vec{c}_i, 0) + \ldots))$$

It remains to show how to compute the estimate r'.

3.7 Estimating Solution Size Bounds

Theorem 2. *Let F_0 be a QFPA formula. Let $\vec{u} = (u_1, \ldots, u_d)$ denote a d-dimensional vector of variables ranging over non-negative integers. Let F be a QFPA formula which does not share any variable with F_0 and $\vec{u}$. If formula $F_0 \wedge \vec{u} \in \{\vec{v} \mid F(\vec{v})\}^*$ is satisfiable, then there exists a non-negative solution vector $\vec{w}$ for variables $\vec{u}$ such that $||\vec{w}||_\infty \leq r' = n(ma)^{2m+1}$ where n, m and a are defined by*

1. *$m := d + m_{F_0}$*
2. *$n := n_{F_0} + 6d(\log(4d) + 2m_F \log(2 + (n_F + 1)a_F))$*
3. *$a := \max\{a_{F_0}, (2 + 2(n_F + 1)a_F)^{2m_F}\}$*

Proof. We establish a bound on the size of the solution vector using two facts. First, as shown in Lemma 3, the fact that $\vec{w}$ is a solution of $\vec{u} \in \{\vec{v} \mid F(\vec{v})\}^*$ implies that $\vec{w}$ is a linear combination of generators of a semilinear set and can be expressed as

$$\vec{w} = \sum_{i=1}^{q} (\mu_i \vec{a}_i + \sum_{j=1}^{q_i} \nu_{ij} \vec{b}_{ij})$$

If we represent the above condition as form $A\vec{x} = \vec{b}$, the matrix A consists of generators of semilinear set and the negative identity matrix $-I$, while the vector $\vec{x}$ consists of $\vec{u}$ as well parameters μ_i and ν_{ij}. The dimensions of the matrix A are $d \times (n_G + d)$, where n_G is the number of generators. By Theorem 1, $n_G \leq 6d(\log(4d) + 2m_F \log(2 + 2(n_F + 1)a_F))$.

Next, observe that $\vec{w}$ is a component of the solution vector of F_0. This implies that there is a matrix B with dimensions m_{F_0} and n_{F_0} and a vector $\vec{v}$ such that $B\vec{w} = \vec{v}$.

Combining matrices A and B we obtain a new matrix C with $d + m_{F_0}$ rows and the number of columns $n_{F_0} + 6d(\log(4d) + 2m_F \log(2 + (n_F + 1)a_F))$. To establish an upper bound on the maximum of absolute values in C, we use an upper bound on the size of generating vectors in a semilinear set given by

Lemma 2. We obtain the final result by applying to C the theorem on upper bounds of smallest solutions of integer linear programming problems [20]. ■

Putting everything together, the bound k on the number of solutions of F and bounds on λ_i enables us to generate, in polynomial time, a QFPA formula equisatisfiable with the original QFPA* formula.

4 Reachability in a Class of Transition Systems

We next show another application of satisfiability checking for extensions of QFPA with star operators. We consider the reachability problem in systems whose state has finite control and an unbounded integer vector, and whose transitions increase the integer vector by a solution of QFPA formula. We show that for systems that have only one loop the problem reduces to QFPA* satisfiability and is therefore NP-complete. We show that for arbitrary graphs, the problem reduces to a generalization of QFPA* with multiple star operators, denoted QFPA$^{\mathsf{REG}}$. We sketch a proof that NP-completeness for QFPA* satisfiability extends to QFPA$^{\mathsf{REG}}$ satisfiability.

A class of transition systems. Let $\mathcal{F}_d$ be the set of all QFPA formulas with the set of free variables $v_1, \ldots, v_d$. If $F \in \mathcal{F}_d$ is such a formula and $a_1, \ldots, a_d \in \mathbb{Z}$, we write $(a_1, \ldots, a_d) \models F$ to denote that F is true when v_i has value a_i for $1 \leq i \leq d$. We consider transition systems described by a tuple (d, Q, E, T) where 1) d is a non-negative integer, denoting the number of integer variables in the state; 2) Q is a finite set, denoting control-flow graph nodes; 3) $E \subseteq Q \times Q$, denoting control-flow graph edges; and 4) $T : E \to \mathcal{F}_d$, specifies possible increments of counters for each control-flow graph edge. Given (d, Q, E, T) we consider the set of states $S \subseteq Q \times \mathbb{Z}^d$ and define the transition relation $R \subseteq S \times S$ such that $(q, \vec{a}), (q', \vec{a}') \in R \iff (q, q') \in E \ \wedge \ (\vec{a}' - \vec{a}) \models T(q, q')$. We are interested in the question of reachability in the transition systems given by relation R. [2]

Single-loop systems. Consider first the case $Q = \{q\}$, $E = \{(q, q)\}$, $T(q, q) = F$. Our definitions then imply that $(q, \vec{a})$ reaches $(q, \vec{a}')$ precisely when the condition $(\vec{a}' - \vec{a}) \in \{\vec{v} \mid F\}^*$ holds. Therefore, the reachability problems that test QFPA relationship between initial $\vec{a}$ and final $\vec{a}'$ state in such systems reduces QFPA* satisfiability.

General case. Now consider arbitrary (d, Q, E, T) and two states $q, q' \in Q$. Let r be a regular expression over the alphabet E describing the set of all paths from q to q' in graph (Q, E), represented as a set of words over language $Q \times Q$. For example, a path q, q_1, q_2, q' is represented by word $(q, q_1)(q_1, q_2)(q_2, q')$. By conversion algorithm from finite state machines to regular expressions we can assume that r exists and its size is polynomial in the number of elements of Q. We map r into a "commutative" regular expression with set addition acting as commutative version of concatenation and closure under vector addition

[2] Note that, unlike in Turing-complete transition systems with integer counters, the set of possible counter increments is given by formula $T(q, q')$ and does not depend on the current values of integer counters $\vec{a}$, but only on control-flow edge (q, q').

acting as Kleene star. We specify this mapping using function h, defined by: $h((q_1, q_2)) = \{x \mid T(q_1, q_2)\}$, $h(r_1 r_2) = h(r_1) + h(r_2)$, $h(r_1 \cup r_2) = h(r_1) \cup h(r_2)$, $h(r*) = h(r)^*$. Due to commutativity of set addition and its consequence $A^* + B^* = (A \cup B)^*$, we can rewrite r in polynomial time to normal form stratified according to star height (the number of nested applications of $*$ operator). We call $\{v \mid T(q_1, q_2)\}$ atomic expressions and denote them a_{kij}. Then each commutative regular expression of star height $k > 0$ has the form $r_k = \cup_{i=1}^{p} (a_{ki1} + ... + a_{kin_i} + r^*_{i,k-1})$ where $r_{i,k-1}$ are expressions of star height $k-1$. If $r = r_1$ i.e. r has no nested stars, then the reachability problem immediately reduces to (1) and is solvable in NP using our algorithm.

More generally, we consider formulas, denoted $\mathsf{QFPA}^{\mathsf{REG}}$, of the form $F_0 \wedge r$ where r_k is a regular expression over atomic expressions. We show that the satisfiability of $\mathsf{QFPA}^{\mathsf{REG}}$ is in NP. First, the condition $\vec{u} \in r_k$ is equivalent to

$$\exists (\vec{v}_{kij} \in a_{kij})_{kij}. \exists (\lambda_{kij})_{kij}. \vec{x} = \sum_{k,i,j} \lambda_{kij} \vec{v}_{kij} \wedge \bigwedge_{\substack{k>1 \\ i,j}} (\lambda_{kij} = 0 \Rightarrow \bigwedge_{i',j'} \lambda_{(k-1)i'j'} = 0)$$

As in Section 3 our goal is then to show that we can select a polynomial subset of vectors in this linear combination and still generate vector $\vec{u}$. The following notion of "star modulo vector dependencies" captures conditions on coefficients of linear combinations that arise from repeatedly applying star to semilinear sets. If $X = \{\vec{x}_1, \ldots, \vec{x}_N\} \subseteq \mathbb{N}^d$ is a finite set of vectors and $W \subseteq X \times X$ a dependency graph on X, define $X^{*(W)} = \{\sum_{i=1}^{N} \lambda_i \vec{x}_i \mid \forall i, j \leq N.\ \lambda_i > 0 \wedge (\vec{x}_i, \vec{x}_j) \in W \Rightarrow \lambda_j > 0\}$. The dependency graph in Theorem 1 would have an edge from each $\vec{b}_{ij}$ to $\vec{a}_i$. The generalization of Fact 1 to the class of graphs W sufficient for the more general result is the following.

Theorem 3. *Let $X \subseteq \mathbb{Z}^d$ be a finite set of integer vectors with acyclic dependency graph $W \subseteq X \times X$ such that for each node $\vec{x} \in X$ the number of nodes reachable from $\vec{x}$ in W is bounded by a constant C. If $\vec{b} \in X^{*(W)}$ then there exists $\tilde{X} \subseteq X$ such that $\vec{b} \in \tilde{X}^{*(W)}$ and $|\tilde{X}| \leq 2C^2 d \log(4dM)$, where $M = \max_{x \in X} ||x||_\infty$.*

Proof sketch. Let $B = 2d \log(4dM)$ from Fact 1. Consider a linear combination $\vec{u} = \sum_i \lambda_i \vec{v}_i$ of vectors from X that satisfies the dependencies in W. Our goal is to find a small number of vectors that generate $\vec{u}$. In the first step we consider the source nodes of W, that is, vectors $Y_0 \subseteq X$ with no incoming edges in the graph. Applying Fact 1 to $\vec{v}_0 = \sum_{\vec{v}_i \in Y_0} \lambda_i \vec{v}_i$ we obtain a subset $Z_0 \subseteq Y_0$, with $|Z_0| \leq B$, such that $\vec{v}_0 = \sum_{\vec{v}_i \in Z_0} \lambda'_i \vec{v}_i$. To enforce the constraints in the graph W, we then take closure of Z_0 under reachability in W and obtain the set Q_0 of size at most CB. Let $\vec{u}_0 = \sum_{\vec{v}_i \in Q_0} \lambda_i \vec{v}_i$.

We repeat the procedure on the vector $\vec{u} - \vec{u}_0$. Only in this step we eliminate all the sources Y_0 and vertices belonging to Q_0 from the graph and consider the vectors that are sources in the subgraph of W induced by the remaining vectors $Y_1 = X \setminus (Y_0 \cup Q_0)$. We repeat this procedure as long as there are nodes in the graph. The number of times we need to repeat it is bounded by the longest path in W, which, by assumption, is bounded by C. At each step we select CB vectors, so the total number of nodes that we need in the linear combination is bounded by $C^2 B$. ■

Using Fact 1 we obtain a polynomial subset of vectors that satisfy given QFPA formulas and whose linear combination is the given vector $\vec{u}$. We then use results from previous sections to show that a linear combination of solutions of a QFPA formula can be represented as a sum of a polynomial number of solutions of this QFPA formula. This allows us to generalize results of Section 3 to formulas that contain not just one star operator but any regular expression over solution sets of QFPA formulas, which in turn proves that the reachability problem for transition systems described in this section is also in NP.

Acknowledgements. We thank Nikolaj Bjørner for useful comments on a draft of this paper and CAV 2008 reviewers for their patience and useful feedback.

References

1. T. Ball, B. Cook, S. K. Lahiri, and L. Zhang. Zapato: Automatic theorem proving for predicate abstraction refinement. In *CAV*, 2004.
2. T. Ball, R. Majumdar, T. Millstein, and S. K. Rajamani. Automatic predicate abstraction of C programs. In *Proc. ACM PLDI*, 2001.
3. M. Balser, W. Reif, G. Schellhorn, K. Stenzel, and A. Thums. Formal system development with KIV. In *FASE*, number 1783 in LNCS, 2000.
4. J.-P. Banâtre and D. L. Métayer. Programming by multiset transformation. *Commun. ACM*, 36(1):98–111, 1993.
5. C. Barrett and S. Berezin. CVC Lite: A new implementation of the cooperating validity checker. In *CAV*, volume 3114 of *LNCS*, 2004.
6. D. Basin and S. Friedrich. Combining WS1S and HOL. In *FROCOS*, volume 7 of *Studies in Logic and Computation*. 2000.
7. L. de Moura and N. Bjørner. Efficient E-matching for SMT solvers. In *CADE*, 2007.
8. L. A. Dennis, G. Collins, M. Norrish, R. Boulton, K. Slind, G. Robinson, M. Gordon, and T. Melham. The PROSPER toolkit. In *TACAS*, number 1785 in LNCS, 2000.
9. F. Eisenbrand and G. Shmonin. Carathéodory bounds for integer cones. *Operations Research Letters*, 34(5):564–568, September 2006. http://dx.doi.org/10.1016/j.orl.2005.09.008.
10. Y. Ge, C. Barrett, and C. Tinelli. Solving quantified verification conditions using satisfiability modulo theories. In *CADE*, 2007.
11. S. Ginsburg and E. Spanier. Semigroups, Pressburger formulas and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.
12. T. A. Henzinger, R. Jhala, R. Majumdar, and G. Sutre. Lazy abstraction. In *POPL*, 2002.
13. V. Kuncak. *Modular Data Structure Verification*. PhD thesis, EECS Department, Massachusetts Institute of Technology, February 2007.
14. V. Kuncak and M. Rinard. Towards efficient satisfiability checking for Boolean Algebra with Presburger Arithmetic. In *CADE-21*, 2007.
15. S. K. Lahiri and S. A. Seshia. The UCLID decision procedure. In *CAV'04*, 2004.

16. D. Lugiez. Multitree automata that count. *Theor. Comput. Sci.*, 333(1-2):225–263, 2005.
17. S. McLaughlin, C. Barrett, and Y. Ge. Cooperating theorem provers: A case study combining HOL-Light and CVC Lite. In *PDPAR*, volume 144(2) of *ENTCS*, 2006.
18. H. H. Nguyen, C. David, S. Qin, and W.-N. Chin. Automated verification of shape, size and bag properties via separation logic. In *VMCAI*, 2007.
19. T. Nipkow, M. Wenzel, L. C. Paulson, and N. Voelker. Multiset theory version 1.30 (Isabelle distribution). http://isabelle.in.tum.de/dist/library/HOL/Library/Multiset.html, 2005.
20. C. H. Papadimitriou. On the complexity of integer programming. *J. ACM*, 28(4):765–768, 1981.
21. R. Piskac and V. Kuncak. Decision procedures for multisets with cardinality constraints. In *VMCAI*, number 4905 in LNCS, 2008.
22. L. Pottier. Minimal solutions of linear diophantine systems: Bounds and algorithms. In *RTA*, volume 488 of *LNCS*, 1991.
23. J. T. Schwartz. On programming: An interim report on the SETL project. Technical report, Courant Institute, New York, 1973.
24. N. Shankar. Using decision procedures with a higher-order logic. In *TPHOLs*, 2001.
25. C. G. Zarba. Combining multisets with integers. In *CADE-18*, 2002.