Collections, Cardinalities, and Relations

Kuat Yessenov[1*], Ruzica Piskac[2], and Viktor Kuncak[2**]

[1] MIT Computer Science and Artificial Intelligence Lab, Cambridge, USA
kuat@csail.mit.edu
[2] EPFL School of Computer and Communication Sciences, Lausanne, Switzerland
firstname.lastname@epfl.ch

Abstract. Logics that involve collections (sets, multisets), and cardinality constraints are useful for reasoning about unbounded data structures and concurrent processes. To make such logics more useful in verification this paper extends them with the ability to compute direct and inverse relation and function images. We establish decidability and complexity bounds for the extended logics.

1 Introduction

Deductive verification of software often involves proving the validity of formulas in expressive logics. Verification condition generation produces such formulas directly from annotated source code [3, 5], whereas predicate abstraction techniques [4] generate many formulas during fixpoint computation. Abstract interpretation [6] precomputes parameterized transfer functions; the automation of this process [25] also reduces to proving formula validity.

As the starting point of this paper we consider decidable logics whose variables denote collections of objects, corresponding to, for example, dynamically allocated objects in the heap, or concurrent processes. Our logics include standard set algebra operations such as $\cap, \cup$ and complement. They also include the *cardinality operator*, to compute the number of elements in the collection, and support linear integer arithmetic constraints on the cardinalities. One such logic is QFBAPA (quantifier-free Boolean Algebra with Presburger Arithmetic), which we recently proved to be in NP [16], an improvement over the previous NEXPTIME algorithms based on quantifier elimination [9,14]. We subsequently generalized this result to quantifier-free constraints on multisets (bags), collections in which an element can occur multiple times [22, 23]. The usefulness of collections and cardinality measures on them has been established through a number of examples from software analysis and verification, including not only decision procedures [14, 16, 29] but also static analyses that operate directly on the set abstraction or the cardinality abstraction [11, 13, 21].

---

* Work done while Kuat Yessenov was visiting EPFL.

** This research is supported in part by the Swiss National Science Foundation Grant "Precise and Scalable Analyses for Reliable Software".

To make the logics of collection more useful, in this paper we generalize them in a natural direction: we introduce functions and relations and supports computing images and inverse images of sets under these functions and relations. Our primary motivation is that in verification problems, collections such as sets and multisets are often defined by computing an image of a more concrete data structure, often itself a set (see Section 2). The resulting logics are extensions of both the logics with cardinalities, but also of certain previously studied constraints (none of which include symbolic cardinality bounds): certain Tarskian set constraints [10], certain Description Logics [1], and set-valued field constraints [15]. Our techniques are also related to the technique of bridging functions [19]. What distinguishes our result from previous ones is the (often optimal) complexity that we achieve in the presence of sets, multisets, relations, and symbolic cardinality constraints. Our NEXPTIME fragment includes images of n-ary relations and is thus not expressible in the two-variable logic with counting [20, 24].

Contributions. We summarize the contributions of this paper as follows:

- We describe a new NEXPTIME-complete logic that includes sets, n-ary relations, unary functions, and symbolic cardinality constraints.
- We sketch the extension of the logic above with cardinalities of relations and with n-ary function symbols; we prove 2-NEXPTIME upper bound for its satisfiability.
- We point to a few simple extensions of the above logic that lead to undecidability.
- We consider an extension of QFBAPA [16] with relation image constraints, for a relation between two disjoint sorts of elements results. We show that the sparse model solution phenomenon of QFBAPA continues to apply in the presence of such relations, and use it to prove that the logic remains inside NP.
- We show NEXPTIME completeness (by reduction to [22]) of a logic that allows computing multisets instead of sets as function images, preserving the multiplicity of elements that occur in the range of the function multiple times. This is a natural definition of the notion of multiset comprehension and arises e.g. when using multisets to abstract the content of Java-like linked data structures.

2 Motivating Examples

In this section we list several examples from verification of data structures that have motivated us to consider extending BAPA with functions and relations.

We start with a dynamically allocated data structure (such as a list or a tree) that manipulates a set of linked nodes denoted by the variable `nodes`. The useful content in the data structure is stored in the `data` fields of the elements of `nodes`. The `nodes` set can be either explicitly manipulated through a library data type or built-in type [7], or it can be verified to correspond to a set of reachable objects using techniques such as [27]. The content of the list, stored in the $\mathsf{content}$ specification variable, is then an image of nodes under the function data. We consider two cases of specification in our example: 1) $\mathsf{content}$ is a *set*, that is, multiple occurrences of elements are ignored and 2) $\mathsf{content}$ is a *multiset*, preserving the counts of occurrences of each element in the data structure.

$$\begin{aligned}&\mathsf{nodes} \subseteq \mathsf{alloc} \land \mathsf{card\ tmp} = 1 \land \mathsf{tmp} \cap \mathsf{alloc} = \emptyset \land \mathsf{data[tmp]} = \mathsf{e} \land \\ &\mathsf{content} = \mathsf{data[nodes]} \land \mathsf{nodes1} = \mathsf{nodes} \cup \mathsf{tmp} \land \mathsf{content1} = \mathsf{data[nodes1]} \rightarrow \\ &\qquad\qquad \mathsf{card\ content1} \leq \mathsf{card\ content} + 1\end{aligned}$$

Fig. 1. Verification condition for verifying that by inserting an element into a list, the size of the list does not decrease. The variables occurring in the formula have the following types: $\mathsf{nodes}, \mathsf{alloc}, \mathsf{tmp}, \mathsf{e}, \mathsf{content}, \mathsf{content1} :: \mathsf{Set}\langle\mathsf{E}\rangle$, $\mathsf{data} :: \mathsf{E} \rightarrow \mathsf{E}$.

$$\begin{aligned}&\mathsf{nodes} \subseteq \mathsf{alloc} \land \mathsf{card\ tmp} = 1 \land \mathsf{tmp} \cap \mathsf{alloc} = \emptyset \land \mathsf{data[tmp]} = \mathsf{e} \land \\ &\mathsf{content} = \mathsf{data[nodes]} \land \mathsf{nodes1} = \mathsf{nodes} \cup \mathsf{tmp} \land \mathsf{content1} = \mathsf{data[nodes1]} \rightarrow \\ &\qquad\qquad \mathsf{card\ content1} = \mathsf{card\ content} + 1\end{aligned}$$

Fig. 2. Verification condition for verifying that by inserting an element into a list, the size of a list increases by one. The variables occurring in the formula have the following types: $\mathsf{nodes}, \mathsf{alloc}, \mathsf{tmp} :: \mathsf{Set}\langle\mathsf{E}\rangle$, $\mathsf{content}, \mathsf{content1}, \mathsf{e} :: \mathsf{Multiset}\langle\mathsf{E}\rangle$, $\mathsf{data} :: \mathsf{E} \rightarrow \mathsf{E}$.

The verification condition generated for the case when the image is a set is given in Figure 1. This formula belongs to the language QFBAPA-Rel defined in Section 3 and a decision procedure presented there checks satisfiability of such formulas. It reduces a formula to a (exponentially larger) quantifier-free BAPA formula [16] by introducing Venn regions [26] and cardinality constraints on them, and eliminating the function symbols such as data. The resulting formula can be decided using the NP algorithm in [16], giving NEXPTIME procedure overall.

A more precise abstraction is obtained if $\mathsf{content}$ is viewed as a multiset. Figure 2 shows the verification condition for this case. Section 6 describes a decision procedure for an extension of QFBAPA with function symbols where functions can also return a multiset, not only set. The approach also rewrites sets as a disjoint union of Venn regions. It then constrains the cardinality of the multiset obtained through the image to be equal to the cardinality of the original set. This final formula is a formula in the NP-complete logic for reasoning about multisets and cardinality constraints [22, 23].

Another motivation in software verification comes from regional logic [2], used for proving correctness of programs with shared mutable objects. Regional logic introduces *region variables*, which are finite sets of object references, and uses to express properties about separation and mutation. Following the example presented in [2], consider a finite binary tree and let x be a variable of type $Node$. A node y, $y \neq \mathbf{null}$, has three fields: `left`, `right` and `item`. We can express that for $x \neq \mathbf{null}$, x has two disjoint subtrees which are closed under `left` and `right` as follows:

$$\begin{aligned} P &\equiv x \neq \mathbf{null} \wedge x.\mathtt{left} \in r_1 \wedge x.\mathtt{right} \in r_2 \wedge r_1 \# r_2 \wedge \mathit{closed} \\ \mathit{closed} &\equiv r_1.\mathtt{left} \subseteq r_1 \wedge r_1.\mathtt{right} \subseteq r_1 \wedge r_2.\mathtt{left} \subseteq r_2 \wedge r_2.\mathtt{right} \subseteq r_2 \end{aligned}$$

Those formulas can be translated into QFBAPA-Rel logic by treating each region as a set and each field as a function defined on a set:

$$\begin{aligned} P_1 &\equiv |X| = 1 \wedge X \neq \emptyset \wedge l(X) \subseteq R_1 \wedge r(X) \subseteq R_2 \wedge R_1 \cap R_2 = \emptyset \wedge \mathit{closed}_1 \\ \mathit{closed}_1 &\equiv l(R_1) \subseteq R_1 \wedge r(R_1) \subseteq R_1 \wedge l(R_2) \subseteq R_2 \wedge r(R_2) \subseteq R_2 \end{aligned}$$

Many of the assertions used in [2] can be easily translated in QFBAPA-Rel. In addition, conditions such as expressing that two regions have the same size can be also expressed in QFBAPA-Rel.

3 QFBAPA-Rel: A Logic of Sets, Cardinalities, Relations, and Unary Functions

This section presents a decision procedure for the language of sets, cardinalities, n-ary relations, and unary total functions. The language we consider is denoted QFBAPA-Rel and is defined by the grammar in Figure 3. It naturally extends quantifier-free fragment of BAPA [16] with unary function symbols (denoted by $f, g, \ldots$) and relations of any arity (denoted by $p, q, r, \ldots$ to distinguish them from functions). The expression $f[B]$ denotes the set $\{y \mid \exists x. x \in B \wedge y = f(x)\}$. Cardinality constraints allow us, in particular, to express whether a function is injective on A (by $|f[A]| = |A|$) or surjective onto A ($f[\mathcal{U}] = A$). For a binary relation r, the expression $r[A]$ is a relational join expression denoting $\{y \mid \exists x. x \in A \wedge (x, y) \in r\}$. Analogously, $r^{-1}[B]$ denotes $\{x \mid \exists y. y \in B \wedge (x, y) \in r\}$. We require functions to be total, whereas relations need not be left-total or right-total. Higher-arity relations have an analogous interpretation with the term $r[B_1, \ldots, B_{i-1}, *, B_{i+1}, \ldots, B_k]$ standing for the set

$$\{x_i \mid \exists x_1 \in B_1, \ldots, x_{i-1} \in B_{i-1}, x_{i+1} \in B_{i+1}, \ldots, x_k \in B_k \wedge (x_1, \ldots, x_k) \in r\}$$

for a relation r of arity k.

The decision problem we are concerned with is the satisfiability problem for QFBAPA-Rel: the question of existence of a finite interpretation α in which formula is true.

An interpretation assigns values to a set, an integer, function and relation variables. If α is an interpretation then $\alpha[x := v]$ is the interpration such that $\alpha[x := v](x) = v$ and $\alpha[x := v](y) = \alpha(y)$ for $x \neq y$.

3.1 Decision Procedure for QFBAPA-Rel

Our decision procedure for QFBAPA-Rel satisfiability is a reduction to the satisfiability of quantifier-free Boolean algebra with Presburger arithmetic (QFBAPA).

$$
\begin{aligned}
F &::= L \mid F_1 \vee F_2 \mid \neg F \\
L &::= B_1 \subseteq B_2 \mid T_1 < T_2 \mid K \,\mathsf{dvd}\, T \\
B &::= x \mid \emptyset \mid \mathcal{U} \mid B_1 \cup B_2 \mid B^c \mid f[B] \mid f^{-1}[B] \mid r[B] \mid r^{-1}[B] \mid \\
&\qquad r[B_1, \ldots, B_{i-1}, *, B_{i+1}, \ldots, B_k] \\
T &::= k \mid K \mid \mathsf{MAXC} \mid T_1 + T_2 \mid \; |B| \\
K &::= \cdots \mid -2 \mid -1 \mid 0 \mid 1 \mid 2 \mid \cdots
\end{aligned}
$$

Fig. 3. Syntax of QFBAPA-Rel

The first step of the reduction is elimination of function inverses and functional and relational composition from the given formula. Because all functions are total, $B = f^{-1}[A]$ is equivalent to $f[B] \subseteq A \wedge f[B^c] \subseteq A^c$. We allocate a fresh set variable for every functional or relational complex expressions. For example, a formula $f[r[A]] \subseteq h[B \cap C]$ becomes $E \subseteq G \wedge D = r[A] \wedge E = f[D] \wedge F = B \cap C \wedge G = h[F]$. This separates functional and relational terms from the rest of the formula. Using these transformations we obtain (in polynomial time) a conjunction of a QFBAPA formula F_{BAPA} and a conjunction of set constraints F_{IMAGE}.

For every function term f, formula F_{IMAGE} contains constraints of the form $A_i = f[B_i]$ where A_i and B_i are set variables. For every relational term r where r is binary, F_{IMAGE} contains constraints of the form $A_i = r[B_i]$ and $A'_i = r^{-1}[B'_i]$ where A_i, A'_i, B_i, B'_i are set variables. For a relation r of arity k the formula F_{IMAGE} contains constraints of the form $A^j_i = r[B^j_{i1}, \ldots, B^j_{i(j-1)}, *, B^j_{i(j+1)}, \ldots, B^j_k]$ for $1 \leq j \leq k$.

Eliminating function applications. Let $s_1, \ldots, s_m$ be the Boolean algebra terms representing the disjoint Venn regions that are formed by taking intersection $\bigcap_{\alpha_i \in \{0,1\}} b_i^{\alpha_i}$ of all set variables b_i appearing in the entire original formula. For a set x, x^1 denotes x and x^0 denotes x^c. We focus on a single function symbol f and its constraints from F_{IMAGE}, We repeat the following algorithm for every function symbol f that appears in F_{IMAGE}.

Let $\bigwedge_i A_i = f[B_i]$ be the constraints for f. Each term B_i may be written as a disjoint union of cubes $s_{i_1} \cup s_{i_2} \cup \ldots \cup s_{i_k}$ so that $f[B_i] = \bigcup f[s_{i_j}]$. Because the cubes are disjoint, we can define the values of the function on each cube independently. Introduce set variables $t_j = f[s_j]$. Replace each term $f[B_i]$ with the corresponding union $\bigcup t_{i_j}$ of a subset of cube images:

$$
A_i = \bigcup_{s_j \subseteq B_i} t_j \tag{1}
$$

After this transformation, the set constraints are reduced to QFBAPA by introducing fresh set variables t_i. Moreover, we introduce the following *functional consistency axioms*:

$$
\bigwedge |t_i| \leq |s_i| \;\wedge\; (|t_i| = 0 \Leftrightarrow |s_i| = 0) \tag{2}
$$

Theorem 1. *The projections of the set of solutions (models) for formulas* (1) $\wedge$ (2) *and the formula* F_{IMAGE} *onto set variables* A_i, B_i *are equal.*

Proof. Given a solution of F_{IMAGE}, define the value of t_j as the value of $f[s_j]$. The result is a model satisfying (1) $\wedge$ (2). Conversely, consider a model α of (1) $\wedge$ (2); we construct a model α' that agrees with α on A_i, B_i and has the value $\alpha'(f)$ such that $\alpha'(f[s_j] = t_j)$ holds. For different s_j such definitions are independent. For $\alpha(s_j) = \emptyset$ also $\alpha(s_j) = \emptyset$, so condition $\alpha'(f[s_j] = t_j)$ holds. Otherwise, $0 < |\alpha(t_j)| \leq |\alpha(s_j)|$ by (2). Then there is a surjective function $h : s_j \to t_j$. Pick any such h and define restriction of $\alpha'(f)$ on s_j to be h. ∎

Eliminating binary relations. Previous procedure does not apply in a straightforward way to relations partly because we do not have a way to express directly inverses for relations that are not total. We instead apply the algorithm in Figure 4 for each relation r. The motivation for this algorithm is as follows.

Let $A_i = r[B_i]$ and $A'_i = r^{-1}[B'_i]$ be the constraints from F_{IMAGE} for r. Similarly to the above, let b_j be Venn regions over B_i. Introduce fresh set variables c_j that are constrained by $c_j = r[b_j]$. Because relational join commutes with set union, $A_i = r[B_i]$ is equivalent to $A_i = \bigcup_{b_j \subseteq B_i} c_j$. Repeat this procedure for B'_i using b'_k as Venn regions over B'_i. We obtain constraints of the form $c_j = r[b_j]$ and $c'_k = r^{-1}[b'_k]$.

INPUT: contrains $\bigwedge_i A_i = r[B_i] \wedge \bigwedge_i A'_i = r^{-1}[B'_i]$
OUTPUT: an equisatisfiable QFBAPA formula
1. define Boolean algebra terms b_j for Venn regions over B_i
2. define Boolean algebra terms b'_k for Venn regions over B'_i
3. introduce fresh set variables L_{jk}, R_{jk} for every pair b_j and b'_k
4. introduce constraints $L_{jk} \subseteq b_j \wedge R_{jk} \subseteq b'_k \wedge (L_{jk} = \emptyset \iff R_{jk} = \emptyset)$
5. replace each set constraint $A_i = r[B_i]$ with $A_i = \bigcup_{b_j \subseteq B_i} \bigcup_k R_{jk}$
6. replace each set constraint $A'_i = r^{-1}[B_i]$ with $A'_i = \bigcup_{b'_k \subseteq B_i} \bigcup_j L_{jk}$
7. take conjunction of all set constraints from steps 4,5,6

Fig. 4. Algorithm for eliminating relations from QFBAPA-Rel

Next, introduce new relation variables r_{jk} meant to denote the restriction $\{(x,y) \mid x \in b_j \wedge y \in b'_k \wedge (x,y) \in r\}$ of the relation r to b_j in the domain and b'_k in the codomain. Then r is the disjoint union of r_{jk} over all pairs of j and k. We rewrite the constraints on c_j, c'_k as:

$$\bigwedge_j \left(c_j = \bigcup_k r_{jk}[b_j] \right) \wedge \bigwedge_k \left(c'_k = \bigcup_j r_{jk}^{-1}[b'_k] \right)$$

The behavior of each relation r_{jk} is unrestricted by any other constraints as long as it is a relation from domain b_j to codomain b'_k. That means that the relation

r_{jk} is determined for our purposes by its domain and range $r_{jk}[b_j]$ and $r_{jk}^{-1}[b'_k]$. We introduce two set variables to encode these as R_{jk} and L_{jk}, respectively. We rewrite the relation constraints as $c_j = \bigcup_k R_{jk}$ and $c'_k = \bigcup_j L_{jk}$.

The *relational consistency condition* amounts to the following axioms:

$$\bigwedge_{j,k} L_{jk} \subseteq b_j \wedge R_{jk} \subseteq b'_k \wedge (L_{jk} = \emptyset \iff R_{jk} = \emptyset)$$

Because both j and k range over singly exponentially many variables, there are singly exponentially many fresh variables and constraints introduced.

Theorem 2. *The algorithm in Figure 4 produces a QFBAPA formula of singly exponential size with the same set of solutions for A_i, B_i, A'_i, B'_i.*

Proof. Because we only made sound syntactic transformations and introduced variables defined by existing terms, it suffices to show that a model of the generated QFBAPA formula extends to a model of the original formula. Assume we are given an interpretation of the QFBAPA formula, that is values of L_{jk} and R_{jk} and the set variables from the original formula A_i, B_i, A'_i, B'_i. Relation consistency axioms allow us to define total relations r_{jk} by mapping every element from L_{jk} to every element from R_{jk}. An interpretation of r is then the union of all these pairwise non-intersecting relations r_{jk}. To see that we satisfied the set constraints, consider, for example, constraint $A_i = r[B_i]$:

$$\begin{aligned} r[B_i] &= \bigcup_{b_j \subseteq B_i} r[b_j] = \bigcup_{b_j \subseteq B_i} \bigcup_{j',k} r_{j'k}[b_j] = \bigcup_{b_j \subseteq B_i} \bigcup_{k} r_{jk}[b_j] = \\ &= \bigcup_{b_j \subseteq B_i} \bigcup_{k} r_{jk}[L_{jk}] = \bigcup_{b_j \subseteq B_i} \bigcup_{k} R_{jk} = A_i \quad \blacksquare \end{aligned}$$

Eliminating higher-arity relations. The algorithm for binary relations extends naturally to higher-arity relations. We sketch the construction in this section. We focus on a single k-arity relation r with set constraints $A_i^j = r[B_{i1}^j, \ldots, B_{i(j-1)}^j, *, B_{i(j+1)}^j, \ldots, B_k^j]$ for $j = 1, \ldots, k$. Similar to above, we introduce Venn regions b_l^j over j-th coordinate set variables B_{ij}. For a k-tuple of Venn regions $\mathbf{v} = (b_{l_1}^1, b_{l_2}^2, \ldots, b_{l_k}^k)$, we consider the restriction $r_{\mathbf{v}}$ of the relation r to $b_{l_j}^j$ on every coordinate.

Observe that every set constraint can be replaced with a union of application of the relations $r_{\mathbf{v}}$ to tuples of Venn regions. The key idea is that each such application is uniquely defined by projections of $r_{\mathbf{v}}$ onto every coordinate. That is we introduce k set variables $\{P_{\mathbf{v}}^i\}_{i=1,\ldots,k}$ for every relation $r_{\mathbf{v}}$ such that:

$$\bigwedge_{j=1,\ldots,k} P_{\mathbf{v}}^j \subseteq b_{i_j}^j \wedge \left(\bigwedge_{j=1,\ldots,k} |P_{\mathbf{v}}^j| = 0 \vee \bigwedge_{j=1,\ldots,k} |P_{\mathbf{v}}^j| > 0 \right)$$

Any model to this condition gives rise to a well-defined relation $r_{\mathbf{v}}$ equal to the Cartesian product of the sets $P_{\mathbf{v}}^1 \times \ldots \times P_{\mathbf{v}}^k$ (or empty if any of them is empty). This way we can reconstruct the original relation r from the pairwise disjoint interpretations of relations $r_{\mathbf{v}}$.

For instance, the following formula represents a set constraint above (after dropping j super-script):

$$A_i = r[B_{i1}, \ldots, B_{i(j-1)}, *, B_{i(j+1)}, \ldots, B_k] = \bigcup_{\text{cube} b_l \subseteq B_{il}} r[b_1, \ldots, b_{j-1}, *, b_{j+1}, \ldots, b_k]$$

$$= \bigcup_{b_l \subseteq B_{il}, l \neq j, \mathbf{v}=(b_l)} r_{\mathbf{v}}[b_1, \ldots, b_{j-1}, *, b_{j+1}, \ldots, b_k] = \bigcup_{b_l \subseteq B_{il}, l \neq j, \mathbf{v}=(b_l)} P^j_{\mathbf{v}}$$

The total number of fresh set variables and the size of the resulting formula are still singly exponential in the size of the formula, since we consider Venn regions for each coordinate and take k-tuples of these regions for k linear in size.

3.2 Complexity of QFBAPA-Rel

Combining results of the previous sections, we obtain a reduction from QFBAPA-Rel to QFBAPA. This reduction produces a formula of a singly exponential size by introducing set variables for Venn regions over set variables in the original formula for each function and relation. Because QFBAPA is known to be NP-complete [16], we conclude that QFBAPA-Rel is in NEXPTIME. Moreover, we obtain EXPTIME *BAPA reduction* from QFBAPA-Rel to QFBAPA [28], which means that the method can be used to combine QFBAPA-Rel with other logics, such as the Weak Monadic Second-Order Logic over Trees.

Theorem 3. *QFBAPA-Rel is NEXPTIME-complete, even with no relation symbols and with only one unary function symbol.*

Proof. The algorithm above established the NEXPTIME upper bound, we next prove the matching lower bound. In [10], NEXPTIME lower bound for Tarskian set constraints with constants and binary functions is shown by reduction of a fragment of first order logic. We adapt this proof to QFBAPA-Rel. The proof relies on the result [17] that acceptance of nondeterministic exponential-time bounded Turing machines can be reduced to satisfiability of formulas of the form $\exists z.F_1 \wedge \forall y \exists x.F_2 \wedge \forall y_1 \forall y_2.F_3$ where F_1, F_2, and F_3 have no quantifiers and are monadic (have only unary predicates). Given a formula of this from, we construct an equisatisfiable QFBAPA-Rel formula as a set of constraints, as follows. We identify monadic predicate symbols with set variables, using the same symbols for both. After Skolemizing the formula by introducing a constant symbol a and a monadic function symbol f, and putting F_1, F_2, and F_3 into the conjunctive normal form, there are three types of clauses (as remarked already in [10]); we describe our encoding of each of these clauses.

1. monadic formulas over the constant symbol a (obtained from $\exists z.F_1$)
   We transform the conjunction of all such formula into a set constraint as follows. For each monadic predicate P replace $P(a)$ with P, replace $\vee$ with $\cup$, replace $\neg$ with $\_^c$, and $\wedge$ with $\cap$. Let the result of this replacement be a set algebra expression S; then generate the QFBAPA-Rel formula $S \neq \emptyset$.

2. clauses of the form:

$$\forall x. P_1(x) \vee P_2(x) \vee \ldots \vee P_m(x) \vee Q_1(f(x)) \vee Q_2(f(x)) \vee \ldots \vee Q_n(f(x))$$

   For each such clause, we generate a constraint:

$$f\,(P_1^c \cap P_2^c \cap \ldots \cap P_m^c) \subseteq Q_1 \cup Q_2 \cup \ldots \cup Q_n$$

3. clauses of the form:

$$\forall y_1 \forall y_2.\ P_1(y_1) \vee P_2(y_1) \vee \ldots \vee P_m(y_1) \vee Q_1(y_2) \vee Q_2(y_2) \vee \ldots \vee Q_n(y_2)$$

   For each such clause we generate the QFBAPA-Rel formula:

$$(P_1 \cup P_2 \cup \ldots \cup P_m = \mathcal{U}) \ \vee \ (Q_1 \cup Q_2 \cup \ldots \cup Q_n = \mathcal{U})$$

   (This last constraint differs from the one in [10] and does not require any binary function symbols).

The resulting QFBAPA-Rel formula is equisatisfiable with the original formula, so NEXPTIME lower bound follows from [17]. ■

Decidable Extensions: n-ary Functions, Relation Cardinalities. We have presented QFBAPA-Rel, as a logic with monadic functions and arbitrary relations and shown it to be NEXPTIME-complete. We next sketch how to extend the decidability to include also the functions of higher arity. Generalizing the method for unary functions, we have for e.g. a binary function $f[p_1 \cup p_2, q_1 \cup q_2] = f[p_1, q_1] \cup f[p_1, q_2] \cup f[p_2, q_1] \cup f[p_2, q_2]$. We apply such reasoning to all Venn regions. This creates a singly exponential blowup in formula size. Given Venn regions p, q and image $f[p, q]$, let their cardinalities be k_p, k_q, k_{fpq}, respectively. Then a necessary condition for a function to be definable on $p \times q$ is $k_{fpq} \leq k_p k_q$, which is a non-linear constraint. In general, the satisfiability of QFBAPA-Rel with n-ary function symbols reduces to the satisfiability of a conjunction of 1) such non-linear constraints $x \leq y_1 \ldots y_n$ and 2) linear integer constraints. Such conjunctions are called *prequadratic* in [10] and their satisfiability is shown to be in NEXPTIME. (The quadratic as opposed to higher-degree monomials on right-hand side suffice because replacing $x \leq y_1 y_2 \ldots y_n$ with $x \leq y_1 z_1 \wedge z_1 \leq y_2 \ldots y_n$ preserves the projection of solution set onto $x, y_1, \ldots, y_n$.) The generated prequadtratic formula is singly exponential, which gives an upper bound of 2-NEXPTIME for QFBAPA-Rel extended with functions of arbitrary arity.

A similar construction works for an extension of QFBAPA-Rel with the cardinality operator applied to relations (computing the number of related pairs of elements). In the notation of Section 3.1, we add the prequadratic constraints $|r_{jk}| \leq |L_{jk}|\ |R_{jk}|$ as well as the appropriate linear constraints.

3.3 Undecidable Extensions: Injective Binary Functions, Quantifiers

Injective binary functions. If in addition to introducing binary function symbols we allow stating that they are injective, then instead of prequadratic constraints of the previous section we obtain constraints of the form $x = yz$. Indeed,

$|f[p,q]| = |p|\ |q|$ for an injective function f. Together with linear constraints, these constraints can express arbitrary Diophantine equations (polynomial integer equations). The satisfiability in such language is undecidable [18] (Hilbert's 10th problem), and thus adding an injective function symbol to QFBAPA gives an undecidable logic.

Relation cardinality with Cartesian product. We noted that decidability is preserved if we allow computing the cardinality of a relation. However, if we can additionally constrain a relation to be full Cartesian product of two sets, then we again obtain the constraint $|p \times q| = |p|\ |q|$, and the undecidability by [18].

Quantification. Note that BAPA with arbitrary set and integer quantifiers is decidable [9,14]. On the other hand, the logic that allows quantification over sets and one function symbols is also decidable [12, Theorem 8.3]. However, a BAPA extension that allows quantified formulas with unary function symbol images is undecidable. Indeed, define a function f mapping A onto B where each inverse image has k elements: $B = f[A] \wedge \forall e.\ e \subseteq B \wedge |e| = 1 \Rightarrow |f^{-1}[e]| = k$. Then $|B| = k|A|$ and the set of values $(|B|, k, |A|)$ contains precisely the solutions (x, y, z) of the equation $x = yz$. Recall that $f^{-1}[e] = u$ is expressible by $f[u] \subseteq e \wedge f[u^c] \subseteq e^c$, so either direct or inverse function image can be used, or a relation restricted to be functional using a quantified formula, in each case resulting in undecidability by [18].

4 NP-Complete Two-Sorted QFBAPA-Rel Fragment

In this section we identify a fragment of the QFBAPA-Rel logic in Figure 3. Remarkably, this fragment has NP instead of NEXPTIME complexity for the satisfiability problem. Figure 5 shows the syntax of this fragment, QFBAPA−R2, which is an extension of QFBAPA with relation image of one two-sorted binary relation symbol. Compared to full QFBAPA-Rel, there are no function symbols, no inverse images, and there is only one relation symbol, denoted r, which is binary. Moreover, each set contains only elements of sort $\mathcal{A}$, or only elements of a disjoint sort $\mathcal{B}$. There are two disjoint universal sets $\mathcal{U}_{\mathcal{A}}$ and $\mathcal{U}_{\mathcal{B}}$ for the corresponding sorts. The boolean operators $\cup, \cap$ and complement apply only to sets of the same sort. We require that the relation r relate sort $\mathcal{A}$ to sort $\mathcal{B}$, that is, the semantic condition $r \subseteq \mathcal{U}_{\mathcal{A}} \times \mathcal{U}_{\mathcal{B}}$ holds. An example formula in this fragment is $x = y \to |r[x]| = |r[y]|$. In this formula x, y have sort $\mathcal{A}$ and the expressions $r[x]$ and $r[y]$ have sort $\mathcal{B}$.

Normal form. Consider an arbitrary QFBAPA−R2 formula F. By introducing fresh variables for sets and integers (similarly as in [16]), we can rewrite the formula in (with only linear increase in size) in the form

$$F_C \wedge F_B \wedge F_A \wedge P \tag{3}$$

where:

$$
\begin{aligned}
F &::= L \mid F_1 \vee F_2 \mid \neg F \\
L &::= B_1 \subseteq B_2 \mid T_1 < T_2 \mid K \,\mathsf{dvd}\, T \\
B &::= x_B \mid \emptyset \mid \mathcal{U}_{\mathcal{B}} \mid B_1 \cup B_2 \mid B^c \mid r[A] \\
A &::= x_A \mid \emptyset \mid \mathcal{U}_{\mathcal{A}} \mid A_1 \cup A_2 \mid A^c \\
T &::= k \mid K \mid \mathsf{MAXC} \mid T_1 + T_2 \mid \ \mathsf{|}B\mathsf{|} \mid \ \mathsf{|}A\mathsf{|} \\
K &::= \cdots \mid -2 \mid -1 \mid 0 \mid 1 \mid 2 \mid \cdots
\end{aligned}
$$

Fig. 5. Syntax of QFBAPA-R2

- F_C is $\bigwedge_{i=1}^{n} B_i = r[A_i]$ and this is the only part of formula containing r;
- F_B is of form $\bigwedge_i L_i$ where each L_i is of the form $|b| = k$ for some integer variable k and some set algebra expression b of sort $\mathcal{B}$ (it is thus a QFBAPA formula);
- F_A is analogously of form $\bigwedge_i L_i$ where each L_i is of the form $|a| = k$ for some integer variable k and some set algebra expression a of sort $\mathcal{A}$ (it is thus also a QFBAPA formula);
- P is a quantifier-free Presburger arithmetic formula.

In the sequel we assume that QFBAPA-R2 formulas are in normal form. The proof of the following Lemma is straightforward.

Lemma 4 (Models Modulo Venn Regions). *Let p be a Venn region over sets A_i and q a Venn region over sets B_i. If α is a model of the* QFBAPA-R2 *formula F and $\alpha(r) \cap (\alpha(p) \times \alpha(q)) \neq \emptyset$, then α' given as $\alpha[r := w]$ is also a model of the formula F where $w = \alpha(r) \cup (\alpha(p) \times \alpha(q))$.*

By repeated application of the above lemma it follows that it suffices to consider *completed models* α, in which $\alpha(r)$ is a union of products of Venn regions, and is thus given by a bipartite graph, denoted E, between Venn regions of sort $\mathcal{A}$ and Venn regions of sort $\mathcal{B}$.

Sparse models. We are interested in the finite satisfiability problem for QFBAPA-R2 formulas. We show that this problem is in NP. This result is a strict a generalization of the proof that QFBAPA is in NP [16] and similarly proceeds by proving a *sparse model property*: if the formula is satisfiable, it has a model in which only polynomially many Venn regions are non-empty. By Lemma 4, models with sparse Venn regions can also be assumed to have polynomial representations that have polynomial sized bipartite graphs E. By *polynomial* in this section we mean polynomial in the size of formula F, where integer constants are denoted in binary. The following theorem builds on the sparse model property for QFBAPA [16]. QFBAPA models can be represented by introducing an integer variable for each Venn region, and the sparse model property for QFBAPA relies on the integer analogue of Carathéodory theorem [8].

Theorem 5. *If a* QFBAPA-R2 *formula has a model, then it has a sparse model.*

Proof. Let α be a completed model of a formula F in form (3). Using α we simplify F_C as follows. For all sets A_i where $\alpha(A_i) = \emptyset$, replace A_i and B_i with

$\emptyset$ and remove such A_i and B_i from consideration. Let K be the number of sets A_i remaining. For the remaining sets A_i, introduce constraint $|A_i| = k_i'$ into F_A and constraint $k_i' > 0$ into P.

Next, apply the sparse model construction of QFBAPA to F_B part, as follows. Consider the result of replacing in F_B each integer variable k with the constant $\alpha(k)$. By [16], consider a sparse solution for the resulting QFBAPA formula that does not introduce any new non-empty Venn regions. That is, consider the Presburger arithmetic formula generated by those Venn regions q over sets B_i for which $\alpha(q) \neq \emptyset$, eliminating the variables corresponding to Venn regions q with $\alpha(q) = \emptyset$. The sparse solution of such Presburger arithmetic formula [8,16] yields a polynomial subset of non-empty Venn regions over B_i for which the integer values of $|b|$ expressions in F_B remain the same. We therefore obtain a set of cubes $C_B = \{q_1, \ldots, q_N\}$ and a model α_1 such that 1) $\alpha_1(q) \neq \emptyset$ iff $q \in C_B$, 2) $\alpha_1(F_B \wedge P)$, and 2) variables other than B_i have same values in α_1 and α.

Next, pick a set C_{A0} of cubes over A_i related to the chosen sparse set of cubes C_B. Let $1 \leq j \leq N$. Let i be any index such that $q_j \subseteq B_i$. Because $\alpha(B_i = r[A_i])$ there exists some pair $(a, b) \in \alpha(r) \cap \alpha(A_i) \times \alpha(q_j)$. Let $a \in p$ where $p \subseteq A_i$ is the cube containing a. Denote such cube p_{ji} and repeat this process for all $1 \leq j \leq N$ and all B_i where $q_j \subseteq B_i$ and let C_{A0} be the resulting set of cubes p_{ji}. The set C_{A0} has at most NK elements, which is polynomially many. In this process we have also identified a bipartite graph $E \subseteq C_{A0} \times C_B$, given by $E = \{(p_{ji}, q_j) \mid 1 \leq j \leq N,\ 1 \leq i \leq K\}$. *Observation about* E: If $(p, q) \in E$ and $p \subseteq A_i$, then $q \subseteq B_i$. *Proof:* Let $(p, q) \in E$ and $p \subseteq A_i$. By construction of E, for some witness elements $a \in \alpha(p)$, $b \in \alpha(q)$ we have $(a, b) \in \alpha(r)$. Because $\alpha(B_i = r[A_i])$, we have $b \in \alpha(B_i)$. Because $\alpha(q)$ and $\alpha(B_i)$ intersect, $q \subseteq B_i$, completing the proof of the observation.

We can now apply the sparse model construction of QFBAPA to the F_A part to pick a sparse set of cubes $C_A \supseteq C_{A0}$. Treat again the values of integer variables in F_A as constant, but then also in the resulting non-redundant integer cone generator replace the cardinalities of variables denoting sizes of each selected cube in $p \in C_{A0}$ by the constant $|\alpha(p)|$, thus removing these variables from the integer equation and removing the corresponding elements from the universe $\mathcal{U}_\mathcal{A}$. Solve the remaining equations to obtain a sparse solution for the simplified F_A formula, again using the results on sparse solutions of such Presburger arithmetic formulas [8, 16]. We obtain a sparse solution that gives a polynomial number of non-empty cubes C_{A1}. We use the obtained values to define $\alpha_1(p)$ for $p \in C_{A1}$. We let $\alpha_1(p) = \alpha(p)$ for $p \in C_{A0}$. Let $C_A = C_{A0} \cup C_{A1}$. Define $\alpha_1(p) = \emptyset$ for $p \notin C_A$. This yields the sparse interpretation α_1, where only cubes in $C_B \cup C_A$ are non-empty and where $\alpha_1(F_B \wedge F_A \wedge P)$ holds.

Finally, define define $\alpha_1(r)$ as a completed model $\alpha_1 = \bigcup\{p \times q \mid (p, q) \in E\}$ where E is defined (by edges (p_{ji}, q_j)) above. We claim $\alpha_1(F_C)$. Indeed, consider a set A_i. Then A_i is union of certain cubes from C_{A0} and certain cubes from C_{A1}. Because E has no outgoing edges for C_{A1}, we have $\alpha(r[\bigcup C_{A1}]) = \emptyset$. Therefore,

$$\alpha_1(r[A_i]) = \alpha_1(r[\cup\{p \mid p \in C_{A0}, p \subseteq A_i\}]) = \alpha_1(\cup\{q \mid \exists p. p \subseteq A_i \wedge (p, q) \in E\}$$

By the above *Observation about* E, we have that for each q above (belonging to $E[\{p\}]$) the condition $q \subseteq B_i$ holds. Therefore $\alpha_1(r[A_i]) \subseteq \alpha(B_i)$. For the converse set inclusion, let $b \in \alpha_1(B_i)$ be arbitrary and let $q_j \in C_B$ be such that $b \in \alpha_1(q_j)$ and $q_j \subseteq B_i$. Note that $\alpha_1(p_{ji}) \neq \emptyset$, so there exists $a \in \alpha_1(p_{ji})$. Then $(a, b) \in \alpha_1(r)$. Because $p_{ji} \subseteq A_i$, we have $b \in \alpha_1(r[A_i])$. Thus, $\alpha(B_i) \subseteq \alpha_1(r[A_i])$ and the therefore $\alpha_1(r[A_i] = B_i)$. Because i was arbitrary, α_1 is a sparse model for the entire formula. ■

Theorem 6. *The satisfiability for* QFBAPA−R2 *is NP complete.*

Proof. (Sketch) NP-hardness follows because QFBAPA−R2 subsumes propositional logic. To show NP membership, we use the sparse model property from the previous theorem: we non-deterministically guess a subset of non-empty sets A_i, then guess a polynomial subset C_B of Venn regions over B_i, using the polynomial bounds from [16]. We then guess the subset C_{A0} bounded by $K|C_B|$ and guess C_{A1} conservatively bounded by the same bound as in [16]. Finally, we guess a graph E whose number of edges is bounded by $|C_B|(|C_{A0}|+|C_{A1}|)$. Given such a guess, we can compute a formula that describes all Boolean Algebra expressions and all images of non-empty relations fragments under non-empty Venn regions, and thus describes the existence of a model for this guess of Venn regions and relation between them. As in [16], the entire guessing process can be compiled into a polynomially large quantifier-free formula of Presburger arithmetic with conditional expressions. ■

We next discuss some extensions of the two-sorted fragment.

NP extensions. Consider any finite number of sorts $s_1, \ldots, s_n$ related by a strict total ordering, and any number of relations of sorts $s_i \times s_{i+1}$ for $0 \leq i < n$. We can then repeat the construction above, starting with relations of sorts $s_{n-1} \times s_n$ and moving towards relations of sort $s_1 \times s_2$. For a fixed number of sorts, we obtain NP complexity. In fact, we can repeatedly apply the sparsity theorem in the case of multiple sorts and multiple relations forming a directed acyclic graph over the sorts.

Limits of membership in NP. Note that if we consider a chain of relations whose sorts form a cycle, through repeated composition we can simulate relations of sort $s \times s$. In this case the above NP construction fails. Moreover, the EXPTIME lower bound follows for such language from the lower bound on the complexity of the ALC Description Logic with general TBox inclusion axioms [1, Theorem 3.27].

5 Logic of Multiset Images of Functions

In this section we illustrate that some of the techniques of the previous section generalize from sets to *multisets*. A multiset M is a function $M : E \to \mathbb{N}$ mapping the set of elements into the non-negative number of their occurrences. The first NP decision procedure for multisets with the cardinality operator was presented in [23]. In this section we extend the logic of multisets with cardinalities to also include a function image operator that maps a set into a multiset.

We define the function image of a set A to be a multiset $f[A] : E \to \mathbb{N}$ such that $(f[A])(e) = |\{x.\ x \in A \wedge f(x) = e\}|$. The set of distinct elements occurring in a multiset is obtained using the set operator: $\mathsf{set}(M) = \{x.\ M(x) > 0\}$. This way $\mathsf{set}(f[B])$ is the set corresponding to the standard notion of function image used in previous sections.

$$
\begin{array}{l}
F ::= A \mid F \vee F \mid \neg F \\
A ::= B \subseteq B \mid M \subseteq M \mid T \leq T \mid K \ \mathsf{dvd}\ T \\
B ::= x \mid \emptyset \mid \mathcal{U} \mid B \cup B \mid B \cap B \mid B^c \mid \mathsf{set}(M) \\
M ::= m \mid \emptyset_M \mid M \cap M \mid M \cup M \mid M \uplus M \mid M \setminus M \mid M \setminus\setminus M \mid \mathsf{mset}(B) \mid f[B] \\
T ::= k \mid K \mid \mathsf{MAXC} \mid T_1 + T_2 \mid K \cdot T \mid |B| \mid |M| \\
K ::= \cdots \mid -2 \mid -1 \mid 0 \mid 1 \mid 2 \mid \cdots
\end{array}
$$

Fig. 6. MAPA-Fun logic of multisets, cardinality operator, and multiset images of sets

Figure 6 shows the logic that embeds the logic of multisets [22, Figure 1], [23], and extends it with the multiset image operator. The logic distinguishes the sorts of sets and multisets, but also includes a casting function $\mathsf{mset}(B)$ which treats a set as a multiset, and an abstraction function $\mathsf{set}(M)$ which extracts the set of distinct elements that occur in a multiset. Unlike the previous section, we do not have disjointness of domains and ranges of functions, and, in terms of expressive power, we effectively treat sets as a special case of multisets.

Given a formula F in the language described in Figure 6, a decision procedure for F works as follows:

1. Apply the algorithm in Figure 7 to translate F into an equisatisfiable multiset formula F' in the syntax given in Figure 1 in [22]. In this step we eliminate function symbols in a way similar to that described in Section 3. The new formula F' has size singly exponential in the size of F;
2. invoke on the formula F' the decision procedure described in [23]. The decision procedure runs in NP time.

The entire procedure runs in NEXPTIME. The lower bound proof from Section 3.2 applies in this case as well, so we conclude that our logic is NEXPTIME-complete.

The correctness of the reduction is stated in the following theorem.

Theorem 7. *Given a formula F as an input to the algorithm described in Figure 7, let the formula F' be its output. Then formulas F and F' are equisatisfiable and their satisfying assignments have the same projections on the set and multiset variables occurring in F.*

Proof. Given a model for F, we construct a model for F' by interpreting M_i as $f[s_i]$. Conversely, let α be a model for F'. We can define f on each disjoint set s_i independently. Because $|s_i| = |M_i|$ holds in the model, we can enumerate

INPUT: formula in the syntax of Figure 6
OUTPUT: multiset formula in the syntax of Figure 1 in [22]

1. Flatten expressions containing the operator set:
   $C[\ldots \mathsf{set}(M) \ldots] \rightsquigarrow (B_F = \mathsf{set}(M) \wedge C[\ldots B_F \ldots])$
   where the occurrence of $\mathsf{set}(M)$ is not already in a top-level conjunct of the form $B = \mathsf{set}(M)$ for some set variable B and B_F is a fresh unused set variable
2. Let S be the set of variables occurring in the formula
   Define the set $S_N = \{s_1, \ldots, s_Q\}$ of Venn regions over elements of S
3. Rewrite each set expression as a disjoint union of the Venn regions from S_N
4. **Eliminate function symbols:**
   $C[\ldots f[s_{i_1} \cup \ldots \cup s_{i_k}] \ldots] \rightsquigarrow C[\ldots (M_{i_1} \uplus \ldots \uplus M_{i_k}) \ldots]$
   where each M_{i_j} is a fresh multiset variable denotes $f[s_{i_j}]$
5. Add the conjuncts which states a necessary condition for $M_{i_j} = f[s_{i_j}]$
   $F \rightsquigarrow F \wedge \bigwedge_{i=1}^{Q} |s_i| = |M_i|$
6. Add the conjuncts which state that s_{i_j} are disjoint sets
   $F \rightsquigarrow F \wedge \forall e. \bigwedge_{i=1}^{Q} (s_i(e) = 0 \vee s_i(e) = 1) \wedge \bigwedge_{i \neq j} (s_i \cap s_j = \emptyset)$

Fig. 7. Algorithm for reducing a MAPA-Fun formula to a MAPA formula

both s_i and M_i into sequences $a_1, \ldots, a_K$ and $b_1, \ldots, b_K$ of same length. This enumeration defines a function assigning a_j to b_j for $1 \leq j \leq K$ such that $f[s_i] = M_i$. ∎

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. CUP, 2003.
2. A. Banerjee, D. A. Naumann, and S. Rosenberg. Regional logic for local reasoning about global invariants. In *ECOOP '08: Proceedings of the 22nd European conference on Object-Oriented Programming*, pages 387–411, Berlin, Heidelberg, 2008. Springer-Verlag.
3. M. Barnett and K. R. M. Leino. Weakest-precondition of unstructured programs. In *PASTE*, pages 82–87, 2005.
4. D. Beyer, T. A. Henzinger, R. Jhala, and R. Majumdar. The software model checker BLAST. *STTT*, 9(5-6):505–525, 2007.
5. E. Cohen, M. Dahlweid, M. Hillebrand, D. Leinenbach, M. Moskal, T. Santen, W. Schulte, and S. Tobies. VCC: A practical system for verifying concurrent c. In *Conf. Theorem Proving in Higher Order Logics (TPHOLs)*, volume 5674 of *LNCS*, 2009.
6. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *POPL*, 1979.
7. R. K. Dewar. Programming by refinement, as exemplified by the SETL representation sublanguage. *ACM TOPLAS*, July 1979.
8. F. Eisenbrand and G. Shmonin. Carathéodory bounds for integer cones. *Operations Research Letters*, 34(5):564–568, September 2006.

9. S. Feferman and R. L. Vaught. The first order properties of products of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.
10. R. Givan, D. McAllester, C. Witty, and D. Kozen. Tarskian set constraints. *Inf. Comput.*, 174(2):105–131, 2002.
11. S. Gulwani, T. Lev-Ami, and M. Sagiv. A combination framework for tracking partition sizes. In *POPL '09*, pages 239–251, 2009.
12. Y. Gurevich and S. Shelah. Spectra of monadic second-order formulas with one unary function. In *LICS*, pages 291–300, 2003.
13. V. Kuncak, P. Lam, K. Zee, and M. Rinard. Modular pluggable analyses for data structure consistency. *IEEE Trans. Software Engineering*, 32(12), December 2006.
14. V. Kuncak, H. H. Nguyen, and M. Rinard. Deciding Boolean Algebra with Presburger Arithmetic. *J. of Automated Reasoning*, 2006.
15. V. Kuncak and M. Rinard. Decision procedures for set-valued fields. In *1st International Workshop on Abstract Interpretation of Object-Oriented Languages*, 2005.
16. V. Kuncak and M. Rinard. Towards efficient satisfiability checking for Boolean Algebra with Presburger Arithmetic. In *CADE-21*, 2007.
17. H. R. Lewis. Complexity results for classes of quantificational formulas. *J. Comput. Syst. Sci.*, 21(3):317–353, 1980.
18. Y. V. Matiyasevich. Enumerable sets are Diophantine. *Soviet Math. Doklady*, 11(2):354–357, 1970.
19. H. J. Ohlbach and J. Koehler. Modal logics, description logics and arithmetic reasoning. *Artificial Intelligence*, 109:1–31, 1999.
20. L. Pacholski, W. Szwast, and L. Tendera. Complexity results for first-order two-variable logic with counting. *SIAM J. on Computing*, 29(4):1083–1117, 2000.
21. J. A. N. Pérez, A. Rybalchenko, and A. Singh. Cardinality abstraction for declarative networking applications. In *CAV*, pages 584–598, 2009.
22. R. Piskac and V. Kuncak. Decision procedures for multisets with cardinality constraints. In *VMCAI*, number 4905 in LNCS, 2008.
23. R. Piskac and V. Kuncak. Linear arithmetic with stars. In *CAV*, 2008.
24. I. Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language and Information*, 14(3):369–395, 2005.
25. T. Reps, M. Sagiv, and G. Yorsh. Symbolic implementation of the best transformer. In *Proc. 5th International Conference on Verification, Model Checking and Abstract Interpretation*, 2004.
26. J. Venn. On the diagrammatic and mechanical representation of propositions and reasonings. *Dublin Philosophical Magazine and Journal of Science*, 9(59):1–18, 1880.
27. T. Wies, V. Kuncak, P. Lam, A. Podelski, and M. Rinard. Field constraint analysis. In *Proc. Int. Conf. Verification, Model Checking, and Abstract Interpratation*, 2006.
28. T. Wies, R. Piskac, and V. Kuncak. Combining theories with shared set operations. In *FroCoS: Frontiers in Combining Systems*, 2009.
29. K. Zee, V. Kuncak, and M. Rinard. Full functional verification of linked data structures. In *ACM PLDI*, 2008.