The Bernays–Schönfinkel–Ramsey Fragment with Bounded Difference Constraints over the Reals Is Decidable

Marco Voigt

The Bernays–Schönfinkel–Ramsey Fragment with Bounded Difference Constraints over the Reals is Decidable

Marco Voigt

Max Planck Institute for Informatics and Saarbrücken Graduate School of Computer Science, Saarland Informatics Campus, Saarbrücken, Germany

Abstract. First-order linear real arithmetic enriched with uninterpreted predicate symbols yields an interesting modeling language. However, satisfiability of such formulas is undecidable, even if we restrict the uninterpreted predicate symbols to arity one. In order to find decidable fragments of this language, it is necessary to restrict the expressiveness of the arithmetic part. One possible path is to confine arithmetic expressions to difference constraints of the form $x - y \triangleleft c$, where $\triangleleft$ ranges over the standard relations $<, \leq, =, \neq, \geq, >$ and x, y are universally quantified. However, it is known that combining difference constraints with uninterpreted predicate symbols yields an undecidable satisfiability problem again. In this paper, it is shown that satisfiability becomes decidable if we in addition bound the ranges of universally quantified variables. As bounded intervals over the reals still comprise infinitely many values, a trivial instantiation procedure is not sufficient to solve the problem.

Keywords: Bernays–Schönfinkel–Ramsey fragment, linear arithmetic constraints, difference constraints, combination of theories

1 Introduction

It has been discovered about half a century ago that linear arithmetic with additional uninterpreted predicate symbols has an undecidable satisfiability problem [15]. Even enriching Presburger arithmetic with only a single uninterpreted predicate symbol of arity one suffices to facilitate encodings of the halting problem for two-counter machines [5, 10]. These results do not change substantially when we use the reals as underlying domain instead of the integers. This means, in order to obtain a decidable subfragment of the combination of linear arithmetic with uninterpreted predicate symbols, the arithmetic part has to be restricted considerably. In this paper, two subfragments with a decidable satisfiability problem are presented. Both are based on the Bernays–Schönfinkel–Ramsey fragment (BSR) of first-order logic, which is the $\exists^*\forall^*$ prefix class. Uninterpreted constant symbols and the distinguished equality predicate are allowed, non-constant function symbols are not. The arity of uninterpreted predicate symbols is not restricted. We extend BSR in two ways and call the obtained clause fragments *BSR modulo simple linear real constraints—BSR(SLR)*—and *BSR modulo bounded difference constraints—BSR(BD).*

The first clause class—defined in Definition 1 and treated in detail in Section 4—adds constraints of the form $s \lhd t$, $x \lhd t$, and $x \lhd y$ to BSR clauses, where x and y are real-valued variables that are implicitly universally quantified, s and t are linear arithmetic terms that are ground, and $\lhd$ ranges over $<, \leq, =, \neq, \geq, >$. We allow Skolem constants in the ground terms s and t. Since their value is not predetermined, they can be conceived as being existentially quantified. The constraints used in this clause fragment are similar to the kind of constraints that appear in the context of the *array property fragment* [4] and extensions thereof (see, e.g., [7, 9]). The main differences are that we use the real domain in this paper instead of the integer domain, and that we allow strict inequalities and disequations between universally quantified variables. In the presence of uninterpreted function symbols, strict inequality or disequations can be used to assert that some uninterpreted function f is injective. This expressiveness prevents certain instantiation-based approaches to satisfiability checking from being applicable, e.g. the methods in [4, 9]. In the context of the array property fragment, this expressiveness even leads to undecidability.

The BSR(BD) clause class—presented in Definition 2 and in Section 5—adds constraints of the form $x \lhd c$, $x \lhd y$ and $x - y \lhd c$ to BSR clauses, where x and y are real-valued variables, c could be any rational number, and $\lhd$ ranges over $<, \leq, =, \neq, \geq, >$ again. We refer to constraints of the form $x - y \lhd c$ as *difference constraints*. Already in the seventies, Pratt identified difference constraints and boolean combinations thereof as an important tool for the formalization of verification conditions [14]. Applications include the verification of timed systems and scheduling problems (see, e.g., [11] for references). As unrestricted combinations of uninterpreted predicate symbols with difference constraints lead to an undecidable satisfiability problem (once more, two-counter machines can be encoded in a simple way [17]), we have to further confine the language. Every difference constraint $x - y \lhd c$ has to be conjoined with four additional constraints $c_x \leq x$, $x \leq d_x$, $c_y \leq y$, $y \leq d_y$, where c_x, d_x, c_y, d_y are rationals. This restriction seems to weaken expressiveness severely. Indeed, it has to, since we aim for a decidable satisfiability problem. Yet, we show in Section 6 that BSR(BD) clause sets are expressive enough to formulate the reachability problem for timed automata. In [13] an encoding of the reachability problem for timed automata in *difference logic* (boolean combinations of difference constraints *without* uninterpreted predicate symbols) is given, which facilitates deciding bounded reachability, i.e. the problem of reaching a given set of states within a bounded number of transition steps. When using BSR(BD) as a modeling language, we do not have to fix an upper bound on the number of steps a priori.

The main result of the present paper is that satisfiability of finite BSR(SLR) clause sets and finite BSR(BD) clause sets is decidable, respectively (Theorems 12 and 19). The proof technique is very similar for the two fragments. It is partially based on methods from Ramsey theory, which are briefly introduced in Section 3. The used approach may turn out to be applicable to other fragments of BSR modulo linear real arithmetic as well. Due to space limitations, most proofs are only sketched. Detailed proofs can be found in [16].

2 Preliminaries and notation

Hierarchic combinations of first-order logic with background theories build upon sorted logic with equality [2, 3, 12]. We instantiate this framework with the BSR fragment and linear arithmetic over the reals as the *base theory*. The *base sort* $\mathcal{R}$ shall always be interpreted by the reals $\mathbb{R}$. For simplicity, we restrict our considerations to a single *free sort* $\mathcal{S}$, which may be freely interpreted as some nonempty domain, as usual.

We denote by $V_{\mathcal{R}}$ a countably infinite set of base-sort variables. *Linear arithmetic (LA) terms* are build from rational constants $0, 1, \frac{1}{2}, -2, -\frac{3}{4}$, etc., the operators $+, -$, and the variables from $V_{\mathcal{R}}$. We moreover allow base-sort constant symbols whose values have to be determined by an interpretation (*Skolem constants*). They can be conceived as existentially quantified. As predicates over the reals we allow the standard relations $<, \leq, =, \neq, \geq, >$.

In order to hierarchically extend the base theory by the BSR fragment, we introduce the free sort $\mathcal{S}$, a countably infinite set $V_{\mathcal{S}}$ of *free-sort variables*, a finite set $\Omega_{\mathcal{S}}$ of *free (uninterpreted) constant symbols of sort* $\mathcal{S}$ and a finite set Π of *free predicate symbols* equipped with sort information. Note that every predicate symbol in Π has a finite, nonnegative arity and can be of a mixed sort over the two sorts $\mathcal{R}$ and $\mathcal{S}$, e.g. $P : \mathcal{R} \times \mathcal{S} \times \mathcal{R}$. We use the symbol $\approx$ to denote the built-in equality predicate on $\mathcal{S}$. To avoid confusion, we tacitly assume that no constant or predicate symbol is overloaded, i.e. they have a unique sort.

Definition 1 (BSR with simple linear real constraints—BSR(SLR)). *A* BSR(SLR) clause *has the form $\Lambda \parallel \Gamma \to \Delta$, where Λ, Γ, Δ are multisets of atoms satisfying the following conditions. (i) Every atom in Λ is an LA constraint of the form $s \lhd t$ or $x \lhd t$ or $x \lhd y$ where s, t are ground (i.e. variable-free) LA terms, $x, y \in V_{\mathcal{R}}$, and $\lhd \in \{<, \leq, =, \neq, \geq, >\}$. (ii) Every atom in Γ and Δ is either an equation $s \approx s'$ over free-sort variables and constant symbols, or a non-equational atom $P(s_1, \ldots, s_m)$ that is well sorted and where the s_i range over base-sort variables, free-sort variables, and free-sort constant symbols.*

Definition 2 (BSR with bounded difference constraints—BSR(BD)). *A* BSR(BD) clause *has the form $\Lambda \parallel \Gamma \to \Delta$, where the multisets Γ, Δ satisfy Condition (ii) of Definition 1, and every atom in Λ is an LA constraint of the form $x \lhd c$, $x \lhd y$, or $x - y \lhd c$ where c may be any rational constant (not a Skolem constant), $x, y \in V_{\mathcal{R}}$, and $\lhd \in \{<, \leq, =, \neq, \geq, >\}$. Moreover, we require that whenever Λ contains a constraint of the form $x - y \lhd c$, then Λ also contains constraints $c_x \leq x$, $x \leq d_x$, $c_y \leq y$, and $y \leq d_y$ with $c_x, d_x, c_y, d_y \in \mathbb{Q}$.*

We omit the empty multiset left of "$\to$" and denote it by $\Box$ right of "$\to$" (where $\Box$ at the same time stands for *falsity*). The introduced clause notation separates arithmetic constraints from the free first-order part. We use the vertical double bar "$\parallel$" to indicate this syntactically. Intuitively, clauses $\Lambda \parallel \Gamma \to \Delta$ can be read as $\left(\bigwedge \Lambda \wedge \bigwedge \Gamma\right) \to \bigvee \Delta$, i.e. the multisets Λ, Γ stand for conjunctions of atoms and Δ stands for a disjunction of atoms. Requiring the free parts Γ and Δ of clauses to not contain any base-sort terms apart from variables does not limit expressiveness. Every base-sort term $t \notin V_{\mathcal{R}}$ in the free part can safely be replaced by a fresh base-sort variable x_t when an atomic constraint $x_t = t$ is added to the constraint part of the clause (a process known as *purification* or *abstraction* [2, 12]).

A *(hierarchic) interpretation* is an algebra $\mathcal{A}$ which interprets the base sort $\mathcal{R}$ as $\mathcal{R}^{\mathcal{A}} = \mathbb{R}$, assigns real values to all occurring base-sort Skolem constants and interprets all LA terms and constraints in the standard way. Moreover, $\mathcal{A}$ comprises a nonempty domain $\mathcal{S}^{\mathcal{A}}$, assigns to each free-sort constant symbol c in $\Omega_{\mathcal{S}}$ a domain element $c^{\mathcal{A}} \in \mathcal{S}^{\mathcal{A}}$, and interprets every sorted predicate symbol $P : \xi_1 \times \ldots \times \xi_m$ in Π by some set $P^{\mathcal{A}} \subseteq \xi_1^{\mathcal{A}} \times \ldots \times \xi_m^{\mathcal{A}}$. Summing up, $\mathcal{A}$ extends the standard model of linear arithmetic and adopts the standard approach to semantics of (sorted) first-order logics when interpreting the free part of clauses.

Given an interpretation $\mathcal{A}$ and a sort-respecting *variable assignment* β : $V_{\mathcal{R}} \cup V_{\mathcal{S}} \to \mathcal{R}^{\mathcal{A}} \cup \mathcal{S}^{\mathcal{A}}$, we write $\mathcal{A}(\beta)(s)$ to mean the *value of the term s under $\mathcal{A}$ with respect to the variable assignment β*. The variables occurring in clauses are implicitly universally quantified. Therefore, given a clause C, we call $\mathcal{A}$ a *(hierarchic) model of C*, denoted $\mathcal{A} \models C$, if and only if $\mathcal{A}, \beta \models C$ holds for every variable assignment β. For clause sets N, we write $\mathcal{A} \models N$ if and only if $\mathcal{A} \models C$ holds for every clause $C \in N$. We call a clause C (a clause set N) *satisfiable* if and only if there exists a model $\mathcal{A}$ of C (of N). Two clauses C, D (clause sets N, M) are *equisatisfiable* if and only if C (N) is satisfiable whenever D (M) is satisfiable and vice versa.

Given a BSR(SLR) or BSR(BD) clause C, we use the following notation: the set of all constant symbols occurring in C is denoted by consts(C). The set bconsts(C) (fconsts(C)) is the restriction of consts(C) to base-sort (free-sort) constant symbols. We denote the set of all variables occurring in a clause C by vars(C). The same notation is used for sets of clauses.

Definition 3 (Normal form of BSR(SLR) and BSR(BD) clauses). *A BSR(SLR) or BSR(BD) clause $\Lambda \,\|\, \Gamma \to \Delta$ is in* normal form *if (1) all non-ground atoms in Λ have the form $x \lhd c$, $x \lhd y$, or $x - y \lhd c$ where c is a rational constant or a Skolem constant, and (2) every variable that occurs in Λ also occurs in Γ or in Δ. A BSR(SLR) or BSR(BD) clause set N is in* normal form *if all clauses in N are in normal form and pairwise variable disjoint. Moreover, we assume that N contains at least one free-sort constant symbol.*

For BSR(SLR) clause sets, we pose the following additional requirement. N can be divided into two parts N_{def} and N' such that (a) every clause in N_{def} has the form $c \neq t \,\| \; \to \Box$ where c is a Skolem constant and t is some ground LA term, and (b) any ground atom $s \lhd t$ in any constraint part Λ in any clause $\Lambda \,\|\, \Gamma \to \Delta$ in N' is such that s and t are constants (Skolem or rational, respectively).

For every BSR(SLR) clause set N there is an equisatisfiable BSR(SLR) clause set N' in normal form, such that $N' \models N$. The same holds for BSR(BD) clause sets. Requirement (2) can be established by any procedure for eliminating existentially quantified variables in LA constraints (see, e.g., [6]). Establishing the other requirements is straightforward.

For two sets $R, Q \subseteq \mathbb{R}$ we write $R < Q$ if $r < q$ holds for all $r \in R$ and $q \in Q$. Given a real r, we denote the *integral part of* r by $\lfloor r \rfloor$, i.e. $\lfloor r \rfloor$ is the largest integer for which $\lfloor r \rfloor \leq r$. By $\mathrm{fr}(r)$ we denote the *fractional part of* r, i.e. $\mathrm{fr}(r) := r - \lfloor r \rfloor$. Notice that $\mathrm{fr}(r)$ is always nonnegative, e.g. $\mathrm{fr}(3.71) = 0.71$, whereas $\mathrm{fr}(-3.71) = 0.29$. Given any tuple $\bar{r}$ of reals, we write $\mathrm{fr}(\bar{r})$ to mean the corresponding tuple of fractional parts, i.e. $\mathrm{fr}\big(\langle r_1, \ldots, r_\mu \rangle\big) := \big\langle \mathrm{fr}(r_1), \ldots, \mathrm{fr}(r_\mu) \big\rangle$. We use the notation $\lfloor \bar{r} \rfloor$ in a component-wise fashion as well.

We write $[k]$ to address the set $\{1, \ldots, k\}$ for any positive integer $k > 0$. Finally, $\mathcal{P}$ denotes the power set operator, i.e. for any set S, $\mathcal{P}(S)$ denotes the set of all subsets of S.

3 Basic tools from Ramsey theory

In this section we establish two technical results based on methods usually applied in Ramsey theory. We shall use these results later on to prove the existence of models of a particular kind for finite and satisfiable BSR(SLR) or BSR(BD) clause sets. These models meet certain uniformity conditions. In order to construct them, we rely on the existence of certain finite subsets of $\mathbb{R}$ that are used to construct prototypical tuples of reals. These finite subsets, in turn, have to behave nicely as well, since tuples that are not distinguishable by BSR(SLR) or BSR(BD) constraints are required to have certain uniformity properties.

A tuple $\langle r_1, \ldots, r_m \rangle \in \mathbb{R}^m$ is called *ascending* if $r_1 < \ldots < r_m$. A *coloring* is a mapping $\chi : S \to \mathcal{C}$ for some arbitrary set S and some finite set $\mathcal{C}$. For the most basic result of this section (Lemma 4), we consider an arbitrary coloring χ of m-tuples of real numbers and stipulate the existence of a finite subset $Q \subseteq \mathbb{R}$ of a given cardinality n such that all ascending m-tuples of elements from Q are assigned the same color by χ.

Lemma 4. *Let $n, m > 0$ be positive integers. Let $\chi : \mathbb{R}^m \to \mathcal{C}$ be some coloring. There is some positive integer $\widehat{n}$ such that for every set $R \subseteq \mathbb{R}$ with $|R| \geq \widehat{n}$—i.e. R needs to be sufficiently large—there exists a subset $Q \subseteq R$ of cardinality n such that all ascending tuples $\langle r_1, \ldots, r_m \rangle \in Q^m$ are assigned the same color by χ.*

Proof (adaptation of the proof of Ramsey's Theorem on page 7 in [8]). For $n < m$ the lemma is trivially satisfied, since in this case Q^m cannot contain ascending tuples. Hence, we assume $n \geq m$. In order to avoid technical difficulties when defining the sequence of elements $s_{m-1}, s_m, s_{m+1}, \ldots$ below, we assume for the rest of the proof that R is finite but sufficiently large. This assumption does not pose a restriction, as we can always consider a sufficiently large finite subset of R, if R were to be infinite.

We proceed by induction on $m \geq 1$. The base case $m = 1$ is easy, since χ can assign only finitely many colors to elements in R and thus some color must be assigned at least $\big\lfloor \frac{|R|}{|\mathcal{C}|} \big\rfloor$ times. Hence, if R contains at least $n|\mathcal{C}|$ elements, we find a uniformly colored subset Q of size n. Suppose $m > 1$. At first, we pick the $m-2$ smallest reals $s_1 < \ldots < s_{m-2}$ from R and set $S_{m-2} := R \setminus \{s_1, \ldots, s_{m-2}\}$. Thereafter, we simultaneously construct two *sufficiently long but finite* sequences

$s_{m-1}, s_m, s_{m+1}, \ldots$ and $S_{m-1}, S_m, S_{m+1}, \ldots$ as follows:
Given S_i, we define s_{i+1} to be the smallest real in S_i.
Given S_i and the element s_{i+1}, we define an equivalence relation $\sim_i$ on the set $S'_i := S_i \setminus \{s_{i+1}\}$ so that $s \sim_i s'$ holds if and only if for every sequence of indices $j_1, \ldots, j_{m-1}$ with $1 \leq j_1 < \ldots < j_{m-1} \leq i+1$, we have $\chi(s_{j_1}, \ldots, s_{j_{m-1}}, s) = \chi(s_{j_1}, \ldots, s_{j_{m-1}}, s')$. This equivalence relation partitions S'_i into at most $|\mathcal{C}|^{\binom{i+1}{m-1}}$ equivalence classes. We choose one such class with largest cardinality to be S_{i+1}.

By construction of the sequence $s_1, s_2, s_3, \ldots$, we must have $\chi(s_{j_1}, \ldots, s_{j_{m-1}}, s_k) = \chi(s_{j_1}, \ldots, s_{j_{m-1}}, s_{k'})$ for every sequence of indices $j_1 < \ldots < j_{m-1}$ and all indices $k, k' \geq j_{m-1} + 1$. Please note that this covers all ascending m-tuples in $\{s_1, s_2, s_3, \ldots\}^m$ starting with $s_{j_1}, \ldots, s_{j_m-1}$, i.e. they all share the same color. We now define a new coloring $\chi' : \{s_1, s_2, s_3, \ldots\}^{m-1} \to \mathcal{C}$ so that $\chi'(s_{j_1}, \ldots, s_{j_{m-1}}) := \chi(s_{j_1}, \ldots, s_{j_{m-1}}, s_{j_{m-1}+1})$ for every sequence of indices $j_1 < \ldots < j_{m-1}$ (in case of j_{m-1} being the index of the last element in the sequence $s_1, s_2, s_3, \ldots$, $\chi'(s_{j_1}, \ldots, s_{j_{m-1}})$ shall be an arbitrary color from $\mathcal{C}$). By induction, there exists a subset $Q \subseteq \{s_1, s_2, s_3, \ldots\}$ of cardinality n, such that every ascending $(m-1)$-tuple $\bar{r} \in Q^{m-1}$ is colored the same by χ'. The definition of χ' entails that now all ascending m-tuples $\bar{r}' \in Q^m$ are colored the same by χ. Hence, Q is the sought set. □

Based on Lemma 4, one can derive similar results for more structured ways of coloring tuples of reals. We shall employ such a structured coloring when proving that the satisfiability problem for finite BSR(SLR) clause sets is decidable. More precisely, the proof of Lemma 10 will rely on such a result. The technical details are elaborated in [16].

4 Decidability of satisfiability for BSR(SLR) clause sets

For the rest of this section we fix two positive integers $m, m' > 0$ and some finite BSR(SLR) clause set N in normal form. For the sake of simplicity, we assume that all uninterpreted predicate symbols P occurring in N have the sort $P : \mathcal{S}^{m'} \times \mathcal{R}^m$. This assumption does not limit expressiveness, as the arity of a predicate symbol P can easily be increased in an (un)satisfiability-preserving way by padding the occurring atoms with additional arguments. For instance, every occurrence of atoms $P(t_1, \ldots, t_m)$ can be replaced with $P(t_1, \ldots, t_m, v, \ldots, v)$ for some fresh variable v that is added sufficiently often as argument.

Given the BSR(SLR) clause set N, every interpretation $\mathcal{A}$ induces a partition of $\mathbb{R}$ into finitely many intervals: the interpretations of all the rational and Skolem constants c occurring in N yield point intervals that are interspersed with and enclosed by open intervals.

Definition 5 **($\mathcal{A}$-induced partition of $\mathbb{R}$).** *Let $\mathcal{A}$ be an interpretation and let $r_1, \ldots, r_k$ be all the values in the set $\{c^{\mathcal{A}} \mid c \in \text{bconsts}(N)\}$ in ascending order. By $\mathcal{J}_{\mathcal{A}}$ we denote the following partition of $\mathbb{R}$:*

$$\mathcal{J}_{\mathcal{A}} := \bigl\{(-\infty, r_1), [r_1, r_1], (r_1, r_2), [r_2, r_2], \ldots, (r_{k-1}, r_k), [r_k, r_k], (r_k, +\infty)\bigr\}.$$

The idea of the following equivalence is that equivalent tuples are indistinguishable by the constraints that we allow in the BSR(SLR) clause set N.

Definition 6 ($\mathcal{J}_\mathcal{A}$-equivalence, $\sim_{\mathcal{J}_\mathcal{A}}$). *Let $\mathcal{A}$ be an interpretation and let k be a positive integer. We call two k-tuples $\bar{r}, \bar{q} \in \mathbb{R}^k$* $\mathcal{J}_\mathcal{A}$-equivalent *if*
(i) for every $J \in \mathcal{J}_\mathcal{A}$ and every i, $1 \leq i \leq k$, we have $r_i \in J$ if and only if $q_i \in J$ and
(ii) for all i, j, $1 \leq i, j \leq k$ we have $r_i < r_j$ if and only if $q_i < q_j$.
The induced equivalence relation on tuples of positive length is denoted by $\sim_{\mathcal{J}_\mathcal{A}}$.

For every positive k the relation $\sim_{\mathcal{J}_\mathcal{A}}$ induces only finitely many equivalence classes on the set of all k-tuples over the reals. We intend to show that, if N is satisfiable, then there is some model $\mathcal{A}$ for N which does not distinguish between different $\mathcal{J}_\mathcal{A}$-equivalent tuples. First, we need some notion that reflects how the interpretation $\mathcal{A}$ treats a given tuple $\bar{r} \in \mathbb{R}^m$. This role will be taken by the coloring $\chi_\mathcal{A}$, which maps $\bar{r}$ to a set of expressions of the form $P\bar{a}$, where P is some predicate symbol occurring in N and $\bar{a}$ is an m'-tuple of domain elements from $\mathcal{S}^\mathcal{A}$. The presence of $P\bar{a}$ in the set $\chi_\mathcal{A}(\bar{r})$ indicates that $\mathcal{A}$ interprets P in such a way that $P^\mathcal{A}$ contains the pair $\langle \bar{a}, \bar{r} \rangle$. In this sense, $\chi_\mathcal{A}(\bar{r})$ comprises all the relevant information that $\mathcal{A}$ contains regarding the tuple $\bar{r}$.

Definition 7 ($\mathcal{A}$-coloring $\chi_\mathcal{A}$). *Given an interpretation $\mathcal{A}$, let $\widehat{\mathcal{S}} := \{a \in \mathcal{S}^\mathcal{A} \mid a = c^\mathcal{A}$ for some $c \in \text{fconsts}(N)\}$ be the set of all domain elements assigned to free-sort constant symbols by $\mathcal{A}$. The* $\mathcal{A}$-coloring of $\mathbb{R}^m$ *is the mapping $\chi_\mathcal{A} : \mathbb{R}^m \to \mathcal{P}\{P\bar{a} \mid \bar{a} \in \widehat{\mathcal{S}}^{m'}$ and P is an uninterpreted predicate symbol in $N\}$ defined such that for every $\bar{r} \in \mathbb{R}^m$ we have $P\bar{a} \in \chi_\mathcal{A}(\bar{r})$ if and only if $\langle \bar{a}, \bar{r} \rangle \in P^\mathcal{A}$.*

Having the coloring $\chi_\mathcal{A}$ at hand, it is easy to formulate a uniformity property for a given interpretation $\mathcal{A}$. Two tuples $\bar{r}, \bar{r}' \in \mathbb{R}^m$ are treated *uniformly* by $\mathcal{A}$, if the colors $\chi_\mathcal{A}(\bar{r})$ and $\chi_\mathcal{A}(\bar{r}')$ agree. Put differently, $\mathcal{A}$ does not distinguish $\bar{r}$ from $\bar{r}'$.

Definition 8 ($\mathcal{J}_\mathcal{A}$-uniform interpretation). *An interpretation $\mathcal{A}$ is* $\mathcal{J}_\mathcal{A}$-uniform *if $\chi_\mathcal{A}$ colors each and every $\sim_{\mathcal{J}_\mathcal{A}}$-equivalence class uniformly, i.e. for all $\sim_{\mathcal{J}_\mathcal{A}}$-equivalent tuples $\bar{r}, \bar{r}'$ we have $\chi_\mathcal{A}(\bar{r}) = \chi_\mathcal{A}(\bar{r}')$.*

We next show that there exists a $\mathcal{J}_\mathcal{B}$-uniform model $\mathcal{B}$ of N, if N is satisfiable. Since such a model does not distinguish between $\mathcal{J}_\mathcal{B}$-equivalent m-tuples, and as there are only finitely many equivalence classes induced by $\sim_{\mathcal{J}_\mathcal{B}}$, only a finite amount of information is required to describe $\mathcal{B}$. This insight will give rise to a decision procedure that nondeterministically guesses how each and every equivalence class shall be treated by the uniform model.

Given some model $\mathcal{A}$ of N, the following lemma assumes the existence of certain finite sets Q_i with a fixed cardinality which are subsets of the open intervals in $\mathcal{J}_\mathcal{A}$. All $\mathcal{J}_\mathcal{A}$-equivalent m-tuples that can be constructed from the reals belonging to the Q_i are required to be colored identically by $\chi_\mathcal{A}$. The existence of the Q_i is the subject of Lemma 10.

Lemma 9. *Let λ be the maximal number of distinct base-sort variables in any single clause in N. In case of $\lambda < m$, we set $\lambda := m$. Let $\mathcal{A}$ be a model of N. Let $J_0, \ldots, J_\kappa$ be an enumeration of all open intervals in $\mathcal{J}_\mathcal{A}$ sorted in ascending order. Moreover, let $r_1, \ldots, r_\kappa$ be all reals in ascending order that define point intervals in $\mathcal{J}_\mathcal{A}$, i.e. $J_0 < [r_1, r_1] < J_1 < \ldots < [r_\kappa, r_\kappa] < J_\kappa$. Suppose we are given a collection of finite sets $Q_0, \ldots, Q_\kappa$ possessing the following properties:*
(i) $Q_i \subseteq J_i$ and $|Q_i| = \lambda$ for every i.
(ii) Let $Q := \bigcup_i Q_i \cup \{r_1, \ldots, r_\kappa\}$. For all $\mathcal{J}_\mathcal{A}$-equivalent m-tuples $\bar{q}, \bar{q}' \in Q^m$ we have $\chi_\mathcal{A}(\bar{q}) = \chi_\mathcal{A}(\bar{q}')$.
Then we can construct a model $\mathcal{B}$ of N that is $\mathcal{J}_\mathcal{B}$-uniform and that interprets the free sort $\mathcal{S}$ as a finite set.

Proof sketch.
Claim I: Let μ be a positive integer with $\mu \leq \lambda$. Every $\sim_{\mathcal{J}_\mathcal{A}}$-equivalence class over $\mathbb{R}^\mu$ contains some representative lying in Q^μ. $\Diamond$

Let $\widehat{\mathcal{S}}$ denote the set $\{a \in \mathcal{S}^\mathcal{A} \mid a = c^\mathcal{A}$ for some $c \in \text{fconsts}(N)\}$. We construct the interpretation $\mathcal{B}$ as follows: $\mathcal{S}^\mathcal{B} := \widehat{\mathcal{S}}$; $c^\mathcal{B} := c^\mathcal{A}$ for every constant symbol c; for every uninterpreted predicate symbol P and for all tuples $\bar{a} \in \widehat{\mathcal{S}}^{m'}$ and $\bar{s} \in \mathbb{R}^m$ we pick some tuple $\bar{q} \in Q^m$ with $\bar{q} \sim_{\mathcal{J}_\mathcal{A}} \bar{s}$, and we define $P^\mathcal{B}$ so that $\langle \bar{a}, \bar{s} \rangle \in P^\mathcal{B}$ if and only if $\langle \bar{a}, \bar{q} \rangle \in P^\mathcal{A}$. By construction, $\mathcal{B}$ is $\mathcal{J}_\mathcal{B}$-uniform.

It remains to show $\mathcal{B} \models N$. Consider any clause $C = \Lambda \parallel \Gamma \rightarrow \Delta$ in N and let β be any variable assignment ranging over $\mathcal{S}^\mathcal{B} \cup \mathbb{R}$. Starting from β, we derive a special variable assignment $\widehat{\beta}_C$ as follows. Let $x_1, \ldots, x_\ell$ be all base-sort variables in C. By Claim I, there is some tuple $\langle q_1, \ldots, q_\ell \rangle \in Q^\ell$ such that $\langle q_1, \ldots, q_\ell \rangle \sim_{\mathcal{J}_\mathcal{A}} \langle \beta(x_1), \ldots, \beta(x_\ell) \rangle$. We set $\widehat{\beta}_C(x_i) := q_i$ for every x_i. For all other base-sort variables, $\widehat{\beta}_C$ can be defined arbitrarily. For every free-sort variable u we set $\widehat{\beta}_C(u) := \beta(u)$.

As $\mathcal{A}$ is a model of N, we get $\mathcal{A}, \widehat{\beta}_C \models C$. By case distinction on why $\mathcal{A}, \widehat{\beta}_C \models C$ holds, one can infer $\mathcal{B}, \beta \models C$. Consequently, $\mathcal{B} \models N$. □

In order to show that uniform models always exist for satisfiable clause sets N, we still need to prove the existence of the sets Q_i mentioned in Lemma 9.

Lemma 10. *Let $\mathcal{A}$ be an interpretation. Let $r_1, \ldots, r_\kappa$ be all the reals defining point intervals in $\mathcal{J}_\mathcal{A}$ and let $J_0, \ldots, J_\kappa$ be all open intervals in $\mathcal{J}_\mathcal{A}$ such that $J_0 < [r_1, r_1] < J_1 < [r_2, r_2] < \ldots < J_{\kappa-1} < [r_\kappa, r_\kappa] < J_\kappa$. Let λ be a positive integer. There is a collection of finite sets $Q_0, \ldots, Q_\kappa$ such that Requirements (i) and (ii) of Lemma 9 are met.*

Proof sketch. We employ a more sophisticated variant of the Ramsey result stated in Lemma 4.
Claim I: There are sets $Q_0, \ldots, Q_\kappa$ satisfying Requirement (i) of Lemma 9 and the following conditions. For every Q_i, $0 \leq i \leq \kappa$, let $s_{\langle i,1 \rangle}, \ldots, s_{\langle i,\lambda \rangle}$ be all the values in Q_i in ascending order. Moreover, we set $s_{\langle \kappa+i,1 \rangle} := r_i$ for every i with $1 \leq i \leq \kappa$. Then, for every mapping $\varrho : [m] \rightarrow \{0, \ldots, 2\kappa\} \times [m]$ we have $\chi_\mathcal{A}(s_{\varrho(1)}, \ldots, s_{\varrho(m)}) = \chi_\mathcal{A}(s'_{\varrho(1)}, \ldots, s'_{\varrho(m)})$. $\Diamond$

One can show that for every $\sim_{\mathcal{J}_\mathcal{A}}$-equivalence class S over $\mathbb{R}^m$ there is some mapping $\varrho : [m] \to \{0, \ldots, 2\kappa\} \times [m]$ such that

(1) whenever $\varrho(i) = \langle k, \ell \rangle$ with $k > \kappa + 1$ then $\ell = 1$, and
(2) for all ascending tuples
$\bar{s}_0 = \langle s_{\langle 0,1 \rangle}, \ldots, s_{\langle 0,m \rangle} \rangle \in J_0^m; \ldots; \bar{s}_\kappa = \langle s_{\langle \kappa,1 \rangle}, \ldots, s_{\langle \kappa,m \rangle} \rangle \in J_\kappa^m$;
$\bar{s}_{\kappa+1} = \langle r_{\langle \kappa+1,1 \rangle} \rangle = \langle r_1 \rangle; \ldots; \bar{s}_{2\kappa} = \langle s_{\langle 2\kappa,1 \rangle} \rangle = \langle r_\kappa \rangle$
we have $\langle s_{\varrho(1)}, \ldots, s_{\varrho(m)} \rangle \in S$, and
(3) for every tuple $\langle q_1, \ldots, q_m \rangle \in S$ there exist ascending tuples $\bar{s}_1, \ldots, \bar{s}_{2\kappa}$ defined as in (2) such that $\langle q_1, \ldots, q_m \rangle = \langle s_{\varrho(1)}, \ldots, s_{\varrho(m)} \rangle$.

Consider any $\bar{q}, \bar{q}' \in S$. By (2), $\bar{q}$ can be written into $\langle s_{\varrho(1)}, \ldots, s_{\varrho(m)} \rangle$ for appropriate values $s_{\langle k,\ell \rangle}$ and $\bar{q}'$ can be represented by $\langle s'_{\varrho(1)}, \ldots, s'_{\varrho(m)} \rangle$ for appropriate $s'_{\langle k,\ell \rangle}$. Claim I entails $\chi_\mathcal{A}(\bar{q}) = \chi_\mathcal{A}(\langle s_{\varrho(1)}, \ldots, s_{\varrho(m)} \rangle) = \chi_\mathcal{A}(\langle s'_{\varrho(1)}, \ldots, s'_{\varrho(m)} \rangle) = \chi_\mathcal{A}(\bar{q}')$. □

Lemmas 9 and 10 together entail the existence of some $\mathcal{J}_\mathcal{A}$-uniform model $\mathcal{A} \models N$ with a finite free-sort domain $\mathcal{S}^\mathcal{A}$, if N is satisfiable.

Corollary 11. *If N has a model, then it has a model $\mathcal{A}$ that is $\mathcal{J}_\mathcal{A}$-uniform and that interprets the sort $\mathcal{S}$ as some finite set.*

Given any interpretation $\mathcal{A}$, the partition $\mathcal{J}_\mathcal{A}$ of the reals is determined by the rational constants in N and by the values that $\mathcal{A}$ assigns to the base-sort Skolem constants in N. Let $d_1, \ldots, d_\lambda$ be all the base-sort Skolem constants in N. If we are given some mapping $\gamma : \{d_1, \ldots, d_\lambda\} \to \mathbb{R}$, then γ induces a partition $\mathcal{J}_\gamma$, just as $\mathcal{A}$ induces $\mathcal{J}_\mathcal{A}$. We can easily verify whether N has a model $\mathcal{B}$ that is *compatible* with γ (i.e. $\mathcal{B}$ assigns the same values to $d_1, \ldots, d_\lambda$) and that is $\mathcal{J}_\mathcal{B}$-uniform. Due to the uniformity requirement, there is only a finite number of candidate interpretations that have to be checked.

Consequently, in order to show decidability of the satisfiability problem for finite BSR(SLR) clause sets in normal form, the only question that remains to be answered is whether it is sufficient to consider a finite number of assignments γ of real values to the Skolem constants in N. Recall that since N is in normal form, we can divide N into two disjoint parts N_{def} and N' such that all ground LA terms occurring in N' are either (Skolem) constants or rationals. Moreover, every clause in N_{def} constitutes a definition $c = t$ of some Skolem constant c. As far as the LA constraints occurring in N' are concerned, the most relevant information regarding the interpretation of Skolem constants is their ordering relative to one another and relative to the occurring rationals. This means, the clauses in N' cannot distinguish two assignments γ, γ' if
(a) for every Skolem constant d_i and every rational r occurring in N' we have (a.1) $\gamma(d_i) \leq r$ if and only if $\gamma'(d_i) \leq r$, and (a.2) $\gamma(d_i) \geq r$ if and only if $\gamma'(d_i) \geq r$, and
(b) for all d_i, d_j we have that $\gamma(d_i) \leq \gamma(d_j)$ if and only if $\gamma'(d_i) \leq \gamma'(d_j)$.

This observation leads to the following nondeterministic decision procedure for finite BSR(SLR) clause sets in normal form:

(1) Nondeterministically fix a total preorder $\preceq$ (reflexive and transitive) on the set of all base-sort Skolem constants and rational constants occurring in N'. Define a clause set $N_{\preceq}$ that enforces $\preceq$ for base-sort Skolem constants, i.e. $N_{\preceq} := \big\{ c > c' \parallel \to \Box \;\big|\; c \preceq c'$, either c or c' or both are Skolem constants$\big\}$.

(2) Check whether there is some mapping $\gamma : \{d_1, \ldots, d_\lambda\} \to \mathbb{R}$ such that γ is a solution for the clauses in $N_{\mathrm{def}} \cup N_{\preceq}$. (This step relies on the fact that linear arithmetic over existentially quantified variables is decidable.)

(3) If such an assignment γ exists, define an interpretation $\mathcal{B}$ as follows.

  (3.1) Nondeterministically define $\mathcal{S}^{\mathcal{B}}$ to be some subset of fconsts(N), i.e. use a subset of the Herbrand domain with respect to the free sort $\mathcal{S}$.

  (3.2) For every $e \in$ fconsts(N) nondeterministically pick some $a \in \mathcal{S}^{\mathcal{B}}$ and set $e^{\mathcal{B}} := a$.

  (3.3) Set $d_i^{\mathcal{B}} := \gamma(d_i)$ for every d_i.

  (3.4) For every uninterpreted predicate symbol P occurring in N nondeterministically define the set $P^{\mathcal{B}}$ in such a way that $\mathcal{B}$ is $\mathcal{J}_{\mathcal{B}}$-uniform.

(4) Check whether $\mathcal{B}$ is a model of N.

Theorem 12. *Satisfiability of finite BSR(SLR) clause sets is decidable.*

5 Decidability of satisfiability for BSR(BD) clause sets

Similarly to the previous section, we fix some finite BSR(BD) clause set N in normal form for the rest of this section, and we assume that all uninterpreted predicate symbols P occurring in N have the sort $P : \mathcal{S}^{m'} \times \mathcal{R}^m$. Moreover, we assume that all base-sort constants in N are integers. This does not lead to a loss of generality, as we could multiply all rational constants with the least common multiple of their denominators to obtain an equisatisfiable clause set in which all base-sort constants are integers. We could even allow Skolem constants, if we added clauses stipulating that every such constant is assigned a value that is (a) an integer and (b) is bounded from above and below by some integer bounds. Dropping any of these two restrictions leads to an undecidable satisfiability problem. For the sake of simplicity, we do not consider Skolem constants in this section.

Our general approach to decidability of the satisfiability problem for finite BSR(BD) clause sets is very similar to the path taken in the previous section. Due to the nature of the LA constraints in BSR(BD) clause sets, the employed equivalence relation characterizing indistinguishable tuples has to be a different one. In fact, we use one equivalence relation $\widehat{\simeq}_\kappa$ on the unbounded space $\mathbb{R}^m$ and another equivalence relation $\simeq_\kappa$ on the subspace $(-\kappa - 1, \kappa + 1)^m$ for some positive integer κ. Our definition of the relations $\simeq_\kappa$ and $\widehat{\simeq}_\kappa$ is inspired by the notion of clock equivalence used in the context of timed automata (see, e.g., [1]).

Definition 13 (bounded region equivalence $\simeq_\kappa$). *Let κ be a positive integer. We define the equivalence relation $\simeq_\kappa$ on $(-\kappa - 1, \kappa + 1)^m$ such that we get $\langle r_1, \ldots, r_m \rangle \simeq_\kappa \langle s_1, \ldots, s_m \rangle$ if and only if the following conditions are met:*
(i) For every i we have $\lfloor r_i \rfloor = \lfloor s_i \rfloor$, and $\mathit{fr}(r_i) = 0$ if and only if $\mathit{fr}(s_i) = 0$.
(ii) For all i, j we have $\mathit{fr}(r_i) \le \mathit{fr}(r_j)$ if and only if $\mathit{fr}(s_i) \le \mathit{fr}(s_j)$.

The relation $\simeq_\kappa$ induces only a finite number of equivalence classes over $(-\kappa-1,\kappa+1)^m$. Over $\mathbb{R}^m$, on the other hand, an analogous equivalence relation $\simeq_\infty$ would lead to infinitely many equivalence classes. In order to overcome this problem and obtain an equivalence relation over $\mathbb{R}^m$ that induces only a finite number of equivalence classes, we use the following compromise.

Definition 14 (unbounded region equivalence $\widehat{\simeq}_\kappa$). *Let κ be a positive integer. We define the equivalence relation $\widehat{\simeq}_\kappa$ on $\mathbb{R}^m$ in such a way that $\langle r_1,\dots,r_m\rangle \widehat{\simeq}_\kappa \langle s_1,\dots,s_m\rangle$ holds if and only if*
(i) for every i either $r_i > \kappa$ and $s_i > \kappa$, or $r_i < -\kappa$ and $s_i < -\kappa$, or the following conditions are met: (i.i) $\lfloor r_i \rfloor = \lfloor s_i \rfloor$ and (i.ii) $fr(r_i) = 0$ if and only if $fr(s_i) = 0$, and (ii) for all i, j
(ii.i) if $r_i, r_j > \kappa$ or $r_i, r_j < -\kappa$, then $r_i \leq r_j$ if and only if $s_i \leq s_j$,
(ii.ii) if $-\kappa \leq r_i, r_j \leq \kappa$, then $fr(r_i) \leq fr(r_j)$ if and only if $fr(s_i) \leq fr(s_j)$.

Obviously, the equivalence relations $\simeq_\kappa$ and $\widehat{\simeq}_\kappa$ coincide on the subspace $(-\kappa,\kappa)^m$. Over $(-\kappa-1,\kappa+1)^m$ the relation $\simeq_\kappa$ constitutes a proper refinement of $\widehat{\simeq}_\kappa$. Figure 1 depicts the equivalence classes induced by $\simeq_\kappa$ and $\widehat{\simeq}_\kappa$ in a two-dimensional setting for $\kappa = 1$. We need both relations in our approach.

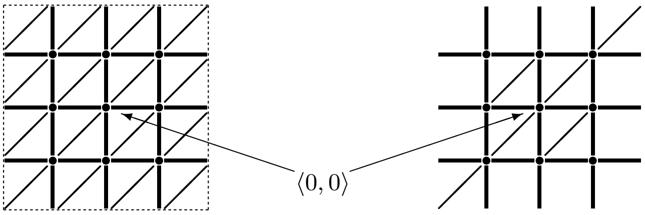

Fig. 1. Left: partition of the set $(-2,2)^2$ induced by $\simeq_1$. Right: partition of $\mathbb{R}^2$ induced by $\widehat{\simeq}_1$. Every dot, line segment, and white area represents an equivalence class.

Definition 15 ($\simeq_\kappa$-uniform and $\widehat{\simeq}_\kappa$-uniform interpretations). *Let κ be a positive integer. Consider a interpretation $\mathcal{A}$. We call $\mathcal{A}$ $\simeq_\kappa$-*uniform *if its corresponding coloring $\chi_\mathcal{A}$ (cf. Definition 7) colors each $\simeq_\kappa$-equivalence class over $(-\kappa-1,\kappa+1)^m$ uniformly, i.e. for all tuples $\bar{q}, \bar{q}' \in (-\kappa-1,\kappa+1)^m$ with $\bar{q} \simeq_\kappa \bar{q}'$ we have $\chi_\mathcal{A}(\bar{q}) = \chi_\mathcal{A}(\bar{q}')$. We call $\mathcal{A}$ $\widehat{\simeq}_\kappa$-*uniform *if $\chi_\mathcal{A}$ colors each $\simeq_\kappa$-equivalence class over $\mathbb{R}^m$ uniformly.*

The parameter κ will be determined by the base-sort constant in N with the largest absolute value. If κ is defined in this way, one can show that the LA constraints occurring in N cannot distinguish between two $\widehat{\simeq}_\kappa$-equivalent m-tuples of reals. This observation is crucial for the proof of Lemma 16.

In order to prove the existence of $\widehat{\simeq}_\kappa$-uniform models for satisfiable N, we start from some model $\mathcal{A}$ of N and rely on the existence of a certain finite set $Q \subseteq [0,1)$ of fractional parts. This set Q can be extended to a set $\widehat{Q} \subseteq (-\kappa-1,\kappa+1)$ by addition of the fractional parts in Q with integral parts k from the range $-\kappa-1 \leq k \leq \kappa$. Hence, $\widehat{Q}$ contains $2(\kappa+1)\cdot|Q|$ reals. We assume that all $\simeq_\kappa$-equivalent tuples $\bar{s}, \bar{s}'$ from $\widehat{Q}^m$ are treated uniformly by $\mathcal{A}$. Put differently, we require $\chi_{\mathcal{A}}(\bar{s}) = \chi_{\mathcal{A}}(\bar{s}')$. We choose to formulate this requirement with respect to $\simeq_\kappa$ because of the more regular structure of its equivalence classes, which facilitates a more convenient way of invoking Lemma 4. Due to the fact that $\simeq_\kappa$ constitutes a refinement of $\widehat{\simeq}_\kappa$ on the subspace $(-\kappa-1, \kappa+1)^m$, and since for every $\widehat{\simeq}_\kappa$-equivalence class $\widehat{S}$ over $\mathbb{R}^m$ there is some $\simeq_\kappa$-equivalence class $S \subseteq (-\kappa-1, \kappa+1)^m$ such that $S \subseteq \widehat{S}$, we can use the color $\chi_{\mathcal{A}}(\bar{r})$ of representative m-tuples $\bar{r}$ constructed from $\widehat{Q}$ to serve as a blueprint when constructing a $\widehat{\simeq}_\kappa$-uniform model $\mathcal{B}$.

Lemma 16. *Let λ be the maximal number of distinct base-sort variables in any single clause in N; in case of $\lambda < m$, we set $\lambda := m$. Let $\mathcal{A}$ be a model of N. Let κ be the maximal absolute value of any rational occurring in N; in case this value is zero, we set $\kappa := 1$. Suppose we are given a finite set $Q \subseteq [0, 1)$ of cardinality $\lambda + 1$ such that $0 \in Q$ and for all tuples $\bar{r}, \bar{s} \in \widehat{Q}^m$, $\bar{r} \simeq_\kappa \bar{s}$ entails $\chi_{\mathcal{A}}(\bar{r}) = \chi_{\mathcal{A}}(\bar{s})$, where*

$$\widehat{Q} := \big\{ q + k \bigm| q \in Q \text{ and } k \in \{-\kappa-1, \ldots, 0, \ldots, \kappa\} \big\}.$$

Then we can construct a model $\mathcal{B}$ of N that is $\widehat{\simeq}_\kappa$-uniform and that interprets the free sort $\mathcal{S}$ as a finite set.

Proof sketch. The construction of $\mathcal{B}$ from $\mathcal{A}$ is similar to the construction of uniform models outlined in the proof of Lemma 9.

Claim I: Let μ be a positive integer with $\mu \leq \lambda$. For every $\widehat{\simeq}_\kappa$-equivalence class S over $\mathbb{R}^\mu$ and every $\bar{r} \in S$ there is some $\bar{q} \in S \cap \widehat{Q}^\mu$ such that $\bar{r} \mathrel{\widehat{\simeq}_\kappa} \bar{q}$ and for all i_1, i_2, i_3 with $r_{i_1} < -\kappa$ and $r_{i_2} > \kappa$ and $-\kappa \leq r_{i_3} \leq \kappa$ we have $\mathrm{fr}(q_{i_1}) < \mathrm{fr}(q_{i_2}) < \mathrm{fr}(q_{i_3})$. $\Diamond$

Let $\widehat{\mathcal{S}}$ denote the set $\{a \in \mathcal{S}^{\mathcal{A}} \mid a = c^{\mathcal{A}} \text{ for some } c \in \mathrm{fconsts}(N)\}$. We construct the interpretation $\mathcal{B}$ as follows: $\mathcal{S}^{\mathcal{B}} := \widehat{\mathcal{S}}$; $c^{\mathcal{B}} := c^{\mathcal{A}}$ for every constant symbol c; for every uninterpreted predicate symbol P occurring in N and for all tuples $\bar{a} \in \widehat{\mathcal{S}}^{m'}$ and $\bar{r} \in \mathbb{R}^m$ we pick some tuple $\bar{q} \in \widehat{Q}^m$ in accordance with Claim I—i.e. $\bar{q}$ satisfies $\bar{r} \mathrel{\widehat{\simeq}_\kappa} \bar{q}$—and define $P^{\mathcal{B}}$ in such a way that $\langle \bar{a}, \bar{r} \rangle \in P^{\mathcal{B}}$ if and only if $\langle \bar{a}, \bar{q} \rangle \in P^{\mathcal{A}}$.

Claim II: The interpretation $\mathcal{B}$ is $\widehat{\simeq}_\kappa$-uniform. $\Diamond$

It remains to show $\mathcal{B} \models N$. We use the same approach as in the proof for Lemma 9, this time based on the equivalence relation $\widehat{\simeq}_\kappa$ instead of $\sim_{\mathcal{J}_{\mathcal{A}}}$. $\square$

We employ Lemma 4 to prove the existence of the set Q used in Lemma 16.

Lemma 17. *Let $\mathcal{A}$ be an interpretation and let κ, λ be positive integers with $\lambda \geq m$. There exists a finite set $Q \subseteq [0, 1)$ of cardinality $\lambda + 1$ such that $0 \in Q$ and for all tuples $\bar{s}, \bar{s}' \in \widehat{Q}^m$, $\bar{s} \simeq_\kappa \bar{s}'$ entails $\chi_{\mathcal{A}}(\bar{s}) = \chi_{\mathcal{A}}(\bar{s}')$, where*

$$\widehat{Q} := \big\{ q + k \bigm| q \in Q \text{ and } k \in \{-\kappa-1, \ldots, 0, \ldots, \kappa\} \big\}.$$

Proof sketch. One can show that every $\simeq_\kappa$-equivalence class S over $(-\kappa-1, \kappa+1)^m$ can be represented by a pair of mappings $\varrho : [m] \to \{0, 1, \ldots, m\}$ and $\sigma : [m] \to \{-\kappa-1, \ldots, 0, \ldots, \kappa\}$ such that

(i) for any ascending tuple $\langle r_0, r_1, \ldots, r_m\rangle \in [0,1)^{m+1}$ with $r_0 = 0$ we have $\big\langle r_{\varrho(1)} + \sigma(1), \ldots, r_{\varrho(m)} + \sigma(m)\big\rangle \in S$, and
(ii) for every tuple $\langle s_1, \ldots, s_m\rangle \in S$ there is an ascending tuple $\langle r_0, r_1, \ldots, r_m\rangle \in [0,1)^{m+1}$ with $r_0 = 0$ such that $\big\langle s_1, \ldots, s_m\big\rangle = \big\langle r_{\varrho(1)} + \sigma(1), \ldots, r_{\varrho(m)} + \sigma(m)\big\rangle$.

Having an enumeration $\langle \varrho_1, \sigma_1\rangle, \ldots, \langle \varrho_k, \sigma_k\rangle$ of pairs of such mappings in which every $\simeq_\kappa$-equivalence class over $(-\kappa-1, \kappa+1)^m$ is represented, we construct a coloring $\widehat{\chi} : \mathbb{R}^m \to \big(\mathcal{P}\{P_i\bar{a} \mid \bar{a} \in \widehat{\mathcal{S}}^{m'} \text{ and } P_i \text{ occurs in } N\}\big)^k$ by setting

$$\widehat{\chi}(\bar{r}) := \big\langle \chi_{\mathcal{A}}\big(\langle r_{\varrho_1(1)} + \sigma_1(1), \ldots, r_{\varrho_1(m)} + \sigma_1(m)\rangle\big), \\ \ldots, \chi_{\mathcal{A}}\big(\langle r_{\varrho_k(1)} + \sigma_k(1), \ldots, r_{\varrho_k(m)} + \sigma_k(m)\rangle\big)\big\rangle$$

for every tuple $\bar{r} = \langle r_1, \ldots, r_m\rangle \in (0,1)^m$, where we define r_0 to be 0. By virtue of Lemma 4, there is a set $Q' \subseteq (0,1)$ of cardinality λ such that all ascending tuples $\langle r_1, \ldots, r_m\rangle \in {Q'}^m$ are assigned the same color by χ. Then $Q := Q' \cup \{0\}$ is the sought set. □

Lemmas 16 and 17 together entail the existence of $\widehat{\simeq}_\kappa$-uniform models for finite satisfiable BSR(BD) clause sets, where κ is defined as in Lemma 16.

Corollary 18. *Let κ be defined as in Lemma 16. If N is satisfiable, then it has a model $\mathcal{A}$ that is $\widehat{\simeq}_\kappa$-uniform and that interprets the sort $\mathcal{S}$ as some finite set.*

By virtue of Corollary 18, we can devise a nondeterministic decision procedure for finite BSR(BD) clause sets N. We adapt the decision procedure for BSR(SLR) as follows. Since base-sort Skolem constants do not occur in N, Steps (1), (2), and (3.3) are skipped. Moreover, Step (3.4) has to be modified slightly. The interpretations of uninterpreted predicate symbols need to be constructed in such a way that the candidate interpretation $\mathcal{B}$ is $\widehat{\simeq}_\kappa$-uniform for $\kappa := \max\big(\{1\} \cup \{|c| \bigm| c \in \text{bconsts}(N)\}\big)$.

Theorem 19. *Satisfiability of finite BSR(BD) clause sets is decidable.*

6 Formalizing reachability for timed automata

In this section we show that reachability for timed automata (cf. [1]) can be formalized using finite BSR(BD) clause sets. In what follows, we fix a finite sequence $\bar{x}$ of pairwise distinct *clock variables* that range over the reals. For convenience, we occasionally treat $\bar{x}$ as a set and use set notation such at $x \in \bar{x}$, $|\bar{x}|$, and $\mathcal{P}(\bar{x})$. A *clock constraint over* $\bar{x}$ is a finite conjunction of LA constraints of the form `true`, $x \lhd c$, or $x - y \lhd c$, where $x, y \in \bar{x}$, c is an integer and $\lhd \in \{<, \leq, =, \neq, \geq, >\}$. We denote the *set of all clock constraints over* $\bar{x}$ by $\text{CC}(\bar{x})$. A *timed automaton* is a tuple $\langle \text{Loc}, \ell_0, \bar{x}, \langle \text{inv}_\ell\rangle_{\ell \in \text{Loc}}, \mathcal{T}\rangle$, where Loc is a finite set of locations; $\ell_0 \in \text{Loc}$ is the initial location; $\langle \text{inv}_\ell\rangle_{\ell \in \text{Loc}}$ is a family of clock constraints from $\text{CC}(\bar{x})$ where each inv_ℓ describes the *invariant at location* ℓ; $\mathcal{T} \subseteq \text{Loc} \times \text{CC}(\bar{x}) \times \mathcal{P}(\bar{x}) \times \text{Loc}$ is the location transition relation within the automaton, including guards with respect to clocks and the set of clocks that are being reset when the transition is taken.

Although the control flow of a timed automaton is described by finite means, the fact that clocks can assume uncountably many values yields an infinite state space, namely, $\mathrm{Loc} \times [0, \infty)^{|\bar{x}|}$. Transitions between states fall into two categories:

delay transitions $\langle \ell, \bar{r} \rangle \hookrightarrow \langle \ell, \bar{r}' \rangle$ with $\bar{r}' = \bar{r} + t$ for some $t \geq 0$ and $[\bar{x}' {\mapsto} \bar{r}'] \models \mathrm{inv}_\ell[\bar{x}']$; and

location transitions $\langle \ell, \bar{r} \rangle \hookrightarrow \langle \ell', \bar{r}' \rangle$ for some $\langle \ell, g, Z, \ell' \rangle \in \mathcal{T}$ with $[\bar{x} {\mapsto} \bar{r}] \models g[\bar{x}]$, $\bar{r}' = \bar{r}[Z \mapsto 0]$, and $[\bar{x}' {\mapsto} \bar{r}'] \models \mathrm{inv}_{\ell'}[\bar{x}']$.

The operation $\bar{r}' := \bar{r} + t$ is defined by setting $r_i' := r_i + t$ for every i, and $\bar{r}' := \bar{r}[Z \mapsto 0]$ means that $r_i' = 0$ for every $x_i \in Z$ and $r_i' = r_i$ for every $x_i \notin Z$.

In [6] Fietzke and Weidenbach present an encoding of reachability for a given timed automaton $\mathbf{A}$ in terms of *first-order logic modulo linear arithmetic.*

Definition 20 (FOL(LA) encoding of a timed automaton, [6]). *Given a timed automaton* $\mathbf{A} := \langle \mathrm{Loc}, \ell_0, \bar{x}, \langle \mathrm{inv}_\ell \rangle_{\ell \in \mathrm{Loc}}, \mathcal{T} \rangle$, *the* FOL(LA) *encoding of* $\mathbf{A}$ *is the following clause set* $N_\mathbf{A}$, *where* Reach *is a* $(1 + |\bar{x}|)$*-ary predicate symbol:*

the initial clause $\bigwedge_{x \in \bar{x}} x = 0 \;\wedge\; \mathrm{inv}_{\ell_0}[\bar{x}] \;\|\rightarrow \mathrm{Reach}(\ell_0, \bar{x})$*;*

delay clauses $z \geq 0 \;\wedge\; \bigwedge_{x \in \bar{x}} x' = x + z \;\wedge\; \mathrm{inv}_\ell[\bar{x}'] \;\|\; \mathrm{Reach}(\ell, \bar{x}) \rightarrow \mathrm{Reach}(\ell, \bar{x}')$
for every location $\ell \in \mathrm{Loc}$*;*

transition clauses $g[\bar{x}] \;\wedge\; \bigwedge_{x \in Z} x' = 0 \;\wedge \bigwedge_{x \in \bar{x} \setminus Z} x' = x \;\wedge\; \mathrm{inv}_{\ell'}[\bar{x}'] \;\|\; \mathrm{Reach}(\ell, \bar{x}) \rightarrow \mathrm{Reach}(\ell', \bar{x}')$
for every location transition $\langle \ell, g, Z, \ell' \rangle \in \mathcal{T}$*.*

Corollary 4.3 in [6] states that for any model of $N_\mathbf{A}$, every location $\ell \in \mathrm{Loc}$, and every tuple $\bar{r} \in \mathbb{R}^{|\bar{x}|}$ we have $\mathcal{A}, [\bar{x} {\mapsto} \bar{r}] \models \mathrm{Reach}(\ell, \bar{x})$ if and only if $\mathbf{A}$ can reach the state $\langle \ell, \bar{r} \rangle$ from its initial configuration.

Given any clock constraint $\psi \in \mathrm{CC}(\bar{x})$ and some location ℓ, the timed automaton $\mathbf{A}$ can reach at least one of the states $\langle \ell, \bar{r} \rangle$ with $[\bar{x} {\mapsto} \bar{r}] \models \psi[\bar{x}]$ from its initial configuration if and only if the clause set $N_\mathbf{A} \cup \big\{ \psi[\bar{x}] \,\|\, \mathrm{Reach}(\ell, \bar{x}) \rightarrow \Box \big\}$ is unsatisfiable (cf. Proposition 4.4 in [6]).

Next, we argue that the passage of time does not have to be formalized as a synchronous progression of all clocks. Instead, it is sufficient to require that clocks progress in such a way that their valuations do not drift apart excessively. Although this weakens the semantics slightly, reachability remains unaffected.

Lemma 21. *Consider any delay clause*

$$C := \quad z \geq 0 \;\wedge\; \textstyle\bigwedge_{x \in \bar{x}} x' = x + z \;\wedge\; \mathrm{inv}_\ell[\bar{x}'] \;\|\; \mathrm{Reach}(\ell, \bar{x}) \rightarrow \mathrm{Reach}(\ell, \bar{x}')$$

that belongs to the FOL(LA) encoding of some timed automaton $\mathbf{A} := \langle \mathrm{Loc}, \ell_0, \bar{x}, \langle \mathrm{inv}_\ell \rangle_{\ell \in \mathrm{Loc}}, \mathcal{T} \rangle$. *Let* λ *be some positive integer. Let* M *be a finite clause set corresponding to the following formula*

$$\varphi := \bigwedge_{x_1, x_2 \in \bar{x}} \;\bigwedge_{-\lambda \leq k \leq \lambda} \big(x_1 - x_2 \leq k \;\leftrightarrow\; x_1' - x_2' \leq k \big) \\ \wedge\; \big(x_1 - x_2 \geq k \;\leftrightarrow\; x_1' - x_2' \geq k \big) \\ \wedge \bigwedge_{x \in \bar{x}} x' \geq x \;\wedge\; \mathrm{inv}_\ell[\bar{x}'] \;\|\; \mathrm{Reach}(\ell, \bar{x}) \rightarrow \mathrm{Reach}(\ell, \bar{x}') \;.$$

For every $\simeq_\lambda$-uniform interpretation $\mathcal{A}$ we have $\mathcal{A}, [\bar{x} \mapsto \bar{r}, \bar{x}' \mapsto \bar{r}'] \models C$ for all tuples $\bar{r}, \bar{r}' \in [0, \lambda + 1)^{|\bar{x}|}$ if and only if $\mathcal{A}, [\bar{x} \mapsto \bar{q}, \bar{x}' \mapsto \bar{q}'] \models M$ holds for all tuples $\bar{q}, \bar{q}' \in [0, \lambda + 1)^{|\bar{x}|}$.

Our approach to decidability of BSR(BD)-satisfiability exploits the observation that the allowed constraints cannot distinguish between tuples from one and the same equivalence class with respect to $\widehat{\simeq}_\lambda$, which induces only a finite number of such classes. Decidability of the reachability problem for timed automata can be argued in a similar fashion, using a suitable equivalence relation on clock valuations [1]. We refer to the induced classes of indistinguishable clock valuations over $\mathbb{R}^{|\bar{x}|}$, which are induced by a given timed automaton $\mathbf{A} = \langle \text{Loc}, \ell_0, \bar{x}, \langle \text{inv}_\ell \rangle_{\ell \in \text{Loc}}, \mathcal{T} \rangle$, as *TA regions* of $\mathbf{A}$. Figure 2 illustrates the TA regions for some timed automaton with two clocks and in which all integer constants have an absolute value of at most 2. For every TA region $R \subseteq \mathbb{R}^2$ of such an automaton, there is at least one representative $\bar{r} \in R$ which lies in $[0, 5)^2$.

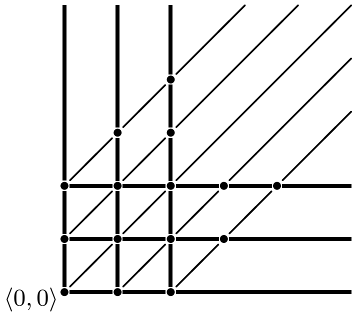

Fig. 2. Partition of the set $[0, \infty)^2$ into classes of clock valuations that cannot be distinguished by a timed automaton with two clocks in which the absolute value of integer constants occurring in location invariants and transition guards does not exceed 2. Every dot, line segment, and white area represents an equivalence class.

Let k be the maximal absolute value of any integer constant occurring in the invariants and the transition guards of $\mathbf{A}$. Moreover, let $x_1, \ldots, x_\ell$ be the clock variables in $\bar{x}$. Consider a constraint of the form

$$\psi := \;\; x_1 - x_2 = k \;\wedge\; x_2 - x_3 = k \;\wedge \ldots \wedge\; x_{\ell-1} - x_\ell = k.$$

We observe that ψ entails $x_1 - x_\ell = (\ell - 1) \cdot k$. Of course, ψ can also be conjoined with the constraint $x_1 < -k$, say, which entails $x_\ell < -k - (\ell - 1) \cdot k$. This example illustrates that one can combine several difference constraints $x - y \triangleleft c$ over different clock variables in such a way that bounds are entailed which cannot be formulated with a single constraint $u - v \triangleleft d$ with $|d| \leq k$. However, all of these combined constraints can be equivalently represented with atomic constraints $x - y \triangleleft c$ or $x \triangleleft c$, where $|c| \leq |\bar{x}| \cdot k$.

In order to decide reachability for $\mathbf{A}$, it is sufficient to consider a bounded subspace of $\mathbb{R}^{|\bar{x}|}$. More precisely, there exists a computable integer λ, namely $|\bar{x}| \cdot k$, such that any valuation $\bar{r}$ of $\mathbf{A}$'s clocks can be projected to some valuation $\bar{r}' \in [0, \lambda + 1)^{|\bar{x}|}$ that $\mathbf{A}$ cannot distinguish from $\bar{r}$. In the subspace $[0, \lambda + 1)^{|\bar{x}|}$, $\mathbf{A}$'s TA regions coincide with (finite unions of) equivalence classes with respect to $\simeq_\lambda$. In fact, the quotient $[0, \lambda{+}1)^{|\bar{x}|}/_{\simeq_\lambda}$ constitutes a refinement of the division of $[0, \lambda + 1)^{|\bar{x}|}$ into TA regions. Since any pair $\langle \ell, \bar{r} \rangle$ with $\bar{r} \in R$ for some TA region R is reachable if and only if all pairs $\langle \ell, \bar{r}' \rangle$ with $\bar{r} \in R$ are reachable, any minimal model $\mathcal{A}$ of the encoding $N_{\mathbf{A}}$ is $\simeq_\lambda$-uniform (where minimality of $\mathcal{A}$ refers to the minimality of the set $\text{Reach}^{\mathcal{A}}$ with respect to set inclusion). This is why Lemma 21 may focus on $\simeq_\lambda$-uniform models.

This means, given the FOL(LA) encoding $N_{\mathbf{A}}$ of $\mathbf{A}$, we obtain a BSR(BD) encoding $N'_{\mathbf{A}}$ of reachability with respect to $\mathbf{A}$ in the following two steps:
(1) Replace every delay clause in $N_{\mathbf{A}}$ with a corresponding finite set of clauses M in accordance with Lemma 21, where we set $\lambda := |\bar{x}| \cdot k$.
(2) Conjoin the constraints $0 \leq x \ \wedge \ x < \kappa$ for $\kappa := \lambda + 1 = |\bar{x}| \cdot k + 1$ to every constraint in which a base-sort variable x occurs.
Since any $\widehat{\simeq}_{\lambda+1}$-uniform model of $N'_{\mathbf{A}}$ is $\simeq_\lambda$-uniform over the subspace $(-\lambda - 1, \lambda + 1)^{|\bar{x}|}$, Lemma 21 entails that $N'_{\mathbf{A}}$ faithfully encodes reachability for $\mathbf{A}$.

Theorem 22. *The reachability problem for a given timed automaton can be expressed in terms of satisfiability of a finite BSR(BD) clause set.*

7 Discussion

We have shown that satisfiability for the clause fragments BSR(SLR) and BSR(BD) is decidable. Both fragments hierarchically combine the Bernays–Schönfinkel–Ramsey fragment over uninterpreted predicate symbols with restricted forms of linear arithmetic over the reals.

Since the syntax of BSR(SLR) allows only a very restricted form of arithmetic on universally quantified variables, this part of the fragment seems to reduce to the theory of (dense) orderings. Except for density, all characteristic properties of orderings (e.g. transitivity) are already definable in the non-extended BSR fragment. On the other hand, regarding existentially quantified variables—which appear in the form of Skolem constants—, BSR(SLR) allows linear arithmetic expressions without notable restrictions, as long as no universally quantified variables are involved in the arithmetic expressions, and as long as no existential quantifier lies within the scope of a universal quantifier. Unfortunately, a more liberal syntax quickly leads to undecidability, as already pointed out in Section 1.

With BSR(BD) we have investigated another decidable fragment that is a hierarchic combination of the Bernays–Schönfinkel–Ramsey fragment with restricted arithmetic over the reals. Since difference constraints have been of use in the analysis and verification of timed systems, the idea suggested itself that BSR(BD) may find applications in this area. Indeed, we have seen that reachability for timed automata can be expressed with BSR(BD), although not entirely in a straightforward fashion. To this end, we have slightly weakened the usual notion of synchronous progression of all clocks. Our modifications do not affect the reachability relation. It is to be expected that BSR(BD) lends itself to more sophisticated applications in the area of timed systems or other fields.

Acknowledgement The present author is indebted to the anonymous reviewers for their constructive criticism and valuable suggestions.

References

1. Rajeev Alur and David L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
2. Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Refutational Theorem Proving for Hierarchic First-Order Theories. *Applicable Algebra in Engineering, Communication and Computing*, 5:193–212, 1994.
3. Peter Baumgartner and Uwe Waldmann. Hierarchic Superposition with Weak Abstraction. In *Automated Deduction (CADE-24)*, pages 39–57, 2013.
4. Aaron R. Bradley, Zohar Manna, and Henny B. Sipma. What's Decidable About Arrays? In *Verification, Model Checking, and Abstract Interpretation (VMCAI'06)*, pages 427–442, 2006.
5. Peter J. Downey. Undecidability of Presburger Arithmetic with a Single Monadic Predicate Letter. Technical report, Center for Research in Computer Technology, Harvard University, 1972.
6. Arnaud Fietzke and Christoph Weidenbach. Superposition as a Decision Procedure for Timed Automata. *Mathematics in Computer Science*, 6(4):409–425, 2012.
7. Yeting Ge and Leonardo Mendonça de Moura. Complete Instantiation for Quantified Formulas in Satisfiabiliby Modulo Theories. In *Computer Aided Verification (CAV'09)*, pages 306–320, 2009.
8. Ronald L. Graham, Bruce L. Rothschild, and Joel H. Spencer. *Ramsey Theory.* A Wiley-Interscience publication. Wiley, second edition, 1990.
9. Matthias Horbach, Marco Voigt, and Christoph Weidenbach. On the Combination of the Bernays–Schönfinkel–Ramsey Fragment with Simple Linear Integer Arithmetic. In *Automated Deduction (CADE-26).* To appear.
10. Matthias Horbach, Marco Voigt, and Christoph Weidenbach. The Universal Fragment of Presburger Arithmetic with Unary Uninterpreted Predicates is Undecidable. *ArXiv preprint*, arXiv:1703.01212 [cs.LO], 2017.
11. Daniel Kroening and Ofer Strichman. *Decision Procedures.* Texts in Theoretical Computer Science. An EATCS Series. Springer, second edition, 2016.
12. Evgeny Kruglov and Christoph Weidenbach. Superposition Decides the First-Order Logic Fragment Over Ground Theories. *Mathematics in Computer Science*, 6(4):427–456, 2012.
13. Peter Niebert, Moez Mahfoudh, Eugene Asarin, Marius Bozga, Oded Maler, and Navendu Jain. Verification of Timed Automata via Satisfiability Checking. In *Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'02)*, pages 225–244, 2002.
14. Vaughan R. Pratt. Two Easy Theories Whose Combination is Hard. Technical report, Massachusetts Institute of Technology, 1977.
15. Hilary Putnam. Decidability and Essential Undecidability. *Journal of Symbolic Logic*, 22(1):39–54, 1957.
16. Marco Voigt. The Bernays–Schönfinkel–Ramsey Fragment with Bounded Difference Constraints over the Reals is Decidable. *ArXiv preprint*, arXiv:1706.08504 [cs.LO], 2017.
17. Marco Voigt and Christoph Weidenbach. Bernays-Schönfinkel-Ramsey with Simple Bounds is NEXPTIME-complete. *ArXiv preprint*, arXiv:1501.07209 [cs.LO], 2015.