Variable Independence, Quantifier Elimination, and Constraint Representations

Leonid Libkin[1]*

Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974, USA.
Email: `libkin@research.bell-labs.com`

Abstract. Whenever we have data represented by constraints (such as order, linear, polynomial, etc.), running time for many constraint processing algorithms can be considerably lowered if it is known that certain variables in those constraints are independent of each other. For example, when one deals with spatial and temporal databases given by constraints, the projection operation, which corresponds to quantifier elimination, is usually the costliest. Since the behavior of many quantifier elimination algorithms becomes worse as the dimension increases, eliminating certain variables from consideration helps speed up those algorithms.

While these observations have been made in the literature, it remained unknown when the problem of testing if certain variables are independent is decidable, and how to construct efficiently a new representation of a constraint-set in which those variables do not appear together in the same atomic constraints. Here we answer this question. We first consider a general condition that gives us decidability of variable independence; this condition is stated in terms of model-theoretic properties of the structures corresponding to constraint classes. We then show that this condition covers the domains most relevant to spatial and temporal applications. For some of these domains, including linear and polynomial constraints over the reals, we provide a uniform decision procedure which gives us tractability, and present a polynomial-time algorithm for producing nice constraint representations.

1 Introduction

We start with a simple example. Suppose we have a set $S \subseteq \mathbb{R}^2$ given by simple order-constraints $\varphi(x,y) = (0 < x < 1) \wedge (0 < y < 1)$. Suppose we want to find its projection on the x axis. This means writing the formula $\exists y\ \varphi(x,y)$ as a quantifier-free formula. This can be done, in general, because the theory of $\langle \mathbb{R}, <, (r)_{r \in \mathbb{R}} \rangle$ admits quantifier elimination. But in this particular case it is very easy to find a quantifier-free formula equivalent to $\exists y\ \varphi(x,y)$ using just standard rules for equivalence of first-order formulae:

$$\exists y\ \varphi(x,y) \ \leftrightarrow\ (0 < x < 1) \wedge \exists y\ (0 < y < 1) \ \leftrightarrow\ (0 < x < 1) \wedge \text{true} \ \leftrightarrow\ 0 < x < 1.$$

* Part of this work was done while visiting INRIA.

Now notice that φ can be considered as a formula in the language of the real field $\langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$ whose theory also admits quantifier elimination. Suppose then that instead of φ, we are given an equivalent formula $\psi(x, y)$:

$$\begin{array}{l} \big((0 < x < 1) \wedge (0 < y < 1) \wedge (4x^2 - y - 1 \geq 0)\big) \\ \vee \big((0 < x < 1) \wedge (0 < y < 1) \wedge (4x^2 - y - 1 \leq 0)\big). \end{array}$$

The first step of quantifier elimination for $\exists y\ \psi$ is easy, as we propagate $\exists y$ inside the disjunction. However, trying to find a quantifier-free equivalent for the first disjunct, that is, a formula equivalent to $\exists y\ \big((0 < x < 1) \wedge (0 < y < 1) \wedge (4x^2 - y - 1 \geq 0)\big)$, one immediately encounters obstacles. Unlike the earlier example, this one requires a bit of thought to come up with the answer $(0.5 \leq x < 1)$. Similarly, some work is needed to compute the answer $(0 < x \leq 1/\sqrt{2})$ for the second disjunct.

Why is it that the first quantifier-elimination procedure is completely elementary, and the second is not, even though both φ and ψ define the same set? The reason is that in the first representation of S, variables x and y are independent, that is, they do not appear in the same atomic formulae. This makes quantifier elimination easy. In the second case, x and y do appear together in the same term $x^2 - 4y - 1$, and this is what causes the problem.

This extremely simple observation can often make constraint processing easier. While it can conceivably be useful in various tasks such as more efficient variable elimination in constraint logic programming [8, 12], here we concentrate on one application area, namely *constraint databases* [15, 14] where it found its way into a practical system for querying spatio-temporal databases [9]. The main goal of constraint databases is to model infinite database objects, which arise in a variety of applications, for example, in Geographical Information Systems.

A particular constraint model is defined over a structure $\mathcal{M} = \langle U, \Omega \rangle$ (where U is the universe and Ω is the vocabulary) which is typically required to have quantifier elimination. Those considered most often in spatial application are the real field $\mathbf{R} = \langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$ and the real ordered group $\mathbf{R}_{\text{lin}} = \langle \mathbb{R}, +, -, 0, 1\ < \rangle$, which give rise to polynomial and linear constraint databases, respectively. A constraint relation of arity n is simply a definable subset of U^n, that is, a set of tuples $\vec{a} \in U^n$ that satisfy a first-order formula. For the above structures, constraint relations are *semi-algebraic* sets for $\mathbf{R}$, and *semi-linear* sets for $\mathbf{R}_{\text{lin}}$ [2]. A constraint database is a finite set of constraint relations.

A standard constraint query language over $\mathcal{M}$ is $\text{FO} + \mathcal{M}$, that is, first-order logic in the language of $\mathcal{M}$ and symbols for relations in a constraint database. For example, if a database contains a single ternary symbol S, the query $\varphi(x) \equiv \exists u, v\ \forall y, z\ (S(x, y, z) \leftrightarrow z = u \cdot y + v)$ finds all a such that the intersection of S with the plane $x = a$ is a line. Note that if S is a semi-algebraic set, then so is $\varphi(S)$.

One of the standard database operations is projection. In the language of constraint processing, it corresponds to quantifier elimination. That is, given a quantifier-free formula $\varphi(y, x_1, \ldots, x_{n-1})$, one wishes to find a quantifier-free formula $\psi(\vec{x})$ equivalent to $\exists y\ \varphi(y, \vec{x})$. In many cases, the complexity of algorithms to find such a ψ is of the form $O(N^{f(n)})$, where N is the size of the formula, and f is some function. For example, if one uses cylindrical algebraic decomposition [3] for the real field, f is $O(2^n)$. In general, even if better algorithms are available, the complexity of constraint processing often increases with dimension to such an extent that it becomes unmanageable for large datasets (see, e.g., [10]).

Assume now that $\vec{x}$ is split into two disjoint tuples $\vec{u}$ and $\vec{v}$ such that $(y, \vec{u})$ and $\vec{v}$ are independent, that is, they do not appear in the same atomic formulae. Then φ is equivalent to a formula of the form

$$\bigvee_{i=1}^{k} \alpha_i(y, \vec{u}) \wedge \beta_i(\vec{v}). \tag{1}$$

Therefore, the formula $\exists y \ \varphi$ is equivalent to

$$\bigvee_{i=1}^{k} (\exists y \ \alpha_i(y, \vec{u})) \wedge \beta_i(\vec{v}). \tag{2}$$

For a number of operations this is a significant improvement, as the exponent becomes lower. For example, in addition to quantifier elimination, data often has to be represented in a nice format (essentially, as union of cells [3]), and algorithms for doing this also benefit from reduction in the dimension [9, 10].

Even though such a notion of independence may seem to be too much of a restriction, from the practical point of view it is sometimes necessary to insist on it, as the cost of general quantifier elimination and other operations could be prohibitively expensive. For example, the DEDALE constraint database system [9] requires that the projection operation only be applied when $\vec{u}$ consists of a single variable. Dealing with spatio-temporal applications, one often queries trajectories of objects, or cadastral (land-ownership) information. These are typically represented as objects in $\mathbb{R}^3$ given by formulae $\varphi(x, y, t)$. To be able to compute $\exists y \ \varphi(x, y, t)$, one approximates φ by a formula $\psi(x, y, t)$ which is a Boolean combination of formulae $\alpha_i(x, y)$ and $\beta_i(t)$. For trajectories, this amounts to saying that an object is in a given region during a given interval of time; thus, it is the information about the speed that is lost in order to have efficient query evaluation. As was further demonstrated in [10], the difference between the case when at most 2 variables are dependent, and that of 3 or more variables being dependent, is quite dramatic, in the case of linear and polynomial constraints.

What is missing, however, in this picture, is the ability to determine whether a given constraint representation of the data can be converted to the one in the right format, just as in our first example, $\psi(x, y)$ is equivalent to $\varphi(x, y)$, in which variables x and y are independent. It was claimed in [5] that such a procedure exists for linear constraints, and then [10] gave a simpler algorithm. However, [16] then showed that both claims were incorrect. It was thus not known if variable independence can be tested for relevant classes of constraints.

Our main goal here is to show that variable independence can be tested for many classes of constraints, and that algorithms for converting a given formula into a one in the right form can be obtained. Moreover, those algorithms often work in time polynomial in the size of the formula (assuming the total number of variables is fixed). Among structures for which we prove such results are the real ordered group, the real field, as well as $\langle \mathbb{Z}, +, 0, 1, < \rangle$ extended with all the relations $x = y(\text{mod } k)$, $k > 1$ (which is used in temporal applications). Even if those algorithms are relatively expensive, it is worth putting data in a nice format for two reasons. First, such an algorithm works only once, and then the data is repeatedly queried by different queries, which can be evaluated faster. Secondly, some queries are known to preserve variable independence; hence, this information can be used for further processing the query output.

Organization In Section 2, we define the notion of variable independence, and more generally, the notion $\varphi \sim P$ of a formula φ respecting a certain partition P of its free variables. Then, in Section 3, we discuss requirements on the theory of $\mathcal{M}$ that guarantee decidability of this notion, as well as the existence of an algorithm that converts a given formula into a one in the right shape. In Section 4, we discuss specific classes of structures and derive some complexity bounds. In particular, we look at *o-minimal* structures [23] (which include linear and polynomial constraints over the reals) and give a *uniform* decision procedure. This procedure gives us tractability, and we also show how to find an equivalent formula in the right shape in polynomial time. All proofs are only sketched here; complete proofs are in the full version [17].

2 Notations

All the definitions can be stated for arbitrary first-order structures, although for the algorithmic considerations we shall require at least decidability of the theory, and often quantifier elimination.

Given a structure $\mathcal{M} = \langle U, \Omega \rangle$ (where U is a set always assumed to be infinite, and Ω can contain predicate, function, and constant symbols, and is always assumed to be a recursive set), we say that the theory of $\mathcal{M}$ is decidable if for every first-order sentence Φ in the language of $\mathcal{M}$ it decidable if $\mathcal{M} \models \Phi$. We say that $\mathcal{M}$ admits (effective) quantifier elimination if for every formula $\varphi(\vec{x})$ in the language of $\mathcal{M}$, there exists (and can be effectively found) a quantifier-free formula $\psi(\vec{x})$ such that $\mathcal{M} \models \forall \vec{x}\ \varphi(\vec{x}) \leftrightarrow \psi(\vec{x})$.

Given a formula $\varphi(\vec{x}, \vec{y})$ in the language of $\mathcal{M}$, with $\vec{x}$ of length n and $\vec{y}$ of length m, and $\vec{a} \in U^n$, we write $\varphi(\vec{a}, \mathcal{M})$ for the set $\{\vec{b} \in U^m \mid \mathcal{M} \models \varphi(\vec{a}, \vec{b})\}$. In the absence of variables $\vec{x}$ we write $\varphi(\mathcal{M})$ for $\{\vec{b} \mid \mathcal{M} \models \varphi(\vec{b})\}$. Sets of the form $\varphi(\mathcal{M})$ are called *definable*. A function $f : U^n \to U^m$ is definable if its graph $\{(\vec{a}, \vec{b}) \in U^{n+m} \mid \vec{b} = f(\vec{a})\}$ is a definable set.

Given a tuple of variables $\vec{x} = (x_1, \ldots, x_n)$ and a partition $P = \{B_1, \ldots, B_m\}$ on $\{1, \ldots, n\}$, we let $\vec{x}_{B_i}$ stand for the subtuple of $\vec{x}$ consisting of the x_js with $j \in B_i$. For a formula $\varphi(x_1, \ldots, x_n)$, we then say that φ *respects the partition* P (over $\mathcal{M}$) if φ is equivalent to a Boolean combination of formulae each having its free variables among $\vec{x}_{B_i}$ for some $i \leq k$. This will be written as $\varphi \sim_{\mathcal{M}} P$, or just $\varphi \sim P$ if $\mathcal{M}$ is clear from the context.

In other words (by putting a Boolean combination into DNF), $\varphi \sim_{\mathcal{M}} P$ if there exists a family of formulae $\alpha^i_j(\vec{x}_{B_i})$, $i = 1, \ldots, m$, $j = 1, \ldots, k$, such that

$$(*) \qquad \mathcal{M} \models \varphi(\vec{x}) \leftrightarrow \bigvee_{j=1}^{k} (\alpha^1_j(\vec{x}_{B_1}) \wedge \ldots \wedge \alpha^m_j(\vec{x}_{B_m}))$$

When $\mathcal{M}$ has quantifier elimination, all α^i_js are quantifier free. In fact, under the quantifier-elimination assumption, the definition of $\varphi \sim_{\mathcal{M}} P$ can be restated as the equivalence of φ to a quantifier-free formula ψ such that every atomic subformula of ψ uses variables from only one block of P.

We say that in φ, two variables x_i and x_j are *independent* if there exists a partition P such that $\varphi \sim_{\mathcal{M}} P$, and x_i and x_j are in two different blocks of P. Equivalently, x_i and x_j are independent if there exists a partition $P = (\vec{y}, \vec{z})$ of $\vec{x}$ such that $\varphi \sim_{\mathcal{M}} P$, x_i is in $\vec{y}$ and x_j is in $\vec{z}$. (When convenient notationally, we identify partitions on the indices of variables and variables themselves.)

Structures. After presenting a general decidability result, we shall deal with several important classes of structures. Two of them were mentioned already: the real ordered group $\mathbf{R}_{\text{lin}} = \langle \mathbb{R}, +, -, 0, 1, < \rangle$ and the real field $\mathbf{R} = \langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$, corresponding to linear and polynomial constraints over the reals. Some of the results for these structures extend to a larger class of *o-minimal* structures: $\mathcal{M} = \langle U, \Omega \rangle$ is called o-minimal [19, 23] if one of the symbols in Ω is $<$, interpreted as a linear order on U, and every definable subset of U, $\{a \mid \mathcal{M} \models \varphi(a)\}$, is a finite union of points and open intervals. Both $\mathbf{R}_{\text{lin}}$ and $\mathbf{R}$ have quantifier elimination (by Fourier elimination [25], and Tarski's theorem [2, 3], respectively), which easily implies that they are o-minimal. The exponential field $\langle \mathbb{R}, +, \cdot, e^x \rangle$ is an example of a structure which is o-minimal [24] but does not have quantifier elimination [22]. For other o-minimal structures on the reals, see [23].

We shall deal with some structures on the integers. Of most interest to us is $\mathcal{Z}_0 = \langle \mathbb{Z}, +, -, 0, 1, <, (\equiv_k)_{k>1} \rangle$ where $n \equiv_k m$ iff $n = m (\text{mod } k)$. This structure corresponds to constraints given by linear repeating points, which are used for modeling temporal databases [13]. The structure $\mathcal{Z}_0$ admits effective quantifier elimination, and its theory is decidable [7].

3 General conditions for deciding variable independence

Given a structure $\mathcal{M}$, we consider two problems. The *variable independence problem* $\text{VI}_{\mathcal{M}}(\varphi, x_i, x_j)$ is to decide, for $\varphi(x_1, \ldots, x_n)$ in the language of $\mathcal{M}$, if x_i and x_j are independent. The *variable partition problem* $\text{VP}_{\mathcal{M}}(\varphi, P)$ is to decide, for a given formula $\varphi(x_1, \ldots, x_n)$ and a partition P on $\{1, \ldots, n\}$, if $\varphi \sim_{\mathcal{M}} P$.

Note that the variable independence problem is a special case of the variable partition problem, as to solve the former, one needs to solve the latter for some partition $P = (B_1, B_2)$ with $i \in B_1$ and $j \in B_2$.

The above problems are just decision problems, but if the theory of $\mathcal{M}$ is decidable, and the answer to $\mathrm{VP}_{\mathcal{M}}(\varphi, P)$ is 'yes', one can effectively find a representation in the form $(*)$, simply by enumerating all the formulae $\langle\psi(\vec{x})\rangle_i$ which are Boolean combinations of formulae having free variables from at most one block of P, and then checking if $\mathcal{M} \models \forall\vec{x}\ (\varphi(\vec{x}) \leftrightarrow \psi_i(\vec{x}))$. Since $\varphi \sim_{\mathcal{M}} P$, for some finite i, we get a positive answer. In many interesting cases, we shall see better algorithms for finding representation $(*)$ than simple enumeration.

The first easy result shows that the problems $\mathrm{VI}_{\mathcal{M}}(\varphi, x_i, x_j)$ and $\mathrm{VP}_{\mathcal{M}}(\varphi, P)$ are equivalent; this allows us to deal then only with two-block partitions.

Lemma 1. *For any $\mathcal{M}$, the variable independence problem is decidable over $\mathcal{M}$ iff the variable partition problem is decidable over $\mathcal{M}$.*

Next, we discuss conditions for decidability of the variable independence problem. It is clear that one needs decidability of the theory of $\mathcal{M}$. However, decidability alone (and even effective quantifier elimination) are not sufficient.

Proposition 1. *a) If the theory of $\mathcal{M}$ is undecidable, then the variable independence problem is undecidable over $\mathcal{M}$.*

b) There exists a structure $\mathcal{M}$ with a decidable theory and effective quantifier elimination such that the variable independence problem is undecidable over $\mathcal{M}$.

Proof sketch. a) If Φ is a sentence and $\varphi(x, y)$ is $(x = y) \wedge \neg\Phi$, then x and y are independent in φ iff $\mathcal{M} \models \Phi$.

b) An example is provided by the theory of traces from [21]. Let U be a union of three disjoint sets: descriptions of Turing machines, input words, and traces, or partial computations of machines on input words, all appropriately coded as strings. Let Ω contain a constant symbol for every element of U, and a single ternary predicate $P(m, w, t)$ saying that t is a trace of the machine m on the input word w. This signature can be expanded by finitely many symbols so that the expanded model has effective quantifier elimination.

Now fix a Turing machine m_0 and an input word w_0 and consider the formula $\varphi(t, t') = (P(m_0, w_0, t) \wedge (t = t'))$. We then show that t and t' are independent iff m_0 halts on w_0.

The proof of Proposition 1, b), shows that it is essential to be able to decide finiteness in order to decide $\mathrm{VI}(\varphi, x_i, x_j)$ (as it is the finiteness of the number of traces that turns out to be equivalent to variable independence). Recall that a formula $\varphi(x)$ is *algebraic* if $\varphi(\mathcal{M})$ is finite. We say that there is an *effective test for algebraicity* in $\mathcal{M}$ if for every $\varphi(x)$ in the language of $\mathcal{M}$, it is decidable if φ is algebraic. Note that this somewhat technical notion will trivially hold for most relevant classes of constraints.

While the notion of variable independence is needed in the context of constraint databases, for finite relational structures it is assumed to be meaningless as every tuple is represented as a conjunction of constraints of the form $x_i = c_i$, where c_is are constants. For example, the graph $\{(1,2),(3,4)\}$ is given by the formula $((x=1)\wedge(y=2))\vee((x=3)\wedge(y=4))$. Clearly, variables x and y are independent.

However, over arbitrary structures, not every finite definable set would satisfy the variable independence condition. To see this, let $\mathcal{M}=\langle\mathbb{N},C,E\rangle$, where C is a unary relation interpreted as $\{1,2\}$ and E is a binary relation symbol interpreted as $\{(1,2),(2,1)\}$. A routine argument shows that this $\mathcal{M}$ has quantifier elimination, decidable theory, and there is a test for algebraicity. The formula $\varphi(x,y)\equiv E(x,y)$ then defines a finite set, but variables x and y are not independent: this is because the only definable proper subsets of $\mathbb{N}$ are $\{1,2\}$ and $\mathbb{N}-\{1,2\}$, and no Boolean combination of those gives us E. As another example, consider the field of complex numbers, whose theory is decidable and has quantifier elimination [18]. Let $\varphi(x,y)=(x^2+1=0)\wedge(y^2+1=0)\wedge(x+y=0)$. It defines the finite set $\{(i,-i),(-i,i)\}$ but nevertheless x and y are not independent (since i is not definable).

To avoid similar situations, we impose an extra condition on a structure, again, well known in model theory [4,11]. We say that $\mathcal{M}$ has *definable Skolem functions* if for every formula $\varphi(\vec{x},\vec{y})$ there exists a definable function $f_\varphi(\vec{x})$ with the property that $\mathcal{M}\models\forall\vec{x}\;(\exists\vec{y}\;\varphi(\vec{x},\vec{y})\rightarrow\varphi(\vec{x},f_\varphi(\vec{x})))$. In other words, $f_\varphi(\vec{a})$ is an element of $\varphi(\vec{a},\mathcal{M})$, assuming $\varphi(\vec{a},\mathcal{M})$ is not empty. We say that a Skolem function f_φ is *invariant* [18], if $\varphi(\vec{a}_1,\mathcal{M})=\varphi(\vec{a}_2,\mathcal{M})$ implies $f_\varphi(\vec{a}_1)=f_\varphi(\vec{a}_2)$. If the existence of such a Skolem function can be guaranteed for every φ, we say that $\mathcal{M}$ has definable invariant Skolem functions.

Theorem 1. *Assume that $\mathcal{M}$ has the following properties:*

(a) its theory is decidable;
(b) $\mathcal{M}$ has effective test for algebraicity; and
(c) $\mathcal{M}$ has definable invariant Skolem functions.

Then the variable partition and independence problems are decidable over $\mathcal{M}$.

Proof sketch. We consider the case of two block partitions; that is, deciding if a formula $\varphi(\vec{x},\vec{y})$ respects the partition P with blocks $\vec{x}$ and $\vec{y}$. Let $\vec{x}$ have length n and $\vec{y}$ have length l. Define an equivalence relation on U^n by

$$\vec{a}_1\equiv\vec{a}_2 \quad\text{iff}\quad \varphi(\vec{a}_1,\mathcal{M})=\varphi(\vec{a}_2,\mathcal{M}).$$

Lemma 2. *For φ, P and $\equiv$ as above, $\varphi\sim_{\mathcal{M}}P$ iff $\equiv$ has finitely many equivalence classes.*

Using this and the assumptions on $\mathcal{M}$, we show how to define a formula $\chi(\vec{x})$ finding a set of representatives of the equivalence classes of $\equiv$; then again using the assumptions on $\mathcal{M}$ we show that it is decidable if $\chi(\mathcal{M})$ is finite.

The proof of Theorem 1 gives an explicit construction for a formula witnessing $\varphi\sim_{\mathcal{M}}P$, where P has two blocks. We now extend it to arbitrary partitions.

Let $\varphi(x_1,\ldots,x_n)$ be given, and let $B\subset\{1,\ldots,n\}$. Let $card(B)=k$. For $\vec{a}\in U^k$, by $\varphi_B(\vec{a},\mathcal{M})$ we denote the set of $\vec{b}\in U^{n-k}$ such that $\varphi(\vec{c})$ holds, where

$\vec{c}$ is obtained from $\vec{a}$ and $\vec{b}$ by putting their elements in the appropriate position, $\vec{a}$ being in the positions specified by B. For example, if $n = 4$, $B = \{2, 4\}$, and $\vec{a} = (a_1, a_2)$, $\vec{b} = (b_1, b_2)$, then $\vec{c}$ is (b_1, a_1, b_2, a_2). Formally, for $i \in [1, n]$, let k_1 be the number of $j \in B$ with $j \leq i$, and k_2 be the number of $j \notin B$ with $j \leq i$. Then c_i is a_{k_1} if $i \in B$, and b_{k_2}, if $i \notin B$.

We use the notation

$$\vec{a}_1 \equiv^{\varphi}_{B_i} \vec{a}_2 \quad \text{iff} \quad \varphi_{B_i}(\vec{a}_1, \mathcal{M}) = \varphi_{B_i}(\vec{a}_2, \mathcal{M}).$$

We now obtain the following characterization of $\mathrm{VP}_{\mathcal{M}}(\varphi, P)$.

Corollary 1. *Let $\mathcal{M}$ be as in Theorem 1, and let $\varphi(x_1, \ldots, x_n)$ and a partition $P = (B_1, \ldots, B_m)$ on $\{1, \ldots, n\}$ be given. Then:*

1. *For each $i \leq m$, it is decidable if the equivalence relation $\equiv^{\varphi}_{B_i}$ has finitely many equivalence classes. Furthermore, $\varphi \sim_{\mathcal{M}} P$ iff each $\equiv^{\varphi}_{B_i}$ has finitely many classes.*
2. *If $\varphi \sim_{\mathcal{M}} P$, then one can further effectively find integers $N_1, \ldots, N_m > 0$ and formulae $\alpha^i_j(\vec{x}_{B_i})$, $i = 1, \ldots, m$, $j = 1, \ldots, N_i$, such that $\equiv^{\varphi}_{B_i}$ has N_i equivalence classes, which are definable by the formulae $\alpha^i_j(\vec{x}_{B_i})$, $j \leq N_i$. Furthermore,*

$$\mathcal{M} \models \forall \vec{x} \left(\varphi(\vec{x}) \leftrightarrow \bigvee_{(j_1, \ldots, j_m) \in K} \alpha^1_{j_1}(\vec{x}_{B_1}) \wedge \ldots \wedge \alpha^m_{j_m}(\vec{x}_{B_m}) \right)$$

where

$$K = \{(j_1, \ldots, j_m) \mid \mathcal{M} \models \exists \vec{x} \left(\alpha^1_{j_1}(\vec{x}_{B_1}) \wedge \ldots \wedge \alpha^m_{j_m}(\vec{x}_{B_m}) \wedge \varphi(\vec{x}) \right)\}.$$

4 Decidability for specific classes of constraints

The general decidability result can be applied to a variety of structures, most notably, those that we listed earlier as the ones particularly relevant to constraint database applications (especially to spatial and temporal databases). In fact, the problem will be shown to be decidable for linear constraints over the rationals and the reals (this corresponds to structures $\langle \mathbb{Q}, +, -, 0, 1, < \rangle$ and $\mathbf{R}_{\text{lin}}$), polynomial constraints over the reals ($\mathbf{R}$), and linear repeating points [13] ($\mathcal{Z}_0$).

4.1 Constraints on the integers

Here the result follows easily form Theorem 1.

Proposition 2. *Let $\mathcal{M}$ be $\langle \mathbb{N}, <, \ldots \rangle$ or $\langle \mathbb{Z}, <, \ldots \rangle$, and let its theory be decidable. Assume, in the latter case, that there is at least one definable constant in $\mathcal{M}$. Then the variable partition and independence problems are decidable over $\mathcal{M}$.*

Corollary 2. *The variable partition problem is decidable over $\mathcal{Z}_0 = \langle \mathbb{Z}, +, -, 0, 1, <, (\equiv_k)_{k>1} \rangle$.*

4.2 Linear and polynomial constraints over the reals

The linear constraints over the reals (corresponding to the structure $\mathbf{R}_{\text{lin}} = \langle \mathbb{R}, +, -, 0, 1, <\rangle$) and the polynomial constraints over the reals (corresponding to $\mathbf{R} = \langle \mathbb{R}, +, \cdot, 0, 1, <\rangle$) are the most useful constraints for spatial and spatio-temporal applications, where the problem of variable independence originated, and where variable independence is used in system prototypes. We thus concentrate on these constraints.

In many cases, however, we can state the results in greater generality using the concept of o-minimality (cf. section 2).

It is known that every o-minimal expansion of the $\mathbf{R}_{\text{lin}}$ has definable invariant Skolem functions [18, 23]. Since every definable subset of U is a finite union of points and open intervals, one can test algebraicity, assuming that the order is dense: given $\varphi(x)$, the sentence $\exists u \exists v \forall x\ (u < x < v \rightarrow \varphi(x))$ tests if $\varphi(\mathcal{M})$ is infinite. This shows

Corollary 3. *Let $\mathcal{M} = \langle \mathbb{R}, +, 0, 1, <, \ldots\rangle$ be o-minimal, and have a decidable theory. Then the variable partition and independence problems are decidable over $\mathcal{M}$. In particular, these problems are decidable over $\mathbf{R}_{\text{lin}}$ and $\mathbf{R}$.*

Since $\langle \mathbb{Q}, +, -, 0, 1, <\rangle$ is elementarily equivalent to $\mathbf{R}_{\text{lin}}$, we conclude that the variable partition problem is decidable over it, too.

Uniform decidability and complexity bounds Our next goal is to present a *uniform* procedure for solving the problem $\mathrm{VI}_{\mathcal{M}}(\varphi, P)$. More precisely, we say that the variable independence problem is *uniformly decidable* over $\mathcal{M}$ if the theory of $\mathcal{M}$ is decidable, and for every partition P on $\{1, \ldots, n\}$, there exists a single sentence Φ_P in the language of $\mathcal{M}$ expanded with an n-ary relation symbol S such that for any formula $\varphi(x_1, \ldots, x_n)$,

$$\varphi \sim_{\mathcal{M}} P \quad \text{iff} \quad (\mathcal{M}, \varphi(\mathcal{M})) \models \Phi_P.$$

Here $(\mathcal{M}, \varphi(\mathcal{M}))$ is the expansion of $\mathcal{M}$ where the new symbol S is interpreted as $\{\vec{a} \mid \mathcal{M} \models \varphi(\vec{a})\}$. Note that the decidability of the theory of $\mathcal{M}$ implies that $(\mathcal{M}, \varphi(\mathcal{M})) \models \Phi_P$ is decidable.

Proposition 3. *Let $\mathcal{M} = \langle \mathbb{R}, +, 0, 1, <, \ldots\rangle$ be o-minimal and have a decidable theory. Then the variable independence problem and partition problems are uniformly decidable over $\mathcal{M}$.*

Proof sketch. We show explicitly how to construct invariant Skolem function for a given equivalence relation. Given a (definable) set Y of representatives of a definable equivalence relation, its finiteness is tested as follows: Let X be the set of all coordinates of elements of Y. Since this is a definable set, it is finite iff it does not contain an open interval (by o-minimality). This condition can be tested by a sentence in the language of $\mathcal{M}$.

Proposition 3 implies that the variable independence problem is uniformly decidable over $\mathbf{R}_{\text{lin}}$ and $\mathbf{R}$. The main application of this result is in establishing complexity bounds.

Since $\mathbf{R}$ admits quantifier elimination, every semi-algebraic set is given by a Boolean combination of polynomial inequalities. Thus, a standard way to represent a semi-algebraic set in $\mathbb{R}^n$ [1, 3, 20] is by specifying a collection of polynomials $p_1, \ldots, p_k \in \mathbb{Z}[x_1, \ldots, x_n]$, and defining a set X as a Boolean combination of sets of the form $\{\vec{a} \mid p_i(\vec{a}) \; \theta \; 0\}$, where θ is either $=$ or $>$. Here $\mathbb{Z}[x_1, \ldots, x_n]$, as usual, is the set of all polynomials in n variables with coefficients from $\mathbb{Z}$. One can use coefficients from $\mathbb{Q}$ as well, but this would not affect the class of definable sets.

Thus, when we study complexity of $\text{VP}_{\mathbf{R}}(\varphi, P)$, we assume that φ is given a Boolean combination of polynomial equalities and inequalities, with all polynomials having integer coefficients. The size of the input formula is then defined in a standard way, assuming that all integer coefficients are given in binary. All the above applies to semi-linear sets (that is, sets definable over $\mathbf{R}_{\text{lin}}$); we just restrict our attention to polynomials of degree 1.

Corollary 4. *Let $\mathcal{M}$ be $\mathbf{R}_{\text{lin}}$ or $\mathbf{R}$. Let P be a fixed partition on $\{1, \ldots, n\}$. Then, for a semi-algebraic (semi-linear) set given by a Boolean combination $\varphi(\vec{x})$ of polynomial inequalities (of degree 1), the problem $\text{VI}_{\mathcal{M}}(\varphi, P)$ is solvable in time polynomial in the size of φ.*

Proof sketch. This follows from Proposition 3 and complexity bounds on quantifier elimination [1, 20].

Another reason to consider the uniform decision procedure for variable independence is that it gives us a test for variable independence under some transformations. For example, linear coordinate change in general would destroy variable independence, although it has relatively little effect on shapes on objects in $\mathbb{R}^n$. Consider, for example, the following version of the variable independence problem $\text{LVI}(X, x_i, x_j)$: Given a semi-algebraic set $X \subseteq \mathbb{R}^n$ (defined by a formula over $\mathbf{R}$), is there a linear change of coordinates such that in the new coordinate system, variables x_i and x_j are independent?

The general decision procedure of Theorem 1 does not give us a decision procedure for LVI. However, using uniformity, we easily obtain:

Corollary 5. *The problem $\text{LVI}(X, x_i, x_j)$ is decidable.*

It turns out that not only the decision part of $\text{VI}_{\mathcal{M}}(\varphi, P)$ and $\text{VP}_{\mathcal{M}}(\varphi, P)$ can be solved in polynomial time for a fixed P over $\mathbf{R}_{\text{lin}}$ and $\mathbf{R}$, but there is also a polynomial time algorithm for finding a formula equivalent to φ that witnesses $\varphi \sim_{\mathcal{M}} P$.

Theorem 2. *1. Given $n > 1$, and a partition $P = (B_1, \ldots, B_k)$ on $\{1, \ldots, n\}$, there exists an algorithm that, for every semi-algebraic set given by a formula $\varphi(x_1, \ldots, x_n)$ which is a Boolean combination of polynomial equalities and inequalities, tests if $\varphi \sim_{\mathcal{M}} P$, and in the case of the positive answer, computes quantifier-free formulae $\alpha^i_j(\vec{x}_{B_i})$ such that each $\alpha^i_j(\vec{x}_{B_i})$ is* *a Boolean combination of polynomial (in)equalities (where polynomials depend only on $\vec{x}_{B_i}$ and all coefficients are integers), and $\varphi(\vec{x})$ is equivalent to $\bigvee_j \bigwedge_i \alpha_j^i(\vec{x}_{B_i})$. Moreover the algorithm works in time polynomial in the size of φ.*

2. *The same statement is true when on replaces semi-algebraic by semi-linear, and all polynomials are of degree* 1*.*

Proof combines Corollary 1, uniform decidability (Proposition 3), complexity bounds for quantifier elimination [1, 20] and, for 1), algorithms for polynomial root isolation [6].

In the full version, we also consider the most typical case of spatio-temporal applications: sets in $\mathbb{R}^3$ given by formulae $\varphi(x, y, t)$, where x, y describe the spatial component and t describes the temporal component. For this case, we present an algorithm based on cylindrical algebraic decomposition [3] for faster testing of variable independence.

5 Conclusion

We looked at the problem of deciding, for a set represented by a collection of constraints, whether some variables in those constraints are independent of each other. Knowing this can considerably improve the running time of several constraint processing algorithms, in particular, quantifier elimination. The problem originated in the field of spatio-temporal databases represented by constraints (linear or polynomial over the reals, for example); it was demonstrated that on large datasets, reasonable performance can only be achieved if variables comprise small independent groups. It had not been known, however, if such independence conditions are decidable.

Here we showed that these conditions are decidable for a large class of constraints, including those relevant to spatial and temporal applications. Moreover, for linear and polynomial constraints over the reals, we gave a uniform decision procedure that implies tractability, and we showed that a given constraint set can be converted into one in a nice shape in polynomial time, too.

Acknowledgements I thank Stavros Cosmadakis and Gabi Kuper for bringing the problem to my attention, and Michael Benedikt and Luc Segoufin for their comments.

References

1. S. Basu. New results on quantifier elimination over real closed fields and applications to constraint databases. *Journal of the ACM*, 46 (1999), 537–555.
2. J. Bochnak, M. Coste, M.-F. Roy. *Real Algebraic Geometry.* Springer Verlag, 1998.
3. B.F. Caviness and J.R. Johnson, Eds. *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Springer Verlag, 1998.
4. C.C. Chang and H.J. Keisler. *Model Theory.* North Holland, 1990.

5. J. Chomicki, D. Goldin and G. Kuper. Variable independence and aggregation closure. In *PODS'96*, pages 40–48.
6. G. E. Collins and R. Loos. Real zeros of polynomials. In *Computer Algebra: Symbolic and Algebraic Computation*, Springer-Verlag, 1983, pages 83–94.
7. H. B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, New York, 1972.
8. A. Fordan and R. Yap. Early projection in CLP(R). In *CP'98*, pages 177–191.
9. S. Grumbach, P. Rigaux, L. Segoufin. The DEDALE system for complex spatial queries. In *SIGMOD'98*, pages 213–224.
10. S. Grumbach, P. Rigaux, L. Segoufin. On the orthographic dimension of constraint databases. In *ICDT'99*, pages 199–216.
11. W. Hodges. *Model Theory.* Cambridge, 1993.
12. J.-L. Imbert. Redundancy, variable elimination and linear disequations. In *SLP'94*, pages 139–153.
13. F. Kabanza, J.-M. Stevenne and P. Wolper. Handling infinite temporal data. *Journal of Computer and System Sciences*, 51 (1995), 3–17.
14. P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51 (1995), 26–52. Extended abstract in *PODS'90*, pages 299–313.
15. G. Kuper, L. Libkin and J. Paredaens, eds. *Constraint Databases.* Springer Verlag, 2000.
16. L. Libkin. Some remarks on variable independence, closure, and orthographic dimension in constraint databases. *SIGMOD Record* 28(4) (1999), 24–28.
17. L. Libkin. Variable independence, quantifier elimination, and constraint representations. Bell Labs Technical Memo, 1998.
18. D. Marker, M. Messmer and A. Pillay. *Model Theory of Fields.* Springer Verlag, 1996.
19. A. Pillay, C. Steinhorn. Definable sets in ordered structures. III. *Trans. AMS* 309 (1988), 469–476.
20. J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. *J. Symb. Comp.* 13 (1992), 255–352.
21. A. Stolboushkin and M. Tsaitlin. Finite queries do not have effective syntax. *Information and Computation*, 153 (1999), 99–116.
22. L. van den Dries. Remarks on Tarski's problem concerning (R,+,*,exp). In *Logic Colloquium'82*, North Holland, 1984, pages 97–121.
23. L. van den Dries. *Tame Topology and O-Minimal Structures.* Cambridge, 1998.
24. A.J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponential function. *J. Amer. Math. Soc.* 9 (1996), 1051–1094.
25. G.M. Ziegler. *Lectures on Polytopes.* Springer-Verlag, 1994.