Computational Complexity of Linear Constraints over the Integers

Peter Jonsson[a,*], Tomas Lööw[b]

[a]*Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden. Phone +46 (0)13 282415, fax +46 (0)13 142231*
[b]*Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden. Phone +46 (0)13 286607, fax +46 (0)13 142231*

Abstract

Temporal reasoning problems arise in many areas of AI, including planning, natural language understanding, and reasoning about physical systems. The computational complexity of continuous-time temporal constraint reasoning is fairly well understood. There are, however, many different cases where discrete time must be considered; various scheduling problems and reasoning about sampled physical systems are two examples. Here, the complexity of temporal reasoning is not as well-studied nor as well-understood. In order to get a better understanding, we consider the powerful Horn Disjunctive Linear Relations (Horn DLR) formalism adapted for discrete time and study its computational complexity. We show that the full formalism is NP-hard and identify several maximal tractable subclasses. We also 'lift' the maximality results to obtain hardness results for other families of constraints. Finally, we discuss how the results and techniques presented in this paper can be used for studying even more expressive classes of temporal constraints.

Keywords: Temporal reasoning, discrete time, computational complexity

1. Introduction

Reasoning about time is ubiquitous in artificial intelligence and many different branches of computer science. Noteworthy examples include planning, diagnosis, and temporal databases. For a general overview of temporal rea-

*Corresponding author.
Email addresses: `peter.jonsson@liu.se` (Peter Jonsson), `tomas.loow@liu.se` (Tomas Lööw)

soning, see, for instance, the survey by Chittaro & Montanari [11], the handbook edited by Fisher et al. [16], or the handbook edited by Rossi et al. [37]. The temporal constraint satisfaction problem is very well-studied and there has lately been substantial progress in understanding the complexity of this problem. Bodirsky and Kára [8] have presented a complete classification of the temporal constraint problem for relations that are first-order definable in the structure $(\mathbb{Q}; <)$. This result subsumes a large portion of previous work on *qualitative* (that is, the case where we cannot refer to individual time points in the underlying time structure) temporal constraints based on time *points*; one may note that this result does not cover formalisms such as the Allen algebra (which is intrinsically based on non-degenerate intervals instead of points). There are no such unifying result for metric temporal constraints, but many partial results are known, cf. Barber [2], Jonsson & Bäckström [22], Krokhin et al. [27], and Wetprasit & Sattar [39].

The situation is very different if we turn our attention to *discrete* temporal constraints where the set of time points is some subset of the set of integers $\mathbb{Z}$. There are some scattered complexity results (see, for example, [3, 25, 30]) but a coherent picture is lacking. This is unsatisfactory since reasoning about discrete time is an important part of AI: let us just mention temporal logics, plan generation, and discrete time Markov chains as three concrete examples. Reasoning about discrete time is also inevitable in many 'industrial' settings: for systems that are repeatedly sampled (for monitoring or other purposes), we are implicitly forced to assume that the underlying model of time is discrete. Our goal with this paper is to initiate a systematic study of temporal constraint satisfaction under the assumption that time is discrete instead of continuous. The focus will be on the computational complexity of such problems; more precisely, we aim at identifying restricted classes of constraints such that the corresponding constraint satisfaction problem can be solved in polynomial time. Obtaining a full classification of hard and easy cases is of course highly desirable — it gives us a very powerful tool for studying the complexity of problems that can be modelled within the language. Since temporal constraint reasoning appears as a subproblem in many different types of automated reasoning, we expect such results to be useful in many other contexts, too. For instance, note that discrete semilinear relations (to be defined later on) have been used intensively for a long time in, for example, constraint databases [25, 35], formal verification [10], distributed computing [1], automata therory [34], and in the study of Presburger arithmetic and other logical formalisms [19]. We also note that results of this kind may be interesting for *satisfiability modulo theories* (SMT), i.e. the satisfiability problem for logical formulas over different background theories. The article by Nieuwenhuis et al. [31] or Gansesh's dissertation [17] may serve as introductions to this highly interesting topic.

We divide the rest of this introduction into three parts: we introduce temporal constraint problems in the first, we briefly discuss computational complexity in the second, and give an outline of the article in the third.

1.1. Temporal constraint problems

In order to introduce temporal constraint reasoning formally, we first define the general constraint satisfaction problem.

Definition 1. *Let Γ be a set of finitary relations over some set D of values. The constraint satisfaction problem over Γ (CSP(Γ)) is defined as follows:*

Instance: *A set V of variables and a set C of* constraints $R(v_1, \ldots, v_k)$ *where k is the arity of R, $v_1, \ldots, v_k \in V$ and $R \in \Gamma$.*
Question: *Is there a total function $f : V \to D$ such that $(f(v_1), \ldots, f(v_k)) \in R$ for each constraint $R(v_1, \ldots, v_k)$ in C?*

The set Γ is referred to as the *constraint language*. Observe that we do not require Γ to be a finite set. Given a set D, we let $\Gamma|_D$ denote Γ restricted to D, i.e. $\Gamma|_D = \{R \cap D^n \mid R \in \Gamma \text{ and } R \text{ has arity } n\}$. We sometimes slightly abuse notation to avoid unnecessary clutter. For instance, we may say 'the relation $x = y + z$' instead of 'the relation $\{(x, y, z) \in \mathbb{Z}^3 \mid x = y + z\}$.'

Let us now turn our attention to temporal constraint problems. We let $D \subseteq \mathbb{R}$ denote a set of *time points*. Let the set $\mathcal{S}_D$ contain all relations $\{(x_1, \ldots, x_n) \in D^n \mid C_1 \wedge \ldots \wedge C_k\}$ where each clause C_i denotes a disjunction $(p_1 r_1 c_1 \vee \ldots \vee p_m r_m c_m)$. Here, c_j is an integer, $r_j \in \{<, \leq, =, \neq, \geq, >\}$ and $p_j(x_1, \ldots, x_n)$ is a linear polynomial (i.e. the degree of p equals one) with integer coefficients. The relations P_1, Q_1, and R_1 below are examples of members in $\mathcal{S}_D$.

- $P_1(x, y, z) \equiv (x = 1 \vee y = 1) \wedge (x = 0 \vee z = 1)$,
- $Q_1(x, y) \equiv 5x + 3y \leq 8 \wedge 3x + 5y \geq 8$, and
- $R_1(x, y, z) \equiv x + y + z \leq 0 \vee x \neq 1 \vee y \neq 1 \vee z \neq 1$.

We adopt a simple representation of relations in $\mathcal{S}_D$: every relation R in $\mathcal{S}_D$ is represented by its defining formula where each coefficient is written in binary. One may note that all possible choices of the relation r_j are not necessary for defining the set $\mathcal{S}_D$; for example, $x \neq y \Leftrightarrow x < y \vee x > y$. However, it simplifies the definition of the forthcoming classes of relations.

Let $\mathcal{D}_D \subseteq \mathcal{S}_D$ contain the relations that are defined by a single clause. Let $\mathcal{H}_D \subseteq \mathcal{D}_D$ contain the relations that are defined by a single clause that contains at most one relation that is not of the type $p(\bar{x}) \neq c$. The names $\mathcal{S}$, $\mathcal{D}$, and $\mathcal{H}$ are chosen to reflect the names given to the corresponding relations in the literature: the relations in $\mathcal{S}_D$ are called *semilinear relations*, the relations in $\mathcal{D}_D$ are called *disjunctive linear relations* (DLRs), and the relations in $\mathcal{H}_D$ are called *Horn DLRs*. DLRs and Horn DLRs were introduced in [22, 26] but only for continuous time structures (in fact, only for the set $\mathbb{R}$ of real numbers). To make things more concrete, $\mathcal{H}_\mathbb{Z}$ contains for example the following relations:

- $P_2(x, y, z) \equiv x - y + z \geq 0 \vee y \neq 1 \vee z \neq 0$,
- $Q_2(x) \equiv x = 17$, and
- $R_2(x, y, z) \equiv x \neq 0 \vee y \neq 0 \vee z \neq 0$.

It is worth noting that the clauses are not Horn clauses in the classical meaning of the word, but the name Horn DLRs is choosen because they are structurally similar. Just like ordinary Horn clauses, a Horn DLR clause can be considered an implication since

$$p(\bar{x})\; r\; c \vee q_1(\bar{x}) \neq d_1 \vee \ldots \vee q_k(\bar{x}) \neq d_k$$

is equivalent to

$$(q_1(\bar{x}) = d_1 \wedge \ldots \wedge q_k(\bar{x}) = d_k) \rightarrow p(\bar{x})\; r\; c.$$

Horn DLRs have appeared in different guises several times in the literature; see [22, 26] for examples and references. We also note that the modelling power (in continuous time) of $\mathcal{H}_\mathbb{R}$ is quite high; many tractable fragments described in the literature are within $\mathcal{H}_\mathbb{R}$ [22, 26]. This indicates that $\mathcal{H}_\mathbb{Z}$ may be interesting from a modelling point of view, too.

1.2. Computational complexity

When studying constraint satisfaction problems with infinite constraint languages, one often makes a distinction between *local* and *global* properties, cf. Bodirsky and Grohe [6].

Definition 2. *A constraint satisfaction problem CSP*(Γ) *is* globally tractable *if CSP*(Γ) *is in P and* locally tractable *if CSP*(Γ') *is in P for every finite set* $\Gamma' \subseteq \Gamma$. *Similarly, CSP*(Γ) *is* globally NP-hard *if CSP*(Γ) *is NP-hard and* locally NP-hard *if CSP*(Γ') *is NP-hard for some finite set* $\Gamma' \subseteq \Gamma$.

It is clear that global tractability implies local tractability. Something far less obvious is that there are infinite constraint langauges that are globally NP-hard but locally tractable. We will discuss the implications of this in Section 6.2, and also present a concrete temporal language with this property.

The separation of local and global tractability/NP-hardness is, among other things, motivated by Theorem 3 below. We need some machinery to state this result. Given a constraint language Γ, we say that a relation R has a *positive primite definition* (pp-definition) in Γ if it can be defined by a first-order formula over Γ without using disjunction and negation, and with only existential quantification. As an example, consider the language

$$\Theta = \{\{(x, y, z) \in \mathbb{Z}^3 \mid x = y + z\}, \{(x, y) \in \mathbb{Z}^2 \mid x \neq y\}, \{1\}\}$$

and note that the relations $x \neq y + 2$ and $x = 5y$ can be pp-defined in Θ:

- $x \neq y + 2 \Leftrightarrow \exists v, w, z. w = 1 \wedge z = w + w \wedge v = y + z \wedge x \neq v$
- $x = 5y \Leftrightarrow \exists v, w. v = y + y \wedge w = v + v \wedge x = w + y$

Let $\langle\Gamma\rangle$ (the *closure* or *co-clone* of Γ) denote all relations that are pp-definable in Γ. The following theorem is due to Jeavons [21].

Theorem 3. *For every finite $\Theta \subseteq \langle\Gamma\rangle$, CSP($\Theta$) is polynomial-time reducible to CSP(Γ). Furthermore, if $R \in \langle\Gamma\rangle$, then CSP($\Gamma \cup \{R\}$) and CSP($\Gamma$) are polynomial-time equivalent problems.*

An immediate consequence is that if CSP(Γ) is globally tractable, then CSP($\langle\Gamma\rangle$) is locally tractable. This theorem will be very important in the sequel since it gives us a convenient method for proving many different complexity results.

We continue by providing some complexity results for different temporal formalisms. For $a, b, c \in \mathbb{Z}$, define $T_{a,b} = \{(a, a, b), (a, b, a), (b, a, a)\}$ and $T'_{a,b,c}(x, y) \equiv \{a, b, c\}^2 \setminus \{(a, a), (b, b), (c, c)\}$.

Proposition 4. *CSP($\{T_{a,b}\}$) and CSP($\{T'_{a,b,c}\}$) are NP-hard problems whenever a, b, c are distinct numbers in $\mathbb{Z}$.*

Proof. CSP($\{T_{a,b}\}$) is an NP-hard problem since it corresponds to 1-IN-3-SAT restricted to clauses without negated literals (see problem LO4 in Garey & Johnson [18]). The problem CSP($\{T'_{a,b,c}\}$) is NP-hard problem since it corresponds to 3-COLOURABILITY (see problem GT4 in Garey & Johnson [18]). ∎

Theorem 5. *CSP($\mathcal{H}_{\mathbb{R}}$) is globally tractable while CSP($\mathcal{H}_{\mathbb{Z}}$) is locally NP-hard. Furthermore, CSP($\mathcal{D}_D$) and CSP($\mathcal{S}_D$) are locally NP-hard when $D \in \{\mathbb{Z}, \mathbb{R}\}$.*

Proof. Global tractability of CSP($\mathcal{H}_{\mathbb{R}}$) and local NP-hardness of CSP($\mathcal{D}_{\mathbb{R}}$) and CSP($\mathcal{S}_{\mathbb{R}}$) follows from [22]. For the remaining cases, it is sufficient to prove local NP-hardness of CSP($\mathcal{H}_{\mathbb{Z}}$). Simply note that we can pp-define $T_{0,1}$ in $\mathcal{H}_{\mathbb{Z}}$ by

$$T_{0,1}(x, y, z) \equiv x \geq 0 \wedge y \geq 0 \wedge z \geq 0 \wedge x + y + z = 1$$

and thereafter apply Proposition 4 and Theorem 3. □

One should also note that CSP($\mathcal{S}_{\mathbb{Z}}$) (and, consequently, the problems CSP($\mathcal{D}_{\mathbb{Z}}$) and CSP($\mathcal{H}_{\mathbb{Z}}$)) are in NP; this is a folklore result that can be proven without too much effort by using Papadimitriou's [32] observation that integer programming is in NP. Since CSP($\mathcal{H}_{\mathbb{Z}}$) is locally NP-hard, it makes sense to start looking for tractable fragments within $\mathcal{H}_{\mathbb{Z}}$, and this is a natural first step in a bottom-up approach to classifying the complexity of CSP($\mathcal{D}_{\mathbb{Z}}$) and CSP($\mathcal{S}_{\mathbb{Z}}$). We will concentrate on identifying tractable fragments and study their *maximality* in the forthcoming sections. Given constraint languages $\Gamma \subseteq \Theta$, we say that Γ is *maximally tractable* in Θ if CSP(Γ) is globally tractable and CSP($\Gamma \cup \{R\}$) is locally NP-hard for every $R \in \Theta \setminus \Gamma$. Maximality can obviously be defined in different ways with respect to local and global properties but this definition is sufficent for our purposes. Note that if a language Γ is maximal in Θ, then there may be a language Θ' such that $\Theta \subseteq \Theta'$ and Γ is not maximal in Θ'; it is in general very important to state which set the maximality relates to. However, since we are exclusively interested in maximal fragments of $\mathcal{H}_{\mathbb{Z}}$ in this article, we allow ourselves to sometimes write 'Γ is maximal' instead of 'Γ is maximal in $\mathcal{H}_{\mathbb{Z}}$'.

1.3. Outline

The main part of this article is devoted to four different types of tractable temporal constraint problems.

Section 2. We consider problems where solutions can be 'scaled' to integer solutions and we use this property for abstractly defining the contraint language $\Lambda_{\mathbb{Z}}$. The polynomial-time algorithm for CSP($\Lambda_{\mathbb{Z}}$) is simple: check if there is a solution over the real numbers, and this can be done in polynomial time by using an algorithm by Jonsson & Bäckström [22]. The abstract formulation is not quite useable in the maximality proof so we make an alternative concrete characterisation of $\Lambda_{\mathbb{Z}}$; we obtain this by using the concept of *reduced formulas* [4]. Armed with this characterisation, we provide a maximality proof and also present a generalised hardness result for constraint languages that are not necessarily subsets of $\mathcal{H}_{\mathbb{Z}}$. Since the basic result on scalability is applicable to a wide range of constraint languages, we conclude this section by considering the problem of deciding whether a given relation in $\mathcal{D}_{\mathbb{Z}}$ is scalable or not.

Section 3. In this section, we study a constraint language that is based on linear equations extended with certain disjunctions. The polynomial-time algorithm for this problem is based on a result on the solvability of linear equations over integers [24] combined with a general technique for handling disjunctions [12]. The maximality proof is once again based on exploiting reduced formulas.

Section 4. If we do not count relations of the type $p(\bar{x}) \neq c$, then the results in Section 2 are mostly concerned with relations of the type $p(\bar{x}) \geq c$ while the results in Section 3 are concerned with relations of the type $p(\bar{x}) = c$. It is thus natural to study how these two types of relations can be mixed. We give an example of such a 'mixed' class Ψ in this section. The tractable algorithm for CSP(Ψ) is an extension of an algorithm by Bodirsky et al. [9]. The maximality proof is slightly more complicated than in the two previous sections so we have divided it into several parts. The proof is based on fairly complex pp-definitions so we use some elementary number theory and linear algebra in order to simplify both the constructions and their presentations.

Section 5. A relation R is *k-valid* if the tuple $(k, \ldots, k) \in R$. Obviously, the constraint satisfaction problem over the set $\Gamma_k \subseteq \mathcal{H}_{\mathbb{Z}}$ of k-valid relations is tractable. We show that Γ_k is a maximal tractable subclass of $\mathcal{H}_{\mathbb{Z}}$ for every $k \in \mathbb{Z}$, and this demonstrates that there are an infinite number of maximal tractable fragments in $\mathcal{H}_{\mathbb{Z}}$.

We conclude the paper by discussing the results and future research directions. We address, for example, full complexity classifications of fragments within $\mathcal{S}_{\mathbb{Z}}$, certain issues arising when dealing with locally tractable problems, and connections with finite-domain constraint satisfaction problems.

This article is a revised version of a conference paper [23]; one should note that Section 2.3, Section 4 and most of Section 6 do not appear in the earlier version.

2. Scalable constraints

One way to start looking for tractable fragments of $\mathcal{H}_\mathbb{Z}$ is to ask under which circumstances a solution to an instance I of CSP($\mathcal{H}_\mathbb{R}$) implies a solution to the corresponding instance $I|_\mathbb{Z}$ of CSP($\mathcal{H}_\mathbb{Z}$). In Section 2.1, we begin by identifying such a condition (which we refer to as *scalability*) and define $\Lambda_\mathbb{Z}$ to be the scalable relations in $\mathcal{H}_\mathbb{Z}$. We continue, in Section 2.2, by proving that $\Lambda_\mathbb{Z}$ is maximal in $\mathcal{H}_\mathbb{Z}$. Finally, we show how to decide whether a given relation $R \in \mathcal{S}_\mathbb{Z}$ is scalable or not in Section 2.3.

2.1. Scalability and the language $\Lambda_\mathbb{Z}$

Our starting point is the following result[1].

Lemma 6. *Let Γ be a constraint language over $\mathbb{R}$ such that the following holds.*

1. *Every satisfiable instance of CSP(Γ) is satisfied by some rational point.*
2. *For each $R \in \Gamma$, it holds that if $\bar{x} = (x_1, x_2, \ldots, x_k) \in R$, then $(ax_1, ax_2, \ldots, ax_k) \in R$ for all $a \in \{y \in \mathbb{R} \mid y \geq 1\} \setminus X$ where X is a (possibly empty) bounded set. The set X may depend on both R and $\bar{x}$.*
3. *CSP(Γ) is globally (or locally) tractable.*

Then, the problem CSP($\Gamma|_\mathbb{Z}$) is also globally (or locally) tractable.

Proof. Let I be an arbitrary satisfiable instance of CSP(Γ) with a rational solution $\bar{x} = (x_1/y_1, \ldots, x_k/y_k)$ where $x_1, \ldots, x_k \in \mathbb{Z}$ and $y_1, \ldots, y_k \in \mathbb{Z}^+ \setminus \{0\}$. Let $n = \prod_{i=1}^{k} y_i$ and note that $n \geq 1$.

For an arbitrary constraint R in I, we know that it is satisfied by $a\bar{x}$ for every $a \in \{y \in \mathbb{R} \mid y \geq 1\} \setminus X$ where X is bounded. For every constraint C_i in I, let X_i denote the set of 'exception' points, let $u_i = \sup X_i$, and let $u = \max_{i=1}^{m} u_i$ (where m is the number of constraints in I).

It follows that there is an infinite number of $a > u$ such that a is divisible by n. Clearly $a\bar{x}$ satisfies I. The vector $a\bar{x}$ is integral by construction, which concludes the proof. $\square$

[1] Lemma 6 strengthens the corresponding result in the conference version of this article; instead of requiring that X is finite, we now only require that X is bounded. However, this generalisation *does not* change the constraint language $\Lambda_\mathbb{Z}$. We do not exclude the possibility that, in other cases, there may be differences when using the 'old' definition compared with the 'new' definition.

Intuitively, we are looking for CSPs where any rational solution can be scaled by some factor so that we end up on an integer point. Hence, we use the term *scalability* when referring to the second condition of the lemma. To exemplify the concept of scalability, consider the following three binary relations over $\mathbb{R}^2$: $R_1(x,y) \equiv x \neq 1 \vee y \neq 2$, $R_2(x,y) \equiv x + y = 0$, and $R_3(x,y) \equiv x \leq 2 \vee y \neq 0$. The relations R_1 and R_2 are both scalable. We see that $R_1 = \mathbb{R}^2 \setminus \{(1,2)\}$ and the sets of 'exception' points are consequently always finite. In the case of R_2, scalability follows from the fact that the solutions to a homogeneous linear equation are invariant under multiplication with arbitrary real constants. The relation R_3 is, to the contrary, not scalable since $(1,0) \in R$ but $(a \cdot 1, a \cdot 0) \notin R$ whenever $a > 2$. One may additionally note that every homogeneous equation is scalable.

We continue by showing that whenever one is working with relations in $\mathcal{S}_{\mathbb{R}}$, then condition 1 in Lemma 6 always holds. We need some mathematical preliminaries. Given a real vector $\bar{x} = (x_1, \ldots, x_k)$, let $||\bar{x}||$ denote its Euclidean norm, i.e. $\sqrt{x_1^2 + \ldots + x_k^2}$. Recall that $||\bar{x} + \bar{y}|| \leq ||\bar{x}|| + ||\bar{y}||$ (i.e. the triangle inequality) and $||\alpha\bar{x}|| = |\alpha| \cdot ||\bar{x}||$ (i.e. positive homogeneity) for all real vectors $\bar{x}, \bar{y}$ and arbitrary $\alpha \in \mathbb{R}$. We also give a reminder concerning the solution spaces of linear equations: every solvable linear system $A\bar{x} = \bar{b}$ (where A and $\bar{b}$ are rational) has a rational solution and $\bar{x}$ is a solution if and only if it can be expressed as $\bar{x} = \bar{c} + x_1\bar{v}_1 + \ldots + x_k\bar{v}_k$ where $A\bar{v}_i = \bar{0}$, $A\bar{c} = \bar{b}$, $\bar{c}, \bar{v}_1, \ldots, \bar{v}_k$ are rational vectors, and $x_1, \ldots, x_k$ are real numbers. The existence of a rational solution follows from the fact that such a solution can be obtained by Gaussian elimination, and A and $\bar{b}$ contain rational entries only. Let $\bar{c}$ denote any solution to $A\bar{x} = \bar{b}$. Then, the full set of solutions to $A\bar{x} = \bar{b}$ equals the set $\{\bar{c} + \bar{v} \mid \mathrm{A}\bar{v} = \bar{0}\}$ [29, Th.6 in Ch. 1] Furthermore, the set $\{\bar{v} \mid A\bar{v} = \bar{0}\}$ is a linear subspace of $\mathbb{R}^n$ (known as the *null space*) [29, Th. 12 in Ch. 2]. This subspace has a basis with at most n vectors [29, Th. 12 in Ch. 4], say $\bar{v}_1, \ldots, \bar{v}_k$. By once again exploiting the fact that A is a rational matrix, we see that these vectors can be chosen such that they are rational. This gives us that the set of solutions to $A\bar{x} = \bar{b}$ equals $\{\bar{c} + x_1\bar{v}_1 + \ldots + x_k\bar{v}_k \mid x_1, \ldots, x_k \in \mathbb{R}\}$ where $\bar{c}, \bar{v}_1, \ldots, \bar{v}_k$ are rational vectors.

Theorem 7. *If I is a satisfiable instance of CSP($\mathcal{S}_{\mathbb{R}}$), then I is satisfied by at least one rational point.*

Proof. Let $\bar{r}$ be a satisfying real point. Assume I contains the constraints $\{C_0, \ldots, C_n\}$. We may without loss of generality assume that each C_i is a disjunction $l_{i1} \vee l_{i2} \vee \ldots \vee l_{ik}$: if some constraint is a conjuction $D_1 \wedge \ldots \wedge D_m$, then we may split it into m disjunctions. There is (at least) one l_{ij} from each C_i that is satisfied by $\bar{r}$. Since $a \leq b \equiv a < b \vee a = b$, $a \geq b \equiv a > b \vee a = b$, and $a \neq b \equiv a < b \vee a > b$, we can without loss of generality assume that either $l_{ij} \equiv p(\bar{x}) < c$ or $l_{ij} \equiv p(\bar{x}) = c$. It is clearly sufficient to find a rational satisfying point, $\bar{q}$, that satisfies the formula $l_{0j_0} \wedge \ldots \wedge l_{nj_n}$.

First consider literals of the type $p(\bar{x}) < c$. The sets of satisfying points to them are clearly open. Hence, there is some rational number $\delta > 0$ so that all points $\bar{x}$ for which $||\bar{r} - \bar{x}|| < \delta$ satisfy these literals.

The remaining literals are of the form $p(\bar{x}) = c$ and we can view them as a linear equation system $A\bar{x} = \bar{b}$. We know that every satisfiable system of linear equations has a rational solution and a vector $\bar{x}$ is a solution if and only if it can be expressed as $\bar{x} = \bar{c} + x_1\bar{v}_1 + \ldots + x_k\bar{v}_k$ where $A\bar{v}_i = \bar{0}$, $A\bar{c} = \bar{b}$, $\bar{c}, \bar{v}_1, \ldots, \bar{v}_k$ are rational vectors, and $x_1, \ldots, x_k$ are real numbers. Since $\bar{r}$ satisfies $A\bar{r} = \bar{b}$, it can be expressed as $\bar{r} = \bar{c} + r_1\bar{v}_1 + \ldots r_k\bar{v}_k$. The rational numbers are dense in the real numbers so there are rational numbers q_i satisfying $|r_i - q_i| < \delta_e$ for all i and for any $\delta_e > 0$. Let $\bar{q} = \bar{c} + q_1\bar{v}_1 + \ldots + q_k\bar{v}_k$ and we find that

$$||\bar{r} - \bar{q}|| = ||(r_1 - q_1)\bar{v}_1 + \ldots + (r_k - q_k)\bar{v}_k|| \leq$$

$$|r_1 - q_1| \cdot ||\bar{v}_1|| + \ldots + |r_k - q_k| \cdot ||\bar{v}_k|| < \delta_e \cdot (||\bar{v}_1|| + \ldots + ||\bar{v}_k||).$$

By choosing $\bar{q}$ so that δ_e gets sufficiently small, we can achieve $||\bar{r} - \bar{q}|| < \delta$. It follows that $\bar{q}$ satisfies $l_{0j_0} \wedge l_{1j_1} \wedge \ldots \wedge l_{nj_n}$. □

Thus, $\mathcal{H}_{\mathbb{R}}$ satisfies requirement 1) and 3) of Lemma 6. We let $\Lambda_{\mathbb{Z}} \subseteq \mathcal{H}_{\mathbb{Z}}$ contain the relations that satisfy requirement 2) and have thus proved the following.

Theorem 8. *The problem CSP($\Lambda_{\mathbb{Z}}$) is tractable.*

A description of the relations in $\Lambda_{\mathbb{Z}}$ will be given in the next section.

2.2. Maximality of $\Lambda_{\mathbb{Z}}$

We now verify that $\Lambda_{\mathbb{Z}}$ is maximally tractable in $\mathcal{H}_{\mathbb{Z}}$. To do this, we need the concept of *reduced formula* [4]. Reduced formulas will play an important rôle in Section 3, too.

Definition 9. *Let $\theta(x_1, \ldots, x_n)$ be a formula in conjunctive normal form. We call θ reduced if it is not logically equivalent to any of its subformulas, i.e. there is no formula ψ obtained from θ by deleting literals of clauses such that $\theta(\bar{x}) = \psi(\bar{x})$ for all $\bar{x} \in \mathbb{Z}^n$.*

Consider the formula $\varphi \equiv x+y=1 \wedge (x \neq 2 \vee y \neq 0 \vee x+z \leq 0)$. Assume $(x,y,z) \in \mathbb{R}^3$ satisfies φ. If $x \neq 2$, then the second clause holds for any value of z. If $x = 2$, then y has to be -1 (due to the first clause) and the second clause holds for any value of z once again. Consequently, φ is not reduced since it is logically equivalent to $\varphi' \equiv x+y = 1 \wedge (x \neq 2 \vee y \neq 0)$. This formula is not reduced either, though. If $(x,y) \in \mathbb{R}^2$ satisfies $x+y=1$, then it cannot be the case that $x=2$ and $y=0$. Consequently, $\varphi' \Leftrightarrow x+y=1$ and the formula $x+y=1$ is indeed reduced.

An important property of reduced formulas is that if R is defined by a reduced formula $l_1 \vee \ldots \vee l_n$, then for each l_i, we can find a vector $\bar{x}$ that satisfies l_i but not l_j for all $j \neq i$. To see this, note that if such an $\bar{x}$ does not exist then there exists an l_i such that

$$\forall \bar{x}. l_i(\bar{x}) \rightarrow l_1(\bar{x}) \vee \ldots \vee l_{i-1}(\bar{x}) \vee l_{i+1}(\bar{x}) \vee \ldots \vee l_n(\bar{x})$$

which contradicts that the definition of R is reduced; the subformula l_i can obviously be removed in this case.

Theorem 10. *$\Lambda_{\mathbb{Z}}$ is maximally tractable in $\mathcal{H}_{\mathbb{Z}}$.*

Proof. Let R be an arbitrary relation (of arity n) in $\mathcal{H}_{\mathbb{Z}}$ that does not satisfy requirement 2) of Lemma 6. This implies that there exists a real n-vector $\bar{y}$ and an unbounded set $S \subseteq \mathbb{R}$ such that $\bar{y}$ satisfies R but for every $s \in S$, $s\bar{y}$ does not satisfy R. Assume without loss of generality that $R(\bar{x})$ is defined by a reduced formula $l_1 \vee ... \vee l_k$.

Suppose that some $l_i \equiv p(\bar{x}) \neq c$ where $c \neq 0$. If $p(\bar{y}) \neq c$, then $p(k\bar{y}) \neq c$ for all $k \in \mathbb{R}^+$ except at most one, and the same holds for $R(k\bar{y})$. If $p(\bar{y}) = c$, then $p(k\bar{y}) \neq c$ for all $k \in \mathbb{R}^+$ except at most one, and the same holds for $R(k\bar{y})$. This leads to a contradiction and we can assume that if a literal $l_i \equiv p(\bar{x}) \neq c$, then $c = 0$.

If $\bar{y}$ satisfies some literal $l_i \equiv p(\bar{x}) \neq 0$, then $p(k\bar{y}) \neq 0$ for all $k \in \mathbb{R}$ except at most one, and the same holds for $R(k\bar{y})$. Thus, $\bar{y}$ can only satisfy literals $l_j \equiv p(\bar{x}) r a$ where $r \in \{<, \leq, =, \geq, >\}$. Observe that $p(\bar{x}) < a \Leftrightarrow p(\bar{x}) \leq a-1$; this holds since every coefficient in p is required to be an integer. Hence, we may additionally assume that $r \in \{\leq, =, \geq\}$. Assume without loss of generality that $a \geq 0$; if $a < 0$, then consider the equivalent inequality obtained by multiplying with -1. If $r = (\geq)$, then $k\bar{y}$ satisfies R for all $k \geq 1$. Thus, $r \in \{\leq, =\}$. If $p(\bar{y}) = 0$, then $k\bar{y}$ satisfies R for all $k \in \mathbb{R}$ so we can safely assume that $a > 0$. We conclude that R has one of the following forms:

1. $p(\bar{x}) \ = a \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0$ or

2. $p(\bar{x}) \ \leq \ a \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0$

where $a > 0$. Assume first that R is of type (1). In $\Lambda_{\mathbb{Z}} \cup \{R\}$, we can pp-define the following relation:

$$\begin{aligned} S(z) = & \exists \bar{x}.(p(\bar{x}) = a \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0) \wedge \\ & q_1(\bar{x}) = 0 \wedge \ldots \wedge q_n(\bar{x}) = 0 \wedge p(\bar{x}) = z. \end{aligned}$$

The definition of R is reduced so there exists a vector $\bar{x}$ such that $p(\bar{x}) = a$ and $q_i(\bar{x}) = 0$, $1 \leq i \leq n$. Thus, $S(z)$ holds if and only if $z = a$; in other words, we have defined a positive non-zero constant. This implies that we can pp-define the constant 1 since

$$z = 1 \Leftrightarrow \exists x_1, ..., x_a, y.z = x_1 \wedge S(y) \wedge x_1 \geq 1 \wedge \ldots \wedge x_a \geq 1 \wedge y = x_1 + \ldots + x_a.$$

It is now straightforward to pp-define the relation

$$T_{1,2}(x, y, z) \equiv \exists w.w = 1 \wedge x + y + z - 4w = 0 \wedge x \geq 1 \wedge y \geq 1 \wedge z \geq 1$$

and it follows that $\mathrm{CSP}(\Gamma \cup \{R\})$ is locally NP-hard by Proposition 4.

We now consider the second case, i.e. when R is of type (2). Assume that the coeffecient a is as small as possible, i.e. that the relation $p(\bar{x}) \ \leq \ a \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0$ is not logically equivalent to a relation $p(\bar{x}) \ \leq \ \alpha \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0$ for any $\alpha < a$. In particular, we note that if $\alpha \leq 0$, then the relation would in fact be a member of $\Lambda_{\mathbb{Z}}$.

Analogously to the construction of S, we construct a non-empty unary relation S' that is upper bounded by a as follows:

$$\begin{aligned} S'(z) = & \exists \bar{x}.(p(\bar{x}) \leq a \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0) \wedge \\ & q_1(\bar{x}) = 0 \wedge \ldots \wedge q_n(\bar{x}) = 0 \wedge p(\bar{x}) = z. \end{aligned}$$

Thus, S' contains a largest element b. If $b > 0$, then the constant b can be pp-defined since $z = b \Leftrightarrow S(z) \wedge z \geq b$ and $z \geq b$ is a member of $\Lambda_{\mathbb{Z}}$. In this case, the proof proceedes as in the first part of the proof. Assume instead that $b \leq 0$. Then, by the definition of b,

$$p(\bar{x}) \ \leq \ a \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0$$

is logically equivalent to

$$(p(\bar{x}) \ \leq \ b \wedge q_1(\bar{x}) = 0 \wedge \ldots \wedge q_n(\bar{x}) = 0) \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0$$

which, in turn, is logically equivalent to

$$p(\bar{x}) \ \leq \ b \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0.$$

This leads to a contradiction since $b < a$. $\square$

The maximality proof can be generalised to a hardness result for constraint languages that are not necessarily subsets of $\mathcal{H}_{\mathbb{Z}}$.

Corollary 11. *Let Γ be a constraint language over $\mathbb{Z}$ such that the relations $x = y + z$ and $x \geq 1$ are in $\langle\Gamma\rangle$. Then, $\Gamma \cup \{R\}$ is NP-hard whenever $R \in \mathcal{H}_{\mathbb{Z}} \setminus \Lambda_{\mathbb{Z}}$.*

Proof. By Theorem 3, we may without loss of generality assume that $x = y + z$ and $x \geq 1$ are members of Γ. By inspecting the proof of Theorem 10, we see that the hardness proof requires that we pp-define a finite number (that only depends on the constraint language Γ) of homogeneous equations and, if the relation R is of type (2), the relation $x \geq a$ for some $a \in \mathbb{Z}^+$. We first show that any homogeneous equation can be pp-defined in Γ. Note that we can inductively pp-define the relation $M_k(x, y) \equiv y = kx$ for any $k \in \mathbb{Z}^+$ with

$$M_k(x, y) \equiv \exists y'.M_{k/2}(x, y') \wedge y = y' + y'$$

if k is even and

$$M_k(x, y) \equiv \exists y'.M_{(k-1)/2}(x, y') \wedge y'' = y' + y' \wedge y = y'' + x$$

otherwise. The base case is given by $M_1(x, y) \equiv y = x + 0$, and for negative k we can define $M_k(x, y) \equiv \exists y'.0 = y + y' \wedge M_{-k}(x, y')$

For a given set $a_1, \ldots, a_n$ of integers, we can now pp-define

$$E_m(x_1, \ldots, x_m, y) \equiv \sum_{1 \leq i \leq m} a_i x_i = y$$

by the following inductive construction: for any $1 \leq m \leq n$, let

$$\begin{aligned} E_i(x_1, \ldots, x_i, y) \equiv &\exists z_1, z_2.E_{i-1}(x_1, \ldots, x_{i_1}, z_1) \wedge \\ &M_{a_i}(x_i, z_2) \wedge y = z_1 + z_2 \end{aligned}$$

Clearly, the homogeneous equation $\sum_{1 \leq i \leq m} a_i x_i = 0$ is equivalent to $\exists y.E_n(x_1, \ldots, x_n, y) \wedge y = 0$.

We can also pp-define every relation $x \geq a$ with $a \geq 0$ since

$$x \geq a \Leftrightarrow \exists y_1, \ldots, y_a.x = y_1 + \ldots + y_a \wedge y_1 \geq 1 \wedge \ldots \wedge y_a \geq 1.$$

where the equation $x = y_1 + \ldots + y_a$ is homogeneous. This concludes the proof. $\square$

2.3. A test for scalability in $\mathcal{D}_{\mathbb{R}}$

Since Lemma 6 is applicable to a wide range of constraint languages, it would be desirable to have a method for deciding whether a given relation is scalable or not. A fully general method for this problem is out of the scope of this article, but we will sketch a method for checking whether a given relation in $\mathcal{D}_{\mathbb{R}}$ is scalable or not.

Arbitrarily choose a relation R in $\mathcal{D}_{\mathbb{R}}$. The relation R can be written as a disjuction of simpler terms, i.e. $R \equiv l_1 \vee ... \vee l_k$ where l_i, $1 \leq i \leq k$, is of the form $p(\bar{x})rc$ where p is a linear polynomial, $r \in \{<, \leq, =, \neq, \geq, >\}$, and c is an integer. In this section, we will change our representation slightly by repeatedly doing the following.

1. rewrite $p(\bar{x}) \leq a$ as $p(\bar{x}) < a \vee p(\bar{x}) = a$,
2. rewrite $p(\bar{x}) \geq a$ as $p(\bar{x}) > a \vee p(\bar{x}) = a$,
3. rewrite $p(\bar{x}) < 0$ as $-p(\bar{x}) > 0$,
4. rewrite $p(\bar{x}) < a$ as $p(\bar{x}) \leq 0 \vee [0 < p(\bar{x}) < a]$ when $a > 0$,
5. rewrite $p(\bar{x}) \neq a$ as $p(\bar{x}) \leq 0 \vee [0 < p(\bar{x}) < a] \vee p(\bar{x}) > a$ when $a > 0$,
6. rewrite $p(\bar{x}) \neq 0$ as $p(\bar{x}) < 0 \vee -p(\bar{x}) < 0$, and
7. rewrite $p(\bar{x}) \neq a$ as $-p(\bar{x}) \neq -a$ when $a < 0$.

The resulting definition of R will only consist of the following three kinds of terms: $p(\bar{x}) = b$, $p(\bar{x}) > a$ and the 'special' term $[0 < p(\bar{x}) < a]$ where $a, b \in \mathbb{Z}$ and $a \geq 0$. The special term is introduced since it simplifies the forthcoming presentation. We will now decompose $R(\bar{x})$ into its terms and then group these terms into 'good' terms and 'bad' terms.

If we consider each type of possible term, we see that the following terms

- $p(\bar{x}) = 0$, and
- $p(\bar{x}) > a$ with $a \geq 0$

are scalable. Let $R_g(\bar{x})$ be the disjunction of all terms of these types that occur in $R(\bar{x})$. Scalability is on the other hand not satisfied by the terms

- $p(\bar{x}) = a$ with $a \neq 0$, and
- $[0 < p(\bar{x}) < a]$ with $a > 0$.

Let $R_b(\bar{x})$ be the disjunction of the terms of these forms in $R(\bar{x})$. We now have a unique decomposition $R(\bar{x}) \equiv R_g(\bar{x}) \vee R_b(\bar{x})$. We note that if

$\bar{x} \in R_b$ then there is some $K \in \mathbb{Z}^+$ such that $l\bar{x} \notin R_b$ for any $l \geq K$, but if $\bar{x} \in R_g$ then $k\bar{x} \in R_g$ for all $k \in \mathbb{Z}^+ \setminus S$ where S is a bounded set.

We now assume that R is scalable and let $\bar{x}$ be a satisfying point. Clearly either $R_g(\bar{x})$ or $R_b(\bar{x}) \wedge \neg R_g(\bar{x})$ hold.

If $x \in R_g$ then we know from the above that $k\bar{x} \in R_g \subseteq R$ for all but a bounded set of values of k. Assume instead that $R_b(\bar{x}) \wedge \neg R_g(\bar{x})$ holds. Since R is scalable, it follows that $R_b(k\bar{x}) \vee R_g(k\bar{x})$ must hold for all $k \geq K$ for some integer K, but we know by construction that $R_b(k\bar{x})$ will be false for large enough values of k so it must hold that $k\bar{x} \in R_g$. Hence, R is scalable if there exists a K such that

$$\forall k \geq K.(R_b(\bar{x}) \wedge \neg R_g(\bar{x})) \Rightarrow R_g(k\bar{x})$$

We see that if $\neg R_g(\bar{x}) \wedge R_g(k\bar{x})$, then clearly no term in R_g is satisfied by $\bar{x}$ but at least one of them is satisfied by $k\bar{x}$ for sufficiently large k. By considering the types of terms that may appear in R_g, we find that the only terms for which this can happen is $p(\bar{x}) > a$ when $a \geq 0$. Consider the term $p(\bar{x}) = 0$. If $p(\bar{x}) \neq 0$ and there exists a $k \in \mathbb{Z}^+$ such that $p(k\bar{x}) = 0$, then linearity gives that $p(\bar{x}) = 0/k = 0$ which leads to a contradiction. Hence, consider the term $p(\bar{x}) > a$ instead. If $p(k\bar{x}) > a$, then it follows by linearity that $p(\bar{x}) > a/k$. If this is to hold for all but a finite number of k, then we conclude that $a = 0$ and the term is $p(\bar{x}) > 0$.

We conclude that we can verify whether a relation satisfies the scalability condition or not by checking whether $\neg R_g(\bar{x}) \wedge R_b(\bar{x})$ imply $p_1(\bar{x}) > 0 \vee p_2(\bar{x}) > 0 \vee \cdots \vee p_k(\bar{x}) > 0$ where the polynomials $p_i(\bar{x})$ are the left hand sides from the clauses of the type $p_i(\bar{x}) > a_i$ in $R_g(\bar{x})$.

Example 12. *As a simple example, consider a relation R that excludes a rectangle from R^2, that is,*

$$R(x, y) \equiv x < l_x \vee x > u_x \vee y < l_y \vee y > u_y$$

where $l_x \leq u_x$ and $l_y \leq u_y$. Assume without loss of generality that l_x, l_y, u_x, u_y are all positive. We begin by decomposing relation R:

$$\begin{aligned}
R(\bar{x}) &\equiv R_g(\bar{x}) \vee R_b(\bar{x}) \text{ where} \\
R_g(\bar{x}) &\equiv x > u_x \vee y > u_y \vee x < 0 \vee x = 0 \vee y < 0 \vee y = 0 \text{ and} \\
R_b(\bar{x}) &\equiv [0 < x < l_x] \vee [0 < y < l_y]
\end{aligned}$$

Next, we compute $\neg R_g(\bar{x}) \wedge R_b(\bar{x})$ and get $(x \leq u_x \wedge y \leq u_y \wedge x \geq 0 \wedge x \neq 0 \wedge y \geq 0 \wedge y \neq 0) \wedge ([0 < x < l_x] \vee [0 < y < l_y])$. Finally, we check if this relation implies $x > 0 \vee y > 0$—this is of course true in this case. We conclude that R satisfies the scalability condition.

So what is the complexity of checking whether a given relation R in $\mathcal{D}_{\mathbb{R}}$ is scalable or not? First note that decomposing it into good and bad parts can be done in polynomial time. Hence, we may assume that $R(\bar{x}) \equiv R_g(\bar{x}) \vee R_b(\bar{x})$ where $R_g(\bar{x}) \equiv \bigvee_{i=1}^{n} g_i(\bar{x})$ and $R_b(\bar{x}) \equiv \bigvee_{i=1}^{m} b_i(\bar{x})$. We now rewrite $R_b(\bar{x}) \wedge \neg R_g(\bar{x})$ as

$$\bigvee_{i=1}^{m} b_i(\bar{x}) \wedge \neg g_1(\bar{x}) \wedge \cdots \wedge \neg g_n(\bar{x}).$$

Rewriting the formula in this way obviously takes polynomial time, too.

Let $g_1(\bar{x}), \ldots, g_t(\bar{x})$, $t \leq n$, be the good terms that are inequalities, i.e. $g_i(\bar{x}) \equiv p_i(\bar{x}) > a_i$. For each i, $1 \leq i \leq m$, we want to check if $(b_i(\bar{x}) \wedge \neg g_1(\bar{x}) \wedge \cdots \wedge \neg g_n(\bar{x}))$ implies $p_1(\bar{x}) \geq 0 \vee \cdots \vee p_t(\bar{x}) \geq 0$. This is equivalent with testing if

$$(b_i(\bar{x}) \wedge \neg g_1(\bar{x}) \wedge \cdots \wedge \neg g_n(\bar{x})) \wedge p_1(\bar{x}) < 0 \wedge \cdots \wedge p_t(\bar{x}) < 0$$

is not satisfiable. It is not hard to see that this is an instance of CSP($\mathcal{H}_{\mathbb{R}}$): merely note that each term in this conjunction is a of the form $p(\bar{x})rc$ where p is a polynomial of degree one, c is an integer and $r \in \{<, \leq, =, \geq, >, \neq\}$.

Hence, testing the scalability condition can be done in polynomial time since CSP($\mathcal{H}_{\mathbb{R}}$) is a polynomial-time solvable problem.

3. General linear equations

In the previous section, we found a large maximally tractable subset $\Lambda_{\mathbb{Z}}$ of $\mathcal{H}_{\mathbb{Z}}$. Clearly, $\Lambda_{\mathbb{Z}}$ does not contain any linear equations $p(\bar{x}) = a$ with $a \neq 0$. We will now consider fragments of $\mathcal{H}_{\mathbb{Z}}$ that contain such equations. Similar fragments have been considered before: it is known that finding integer solutions to linear equation systems is a tractable problem [24], and other related problems have been discussed in [7]. We will now work 'backwards' compared to the previous section; instead of starting with $\mathcal{H}_{\mathbb{Z}}$ and removing relations, we will extend the set of linear equations.

The algorithmic part will use results from Cohen et al. [12] and, in particular, exploit a property known as 1-*independence*. We note that the original definitions by Cohen et al. are slightly more general than those presented here; they do not restrict themselves to constraint languages. By the notation $\mathrm{CSP}_{\Delta \leq k}(\Gamma \cup \Delta)$, we mean the CSP problem with constraints over $\Gamma \cup \Delta$ but where the number of constraints over Δ is less than or equal to k.

Definition 13. *For two constraint languages* Γ *and* Δ*, we say that* Δ *is* k*-*independent with respect to Γ *if the following condition holds: any instance* I *of CSP(*$\Gamma \cup \Delta$*) has a solution provided every subinstance of* I *belonging to* $CSP_{\Delta \leq k}(\Gamma \cup \Delta)$ *has a solution.*

1-independence gives us a way to handle disjunctions efficiently. For constraint languages Γ and Δ, let the constraint language $\Gamma \overset{\times}{\vee} \Delta^*$ contain all relations $R(\bar{x}) \equiv c(\bar{x}) \vee d_1(\bar{x}) \vee \ldots \vee d_n(\bar{x})$, $n \geq 0$, where $c(\bar{x})$ is a constraint over Γ and $d_1(\bar{x}), \ldots, d_n(\bar{x})$ are constraints over Δ. Cohen et al. have proved the following result.

Theorem 14. *Let* Γ *and* Δ *be constraint languages. If* $CSP_{\Delta \leq 1}(\Gamma \cup \Delta)$ *is globally tractable and* Δ *is 1-independent with respect to* Γ*, then CSP(*$\Gamma \overset{\times}{\vee} \Delta^*$*) is globally tractable.*

Let $\Gamma \subseteq \mathcal{H}_{\mathbb{Z}}$ denote all relations $p(\bar{x}) = b$ and $\Delta \subseteq \mathcal{H}_{\mathbb{Z}}$ denote all relations $p(\bar{x}) \neq b$. We will now prove that CSP($\Gamma \overset{\times}{\vee} \Delta^*$) is globally tractable (Theorem 15) and that it is a maximal tractable fragment of $\mathcal{H}_{\mathbb{Z}}$ (Theorem 16). We will also extend the maximality result in a way similar to Corollary 11; this result can be found in Corollary 17.

Theorem 15. *CSP(*$\Gamma \overset{\times}{\vee} \Delta^*$*) is globally tractable.*

Proof. We first prove that Δ is 1-independent with respect to Γ. Let I_Γ be an instance of CSP(Γ) and I_Δ an instance of CSP(Δ). Assume that $I_\Gamma \cup \{d_i\}$ is satisfiable for each $d_i \in I_\Delta$ with $d_i \equiv p_i(\bar{x}) \neq c_i$.

We will perform an induction on the size of I_Δ. If $|I_\Delta| = 1$, then satisfiability follows from the assumptions. Assume that $|I_\Delta| = d$, $d > 1$, and that the statement holds for all $I'_\Delta \subset I_\Delta$. We show that $I_\Gamma \cup I_\Delta$ is satisfiable, too.

Let $I^i_\Delta = I_\Delta \setminus \{p_i(\bar{x}) \neq c_i\}$ and consider the instance $I_\Gamma \cup I^i_\Delta$ for each i. Let D_i, $1 \leq i \leq d$, be the set of satisfying points to these subproblems. The sets $D_1, \ldots, D_d$ are non-empty due to the induction hypothesis. Arbitrarily choose an element $\bar{x}_i \in D_i$ for each i. If $\bar{x}_i \in D_j$ for any $j \neq i$, then it is a solution to the entire instance and we are done. We can consequently assume that $p_i(\bar{x}_i) = c_i$ for all i.

Take two points $\bar{x}_1 \in D_1$ and $\bar{x}_2 \in D_2$ and define $\bar{x}^k = k\bar{x}_1 + (1-k)\bar{x}_2$ for $k \in \mathbb{Z}$. Observe that $\bar{x}^k$ satisfies I_Γ for all k. We will now show that there is a k such that $\bar{x}^k \in D_i$ for all i; by the previous comment, it is sufficient to consider the disequations.

For $i = 1$ we note that $p_1(\bar{x}^k) \neq c_1 \Leftrightarrow kp_1(\bar{x}_1) + (1-k)p_1(\bar{x}_2) \neq c_1 \Leftrightarrow (1-k)(p_1(\bar{x}_2) - c_1) \neq 0$ and since $p(\bar{x}_2) \neq c_1$ this is true for all $k \neq 1$. In the same way, we see that $\bar{x}^k \in D_2$ when $k \neq 0$.

For $i \neq 1, 2$, we note that if $p_i(\bar{x}_1) = d_1 \neq c_i$ and $p_i(\bar{x}_2) = d_2 \neq c_i$, then $p_i(\bar{x}^k) = k(d_1 - d_2) + d_2$. If $d_1 = d_2$, then the disequation is always true; otherwise, there is at most one value for k such that $p_i(\bar{x}^k) = c_i$. Hence, each disequation is not satisfied by $\bar{x}^k$ for at most one value of k, and we conclude that there is some k' such that $\bar{x}^{k'} \in D_i$ for all i. It follows that $I_\Gamma \cup I_\Delta$ is satisfiable for any size of I_Δ.

By Theorem 14, it is now sufficient to prove that $\mathrm{CSP}_{\Delta\leq 1}(\Gamma \cup \Delta)$ is tractable. Let I be an instance of $\mathrm{CSP}_{\Delta\leq 1}(\Gamma \cup \Delta)$. We view I as an equation system $A\bar{x} = \bar{b}$ together with a disequation $p(\bar{x}) \neq c$. We start by finding a satisfying integer point $\bar{x}$ to $A\bar{x} = \bar{b}$; this is tractable by [24]. If no such point exists, then I is not satisfiable. If the found solution $\bar{x}$ also satisfies $p(\bar{x}) \neq c$, then we have found a solution to I, too. Otherwise, note that if $\bar{y} \neq \bar{x}$ and $A\bar{y} = \bar{b}$, then $A(\bar{y} - \bar{x}) = \bar{b} - \bar{b} = \bar{0}$. By letting $\bar{x}_h = \bar{y} - \bar{x}$, we see that any satisfying point $\bar{z}$ can be written as $\bar{z} = \bar{x} + \bar{x}_h$ for some $\bar{x}_h$ such that $A\bar{x}_h = \bar{0}$. Since $p(\bar{x}) = c$, we note that $p(\bar{z}) \neq c \Leftrightarrow p(\bar{x}) + p(\bar{x}_h) \neq c \Leftrightarrow p(\bar{x}_h) \neq 0$. From this we conclude that we can find a solution to I if and only if we can find a point $\bar{x}_h$ such that $A\bar{x}_h = \bar{0}$ and $p(\bar{x}_h) \neq 0$.

Now we solve the system $A\bar{x} = \bar{0} \wedge p(\bar{x}) = 1$ over the rational numbers. If this system has no solution, then there is no point $\bar{x}_h$ since some rational multiple of $\bar{x}_h$ would have been a solution. If we find a solution $\bar{x}_q$ to this system, then there exists an integer $k \neq 0$ such that $k\bar{x}_q$ is an integer point satisfying $Ak\bar{x}_q = \bar{0}$ and $p(k\bar{x}_q) = k \neq 0$. We see that we can let $\bar{x}_h = k\bar{x}_q$ and conclude that I is satisfiable. As this only requires solving two linear systems, one over the integers and one over the rational numbers, this is a polynomial-time algorithm for solving $\mathrm{CSP}_{\Delta\leq 1}(\Gamma \cup \Delta)$. ☐

Theorem 16. $\Gamma \overset{\times}{\vee} \Delta^*$ *is maximally tractable in* $\mathcal{H}_{\mathbb{Z}}$.

Proof. Arbitrarily choose a relation $R \in \mathcal{H}_{\mathbb{Z}} \setminus (\Gamma \overset{\times}{\vee} \Delta^*)$ such that $R \equiv p(\bar{x}) \leq c \vee \bigvee_{i=1}^{n} (q_i(\bar{x}) \neq a_i)$ and R has arity m. Note that we do not have to consider relations with $<$ separately since those are always equivalent to a relation using $\leq$. We assume without loss of generality that the definition of R is reduced.

We will now show how to pp-define T_{z_0,z_1} for some $z_0 \neq z_1$ in $\mathbb{Z}$. By reasoning as in the proof of Theorem 10, we see that we can pp-define a unary relation $S(z)$ that is a subset of $\{z \in \mathbb{Z} \mid z \leq c\}$ by

$$S(z) \equiv \exists \bar{x}.(z = p(\bar{x})) \wedge (p(\bar{x}) \leq c \vee \bigvee_{i=1}^{n} (q_i(\bar{x}) \neq a_i)) \wedge (\bigwedge_{i=1}^{n} (q_i(\bar{x}) = a_i)).$$

We first prove that $|S| > 1$. The definition of R is reduced so there exists an integral vector $\bar{x}$ such that $p(\bar{x}) \leq c$ and $q_1(\bar{x}) = a_1, \ldots, q_n(\bar{x}) = a_n$. Consequently, $|S| > 0$. If $|S| = 1$, then

$$\left(\bigwedge_{i=1}^{n} q_i(\bar{x}) = a_i \right) \Rightarrow p(\bar{x}) = d \vee p(\bar{x}) > c$$

for some $d \leq c$. Hence,

$$R(\bar{x}) \equiv p(\bar{x}) = d \vee \bigvee_{i=1}^{n} (q_i(\bar{x}) \neq a_i)$$

which leads to a contradiction since $R \notin \Gamma \overset{\times}{\vee} \Delta^*$. We have thus shown that $|S| > 1$.

Let $z_0 = \max\{z \mid S(z)\}$ and $z_1 = \max\{z \mid S(z), z \neq z_0\}$ and recall that both $x = z_0$ and $x = z_1$ are members of Γ. Now,

$$T_{z_0,z_1}(x, y, z) \Leftrightarrow S(x) \wedge S(y) \wedge S(z) \wedge x + y + z = (2z_0 + z_1)$$

so T_{z_0,z_1} is pp-definable in $(\Gamma \overset{\times}{\vee} \Delta^*) \cup \{R\}$ and NP-hardness follows from Theorem 3 and Proposition 4. □

Corollary 17. *Let Γ be a constraint language over $\mathbb{Z}$ such that the relations $x = y + z$ and $x = 1$ are in $\langle \Gamma \rangle$. Then, $\Gamma \cup \{R\}$ is NP-hard whenever $R \in \mathcal{H}_{\mathbb{Z}} \setminus (\Gamma \overset{\times}{\vee} \Delta^*)$.*

Proof. By Theorem 3, we may without loss of generality assume that $x = y+z$ and $x = 1$ are members of Γ. By inspecting the proof of Theorem 16, we see that the hardness proof requires that we pp-define a finite number (that only dependes on the constraint language Γ) of homogeneous equations and relations $x = a$ where $a \in \mathbb{Z}$.

By the proof of Corollary 11, we know that we can pp-define every homogeneous linear equation in Γ by using the relation $x = y + z$. We can also pp-define $x = a$ for any integer a since $x = a \Leftrightarrow \exists y.y = 1 \wedge x = y + y + \cdots + y$ where the sum contains a terms. Similarly, if a is negative, then $x = a \Leftrightarrow \exists y, z.y + y = y \wedge z = a \wedge x + z = y$. □

4. Binary linear equations

As a third fragment we will consider the language of binary equations combined with unary inequalities and unary disequations. We will refer to this language as Ψ. The language Ψ is a strict extension of the language studied by Bodirsky et al. [9] since it does not allow constraints of the type $x \neq c$.

Definition 18. *Let Ψ be the smallest constraint language containing all binary equations, unary inequalities and unary disequations,* ie. $\{(x, y) \in \mathbb{Z}^2 \mid ax + by = c\} \in \Psi$*,* $\{x \in \mathbb{Z} \mid x \leq u\} \in \Psi$*,* $\{x \in \mathbb{Z} \mid x \geq l\} \in \Psi$ *and* $\{x \in \mathbb{Z} \mid x \neq l\} \in \Psi$ *for arbitrary $a, b, c, u, l \in \mathbb{Z}$.*

Our goal is once again to verify that Ψ is a maximal tractable subclass of $\mathcal{H}_{\mathbb{Z}}$, and we start by showing that CSP(Ψ) is tractable.

Lemma 19. *The problem CSP(Ψ) is solvable in polynomial time.*

Proof. Let (V, C) be an arbitrary instance of CSP(Ψ). Let $C_2 \subseteq C$ denote the set of binary constraints in C. Construct a graph (V, E) as follows:

$(x, y) \in E$ if and only if $ax + by = c$ in C_2.

First determine the connected components of this graph; they can easily be identified in polynomial time. The subproblems corresponding to the connected components can clearly be solved independently so we assume, without loss of generality, that there is exactly one component.

We now consider the system of equations that contains variables from this component; denote this system $Ax = b$. Bodirsky et al. [9] have shown the following:

- either there is no $\bar{x} \in \mathbb{Z}^n$ such that $A\bar{x} = b$, or
- there are two vectors $\bar{a}, \bar{h} \in \mathbb{Z}^n$ such that $A\bar{x} = b$ (with $\bar{x} \in \mathbb{Z}^n$) if and only if $\bar{x} \in \{\bar{a} + t\bar{h} \mid t \in \mathbb{Z}\}$.

Furthermore, $\bar{a}$ and $\bar{h}$ can be computed in polynomial time given integral A and b. It follows that every variable can be written as $x_i = a_i + th_i$ for any integer t, and this implies that every unary relation on x_i can be viewed as a unary relation on t. It is now easy to compute lower and upper bounds (l, u), $l, u \in \mathbb{Z} \cup \{-\infty, \infty\}$, on t by using the unary inequalities. Assume, for instance, we have the inequality $x_i \leq b$. We know that $x_i = a_i + th_i$ and, consequently, that $t \leq \frac{b-a_i}{h_i}$.

Suppose now that we have derived that t is in the set $\{l, l+1, \ldots, u-1, u\}$ with $l, u \in \mathbb{Z}$. Given a disequality $x_i \neq b$, it can exclude at most one possible value for t since $x_i \neq b \Leftrightarrow t \neq \frac{b-a_i}{h_i}$ when $x_i = a_i + th_i$. Hence, if the given instance contains k disequalities, then we need to keep track of at most k excluded values $\{m_1, \ldots, m_k\}$ and check that $\{l, l+1, \ldots, u-1, u\} \setminus \{m_1, \ldots, m_k\}$ is non-empty. If $l = -\infty$ or $u = \infty$, then we see that we do not even have to do this test since the k disequalitites cannot exclude an infinite number of points. We conclude that CSP(Ψ) is a polynomial-time solvable problem. $\square$

We continue by showing that Ψ is a maximal tractable subclass of $\mathcal{H}_{\mathbb{Z}}$. The somewhat lengthy proof is divided into three parts (Sec. 4.1–4.3) where we consider equations, inequalities, and disjunctive relations, respectively. It is easy to see that all possible cases are covered by these three cases and this gives us the desired result.

4.1. Ternary equations

We now consider equations of arity 3 or higher. We need some basic number theory.

Lemma 20 (Bezout's identity)**.** *Given two integers a, b with $gcd(a, b) = 1$, then for any integer k there exists integers x, y such that $k = ax + by$.*

For a proof see for example Corollary 3.8.1 in [36]. Also remember that $gcd(a, b, c) = gcd(a, gcd(b, c))$ (Lemma 3.2. in [36]). This identity generalizes to three or more variables as well.

Lemma 21. *Given three integers a, b, c with $gcd(a, b, c) = 1$, then for any integer k there exists integers x, y, z such that $k = ax + by + cz$.*

Proof. Since $gcd(a, b, c) = gcd(a, gcd(b, c))$, it follows by Lemma 20 that we can find integers x, w such that $ax + gcd(b, c)w = k$ and then find integers y, z such that $by + cz = gcd(b, c)$. Combining these equations gives us $ax + (by + cz)w = ax + (bw)y + (cw)z = k$. $\square$

It is well-known that the integers x, y, z above can be computed in polynomial time (in the size of k, a, b, and c) by using Euclid's algorithm repeatedly, cf. Lemma 3.2 in [36].

We will now consider the language $\Psi \cup \{R(x, y, z)\}$ where R is defined by a ternary equation. We exhibit a serie of pp-definitions that show that the relation $T_{0,1}$ can be pp-defined in $\Psi \cup \{R(x, y, z)\}$ and, consequently, that CSP($\Psi \cup \{R\}$) is NP-hard by Proposition 4.

Lemma 22. *Let $p(x, y, z) = ax + by + cz$ and $R(x, y, z) \equiv p(x, y, z) = d$ for some $a, b, c, d \in \mathbb{Z}$ with $a \neq 0, b \neq 0$, and $c \neq 0$. If $R \neq \varnothing$, then the problem CSP($\Psi \cup \{R\}$) is NP-hard.*

Proof. If $gcd(a, b, c) = k \neq 1$, then we rewrite the definition of R such that

$$R(x, y, z) \equiv \frac{a}{k} \cdot kx + \frac{b}{k} \cdot ky + \frac{c}{k} \cdot kz = \frac{d}{k} \cdot k$$

where $\frac{a}{k}, \frac{b}{k}, \frac{c}{k}$ are integers. We divide by k to get the following equivalent definition:

$$R(x, y, z) \equiv \frac{a}{k} \cdot x + \frac{b}{k} \cdot y + \frac{c}{k} \cdot z = \frac{d}{k}$$

If k does not divide d (i.e. $\frac{d}{k}$ is not an integer), then $R = \varnothing$ since the left-hand side is integral for every choice of $x, y, z \in \mathbb{Z}$. Otherwise,

$$R(x, y, z) \equiv a'x + b'y + c'z = d'$$

where $a' = \frac{a}{k}, b' = \frac{b}{k}, c' = \frac{c}{k}, d' = \frac{d}{k}$ and $gcd(a', b', c') = 1$. Hence, we can assume that $gcd(a, b, c) = 1$ without loss of generality.

Recall that all relations of the type $x = x' + d_x$ (where $d_x \in \mathbb{Z}$) are in Ψ, and we can therefore pp-define the relation

$$\begin{aligned} R'(x, y, z) \equiv \exists x'y'z'.R(x', y', z') \wedge \\ x' = x + d_x \wedge \\ y' = y + d_y \wedge \\ z' = z + d_z \end{aligned}$$

for any $d_x, d_y, d_z \in \mathbb{Z}$. We see that $R'(x, y, z) \equiv p(x + d_x, y + d_y, z + d_z) = d \Leftrightarrow R'(x, y, z) \equiv p(x, y, z) = d - (ad_x + bd_y + cd_z)$.

It follows from Lemma 21 that we can choose values for d_x, d_y, d_z so that $ad_x + bd_y + cd_z = k$ for any k. In particular, we can choose the values so that $R'(x, y, z) \equiv p(x, y, z) = abc$.

We continue by pp-defining the relation

$$\begin{aligned} Q(x, y, z) \equiv \exists x'y'z'.R'(x', y', z') \wedge \\ ax' = abcx \wedge \\ by' = abcy \wedge \\ cz' = abcz. \end{aligned}$$

We see that $Q(x, y, z) \equiv abcx + abcy + abcz = abc$ so

$$Q(x, y, z) \equiv x + y + z = 1.$$

Since all unary inequalities are in Ψ, we can now pp-define the relation $T_{0,1}$ by the following construction:

$$\begin{aligned}T_{0,1}(x,y,z) \equiv & x \leq 1 \wedge x \geq 0 \wedge \\ & y \leq 1 \wedge y \geq 0 \wedge \\ & z \leq 1 \wedge z \geq 0 \wedge \\ & Q(x,y,z)\end{aligned}$$

It follows from Theorem 3 that CSP($\Psi \cup \{R\}$) is NP-hard since CSP($T_{0,1}$) is NP-hard by Proposition 4. □

It is straightforward to generalise this result to equations of higher arity.

Corollary 23. *Let $p(\bar{x}) = \sum_{i=1}^{n} a_i x_i$ with $a_i \in \mathbb{Z} \setminus \{0\}$ and $n \geq 3$, and let $R(\bar{x}) \equiv p(\bar{x}) = d$ for some $d \in \mathbb{Z}$. If $R \neq \varnothing$, then the problem CSP($\Psi \cup \{R\}$) is NP-hard.*

Proof. The relation R has at least one satisfying point $\bar{d} = (d_1, d_2, \ldots d_n)$ by assumption. The relations $x_i = d_i$, $4 \leq i \leq n$, are in Ψ so we pp-define

$$\begin{aligned}T(x,y,z) = \exists x_4 x_5 \ldots x_n . R(x,y,z,x_4,x_5,\ldots,x_n) \wedge \\ x_4 = d_4 \wedge, x_5 = d_4 \wedge \cdots \wedge x_n = d_n.\end{aligned}$$

The relation T is non-empty by the choice of $\bar{d}$. Furthermore, $T(x,y,z) \equiv p(x,y,z) = d$ where p is a ternary equation. Since T is pp-definable in $\Psi \cup \{R\}$, it follows from Theorem 3 and Lemma 22 that CSP($\Psi \cup \{R\}$) is NP-hard. □

4.2. Binary inequalities

We will now consider relations of the form $ax + by \leq c$. It has been noted by Hochbaum & Naor [20] that NP-hardness of the following problem is a consequence of Theorem C in Lagarias [28].

MONOTONE SYSTEM INTEGER FEASIBILITY
INSTANCE. Integral matrix A such that each row contains at most one entry > 0 and at most one entry < 0, integral vector $\bar{b}$.
QUESTION. Is there an integral vector $\bar{x}$ such that $A\bar{x} \leq \bar{b}$?

This problem will provide the basis for our next hardness result.

Lemma 24. *Let R be a binary relation defined such that $R(x,y) \equiv ax+by \leq c$ with $a, b \in \mathbb{Z} \setminus \{0\}, c \in \mathbb{Z}$. Then, the problem CSP($\Psi \cup \{R\}$) is NP-hard.*

Proof. Assume that we are given $R(x, y) \equiv ax + by \leq c$ as above. If $gcd(a, b) = k \neq 1$, then R can be equivalently defined as

$$R(x, y) \equiv \frac{a}{k} \cdot x + \frac{b}{k} \cdot y \leq \lfloor \frac{c}{k} \rfloor$$

so we assume that $gcd(a, b) = 1$ without loss of generality.

First we show that it is possible to pp-define the relation $R_k(x, y) \equiv ax + by \leq k$ for any $k \in \mathbb{Z}$. Arbitrarily choose $d_x, d_y \in \mathbb{Z}$ and consider the following pp-definition:

$$R'(x, y) = \exists x'y'.R(x', y') \wedge x' = x + d_x \wedge y' = y + d_y.$$

Clearly, $R'(x, y) = ax + by \leq c - ad_x - bd_y$. It follows from Lemma 20 that there are d_x, d_y such that $ad_x + bd_y = c - k$ and R_k can, consequently, be pp-defined for any $k \in \mathbb{Z}$.

We now extend this idea and show that we can pp-define an arbitrary binary inequality $px + qy \leq r$ for any $p, q, r \in \mathbb{Z}$. Let

$$\begin{aligned} Q(x, y) = \exists x'y'.R_{rab}(x', y') \wedge \\ ax' = pabx \wedge \\ by' = qaby \end{aligned}$$

We see that $Q(x, y) \equiv pabx + qaby \leq rab \equiv px + qy \leq r$ and any given binary inequality can be pp-defined in $\Psi \cup \{R\}$. We also note that this pp-definition can be computed in polynomial time in the size of a, b, p, q, r. First note that the integers d_x, d_y (which are used in defining R_{rab}) can be computed in polynomial time in the size of a,b and r; this follows from the fact that they can be computed by Euclid's algorithm. All other constants can be computed by applying elementary arithmetic operations to a, b, p, q, and r. We conclude that the pp-definition can be computed in polynomial time (in the size of a, b, p, q, r).

We will now prove NP-hardness by a polynomial-time reduction from Monotone System Integer Feasibility. Let $(A, \bar{b})$ denote an arbitrary instance of this problem and consider the system $A\bar{x} \leq \bar{b}$. Each 'row' in this system corresponds to a relation of the type $px - qy \leq r$ where $p, q \in \mathbb{Z}^+$ and $r \in \mathbb{Z}$. We have seen that each such inequality can, in polynomial time, be converted into a equivalent pp-definition over $\Psi \cup \{R\}$. We conclude that CSP($\Psi \cup \{R\}$) is NP-hard. $\square$

4.3. Disjunctive relations

We divide the remaining relations into three types. Let $R \in \mathcal{H}_{\mathbb{Z}}$ be a relation of arity k strictly greater than one.

type 1: $R(x_1, \ldots, x_k) \equiv P(x_1, \ldots, x_k) \vee x_i \neq c$ where $P(\bar{x})$ is either $p(\bar{x}) = d$ or $p(\bar{x}) \leq d$ for a linear polynomial p and integer d.

type 2: $R(x_1, \ldots, x_k) \equiv (l_1 \vee \ldots \vee l_m)$ where $l_1 \equiv x_1 \neq c$ and $l_2 \equiv x_2 \neq d$

type 3: $R(x_1, \ldots, x_k) \equiv (l_1 \vee \ldots \vee l_m)$ where $l_1 \equiv p(x_1, \ldots, x_k) \neq c$ and p is a linear polynomial such that at least two coeffecients in p are non-zero.

We first show that every relation in $\mathcal{H}_{\mathbb{Z}} \setminus \Psi$ that is not covered by the previous two sections are covered by relations in the list above. Hence, let $R \in \mathcal{H}_{\mathbb{Z}} \setminus \Psi$ be chosen such that it has arity k and not being covered by the previous two sections. If R is defined by an equation, then this equation must be of arity less than or equal to two (otherwise, it would be covered by the results in Section 4.1). However, this is impossible since this implies that $R \in \Psi$. If R is defined by an inequality, then this inequality must be of arity one (otherwise, it would be covered by the results in Section 4.2). Once again, $R \in \Psi$ which leads to a contradiction. If R is defined by a disequality, then this disequality must have arity strictly greater than one and then R is of type (3). We conclude that R has to be defined by a disjunction.

Assume now that

$$R(\bar{x}) \equiv P(\bar{x}) \vee Q_1(\bar{x}) \vee \ldots \vee Q_m(\bar{x})$$

where $Q_1, \ldots, Q_m$ denote disequality constraints and P is either the constraint *false* or a constraint that is not a disequality. We assume without loss of generality (since $p(\bar{x}) < a \Leftrightarrow p(\bar{x}) \leq a - 1$ when we work over the integers and all coeffecients are integers) that the relation in P is either $(=)$ or $(\leq)$. If there is an $1 \leq i \leq m$ such that two coeffecients in Q_i are non-zero, then R is of type (3). Hence, we may assume that

$$R(\bar{x}) \equiv P(\bar{x}) \vee x_{i_1} \neq c_{i_1} \vee \ldots \vee x_{i_n} \neq c_{i_n}$$

for some set of indices $I = \{i_1, \ldots, i_n\} \subseteq \{1, \ldots, k\}$. If $|I| \geq 2$, then R is of type (2). If $|I| = 1$ and $P(\bar{x}) \equiv$ *false*, then R is a member of Ψ which leads to a contradiction. If $P(\bar{x}) \not\equiv$ *false*, then R is of type (1).

We finally note that a relation may simultaneously have several types but this will not cause any troubles in the following proofs.

4.3.1. Relations of type 1

Define the relation

$$X_{a_1,a_2}(x_1,x_2) \equiv x_1 \neq a_1 \vee x_2 \neq a_2$$

for arbitrary integers a_1, a_2.

Lemma 25. *The problem CSP($\Psi \cup \{X_{a_1,a_2}\}$) is NP-hard for all choices of $a_1, a_2 \in \mathbb{Z}$.*

Proof. For arbitrary integers b_1, b_2, we can pp-define

$$X_{b_1,b_2}(x,y) \equiv x - b_1 = x' - a_1 \wedge y - b_2 = y' - a_2 \wedge X_{a_1,a_2}(x',y').$$

We can therefore pp-define $T'_{0,1,2}$ by

$$\begin{aligned} T'_{0,1,2}(x,y) \equiv & x \geq 0 \wedge x \leq 2 \wedge y \geq 0 \wedge y \leq 2 \wedge \\ & X_{0,0}(x,y) \wedge X_{1,1}(x,y) \wedge X_{2,2}(x,y). \end{aligned}$$

which shows that CSP($\Psi \cup \{X_{a_1,a_2}\}$) is NP-hard by Proposition 4. $\square$

We are now ready to show that CSP($\Psi \cup \{R\}$) is NP-hard whenever $R \in \mathcal{H}_{\mathbb{Z}} \setminus \Psi$ is a relation of type 1.

Lemma 26. *Let R be a relation (of arity $k > 1$) such that its reduced definition is*

$$R(\bar{x}) \equiv p(\bar{x}) rc \vee x_i \neq d$$

where p is a linear polynomial, $r \in \{=, \leq\}$ and $c, d \in \mathbb{Z}$. Then, the problem CSP($\Psi \cup \{R\}$) is NP-hard.

Proof. Since $k > 1$, there is some variable x_j, $j \neq i$, that occurs in $p(\bar{x})$ with a non-zero coeffecient. Assume without loss of generality that $j < i$ and that the coeffecient is positive. We know that $R(\bar{x})$ is not equivalent to $x_i \neq d$ due to its reduced definition, and this implies that there is a point $\bar{d} = (d_1, \ldots, d_k) \in R$ such that $d_i = d$. We now pp-define the following binary relation

$$R'(x,y) \equiv R(d_1, \ldots, d_{j-1}, y, d_{j+1}, \ldots, d_{i-1}, x, d_{i+1}, \ldots, d_k).$$

We see that $R'(x,y) \equiv y \leq a \vee x \neq d$ or $R'(x,y) \equiv y = a \vee x \neq d$ (depending on whether $r = (\leq)$ or $r = (=)$) for some constant $a \in \mathbb{Z}$.

Given the former case, we can pp-define a relation

$$R''(x,y) \equiv R'(x,y) \wedge R'(x,z) \wedge z = 2a - y$$

and we see that

$$\begin{aligned} R''(x,y) &\equiv (y \leq a \vee x \neq d) \wedge (2a - y \leq a \vee x \neq d) \\ &\equiv (y \leq a \vee x \neq d) \wedge (a \leq y \vee x \neq d) \\ &\equiv y = a \vee x \neq d \end{aligned}$$

so we only need to consider the case where we have the equality relation in the definition of R'. We can now pp-define the relation $X_{d,d}(x,y)$ as follows:

$$\begin{aligned} X_{d,d}(x,y) \equiv \exists z, w, w'.0 \leq z \wedge z \leq 1 \wedge \\ w = az \wedge w' = a - w \wedge \\ R''(x,w) \wedge R''(y,w') \end{aligned}$$

Hence, it follows from Lemma 25 that CSP($\Psi \cup \{R\}$) is NP-hard. $\square$

4.3.2. Relations of type 2

Given a relation $R \in \mathcal{H}_{\mathbb{Z}} \setminus \Psi$, we may without loss of generality assume that its arity is strictly greater than one since every unary relation in $\mathcal{H}_{\mathbb{Z}}$ is a member of Ψ. This observation immediately leads us to the following hardness proof for relations of type 2.

Lemma 27. *Arbitrarily choose a relation $R \in \mathcal{H}_{\mathbb{Z}}$ of arity $k > 1$ such that $R \neq \mathbb{Z}^k$. If R is a relation of type 2, then CSP($\Psi \cup \{R\}$) is NP-hard.*

Proof. Assume that $R(x_1, \ldots, x_{k-2}, y, z) \equiv (l_1 \vee \ldots \vee l_m)$ where $l_1 \equiv y \neq c$ and $l_2 \equiv z \neq d$. There is some $\bar{b} = (b_1, b_2, \ldots, b_k) \notin R$ since $R \neq \mathbb{Z}^k$. Now, pp-define a binary relation R' as

$$R'(y,z) \equiv \exists x_1, \ldots, x_{k-2}.R(x_1, \ldots, x_{k-2}, y, z) \wedge x_1 = b_1 \wedge \cdots \wedge x_{k-2} = b_{k-2}.$$

It is not hard to see that $R'(y,z) = \mathbb{Z}^2 \setminus (b_1, b_2)$, i.e. $R'(y,z) \equiv X_{b_1,b_2}(y,z)$, and Lemma 25 implies that CSP($\Psi \cup \{R\}$) is NP-hard. $\square$

4.3.3. Relations of type 3

The hardness proof for relations of type 3 consists of three distinct parts. In the two first parts (which can be found in Lemma 28), we only consider binary relations. In the first part, we show that R falls into one of three classes based on its definition. NP-hardness for the two first classes follows more or less immediately from the NP-hardness result for type 1 relations (Lemma 26). The third class is a bit more difficult to tackle, though, and the second part of Lemma 28 is devoted to proving NP-hardness in this case. In the third and final step (Corollary 29), we generalize the result to relations of higher arity.

Lemma 28. *Arbitrarily choose a relation $R \in \mathcal{H}_{\mathbb{Z}} \setminus \Psi$ of arity 2 such that $R \neq \mathbb{Z}^2$. If R is a relation of type 3, then CSP$(\Psi \cup \{R\})$ is NP-hard.*

Proof. Assume that $R(x,y) \equiv (l_1 \vee \ldots \vee l_k)$ where $l_1 \equiv p(x,y) \neq c$ and the two coeffecients in p are non-zero. We note that we may view R as having the definition

$$R(x,y) \equiv P(x,y) \vee \neg \left(A \begin{pmatrix} x \\ y \end{pmatrix} = \bar{c} \right)$$

for some matrix A, some vector $\bar{c}$, and where $P(x,y)$ is either $p(x,y) = d$ or $p(x,y) \leq d$ for a linear polynomial p and integer d, or $P(x,y) \equiv false$.

It follows from basic linear algebra that the set of solutions S to the linear system $A\binom{x}{y} = \bar{c}$ are either the empty set, a single point, or all points on a line L in $\mathbb{R}^2$. The first case is obviously ruled out since $R \neq \mathbb{Z}^2$.

Let S' denote the set of integer points in S, i.e. $S' = S \cap \mathbb{Z}^2$. If $S' = \varnothing$, then $R = \mathbb{Z}^2$ which is not possible. Hence, S' contains a single point s or all integer points along L. This leaves us with two possibilities:

1. $R = \mathbb{Z}^2 \setminus \{s\}$, or
2. $R = \mathbb{Z}^2 \setminus L'$ where L' is an infinite subset of $L \cap \mathbb{Z}^2$.

The first case appears when S' is a single point and the second case when S' contains the integer points along L. This is easy to see since the term P will add at most one point if it is an equation or $P = false$, and it will add all the points from a half-plane otherwise. If P defines a halfplane H, then $L' \not\subseteq H$ since this would imply that $R = \mathbb{Z}^2$. Thus, L' will be an unbounded and infinite set.

In the first case, we clearly have $R = X_{a,b}$ for some $a, b \in \mathbb{Z}$ and it follows from Lemma 25 that CSP$(\Psi \cup \{R\})$ is NP-hard. In the second case, R can be defined in one of the following ways

(1) $R(x,y) \equiv P(x,y) \vee x \neq b$
(2) $R(x,y) \equiv P(x,y) \vee y \neq d$
(3) $R(x,y) \equiv P(x,y) \vee \neg(\exists t. x = at + b \wedge y = ct + d)$

where $a, b, c, d \in \mathbb{Z}$ and $a, c \neq 0$.

In case (1) and (2), the line $L \cap \mathbb{Z}^2$ is parallel to one of the axes of $\mathbb{Z}^2$. We concentrate on case (1) in the sequel; case (2) is obviously analogous. Assume that for arbitrary $\alpha \in \mathbb{Z}$, $(b, \alpha) \notin R$. This implies that $R(x, y) \equiv x \neq b$ and, consequently, that $R \in \Psi$ which leads to a contradiction. Hence, we may assume that R has the reduced definition $R(x, y) \equiv P(x, y) \vee x \neq b$ (with $P(x, y) \not\equiv false$) and NP-hardness follows from Lemma 26.

In case (3), the line $L \cap \mathbb{Z}^2$ is described in parametric form by the equations $x = at + b$ and $y = ct + d$. Since $a \neq 0$ and $c \neq 0$, the line is not parallel to any of the axes. We present an NP-hardness proof for this case in the final part of the proof.

First, pp-define $T(x, y) \equiv \exists x', y'.R(x', y') \wedge x' = ax + b \wedge y' = cy + d$ and observe that

$$\begin{aligned}
T(x, y) &\equiv \exists x', y'.[P(x', y') \vee \neg(\exists t.x' = at + b \wedge y' = ct + d)] \wedge \\
&\qquad \wedge x' = ax + b \wedge y' = cy + d \\
&\equiv P(ax + b, cy + d) \vee \neg(\exists t.ax + b = at + b \wedge cy + d = ct + d) \\
&\equiv P(ax + b, cy + d) \vee \neg(\exists t.x = t \wedge y = t) \\
&\equiv P(ax + b, cy + d) \vee x \neq y.
\end{aligned}$$

We note, once again, that the term P will add at most one point if it is an equation or $P = false$, and it will add all the points from a halfplane otherwise. We conclude that there is some integer k such that $(k, k) \notin T, (k + 1, k + 1) \notin T$ and $(k + 2, k + 2) \notin T$. We use this fact to finally pp-define

$$T'_{k,k+1,k+2}(x, y) \equiv T(x, y) \wedge x \geq k \wedge x \leq k + 2 \wedge y \geq k \wedge y \leq k + 2.$$

It follows that $\mathrm{CSP}(\Psi \cup \{R\})$ is NP-hard by Proposition 4.. $\square$

Corollary 29. *Arbitrarily choose a relation $R \in \mathcal{H}_{\mathbb{Z}} \setminus \Psi$ of arity $k > 2$ such that $R \neq \mathbb{Z}^k$. If R is a relation of type 3, then CSP($\Psi \cup \{R\}$) is NP-hard.*

Proof. Assume that $R(\bar{x}) \equiv (l_1 \vee \ldots \vee l_k)$ where $l_1 \equiv p(\bar{x}) \neq c$ and at least two coeffecients in p are non-zero. For simplicity, we assume that the variables x_1 and x_2 have non-zero coeffecients in p. We note that there is some $\bar{b} = (b_1, b_2, \ldots, b_n) \notin R$ by assumption. We pp-define a binary relation R' as

$$R'(x_1, x_2) = \exists x_3, x_4, \ldots, x_n.R(x_1, x_2, \ldots, x_n) \wedge x_3 = b_3 \wedge \cdots \wedge x_n = b_n.$$

We see that $R' \neq \mathbb{Z}^2$ since $(b_1, b_2) \notin R'$. We also see that R' is of type 3; note that the literal l_1 in the definition of R has been transformed into $\alpha x + \beta y \neq \gamma$ (for integers α, β, γ where α, β are non-zero) in the definition of R'. Lemma 28 implies that CSP$(\Psi \cup \{R\})$ is NP-hard. $\square$

5. Constraints that are k-valid

We will now demonstrate that there are an infinite number of distinct maximally tractable fragments within $\mathcal{H}_{\mathbb{Z}}$. This fact makes complexity classifications harder since a description of the tractable cases must be more elaborate than just listing the maximally tractable fragments.

A relation R is said to be *k-valid* (for some $k \in \mathbb{Z}$) if $(k, \ldots, k) \in R$. A constraint language Γ is k-valid if every relation in Γ is k-valid. Let Γ_k, $k \in \mathbb{Z}$, denote the set of k-valid relations in $\mathcal{H}_{\mathbb{Z}}$ together with the empty relation. Clearly, $\Gamma_i \neq \Gamma_j$ whenever $i \neq j$; Γ_i contains the relation $x = i$ but does not contain $x = j$ and vice versa. Solving instances of CSP(Γ_k) is obviously trivial (simply check whether some constraint is based on the empty relation or not) but such classes have to be considered, too, in order to obtain full complexity classifications. The maximality proof for k-valid constraints differs slightly from the proofs in the preceeding sections. There, we managed to construct explicit NP-hard constraint languages. This proof is slightly non-constructive since we obtain a sequence of constraint languages and prove that (at least) one of them is NP-hard. However, we do not know which one.

Theorem 30. *For each $k \in \mathbb{Z}$, Γ_k is a maximal tractable language in $\mathcal{H}_{\mathbb{Z}}$.*

Proof. The problem CSP(Γ_k) is obviously globally tractable. To prove maximality, arbitrarily choose a relation $R \in \mathcal{H}_{\mathbb{Z}}$ that is not k-valid. Let m denote the arity of R and consider the following relations:

$$\begin{aligned}
U_1(z) &\equiv \exists y, x_2, \ldots, x_m . R(z, x_2, x_3, \ldots, x_m) \wedge y = k \\
U_2(z) &\equiv \exists y, x_3, \ldots, x_m . R(y, z, x_3, x_4, \ldots, x_m) \wedge y = k \\
&\vdots \\
U_m(z) &\equiv \exists y . R(y, y, y, \ldots, y, z) \wedge y = k \\
U_{m+1}(z) &\equiv \exists y . R(y, y, y, \ldots, y, y) \wedge y = k
\end{aligned}$$

The relations $U_1, \ldots, U_{m+1}$ are pp-definable in $\Gamma_k \cup \{R\}$ since the relation $y = k$ is k-valid. We claim that there exists an index $1 \leq j \leq m$ such that $U_j \neq \varnothing$ and $k \notin U_j$. Since R is not k-valid, it follows that $U_{m+1} = \varnothing$ so there exists a smallest index $2 \leq j' \leq m+1$ such that $U_{j'} = \varnothing$. Let $j = j' - 1$. Clearly, U_j is non-empty and if $k \in U_j$, then $U_{j+1} = U_{j'}$ is non-empty which leads to a contradiction.

We now let $c_a(z) \equiv z = a$ and show that we can pp-define the relation $c_{k'}(z)$ for some $k' \neq k$. Assume without loss of generality that there is some element in U_j that is larger than k; if not, then there is some element in U_j that is smaller than k and the reasoning is symmetric. Let $k' = \min\{x \in U_j \mid x > k\}$ and note that $z = k' \Leftrightarrow U_j(z) \wedge z \geq k \wedge z \leq k'$. The relations $z \geq k$ and $z \leq k'$ are both k-valid so $z = k$ is pp-definable in $\Gamma_k \cup \{R\}$. Using the relation $z = k'$, we conclude the proof by the following pp-definition where we exploit that the relation $(z = w \vee x \neq y)$ is k-valid:

$$\begin{aligned} T'_{k-1,k,k+1}(x,y) \equiv & \exists z, w.(z = w \vee x \neq y) \wedge c_k(z) \wedge c_{k'}(w) \wedge \\ & k-1 \leq x \wedge x \leq k+1 \wedge k-1 \leq y \wedge y \leq k+1. \end{aligned}$$

NP-hardness of CSP($\{T'_{k-1,k,k+1}\}$) implies NP-hardness of CSP($\Gamma_k \cup \{R\}$) via Theorem 3 and Proposition 4. □

6. Discussion

The results reported in this paper constitute a step towards a better understanding of the complexity of temporal reasoning in discrete time structures. Below, we discuss several different ways of continuing this work.

6.1. The complexity of Horn DLRs

Completely classifying the complexity of CSP($\mathcal{H}_{\mathbb{Z}}$) appears to be possible with current techniques but it is by no means a trivial task. Consider the NP-complete integer feasibility problem: given a system of inequalities $Ax \geq b$, decide whether there exists a satisfying integer vector x or not. Note that each row $\alpha_1 x_1 + \ldots + \alpha_n x_n \geq \beta$ can be viewed as a relation in $\mathcal{H}_{\mathbb{Z}}$. Thus, a complete classification of CSP($\mathcal{H}_{\mathbb{Z}}$) would give us a classification of the integer feasibility problem (parameterised by allowed row vectors). Such a classification is not currently known and, in fact, there are no classifications even if we restrict ourselves to finite domains or if we consider the closely related integer optimisation problem.

One obvious difficulty when classifying CSP($\mathcal{H}_{\mathbb{Z}}$) is that we do not know what algorithmic techniques will be needed. The results in this paper are, to a large extent, based on either solving linear equations or solving linear programming problems over the real numbers. Completely different methods may be needed in other cases, though.

Another difficulty is that there are tractable cases where we have not been able to prove maximality. One example is the following: for arbitrary $a, b \in \{0,1\}$ and $c \in \mathbb{Z}$, we let $T^{=}_{a,b,c} = \{(x,y) \in \mathbb{R}^+ \mid ax - by = c\}$, $T^{\leq}_{a,b,c} = \{(x,y) \in \mathbb{R}^+ \mid ax - by \leq c\}$, and $\Sigma_{\mathbb{R}} = \{T^{=}_{a,b,c}, T^{\leq}_{a,b,c} \mid a,b \in \{0,1\}, c \in \mathbb{Z}\}$. Define $\Sigma_{\mathcal{Z}}$ analogously over the integers. Note that CSP($\Sigma_{\mathcal{Z}}$) is not the same problem as MONOTONE SYSTEM INTEGER FEASIBILITY since the coefficients a, b are restricted to be members of $\{0,1\}$. Now, consider the following result:

Proposition 31. *CSP($\Sigma_{\mathbb{Z}}$) is a globally tractable problem.*

Proof. Given an instance I of CSP($\Sigma_{\mathbb{R}}$), we see that I equivalently can be viewed as a linear feasibility problem $Ax \leq b$; merely note that each constraint $T^{=}_{a,b,c}(x,y)$ can be replaced by the two constraint $T^{\leq}_{a,b,c}, T^{\leq}_{1-a,1-b,-c}$. Obviously, CSP($\Sigma_{\mathbb{R}}$) can be solved in polynomial time. By inspecting the matrix A, we see that A only contains entries from the set $\{0, \pm 1\}$, each row contains at most two non-zero entries, and if a row contains two non-zero entries, then they have opposite signs. This implies that A is *totally unimodular* (TUM)[2]; observe that A is TUM if and only its transpose is TUM and apply Theorem 19.3(iv) in [38]. Hence, $Ax \leq b$ has a solution if and only if it has an integral solution (see Theorem 19.1 in Schrijver [38]). We have thus shown that CSP($\Sigma_{\mathbb{Z}}$) is tractable. □

One may note that $\Sigma_{\mathbb{Z}}$ is closely connected to the *simple temporal problem* (STP) first described by Dechter et al. [14]. The STP can be defined as follows: let S_{ab}, $a, b \in \mathbb{Z}$, denote the relation $\{(x,y) \in \mathbb{R}^2 \mid a \leq x - y \leq b\}$ and define $\text{STP}_{\mathbb{R}} = \{S_{a,b} \mid a,b \in \mathbb{Z}\}$. Now, the simple temporal problem equals CSP($\text{STP}_{\mathbb{R}}$) and Dechter *et al.* [14] have proved that this problem is tractable. Let CSP($\text{STP}_{\mathbb{Z}}$) denote the simple temporal problem over the integers. By Proposition 31, we see that CSP($\text{STP}_{\mathbb{Z}}$) is tractable, too; each constraint $a \leq x - y \leq b$ can be viewed as a conjunction of two constraints $x - y \leq b$ and $y - x \leq -a$.

6.2. Local and global tractability

During our study of $\mathcal{H}_{\mathbb{Z}}$, we have not encountered any globally NP-hard language that are locally tractable. This is fortunate since such lan-

[2] An integer matrix A is TUM if $\det(B) \in \{0, \pm 1\}$ for every square submatrix B.

guages are problematic: their existence indicates that the number of maximally tractable sublanguages is infinite and that they potentially form an intricate structure. To see this, let $\Gamma = \{R_1, R_2, \ldots\}$ be a constraint language that is globally NP-hard and locally tractable. Consider the languages $\Theta_i = \{R_1, R_2, \ldots, R_i\}$ and note that CSP(Θ_i) is tractable for every $i \geq 1$. Arbitrarily choose a Θ_p, $p \geq 1$, such that Θ_p is included in at least one maximal tractable fragment of Γ. Assume (with the aim of reaching a contradiction) that the set of maximal tractable fragments $\mathbb{X}$ that contains Θ_p is finite, i.e. $\mathbb{X} = \{X_0, \ldots, X_k\}$ for some $k \geq 0$. For $i > 0$, define

$$\varphi(i) = \{j \in \mathbb{N} \mid \Theta_j \subseteq X_i\}.$$

Each set $\varphi(i)$ contains at least one element (namely p) since $\mathbb{X}$ contains every maximal tractable fragment that contains Θ_p. Suppose that the set $\varphi(i)$ is unbounded. This implies that

$$\bigcup_{s \in \varphi(i)} \Theta_s = \Gamma$$

due to the the choice of Θ_i. Hence, $X_i = \Gamma$ which contradicts the fact that CSP(X_i) is tractable. It follows that $\max(\varphi(i))$ is a well-defined natural number for every $i \geq 0$. Let $t = \max \bigcup_{i=1}^{k} \varphi(i)$ and note that t is a natural number such that $t \geq p$. Observe that Θ_{t+1} is not included in any set in $\mathbb{X}$, and recall that $\mathbb{X}$ contains all maximally tractable sets that contain Θ_p. This leads to a contradiction since $\Theta_p \subseteq \Theta_{t+1}$ and CSP(Θ_{t+1}) is a tractable problem. Since $\mathbb{X}$ is an infinite set, the full set of maximally tractable languages is an infinite set, too.

It is folklore within the CSP community that globally NP-hard languages that are locally tractable exist when considering infinite-domain CSP, while the existence of such languages in finite-domain CSPs is an important open question. We will now present a concrete and simple example of such a language within the temporal domain. This shows that additional complications are to be expected when studying temporal languages outside $\mathcal{H}_{\mathbb{Z}}$. We want to point out that such languages may exist within $\mathcal{H}_{\mathbb{Z}}$, too; the mere fact that we have not encountered them yet does not exclude their existence.

We consider the *conjunctive closure* (also known as the *weak co-clone*) $\langle \mathcal{H}_{\mathbb{Z}} \rangle_{\nexists}$ of $\mathcal{H}_{\mathbb{Z}}$. Given a constraint language Γ, we define the conjunctive closure $\langle \Gamma \rangle_{\nexists}$ such that $R(x_1, \ldots, x_k) \in \langle \Gamma \rangle_{\nexists}$ if and only if there exist relations $R_1, \ldots, R_n \in \Gamma$ such that $R(x_1, \ldots, x_k)$ is equivalent to a conjunction of applications of the relations $R_1, \ldots, R_n$ to the variable set $\{x_1, \ldots, x_k\}$. One may view this as a pp-definition where one is not allowed to use existential quantification which explains the notation $\langle\cdot\rangle_{\nexists}$. Clearly, $\langle\mathcal{H}_{\mathbb{Z}}\rangle_{\nexists}$ appears to have fairly limited expressive power compared to $\mathcal{S}_{\mathbb{Z}}$.

We now show that there exists a constraint language $\Gamma^{\circ} = \{R_1, R_2, \ldots\} \subseteq \langle\mathcal{H}\rangle_{\nexists}$ that is globally NP-complete but locally tractable. For arbitrarily chosen $a, b, U \in \mathbb{N}$ and $c \in \mathbb{Z}$, we define relations $M_{a,b,c}$ and $M_{a,b,c,U}$ such that

$$M_{a,b,c} = \{(x, y) \in \mathbb{Z}^2 \mid ax - by \leq c \text{ and } 0 \leq x, y\}$$

and

$$M_{a,b,c,U} = \{(x, y) \in \mathbb{Z}^2 \mid ax - by \leq c \text{ and } 0 \leq x, y \leq U\}.$$

We also define a number of constraint languages:

- $\Gamma' = \{M_{a,b,c} \mid a, b \in \mathbb{N},\ c \in \mathbb{Z}\}$;
- $\Gamma'_U = \{M_{a,b,c,U} \mid a, b \in \mathbb{N},\ c \in \mathbb{Z}\}$ where $U \in \mathbb{N}$;
- $\Gamma^{\circ} = \bigcup_{i=0}^{\infty} \Gamma'_i$

We see that Γ', Γ'_U, and Γ° are all subsets of $\langle\mathcal{H}_{\mathbb{Z}}\rangle_{\nexists}$.

Theorem 32. *CSP*(Γ') *is globally NP-hard and CSP*(Γ'_U) *is globally tractable for any* $U \in \mathbb{N}$.

Proof. CSP(Γ') is equivalent to the NP-hard problem Monotone System Integer Feasibility. The tractability result is due to Hochbaum & Naor [20]. ☐

Obviously, we may view CSP(Γ'), CSP(Γ'_U), and CSP(Γ°) as integer program feasibility problems which implies that we can use the following result for bounding solutions. A proof of this result can be found in, for instance, Chapter 13.3 of Papadimitriou & Steiglitz [33].

Theorem 33. *Let A be an integer $n \times m$ matrix and b an m-vector. If the set $X = \{x \in \mathbb{N}^n \mid Ax \leq b\}$ is non-empty, then there is an $(x_1, \ldots, x_n) \in X$ such that $0 \leq x_i \leq (n+m)(ma_1)^{2m+3}(1+a_2)$, $1 \leq i \leq n$, where $a_1 = \max_{i,j}\{|a_{ij}|\}$ and $a_2 = \max_i\{|b_i|\}$.*

We now put the pieces together.

Theorem 34. *CSP*(Γ°) *is globally NP-hard but locally tractable.*

Proof. We first prove that CSP(Γ°) is NP-hard by a polynomial-time reduction from CSP(Γ'). Let $I = (V, \mathbb{N}, C)$ be an arbitrary instance of CSP(Γ'). Note that every constraint $c \in C$ can be rewritten as at most three linear inequalities. Hence, I can equivalently be viewed as the problem of deciding non-emptiness of the set $\{x \in \mathbb{Z}^n \mid Ax \leq b\}$ where $n = |V|$, A is an integral $(n \times 3|C|)$-matrix, and b is an integral $3|C|$-vector. If I has a solution $s : V \to \mathbb{N}$, then $0 \leq s(v) \leq U$ for each variable $v \in V$ and some bound U that can be computed in polynomial time by Theorem 33. Construct an instance $I' = (V, \mathbb{N}, C')$ of CSP(Γ°) as follows: for each constraint $((x, y), M_{a,b,c}) \in C$, add the constraint $((x, y), M_{a,b,c,U})$ to I'. Obviously, I' has a solution if and only if I' has a solution and NP-hardness follows from Theorem 32.

Next, we prove that CSP(Γ°) is polynomial-time solvable whenever $\widehat{\Gamma}$ is a finite subset of Γ°. Let $T = \max\{U \mid M_{a,b,c,U} \in \widehat{\Gamma^\circ}\}$ and note that $\widehat{\Gamma^\circ} \subseteq \Gamma'_T$. Thus, the existence of a solution can be checked in polynomial time by Theorem 32. $\square$

6.3. Semilinear relations and DLRs

If we turn our attention to semilinear relations and DLRs, then we immediately note that they give rise to a much richer class of CSPs than Horn DLRs. The following is an important observation: for every finite constraint language Γ over a finite domain D, there exists a finite set $\Gamma' \subseteq \mathcal{S}_{\mathbb{Z}}$ such that CSP(Γ) and CSP(Γ') are polynomial-time equivalent. This can be proved by using the following construction: given a relation $R \subseteq D^k$ where $D = \{d_1, \ldots, d_m\}$ is finite, define

$$R'(x_1, \ldots, x_k) \equiv (x_1 = d_1 \vee \ldots \vee x_1 = d_m) \wedge \ldots \wedge (x_k = d_1 \vee \ldots \vee x_k = d_m) \wedge$$

$$\wedge \bigwedge_{(t_1,\ldots,t_k) \in D^k, (t_1,\ldots,t_k) \notin R} (x_1 \neq t_1 \vee \ldots \vee x_k \neq t_k)$$

It is now straightforward to see that R' is a semilinear relation and that CSP($\{R\}$) is polynomial-time equivalent to CSP($\{R'\}$). This idea is straightforward to extend to constraint languages, so a complete classification of CSP($\mathcal{S}_{\mathbb{Z}}$) would also constitute a complete classification of finite-domain CSPs. Such a classification has for many years been a major open question within the CSP community [15].

There appear to be other natural links between the complexity of temporal reasoning and the complexity of finite-domain constraint satisfaction.

One example is *distance* constraints, i.e. relations that are first-order definable in $(\mathbb{Z}; succ)$ where $succ$ denotes the successor relation $succ(x, y) \equiv y = x + 1$. Every relation that is first-order definable in $(\mathbb{Z}; succ)$ has a quantifier-free first-order definition in $(\mathbb{Z}; +, 1)$ so every distance constraint is a member of $\mathcal{S}_{\mathbb{Z}}$. The constraint satisfaction problem for distance constraints has been thoroughly studied by Bodirsky et al. [5] and they identify several tractable fragments, but they fail to provide a complete classification. Interestingly, the complexity of distance constraints depends on the complexity of certain finite-domain CSPs (those having a transitive group of automorphisms.)

When studying the complexity of CSP($\mathcal{S}_{\mathbb{Z}}$) and CSP($\mathcal{D}_{\mathbb{Z}}$), one may expect to encounter fundamentally different tractable fragments when compared to $\mathcal{H}_{\mathbb{Z}}$. Consider, for instance, the relations that are first-order definable over $(\mathbb{Q}; <)$. Every finite tractable constraint language has been identified by Bodirsky and Kara [8]. Let Γ be a tractable language of theirs. It is known that the relations in Γ have the scaling property so Lemma 6 is applicable and $\Gamma|_{\mathbb{Z}}$ is tractable, too. Furthermore, the structure $(\mathbb{Q}; <)$ admits quantifier elimination and we can consequently view each relation in Γ as a member of $\mathcal{S}_{\mathbb{Q}}$. Since the languages identified by Bodirsky and Kara are a very diverse family of languages, the same will hold for the tractable languages within $\mathcal{S}_{\mathbb{Z}}$. We also observe that the constraint language $\Gamma|_{\mathbb{Z}}$ is typically not a subset of $\mathcal{D}_{\mathbb{Z}}$. In fact, the simpler structure of $\mathcal{D}_{\mathbb{Z}}$ may very well simplify the classification task. One may, for instance, note that the finite-domain CSP problem for so-called *clausal relations* is completely classified [13]; a clausal constraint is a disjunction of inequalities of the form $x \geq d$ or $x \leq d$. If we instead consider the closely related class of relations that are first-order definable in $(\{0, \ldots, k\}, \leq)$, then there is no corresponding complete classification of complexity.

Acknowledgements

Peter Jonsson is partially supported by the *Swedish Research Council* (VR) under grant 621-2009-4431, and Tomas Lööw is partially supported by the *Swedish National Graduate School in Computer Science* (CUGS).

References

[1] D. Angluin, J. Aspnes, D. Eisenstat, E. Ruppert, The computational power of population protocols, Distributed Computing 20 (2007) 279–304.

[2] F. Barber, Reasoning on interval and point-based disjunctive metric constraints in temporal contexts, Journal of Artificial Intelligence Research 12 (2000) 35–86.

[3] C. Bettini, X. Wang, S. Jajodia, A general framework for time granularity and its application to temporal reasoning, Annals of Mathematics and Artificial Intelligence 22 (1998) 29–58.

[4] M. Bodirsky, H. Chen, M. Pinsker, The reducts of equality up to primitive positive definability, Journal of Symbolic Logic 75 (2010) 1249–1292.

[5] M. Bodirsky, V. Dalmau, B. Martin, M. Pinsker, Distance constraint satisfaction problems, in: Proceedings of 35th International Symposium on Mathematical Foundations of Computer Science (MFCS-2010), 2010, pp. 162–173.

[6] M. Bodirsky, M. Grohe, Non-dichotomies in constraint satisfaction complexity, in: Proceedings of 35th International Colloquium on Automata, Languages and Programming (ICALP-2008), 2008, pp. 184–196.

[7] M. Bodirsky, P. Jonsson, T. von Oertzen, Horn versus full first-order: Complexity dichotomies in algebraic constraint satisfaction, Journal of Logic and Computation (2012). To appear.

[8] M. Bodirsky, J. Kára, The complexity of temporal constraint satisfaction problems, J. ACM 57 (2010).

[9] M. Bodirsky, G. Nordh, T. von Oertzen, Integer programming with 2-variable equations and 1-variable inequalities, Information Processing Letters 109 (2009) 572–575.

[10] A. Boujjani, P. Habermehl, Constrained properties, semilinear systems, and Petri nets, in: Proceddings of the 7th International Conference on Concurrency Theory (CONCUR-96), 1996, pp. 481–497.

[11] L. Chittaro, A. Montanari, Temporal representation and reasoning in artificial intelligence: issues and approaches, Annals of Mathematics and Artificial Intelligence 28 (2000) 47–106.

[12] D. Cohen, P. Jeavons, P. Jonsson, M. Koubarakis, Building tractable disjunctive constraints, J. ACM 47 (2000) 826–853.

[13] N. Creignou, M. Hermann, A. Krokhin, G. Salzer, Complexity of clausal constraints over chains, Theory of Computing Systems 42 (2008) 239–255.

[14] R. Dechter, I. Meiri, J. Pearl, Temporal constraint networks, Artif. Intell. 49 (1991) 61–95.

[15] T. Feder, M.Y. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: a study through datalog and group theory, SIAM J. Comput. 28 (1998) 57–104.

[16] M. Fisher, D. Gabbay, L. Vila (Eds.), Handbook on Temporal Reasoning in Artificial Intelligence, Elsevier, 2005.

[17] V. Ganesh, Decision procedures for bit-vectors, arrays and integers, Ph.D. thesis, Stanford University, 2007.

[18] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.

[19] S. Ginsburg, E. Spanier, Semigroups, Presburger formulas, and languages, Pacific Journal of Mathematics 16 (1966) 285–296.

[20] D.S. Hochbaum, J. Naor, Simple and fast algorithms for linear and integer programs with two variables per inequality, SIAM J. Comput. 23 (1994) 1179–1192.

[21] P. Jeavons, On the algebraic structure of combinatorial problems, Theoret. Comput. Sci. 200 (1998) 185–204.

[22] P. Jonsson, C. Bäckström, A unifying approach to temporal constraint reasoning, Artif. Intell. 102 (1998) 143–155.

[23] P. Jonsson, T. Lööw, Discrete-time temporal reasoning with Horn DLRs, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011), 2011, pp. 931–936.

[24] R. Kannan, A. Bachem, Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix, SIAM J. Comput. 8 (1979) 499–507.

[25] M. Koubarakis, The complexity of query evaluation in indefinite temporal constraint databases, Theoret. Comput. Sci. 171 (1997) 25–60.

[26] M. Koubarakis, Tractable disjunctions of linear constraints: basic results and applications to temporal reasoning, Theoret. Comput. Sci. 266 (1–2) (2001) 311–339.

[27] A. Krokhin, P. Jeavons, P. Jonsson, Constraint satisfaction problems on intervals and lengths, SIAM Journal on Discrete Mathematics 17 (2004) 453–477.

[28] J.C. Lagarias, The computational complexity of simultaneous diophantine approximation problems, SIAM J. Comput. 14 (1985) 196–209.

[29] D. Lay, Linear Algebra and Its Applications (2nd ed.), Addison Wesley, 1999.

[30] I. Meiri, Combining qualitative and quantitative constraints in temporal reasoning, Artif. Intell. 87 (1996) 343–385.

[31] R. Nieuwenhuis, A. Oliveras, C. Tinelli, Solving SAT and SAT modulo theories: from an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T), J. ACM 53 (6) (2010) 937–977.

[32] C.H. Papadimitriou, On the complexity of integer programming, J. ACM 28 (1981) 765–768.

[33] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, New Jersey, 1982.

[34] R. Parikh, On context-free lanugages, J. ACM 13 (1966) 570–581.

[35] P. Revesz, Introduction to Constraint Databases, Springer 2002.

[36] K. Rosen, Elementary Number Theory, Addison Wesley, 6th edition, 2010.

[37] F. Rossi, P. van Beek, T. Walsh (Eds.), Handbook of Constraint Programming, Elsevier, 2006.

[38] A. Schrijver, Theory of linear and integer programming, John Wiley & Sons, 1986.

[39] R. Wetprasit, A. Sattar, Temporal reasoning with qualitative and quantitative information about points and durations, in: Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), AAAI Press, 1998, pp. 656–663.