GBFS and Presburger arithmetic

Emil Jeřábek (NMAG570 Decidable Theories)

April 30, 2026

1 Graded back-and-forth systems

Definition 1.1 Let $\mathcal{K}$ be a class of $\mathcal{L}$-structures. Put $\mathrm{Tup}_t(\mathcal{K}) = \{\langle \mathbf{A}, \vec{a}\rangle : \mathbf{A} \in \mathcal{K}, \vec{a} \in A^t\}$. A *graded back-and-forth system* (*GBFS*) on $\mathcal{K}$ is a sequence $\{E_k : k \in \mathbb{N}\}$ of binary relations $E_k \subseteq \bigcup_t \big(\mathrm{Tup}_t(\mathcal{K}) \times \mathrm{Tup}_t(\mathcal{K})\big)$ such that

(i) E_k is an equivalence relation;

(ii) $\mathbf{A}, \vec{a}\ E_0\ \mathbf{B}, \vec{b} \implies \mathbf{A}, \vec{a} \equiv_0 \mathbf{B}, \vec{b}$;

(iii) $\mathbf{A}, \vec{a}\ E_{k+1}\ \mathbf{B}, \vec{b} \implies \forall c \in A\ \exists d \in B\ \mathbf{A}, \vec{a}, c\ E_k\ \mathbf{B}, \vec{b}, d$

for all $k, t \in \mathbb{N}$ and $\langle \mathbf{A}, \vec{a}\rangle, \langle \mathbf{B}, \vec{b}\rangle \in \mathrm{Tup}_t(\mathcal{K})$. If $E_k = E$ for all k, then E is a *back-and-forth system* (*BFS*).

A *weak*[1] (*G*)*BFS* satisfies the conditions above with (iii) relaxed to

(iii′) $\mathbf{A}, \vec{a}\ E_{k+1}\ \mathbf{B}, \vec{b} \implies \forall c \in A\ \exists \mathbf{B}' \in \mathcal{K}, \mathbf{B}' \succcurlyeq \mathbf{B}\ \exists d \in B'\ \mathbf{A}, \vec{a}, c\ E_k\ \mathbf{B}', \vec{b}, d.$

In the last talk, we have shown:

Lemma 1.2 *If $\{E_k : k \in \mathbb{N}\}$ is a weak GBFS on $\mathcal{K}$, then $\mathbf{A}, \vec{a}\ E_k\ \mathbf{B}, \vec{b} \implies \mathbf{A}, \vec{a} \equiv_k \mathbf{B}, \vec{b}$ for all $k, t \in \mathbb{N}$ and $\langle \mathbf{A}, \vec{a}\rangle, \langle \mathbf{B}, \vec{b}\rangle \in \mathrm{Tup}_t(\mathcal{K})$.* □

Theorem 1.3 *Let Γ be a set of $\mathcal{L}$-formulas closed under finite disjunctions (incl. empty), and $\{E_k : k \in \mathbb{N}\}$ a weak GBFS on a class $\mathcal{K}$ of $\mathcal{L}$-structures such that for all $k, t \in \mathbb{N}$, E_k has only finitely many equivalence classes on $\mathrm{Tup}_t(\mathcal{K})$, each of which is definable[2] by a formula from Γ.*

Then every formula is equivalent to a formula from Γ over $\mathcal{K}$. □

Let us write $\mathbf{A}, \vec{a} \equiv_\Gamma \mathbf{B}, \vec{b}$ if $\mathbf{A} \vDash \varphi(\vec{a}) \iff \mathbf{B} \vDash \varphi(\vec{b})$ for all $\varphi \in \Gamma$.

Theorem 1.4 *Let Γ be a set of $\mathcal{L}$-formulas closed under Boolean combinations, T an $\mathcal{L}$-theory, and $\{E_k : k \in \mathbb{N}\}$ a weak GBFS on $\mathcal{K} = \mathrm{Mod}(T)$ such that for all $k, t \in \mathbb{N}$ and $\langle \mathbf{A}, \vec{a}\rangle, \langle \mathbf{B}, \vec{b}\rangle \in \mathrm{Tup}_t(\mathcal{K})$, $\mathbf{A}, \vec{a} \equiv_\Gamma \mathbf{B}, \vec{b}$ implies $\mathbf{A}, \vec{a}\ E_k\ \mathbf{B}, \vec{b}$.*

Then every formula is equivalent to a formula from Γ over T. □

[1] This is a nonstandard term.

[2] Meaning that it is of the form $\{\langle \mathbf{A}, \vec{a}\rangle \in \mathrm{Tup}_t(\mathcal{K}) : \mathbf{A} \vDash \varphi(\vec{a})\}$ for some $\varphi \in \Gamma$.

2 Presburger arithmetic

One of the early successes of Hilbert's program (before it was shattered by Gödel) was a complete axiomatization of integer arithmetic with + (but no ·) by M. Presburger [1], who also proved a form of quantifier elimination for the theory, and provided a decision procedure (albeit the notion of an algorithm was not formalized at that point yet).

The precise definition of Presburger arithmetic varies in the literature, depending on whether the domain is taken to be $\mathbb{Z}$ or $\mathbb{N}$, and what symbols are included in the language: the theories of the structures $\langle \mathbb{N}, + \rangle$, $\langle \mathbb{N}, 0, 1, +, \leq \rangle$, $\langle \mathbb{Z}, +, \leq \rangle$, $\langle \mathbb{Z}, 0, 1, +, -, \leq \rangle$ are essentially the same, as 0, 1, and $\leq$ are definable in $\langle \mathbb{N}, + \rangle$, 0, 1, and $-$ are definable in $\langle \mathbb{Z}, +, \leq \rangle$, and the structures based on $\mathbb{N}$ and $\mathbb{Z}$, respectively, are bi-interpretable by the usual construction of negative numbers. One can also consider the unordered structure $\langle \mathbb{Z}, 0, 1, +, - \rangle$, which has weaker expressive power (it cannot define $\leq$). These are all variously called "Presburger arithmetic" in the literature[3]; we will concentrate on the theory of $\langle \mathbb{Z}, 0, 1, +, -, \leq \rangle$, which is often called more specifically the theory of $\mathbb{Z}$-*groups*. Thus, in this section, we fix $\mathcal{L} = \langle 0, 1, +, -, \leq \rangle$.

Presburger arithmetic can be analyzed in various ways, both syntactic and model-theoretic. We will present an argument (or rather, several related arguments) using GBFS, which is one of the most convenient methods.

2.1 Quantifier elimination

As is usual in the theory of abelian groups, we introduce unary terms nx for each constant $n \in \mathbb{Z}$ by $0x = 0$, $nx = \underbrace{x + \cdots + x}_{n}$ for $n > 0$, and $(-n)x = -(nx)$. We also write just n for the constant terms $n1$. Any term in variables $\{x_i : i < t\}$ can be written as an inhomogeneous linear form $\sum_{i<t} n_i x_i + r$ for some $n_i, r \in \mathbb{Z}$, and atomic formulas can be written as integer linear equalities and inequalities $\sum_i n_i x_i = r$, $\sum_i n_i x_i \geq r$. However, Presburger arithmetic does not have quantifier elimination in this language: it can also define congruences

$$x \equiv y \pmod{m}$$

for $m \in \mathbb{N}_{>0}$ by the formulas $\exists z\, y = x + mz$, and these are not equivalent to any quantifier-free formulas. Thus, we will have to add them to the language to get quantifier elimination.

Recall that a propositional formula is *monotone* if it only uses the connectives $\wedge, \vee, \bot, \top$.

Theorem 2.1 *Every formula φ in variables $\{x_i : i < t\}$ is over $\langle \mathbb{Z}, 0, 1, +, -, \leq \rangle$ equivalent to a monotone Boolean combination of formulas of the form*

$$\sum_{i<t} n_i x_i \geq r,$$
$$x_i \equiv a \pmod{m}$$

for some $\vec{n} \in \mathbb{Z}^t$, $r, m, a \in \mathbb{Z}$, $0 \leq a < m$.

[3] In fact, in the original paper, Presburger [1] treats the unordered arithmetic $\langle \mathbb{Z}, 0, 1, + \rangle$, and only mentions the extension of his methods to $\langle \mathbb{Z}, 0, 1, +, < \rangle$ in an end-note.

Proof: We intend to construct a GBFS on $\mathcal{K} = \{\mathbb{Z}\}$ and apply Theorem 1.3. To this end, we should take for $\vec{a}\ E_k\ \vec{b}$ something like "$\vec{a}$ and $\vec{b}$ satisfy the same integer linear inequalities and congruences", except that we also need to ensure that the number of equivalence classes is finite: thus, each E_k will be defined using only *finitely many* linear inequalities with suitably bounded coefficients, and congruences with suitably bounded moduli.

More precisely, we define $\vec{a}\ E_k\ \vec{b}$ (for $\vec{a}, \vec{b} \in \mathbb{Z}^t$) to hold iff

$$a_i \equiv b_i \pmod{M_k} \quad \text{for each } i < t \tag{C_k}$$

and

$$\sum_{i<t} n_i a_i + r \geq 0 \iff \sum_{i<t} n_i b_i + r \geq 0 \quad \text{for all } \vec{n} \in \mathbb{Z}^t,\ r \in \mathbb{Z},\ \text{s.t. } \|\vec{n}\|_1 \leq N_k,\ |r| \leq R_k \tag{L_k}$$

where $\|\vec{n}\|_1 = \sum_i |n_i|$ is the ℓ^1 norm, and

$$M_k, N_k, R_k \in \mathbb{N}_{>0} \tag{0}$$

are certain as-yet unspecified constants. We could pull these constants out of thin air and just define them right away, but it will be more instructive to collect properties that they need to satisfy as we progress through the proof, and fix them at the end when we will already know *why* they need to have a particular value.

It is obvious from the definition that each E_k is an equivalence relation with finitely many classes on t-tuples for any t. Moreover, each equivalence class of E_k is defined by a conjunction of $x_i \equiv c \pmod{M_k}$ for some c, linear inequalities $\sum_i n_i x_i \geq r$, and their negations; but $\sum_i n_i x_i \not\geq r \iff \sum_i (-n_i) x_i \geq 1 - r$ is itself a linear inequality. Thus, it remains to show that E_k satisfies conditions (ii)&(iii) from Definition 1.1.

For (ii), unnested atomic formulas are equivalent to linear inequalities and equalities (each of which can be written as two inequalities) such as $x + y - z = 0$, $x = 1$, $y - x \geq 0$, where the linear coefficients have ℓ^1 norm ≤ 3 and the constant coefficient ≤ 1; thus, we will have $\vec{a}\ E_0\ \vec{b} \implies \vec{a} \equiv_0 \vec{b}$ if

$$N_0 \geq 3. \tag{1}$$

For (iii), assume we are given $\vec{a}\ E_{k+1}\ \vec{b}$ and c; we need to find d such that $\vec{a}, c\ E_k\ \vec{b}, d$. If

$$M_k \mid M_{k+1}, \tag{2}$$

then $\vec{a}\ E_{k+1}\ \vec{b}$ implies $a_i \equiv b_i \pmod{M_k}$, thus condition (C_k) reduces to

$$d \equiv c \pmod{M_k}.$$

Condition (L_k) is an equivalence, but using again that negated inequalities are themselves inequalities with the constant coefficient shifted by 1, it suffices to obtain the implication

$$\sum_{i<t} n_i a_i + r \geq nc \implies \sum_{i<t} n_i b_i + r \geq nd \qquad \text{for } \|\vec{n}\|_1 + |n| \leq N_k,\ |r| \leq R_k + 1.$$

When $n = 0$, the conclusion does not involve d, and follows automatically from $\vec{a}\ E_{k+1}\ \vec{b}$ if

$$N_k \le N_{k+1}, \qquad R_k < R_{k+1}. \tag{3}$$

The constraints with $n > 0$ give upper bounds on d of the form

$$d \le \frac{1}{n}\Big(\sum_{i<t} n_i b_i + r\Big).$$

To satisfy *all* of them, it suffices to pick *one* such constraint with the minimal right-hand side. Likewise, the constraints with $n < 0$ give lower bounds on d, and it suffices to consider only one of them that gives the maximal bound. (If there are only upper bounds, all constraints are satisfied by any sufficiently small d such that $d \equiv c \pmod{M_k}$, and we are done; likewise if there are only lower bounds. Thus, we may assume w.l.o.g. that one lower bound and one upper bound are present.)

All in all, our task reduces to the following (after renaming and negating some of the coefficients for convenience): given $\vec{n}, \vec{m} \in \mathbb{Z}^t$, $n, m \in \mathbb{Z}_{>0}$, and $r, s \in \mathbb{Z}$ such that

$$\begin{aligned} \|\vec{n}\|_1 + n \le N_k, \quad |r| \le R_k + 1, \\ \|\vec{m}\|_1 + m \le N_k, \quad |s| \le R_k + 1, \end{aligned} \qquad \frac{1}{n}\Big(\sum_{i<t} n_i a_i + r\Big) \le c \le \frac{1}{m}\Big(\sum_{i<t} m_i a_i + s\Big),$$

find d such that

$$d \equiv c \pmod{M_k} \qquad \text{and} \qquad \frac{1}{n}\Big(\sum_{i<t} n_i b_i + r\Big) \le d \le \frac{1}{m}\Big(\sum_{i<t} m_i b_i + s\Big).$$

In order to eventually apply the assumption $\vec{a}\ E_{k+1}\ \vec{b}$, we will need to convert these rational inequalities to integer inequalities. Putting $n' = n/\gcd(n,m)$, $m' = m/\gcd(n,m)$, $\ell = \operatorname{lcm}(n,m) = n'm = nm'$, we have

$$\sum_{i<t} m' n_i a_i + m' r \le \ell c \le \sum_{i<t} n' m_i a_i + n' s,$$

and it suffices to find d' such that

$$d' \equiv \ell c \pmod{\ell M_k} \qquad \text{and} \qquad \sum_{i<t} m' n_i b_i + m' r \le d' \le \sum_{i<t} n' m_i b_i + n' s$$

(this ensures that $\ell \mid d'$, and then $d = d'/\ell$ is as required). We distinguish two cases depending on how large is the interval given by the bounds.

Case 1: $\sum_{i<t} m' n_i a_i + m' r + \ell M_k \le \sum_{i<t} n' m_i a_i + n' s$**.** This is a linear inequality with linear coefficients of norm

$$\|m'\vec{n} - n'\vec{m}\|_1 \le m\|\vec{n}\|_1 + n\|\vec{m}\|_1 \le \big(\|\vec{n}\|_1 + n\big)\big(\|\vec{m}\|_1 + m\big) \le N_k^2$$

and constant coefficient of norm

$$|m'r - n's + \ell M_k| \le \big(N_k + (N_k - 1)\big)(R_k + 1) + N_k(N_k - 1)M_k$$

(NB: we cannot have $|n'| = |m'| = N_k$ as $\gcd(n', m') = 1$), thus if

$$N_k^2 \le N_{k+1}, \qquad (2N_k - 1)(R_k + 1) + (N_k^2 - N_k)M_k \le R_{k+1}, \tag{4}$$

then $\vec{a}\ E_{k+1}\ \vec{b}$ implies

$$\sum_{i<t} n_i m' b_i + m' r + \ell M_k \le \sum_{i<t} n' m_i b_i + n' s.$$

That is, our interval for d' has length $\ge \ell M_k$, hence it contains integers in arbitrary residue classes modulo ℓM_k, and specifically some $d' \equiv \ell c \pmod{\ell M_k}$ as required.

Case 2: Otherwise. Using (4) and $\vec{a}\ E_{k+1}\ \vec{b}$, we still have at least

$$\sum_{i<t} m' n_i b_i + m' r \le \sum_{i<t} n' m_i b_i + n' s.$$

Moreover, assuming

$$M_k \operatorname{lcm}\{1, \dots, N_k\} \mid M_{k+1}, \tag{5}$$

we have $\ell M_k \mid M_{k+1}$, i.e., $a_i \equiv b_i \pmod{\ell M_k}$, whence

$$\sum_{i<t} m' n_i a_i + m' r \equiv \sum_{i<t} m' n_i b_i + m' r \pmod{\ell M_k},$$
$$\sum_{i<t} n' m_i a_i + n' s \equiv \sum_{i<t} n' m_i b_i + n' s \pmod{\ell M_k}.$$

A moment's reflection shows that in the cyclic order of residue classes modulo ℓM_k, if we start from $\sum_{i<t} m' n_i a_i + m' r \pmod{\ell M_k}$, we encounter $\ell c \pmod{\ell M_k}$ before $\sum_{i<t} n' m_i a_i + n' s \pmod{\ell M_k}$, therefore the same is true for the $\vec{b}$: if we start from $\sum_{i<t} m' n_i b_i + m' r$, we encounter an integer congruent to $\ell c \pmod{\ell M_k}$ before any integer congruent to $\sum_{i<t} n' m_i b_i + n' s \pmod{\ell M_k}$, and in particular, before we reach $\sum_{i<t} n' m_i b_i + n' s$ itself. Thus, a suitable d' exists.

This finishes the proof, except that we have to observe that there exist constants N_k, M_k, R_k satisfying requirements (0)–(5). We may ignore (2) and (3) as they are subsumed by (4) and (5), respectively. Clearly, the minimal N_k and M_k satisfying the requirements are

$$N_k = 3^{2^k}, \qquad M_k = \prod_{h<k} \operatorname{lcm}\{1, \dots, N_h\} = \prod_{h<k} \operatorname{lcm}\{1, \dots, 3^{2^h}\}.$$

(For a somewhat larger but simpler bound on M_k, we could take its multiple $\operatorname{lcm}\{1, \dots, 3^{2^k}\}$.) The recurrence for R_k we get from (4) is quite messy, and the minimal solution does not have a simple closed form, but it is in any case obvious that there *exists* a solution. □

Remark 2.2 $\log(\operatorname{lcm}\{1,\dots,n\})$ is Chebyshev's function $\psi(n) = \sum_{p^k \le n} \log p$ (where the sum is over prime powers), which has the order of growth $\psi(n) = \Theta(n)$ (the prime number theorem even shows $\psi(n) \sim n$). Thus, our M_k is of magnitude roughly $e^{3^{2^{k-1}}}$.

Since the GBFS we constructed can be useful even outside the proof of Theorem 2.1, let us try to get a reasonable bound on R_k anyway. Since (4) makes R_k larger than M_{k-1}, a natural idea to try is whether we could just take $R_k = M_k$, simplifying the number of parameters. There is a small exception: we have $M_0 = 1$, $M_1 = 6$, $M_2 = 15\,120$, while the minimal solution for R_k has $R_0^{\min} = 1$, $R_1^{\min} = 16$, $R_2^{\min} = 721$. Thus, R_1 has to be larger than M_1, but we can take $R_2 = M_2$, and from that point on it works: observing that for odd $n \ge 5$, we have

$$\operatorname{lcm}\{1,\dots,n\} \ge n(n-1)(n-2) \ge (n^2-3n)n \ge 2n^2,$$

we get for all $k \ge 2$ that $M_k \ge 2N_{k-1}^2 = 2N_k$ and

$$(2N_k - 1)(M_k + 1) + (N_k^2 - N_k)M_k \le (N_k^2 + N_k)M_k \le \operatorname{lcm}\{1,\dots,N_k\}M_k = M_{k+1},$$

i.e., M_k satisfies the recurrence (4) for R_k for $k \ge 2$. Thus, we can take

$$R_k = \begin{cases} 16 & k = 1 \\ M_k & \text{otherwise} \end{cases}$$

in the definition of E_k. In fact, if we use the alternative value $M_k = \operatorname{lcm}\{1,\dots,3^{2^k}\}$ (which makes $M_1 = 2\,520$), we can take $R_k = M_k$ for all k.

2.2 Axiomatization and decidability

The quantifier elimination result in Theorem 2.1 does not yet imply decidability of Presburger arithmetic. It is in fact possible to extract an algorithm from the given GBFS, as we will see in Section 3. But we can also obtain decidability from an explicit recursive axiomatization of the theory, which is something we are interested in for its own sake.

An axiomatization of $\mathrm{Th}(\mathbb{Z}, 0, 1, +, -, \le)$ could be constructed by carefully examining what properties of $\mathbb{Z}$ we used in the proof of Theorem 2.1, but we will just present it directly.

Definition 2.3 $\mathbf{A} = \langle A, 0, +, -, \le\rangle$ is an *ordered abelian group* (*OAG*) if $\langle A, 0, +, -\rangle$ is an abelian group, and $\le$ a linear order[4] on A that satisfies

$$x \le y \to x + z \le y + z.$$

Notice that any OAG is torsion-free.

A *discrete[ly]*[5] *ordered abelian group* (*DOAG*) is an OAG that has a minimal positive element, denoted 1. As a first-order theory, we formulate it in the language $\mathcal{L} = \langle 0, 1, +, -, \le\rangle$; it has the axioms of OAG and $1 \not\le 0$, $\forall x\,(x \le 0 \vee 1 \le x)$.

[4] We generally formulate linear orders in language $\{<\}$ in this course, but for discretely ordered structures, the choice of $\le$ appears more convenient.

[5] The order of a DOAG is indeed discrete as any element x has an immediate successor $x + 1$ and immediate predecessor $x - 1$; in other words, the interval topology is discrete as $(x - 1, x + 1) = \{x\}$ isolates x. There is a dichotomy that every OAG is either discrete or densely ordered.

A $\mathbb{Z}$-*group* is a DOAG $\mathbf{A}$ such that the quotient group $\mathbf{A}/\mathbb{Z}$ is divisible, where we identify $\mathbb{Z}$ with the subgroup generated by 1. This is equivalent to the infinite schema of axioms

$$\forall x\, \exists y \bigvee_{i<m} x = my + i$$

for each $m \in \mathbb{N}_{>0}$ (it suffices to postulate these axioms only for prime m). We denote the theory of $\mathbb{Z}$-groups as ZG.

Discreteness implies that 1 is not divisible by any $m \geq 2$, hence for a given x, the i (and y) supplied by the axiom is *unique*, and it is clearly additive modulo m. That is, if $\mathbf{A}$ is a $\mathbb{Z}$-group and $m \in \mathbb{N}_{>0}$, there is a unique group homomorphism $\mathrm{rem}_m \colon \mathbf{A} \to \mathbb{Z}/m\mathbb{Z}$ such that $\mathrm{rem}_m(1) = 1 + m\mathbb{Z}$; its kernel is $m\mathbf{A}$, thus it provides a canonical isomorphism $\mathbf{A}/m\mathbf{A} \simeq \mathbb{Z}/m\mathbb{Z}$.

The relation $x \equiv y \pmod m$ we introduced earlier is only defined if x, y belong to the same $\mathbb{Z}$-group $\mathbf{A}$ (it just says $x - y \in m\mathbf{A}$). However, it is equivalent to $\mathrm{rem}_m(x) = \mathrm{rem}_m(y)$, and this makes perfect sense even if x and y belong to different groups, hence we can extend the notation to this situation.

Theorem 2.4 *Every formula φ in variables $\{x_i : i < t\}$ is over ZG equivalent to a monotone Boolean combination of formulas of the form*

$$\sum_{i<t} n_i x_i \geq r,$$
$$x_i \equiv a \pmod m$$

for some $\vec{n} \in \mathbb{Z}^t$, $r, m, a \in \mathbb{Z}$, $0 \leq a < m$. Consequently, $\mathsf{ZG} = \mathrm{Th}(\mathbb{Z})$ is complete and decidable.

For $t = 0$, the quantifier elimination result states that every sentence is equivalent to a Boolean combination of sentences $0 \geq r$ for some $r \in \mathbb{Z}$, and these are all decidable in ZG. This implies the completeness of the theory, and its decidability as well, as it is recursively axiomatized. Thus, it suffices to prove quantifier elimination. We sketch two different proofs.

Proof 1: The definition of the GBFS $\{E_k : k \in \mathbb{N}\}$ in the proof of Theorem 2.1 readily extends to the class of all $\mathbb{Z}$-groups: that is, if $\mathbf{A}$ and $\mathbf{B}$ are $\mathbb{Z}$-groups, $\vec{a} \in A^t$, and $\vec{b} \in B^t$, we define $\mathbf{A}, \vec{a}\ E_k\ \mathbf{B}, \vec{b}$ iff

$$\mathrm{rem}_{M_k}(a_i) = \mathrm{rem}_{M_k}(b_i) \quad \text{for each } i < t, \text{ and}$$
$$\sum_{i<t} n_i a_i + r \geq 0 \iff \sum_{i<t} n_i b_i + r \geq 0 \quad \text{for all } \vec{n} \in \mathbb{Z}^t,\ r \in \mathbb{Z}, \text{ s.t. } \|\vec{n}\|_1 \leq N_k,\ |r| \leq R_k.$$

A careful examination of the proof shows that it goes through[6] in this more general setting. □

In the context of Theorem 2.1, we genuinely need the GBFS to be graded, otherwise we do not get anywhere: $E = \bigcap_k E_k$ is just the identity relation, which would not help us prove anything. However, things are different when we work with a class of structures closed under

[6]The usage of rational fractions can be circumvented in one way or another, or we may indeed work with them literally in the divisible hulls $\mathbf{A} \otimes_{\mathbb{Z}} \mathbb{Q}$, $\mathbf{B} \otimes_{\mathbb{Z}} \mathbb{Q}$.)

elementary extensions such as the class of all $\mathbb{Z}$-groups: then ungraded *weak* BFS can be put to good use, exploiting compactness in lieu of finitarity. We will exhibit this method in our second proof of Theorem 2.4. While the overall structure of the argument is similar to the proof of Theorem 2.1, it avoids most of the technicalities as it does not require any bounds on the coefficients. This makes the argument simpler and a bit more abstract, at the expense of being nonconstructive (thus, e.g., it cannot be used to get an explicit algorithm such as in Section 3).

Proof 2: For any $\mathbb{Z}$-groups $\mathbf{A}$ and $\mathbf{B}$, and $\vec{a} \in A^t$, $\vec{b} \in B^t$, we define

$$\mathbf{A}, \vec{a} \mathrel{E} \mathbf{B}, \vec{b} \iff \operatorname{rem}_m(a_i) = \operatorname{rem}_m(b_i) \quad \text{for all } i < t \text{ and } m \in \mathbb{N}_{>0}, \text{ and}$$
$$\sum_{i<t} n_i a_i + r \geq 0 \iff \sum_{i<t} n_i b_i + r \geq 0 \quad \text{for all } \vec{n} \in \mathbb{Z}^t,\ r \in \mathbb{Z}.$$

We claim that E is a weak BFS, which will prove the result[7] by Theorem 1.4.

We need to show that if $\mathbf{A}, \vec{a} \mathrel{E} \mathbf{B}, \vec{b}$ and $c \in A$, there is $\mathbf{B}' \succcurlyeq \mathbf{B}$ and $d \in B'$ such that $\mathbf{A}, \vec{a}, c \mathrel{E} \mathbf{B}', \vec{b}, d$. Elementary extensions of $\mathbf{B}$ can be constructed as models of the elementary diagram $\operatorname{Th}(\mathbf{B}_B)$, hence it suffices to show that the theory

$$T = \operatorname{Th}(\mathbf{B}_B) + \big\{ d \equiv \varrho \pmod{M} : M \in \mathbb{N}_{>0}, \varrho = \operatorname{rem}_M(c) \big\}$$
$$+ \Big\{ \sum_{i<t} n_i b_i + nd \geq 0 : \vec{n} \in \mathbb{Z}^t, n \in \mathbb{Z}, \sum_{i<t} n_i a_i + nc \geq 0 \Big\}$$

(in language $\mathcal{L}_B \cup \{d\}$ with a fresh constant d) is consistent. By compactness, it is enough to show that any finite subtheory of T is consistent, i.e., given finitely many axioms of T, there is $d \in B$ that satisfies them (so now we are back to just $\mathbf{B}$ itself).

By the same manipulation as in the proof of Theorem 2.1, this amounts to the following: if $\vec{n}, \vec{m} \in \mathbb{Z}^t$, $r, s \in \mathbb{Z}$, $n, m, M \in \mathbb{N}_{>0}$ are such that

$$\frac{1}{n}\Big(\sum_{i<t} n_i a_i + r\Big) \leq c \leq \frac{1}{m}\Big(\sum_{i<t} m_i a_i + s\Big),$$

there is $d \in B$ such that

$$d \equiv c \pmod{M} \qquad \text{and} \qquad \frac{1}{n}\Big(\sum_{i<t} n_i b_i + r\Big) \leq d \leq \frac{1}{m}\Big(\sum_{i<t} m_i b_i + s\Big),$$

or equivalently, $d' \in B$ such that

$$d' \equiv nmc \pmod{nmM} \qquad \text{and} \qquad \sum_{i<t} mn_i b_i + mr \leq d' \leq \sum_{i<t} nm_i b_i + ns. \tag{6}$$

Now, if

$$\sum_{i<t} mn_i a_i + mr + nmM \leq \sum_{i<t} nm_i a_i + ns,$$

[7] Strictly speaking, Theorem 1.4 will given that any formula is equivalent to a not necessarily monotone Boolean combination of linear inequalities and congruences $x_i \equiv a \pmod{m}$. However, it is then simple to eliminate negations.

then $\mathbf{A}, \vec{a}\ E\ \mathbf{B}, \vec{b}$ implies

$$\sum_{i<t} mn_i b_i + mr + nmM \leq \sum_{i<t} nm_i b_i + ns,$$

hence a suitable d' exists. Otherwise, we have

$$\begin{aligned}
&\sum_{i<t} mn_i a_i + mr \leq c \leq \sum_{i<t} nm_i a_i + ns < \sum_{i<t} mn_i a_i + mr + nmM,\\
&\sum_{i<t} mn_i b_i + mr \leq \sum_{i<t} nm_i b_i + ns,\\
&\sum_{i<t} mn_i a_i + mr \equiv \sum_{i<t} mn_i b_i + mr \pmod{nmM},\\
&\sum_{i<t} nm_i a_i + ns \equiv \sum_{i<t} nm_i b_i + ns \pmod{nmM},
\end{aligned}$$

hence d' satisfying (6) exists by considering the cyclic order of residue classes modulo nmM as in the proof of Theorem 2.1. □

Exercise 2.5 Using a suitable GBFS or weak BFS, prove that the theory of nontrivial divisible OAG has quantifier elimination (in its basic language) and it is complete and decidable.

Remark 2.6 The theory of *all* OAG is itself decidable. This is a highly nontrivial result of Gurevich.

Dropping the order, the theory of abelian groups is also decidable, and has quantifier elimination in a language with divisibility predicates, as proved originally by Szmielew. This is a fairly accessible result; it can be found in Hodges, *Model theory*, along with the more general Baur–Monk quantifier elimination theorem for R-modules. We might get to some of these results later in the course.

3 Computational complexity

In the beginning of the course, I said that we will not be concerned with computational complexity of algorithms, only with their existence. But we will make an exception in this section, as it turns out that one can construct decision procedures with somewhat reasonable efficiency on the basis of Ehrenfeucht–Fraïssé games/GBFS.

I am not going to define complexity measures very formally, but we need at least a brief overview of the basic terminology and notation.

An algorithm A (formally, a multitape Turing machine) computing a decision problem $D \subseteq \Sigma^*$ (i.e., $D\colon \Sigma^* \to \{\mathrm{YES}, \mathrm{NO}\}$) or a function $F\colon \Sigma^* \to \Sigma^*$ works in *time* $t(n)$, where $t\colon \mathbb{N} \to \mathbb{N}$, if for every input x of length $|x| = n$ (i.e., $x \in \Sigma^n$), A computes the answer after performing at most $t(n)$ basic steps. Likewise, A works in *space* $s(n)$ if for every x of length n, the computation of A on input n only uses $\leq s(n)$ distinct memory units (of bounded size; on a Turing machine, these would be the cells on work tapes). Let $\mathsf{DTIME}(t(n))$ and $\mathsf{DSPACE}(s(n))$ denote the classes of all decision problems that are decidable by an algorithm working in time $t'(n)$ for

some function $t'(n) = O(t(n))$, or in space $s'(n)$ for some function $s'(n) = O(s(n))$, respectively. (The D stands for "deterministic". Asymptotic bounds are used mostly for convenience.) The basic relationships between these classes are

$$\mathsf{DTIME}(t(n)) \subseteq \mathsf{DSPACE}(t(n)), \qquad \mathsf{DSPACE}(s(n)) \subseteq \mathsf{DTIME}\big(2^{O(s(n))}\big). \tag{7}$$

(An algorithm working in time $O(t(n))$ can only use $O(t(n))$ memory cells, as each single-step operation can only write in one cell. An algorithm working in space $O(s(n))$ can only be in $2^{O(s(n))}$ distinct states/configurations based on the content of the memory, and it cannot repeat any configuration twice during the computation, as this would mean that it is stuck in an endless loop.)

A *polynomial-time algorithm* is one that works in time $p(n)$ for some polynomial p with integer coefficients; that is, the class of polynomial-time computable decision problems is

$$\mathsf{P} = \bigcup_{c \in \mathbb{N}} \mathsf{DTIME}(n^c).$$

Other popular classes relevant for us include

$$\mathsf{PSPACE} = \bigcup_{c \in \mathbb{N}} \mathsf{DSPACE}(n^c),$$
$$\mathsf{EXP} = \bigcup_{c \in \mathbb{N}} \mathsf{DTIME}\big(2^{n^c}\big).$$

A decision problem A is polynomial-time reducible to a decision problem B, written $A \leq_p B$, if there exists a polynomial-time computable function f such that

$$x \in A \iff f(x) \in B.$$

If $\mathcal{C}$ is a complexity class (i.e., a set of decision problems), then a problem B is *$\mathcal{C}$-hard* if $A \leq_p B$ for every $A \in \mathcal{C}$, and it is *$\mathcal{C}$-complete* if additionally $B \in \mathcal{C}$.

PSPACE includes the famous class NP (which is defined using nondeterministic algorithms that I will not introduce),

$$\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXP},$$

hence PSPACE-hard problems are already viewed as intractable. However, the most popular PSPACE-complete problem is truth of so-called quantified Boolean sentences, which almost trivially reduces to truth of first-order formulas (in a language with only $=$) referring to two distinct objects; that is:

Fact 3.1 *If T is any theory consistent with $\exists x\, \exists y\, x \neq y$, then T is PSPACE-hard.* □

This means that decision procedures for first-order theories can be efficient only to a rather limited degree. But still, there may be a difference between decision algorithms that are somewhat usable (say, for formulas of bounded quantifier complexity), and those that are completely intractable.

```
1  function Sat_A(φ(x⃗) : formula, a⃗ : A) : {0,1}
2      if φ is atomic: directly evaluate φ(a⃗)
3      if φ is ¬ψ:
4          u ← Sat_A(ψ, a⃗)
5          return ¬u
6      if φ is (ψ ∧ χ):
7          u ← Sat_A(ψ, a⃗)
8          v ← Sat_A(χ, a⃗)
9          return u ∧ v
10     if φ is ∃y ψ(x⃗, y):
11         for each b ∈ A:
12             if Sat_A(ψ, a⃗, b) = 1 then return 1
13         return 0
```

Figure 1: An "algorithm" for satisfaction in $\mathbf{A}$.

Algorithms based on naïve syntactic quantifier elimination, such as we have seen in the first lecture, are firmly in the "completely intractable" category. The reason is that each step of eliminating one quantifier typically starts by converting the quantifier-free part of the formula to DNF and distributing the $\exists$ quantifier over the disjunctions, but this already incurs an exponential blow-up. Thus, if we do this for a formula with n quantifiers, the running time will be at least a tower of exponentials of height $\approx n$. For many theories, algorithms based on model-theoretic methods (variants of GBFS) can do much better.

The starting point is the simple recursive procedure for testing satisfaction in a fixed model $\mathbf{A}$ in Fig. 1. It clearly works, and the recursion depth is at most $n = |\varphi|$, hence it looks fairly space-efficient, except for the "minor" problem that it may not terminate at all because of the unbounded infinite search on line 11.

But we can still do something with it. To begin with, it *does* literally work if $\mathbf{A}$ is finite, in which case it indeed terminates, and uses only space $O(n)$:

Corollary 3.2 *If T has only finite models, then T is decidable in* PSPACE. □

Next, how about a simple infinite structure such as our favourite $\langle \mathbb{Q}, < \rangle$? Well, there is no point in trying *all* elements $b \in \mathbb{Q}$ on line 11, as they will mostly get the same answer: the only thing that matters is the relative order of b w.r.t. the elements of $\vec{a}$, thus if $\vec{a}$ has t elements, it suffices to test $t + 1$ elements b—one in each interval determined by $\vec{a}$. Furthermore, if we start with a sentence of length n, then the recursive subcalls will only ask about subformulas of φ, each of which has $< n$ free variables, hence we only need to go through $\leq n$ elements in each recursive call. We do not need to store them as rationals, but just indicate the ordering relation on them. Thus, the total work space needed is only polynomial in n, i.e., DLO has the lowest possible complexity in view of Fact 3.1:

Example 3.3 The theory DLO is PSPACE-complete. □

```
1  function Sat_A(φ(x⃗) : formula, a⃗ : A) : {0,1}
2      if φ is atomic: directly evaluate φ(a⃗)
3      if φ is ¬ψ:
4          u ← Sat_A(ψ, a⃗)
5          return ¬u
6      if φ is (ψ ∧ χ):
7          u ← Sat_A(ψ, a⃗)
8          v ← Sat_A(χ, a⃗)
9          return u ∧ v
10     if φ is ∃y ψ(x⃗, y):
11         k ← qr(ψ)
12         let {b_i : i < s} be representants of E_k-classes of ⟨a⃗, b⟩
13         for each i < s:
14             if Sat_A(ψ, a⃗, b_i) = 1 then return 1
15         return 0
```

Figure 2: An algorithm for satisfaction in $\mathbf{A}$ with a GBFS $\{E_k : k \in \mathbb{N}\}$.

In other words, what we did was that instead of all $b \in A$, we only took representatives of extensions of $\vec{a}$ to $\vec{a}, b$ up to elementary equivalence, which amounts to equivalence of quantifier-free types due to quantifier elimination.

Exercise 3.4 Prove along similar lines that the theory of random graphs is PSPACE-complete.

Exercise 3.5 If $\mathbf{F}$ is the Fraïssé limit of a finitely axiomatized (relative to finite structures) class $\mathcal{K}$ of finite relational structures, then $\mathrm{Th}(\mathbf{F})$ is decidable in PSPACE.

More generally, if $\{E_k : k \in \mathbb{N}\}$ is a GBFS on $\{\mathbf{A}\}$ such that for each k, t, E_k has only finitely many equivalence classes on t-tuples, we can decide $\mathrm{Th}(\mathbf{A})$ using the modified algorithm in Fig. 2. The exact complexity of this algorithm (or whether it *is* a computable algorithm at all) will depend on how many equivalence classes E_k has, and how efficiently they can be represented and enumerated. But for example, we get:

Proposition 3.6 *Presburger arithmetic is decidable in space* $2^{2^{O(n)}}$ *and time* $2^{2^{2^{O(n)}}}$.

Proof: We use the algorithm from Fig. 2 with the GBFS from the proof of Theorem 2.1. For representants of E_k-classes, we can simply take all b of absolute value up to a suitable bound that we need to estimate.

For a fixed $\vec{a}$, an E_k-equivalence class of tuples $\langle \vec{a}, b\rangle$ is determined by b mod M_k and bounds of the form $b \le z$ or $b \ge z$ where $z = \sum_i q_i a_i + r$ for certain rational q_i, r with $\|\vec{q}\|_1 \le N_k$ and $|r| \le R_k$. The maximal such bound has $|z| \le N_k \|\vec{a}\|_\infty + R_k$, hence any tuple $\langle \vec{a}, b\rangle$ is E_k-equivalent to one with $|b| \le N_k\|\vec{a}\|_\infty + R_k + M_k \le N_k\|\vec{a}\|_\infty + 2M_k$ (in view of Remark 2.2).

If we start with a sentence of length n, the initial $\vec{a}$ is empty, and we iterate this bound up to n times, each time with $k \leq n$, thus all integers used by the algorithms are bounded by $O(M_n N_n^n) = 2^{2^{2^{O(n)}}}$.

This means that when written in binary, each number takes $2^{2^{O(n)}}$ bits, and we only need $O(n)$ of them at a time, hence the overall space usage is $2^{2^{O(n)}}$. The recursive invocations of the algorithm form a tree of branching $2^{2^{2^{O(n)}}}$ and depth n, and each step involves simple computation with numbers of bit-length $2^{2^{O(n)}}$ which can be done in time $2^{2^{O(n)}}$. Thus, the total time spent by the algorithm is $2^{2^{2^{O(n)}}}$ (alternatively, this follows from (7)). □

Remark 3.7 The algorithm for Presburger arithmetic from the proof of Proposition 3.6 is more-or-less optimal, but deterministic space is not the best measure how to express its complexity; it can be implemented on a so-called *alternating Turing machine* working in time $2^{2^{O(n)}}$ with $O(n)$ alternations, and by a result of Berman, Presburger arithmetic is complete for the class of all problems decidable by such algorithms.

The bounds become considerably better (to the point of being practically usable) for formulas with a bounded number of quantifier alternations. This can be done by an adaptation of the algorithm in Fig. 2 to handle blocks of quantifiers in one go, and a corresponding modification of EF games and GBFS.

Exercise 3.8 Devise a suitable GBFS for the theory $\mathrm{Th}(\mathbb{Z}, <)$ of discrete linear orders and show that it is PSPACE-complete.

References

[1] Mojżesz Presburger, *Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt*, in: Sprawozdanie z I Kongresu matematyków krajów słowiańskich, Warszawa 1929 (Comptes-rendus du I Congrès des Mathématiciens des Pays Slaves, Varsovie 1929) (Warsaw), 1930, pp. 92–101.