Linear arithmetic: Geometry, algorithms, and logic

Friday

Dmitry Chistikov Christoph Haase

Centre for Discrete Mathematics and its Applications (DIMAP) &
Department of Computer Science, University of Warwick, UK

Department of Computer Science
University of Oxford, UK

ESSLLI 2018

Today's lecture

What makes Presburger arithmetic computationally hard:

- Hardness results for fragments of Presburger arithmetic
- How to encode big numbers in Presburger arithmetic

Computationally hard problems

NP-hard problems

3CNF satisfiability:

- Given $\Phi(x_1, \ldots, x_k) = C_1 \wedge \cdots \wedge C_k$ with

$$C_i = \ell_1^i \vee \ell_2^i \vee \ell_3^i$$

- Find $\mathcal{A}\colon X \to \{0, 1\}$ such that Φ evaluates to true

NP-hard problems

3CNF satisfiability:

- Given $\Phi(x_1, \ldots, x_k) = C_1 \wedge \cdots \wedge C_k$ with

$$C_i = \ell_1^i \vee \ell_2^i \vee \ell_3^i$$

- Find $\mathcal{A}\colon X \to \{0, 1\}$ such that Φ evaluates to true

Subset Sum:

- Given $M = \{m_1, \ldots, m_n\} \subseteq \mathbb{N}$ and $t \in \mathbb{N}$
- Is there $M' \subseteq M$ such that

$$t = \sum_{m \in M'} m$$

NP-hard problems

3CNF satisfiability:

- Given $\Phi(x_1, \ldots, x_k) = C_1 \wedge \cdots \wedge C_k$ with

$$C_i = \ell_1^i \vee \ell_2^i \vee \ell_3^i$$

- Find $\mathcal{A}\colon X \to \{0, 1\}$ such that Φ evaluates to true

Subset Sum:

- Given $M = \{m_1, \ldots, m_n\} \subseteq \mathbb{N}$ and $t \in \mathbb{N}$
- Is there $M' \subseteq M$ such that

$$t = \sum_{m \in M'} m$$

Corollary

Integer programming is NP-hard.

Even harder problems

Σ_i-3CNF satisfiability, i odd

- Given $\Phi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = C_1 \wedge \cdots \wedge C_k$ with

$$C_i = \ell_1^i \vee \ell_2^i \vee \ell_3^i$$

- Does the following hold

$$\exists \boldsymbol{x}_k \, \forall \boldsymbol{x}_{k-1}, \cdots \exists \boldsymbol{x}_1 : \Phi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)?$$

Even harder problems

Σ_i-3CNF satisfiability, i odd

- Given $\Phi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = C_1 \wedge \cdots \wedge C_k$ with

$$C_i = \ell_1^i \vee \ell_2^i \vee \ell_3^i$$

- Does the following hold

$$\exists \boldsymbol{x}_k \, \forall \boldsymbol{x}_{k-1}, \cdots \exists \boldsymbol{x}_1 : \Phi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)?$$

Σ_i-Subset Sum, i odd:

- Given $M_1 = \{m_1^1, \ldots, m_{n_1}^1\}, \ldots, M_k = \{m_1^k, \ldots, m_{n_k}^k\} \subseteq \mathbb{N}$ and $t \in \mathbb{N}$
- Does the following hold:

$$\exists M_k' \subseteq M_k \, \forall M_{k-1}' \subseteq M_{k-1} \cdots \exists M_1' \subseteq M_1 : t = \sum_{i=1}^{k} \sum_{m \in M_i'} m$$

Chinese remainder representation

The Chinese remainder theorem

Theorem

Let $m_1, \ldots, m_k > 0$ be relatively co-prime. Then the system of linear congruences

$$\begin{aligned} x &\equiv a_1 \mod m_1 \\ &\vdots \\ x &\equiv a_k \mod m_k \end{aligned}$$

has a unique solution modulo $\prod_{i=1}^{k} m_i$.

The Chinese remainder theorem

Theorem

Let $m_1, \ldots, m_k > 0$ be relatively co-prime. Then the system of linear congruences

$$\begin{aligned} x &\equiv a_1 \mod m_1 \\ &\vdots \\ x &\equiv a_k \mod m_k \end{aligned}$$

has a unique solution modulo $\prod_{i=1}^{k} m_i$.

- satisfying x can be computed in polynomial time
- can choose m_i to be prime number p_i
- prime number theorem guarantees that $p_i \sim i \cdot \log i$

Chinese remainder representation

Word $b_1 \cdots b_k \in \{0,1\}^k$ can uniquely encoded into $x \in \mathbb{N}$ using system of linear congruences:

$$\begin{aligned} x &\equiv b_1 \mod p_1 \\ &\vdots \\ x &\equiv b_k \mod p_k \end{aligned}$$

Chinese remainder representation

Word $b_1 \cdots b_k \in \{0,1\}^k$ can uniquely encoded into $x \in \mathbb{N}$ using system of linear congruences:

$$\begin{aligned} x &\equiv b_1 \mod p_1 \\ &\vdots \\ x &\equiv b_k \mod p_k \end{aligned}$$

Problem: not every $x \in \mathbb{N}$ encodes a word in $\{0,1\}^k$

Chinese remainder representation

Word $b_1 \cdots b_k \in \{0,1\}^k$ can uniquely encoded into $x \in \mathbb{N}$ using system of linear congruences:

$$\begin{aligned} x &\equiv b_1 \mod p_1 \\ &\vdots \\ x &\equiv b_k \mod p_k \end{aligned}$$

Problem: not every $x \in \mathbb{N}$ encodes a word in $\{0,1\}^k$

Key observation:

- can rule out exponential number of invalid encodings with single non-congruence
- Constraint $x \not\equiv \ell \bmod p_i$ for $1 < \ell < p_i$ rules out

$$\prod_{1 \le j \ne i \le k} p_j$$

invalid encodings

Lower bound for $\exists\forall$ quantifier prefix

Presburger arithmetic becomes hard quickly

Theorem (Schöning, 1997)

Presburger arithmetic with an $\exists x \, \forall y$ quantifier prefix is NP-hard.

Presburger arithmetic becomes hard quickly

Theorem (Schöning, 1997)

Presburger arithmetic with an $\exists x \, \forall y$ quantifier prefix is NP-hard.

Reduction from 3-CNF satisfiability:

- Use Chinese remainder representation encoding assignment represented by $x \in \mathbb{N}$
- Ensure x is a valid encoding:

$$\forall y : \bigwedge_{k=1}^{k} \bigwedge_{\ell=2}^{p_j - 1} \neg(x = p_j \cdot y + \ell)$$

- Ensure x encodes satisfying assignment: for $C = x_i \vee \neg x_j \vee x_k$ add constraint

$$\neg(x \equiv 0 \bmod p_i \wedge x \equiv 1 \bmod p_j \wedge x \equiv 0 \bmod p_k)$$
$$\iff \forall z : \neg(x = p_i \cdot z) \wedge \neg(x = p_j \cdot z + 1) \wedge \neg(x = p_k \cdot z)$$

- Melt universal quantifiers

Quantified Integer Programming

Quantified integer programming is hard

Quantified integer programming for k odd:

$$\begin{aligned} & \exists \boldsymbol{x}_k \forall \boldsymbol{x}_{k-1} \cdots \exists \boldsymbol{x}_1 : A \cdot \boldsymbol{x} \geq \boldsymbol{c} \\ \iff & \exists \boldsymbol{x}_k \forall \boldsymbol{x}_{k-1} \cdots \exists \boldsymbol{x}_1 : A_k \cdot \boldsymbol{x}_k + \cdots + A_1 \cdot \boldsymbol{x}_1 \geq \boldsymbol{c} \end{aligned}$$

Quantified integer programming is hard

Quantified integer programming for k odd:

$$\begin{aligned} & \exists \boldsymbol{x}_k \forall \boldsymbol{x}_{k-1} \cdots \exists \boldsymbol{x}_1 : A \cdot \boldsymbol{x} \geq \boldsymbol{c} \\ \iff & \exists \boldsymbol{x}_k \forall \boldsymbol{x}_{k-1} \cdots \exists \boldsymbol{x}_1 : A_k \cdot \boldsymbol{x}_k + \cdots + A_1 \cdot \boldsymbol{x}_1 \geq \boldsymbol{c} \end{aligned}$$

Reduce from Σ_i-Subset Sum:

$$\exists M'_k \subseteq M_k \, \forall M'_{k-1} \subseteq M_{k-1} \cdots \exists M'_1 \subseteq M_1 : t = \sum_{i=1}^{k} \sum_{m \in M'_i} m$$

Quantified integer programming is hard

Quantified integer programming for k odd:

$$\begin{aligned} & \exists \boldsymbol{x}_k \forall \boldsymbol{x}_{k-1} \cdots \exists \boldsymbol{x}_1 : A \cdot \boldsymbol{x} \geq \boldsymbol{c} \\ \iff & \exists \boldsymbol{x}_k \forall \boldsymbol{x}_{k-1} \cdots \exists \boldsymbol{x}_1 : A_k \cdot \boldsymbol{x}_k + \cdots + A_1 \cdot \boldsymbol{x}_1 \geq \boldsymbol{c} \end{aligned}$$

Reduce from Σ_i-Subset Sum:

$$\exists M_k' \subseteq M_k \, \forall M_{k-1}' \subseteq M_{k-1} \cdots \exists M_1' \subseteq M_1 : t = \sum_{i=1}^{k} \sum_{m \in M_i'} m$$

- Choice of i-th subset isomorphic to $\boldsymbol{v}_i \in \{0,1\}^{|M_i|}$
- Variables in $\boldsymbol{x}_i$ range over all of $\mathbb{Z}$

Constraining ranges without disjunction

For existentially quantified x add constraint:

$$0 \leq x \leq 1$$

Constraining ranges without disjunction

For existentially quantified x add constraint:

$$0 \leq x \leq 1$$

For universally quantified y would like:

$$\Phi(y) \equiv \forall y : 0 \leq y \leq 1 \rightarrow \Psi(y)$$

Constraining ranges without disjunction

For existentially quantified x add constraint:

$$0 \leq x \leq 1$$

For universally quantified y would like:

$$\Phi(y) \equiv \forall y : 0 \leq y \leq 1 \rightarrow \Psi(y)$$

Idea: remove slack with extra existentially quantified variable:

$$\Phi(y) \equiv \forall y \, \exists z : 0 \leq y - 2z \leq 1 \wedge \Psi(y - 2z)$$

Constraining ranges without disjunction

For existentially quantified x add constraint:

$$0 \leq x \leq 1$$

For universally quantified y would like:

$$\Phi(y) \equiv \forall y : 0 \leq y \leq 1 \rightarrow \Psi(y)$$

Idea: remove slack with extra existentially quantified variable:

$$\Phi(y) \equiv \forall y \, \exists z : 0 \leq y - 2z \leq 1 \wedge \Psi(y - 2z)$$

Theorem (Chistikov, Haase, 2017)
Quantified integer programming with $\boldsymbol{k}$ quantifier blocks, $\mathrm{Q}\boldsymbol{x}_k \forall \boldsymbol{x}_{k-1} \cdots \exists \boldsymbol{x}_1 : A \cdot \boldsymbol{x} \geq \boldsymbol{c}$, is complete for $\boldsymbol{k}^{\text{th}}$ level of $\mathbf{PH}$.

A puzzle: Can QIP express disjunction?

$(x = 0) \vee (y = 0)$:

- cannot be expressed by IP ($=$ ANDs of linear constraints)

What if we can use additional variables?

A puzzle: Can QIP express disjunction?

$(x = 0) \vee (y = 0)$:

- cannot be expressed by IP ($=$ ANDs of linear constraints)

What if we can use additional variables?

$$\exists\, z_1\ \forall\, z_2\ \exists\, z_3.\ \varphi(x, y, z_1, z_2, z_3)$$

A puzzle: Can QIP express disjunction?

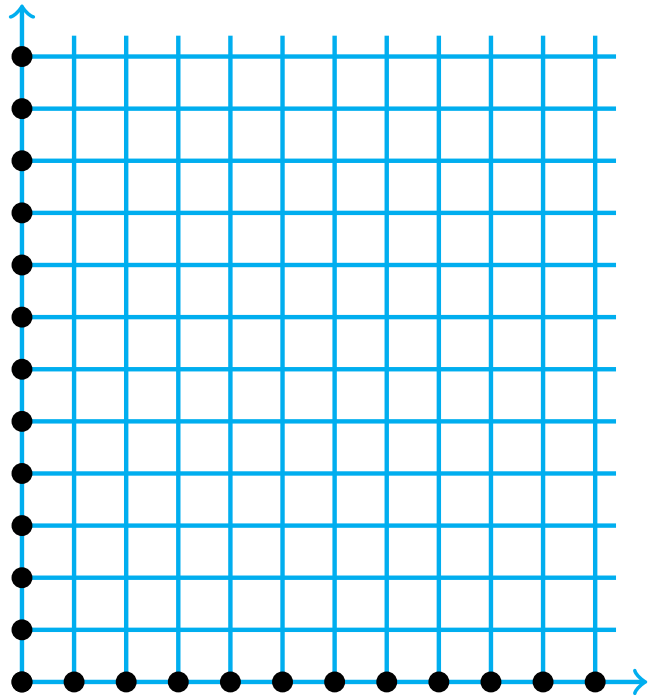
$(x = 0) \vee (y = 0)$:

- cannot be expressed by IP ($=$ ANDs of linear constraints)

What if we can use additional variables?

$$\begin{aligned} &\{(x, y) \colon \exists\, z_1\ \forall\, z_2\ \exists\, z_3.\ \varphi(x, y, z_1, z_2, z_3)\} \\ = &\{(x, y) \colon (x = 0) \text{ or } (y = 0)\} \end{aligned}$$

?



Quantified IP: Quantifier elimination

$$\exists\, x\ \forall\, y\ \exists\, z:\ \ A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \boldsymbol{c}$$

Quantified IP: Quantifier elimination

$$\exists\, x\ \forall\, y\ \exists\, z : \boxed{A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \boldsymbol{c}}$$

$L(C_0, Q_0) \subseteq \mathbb{N}^3$

Quantified IP: Quantifier elimination

$$\exists\, x\ \forall\, y \boxed{\exists\, z :\ A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \boldsymbol{c}}$$

$L(C_0, Q_0) \subseteq \mathbb{N}^3$

$L(C_1, Q_1) \subseteq \mathbb{N}^2$: projection of $L(C_0, Q_0)$

Quantified IP: Quantifier elimination

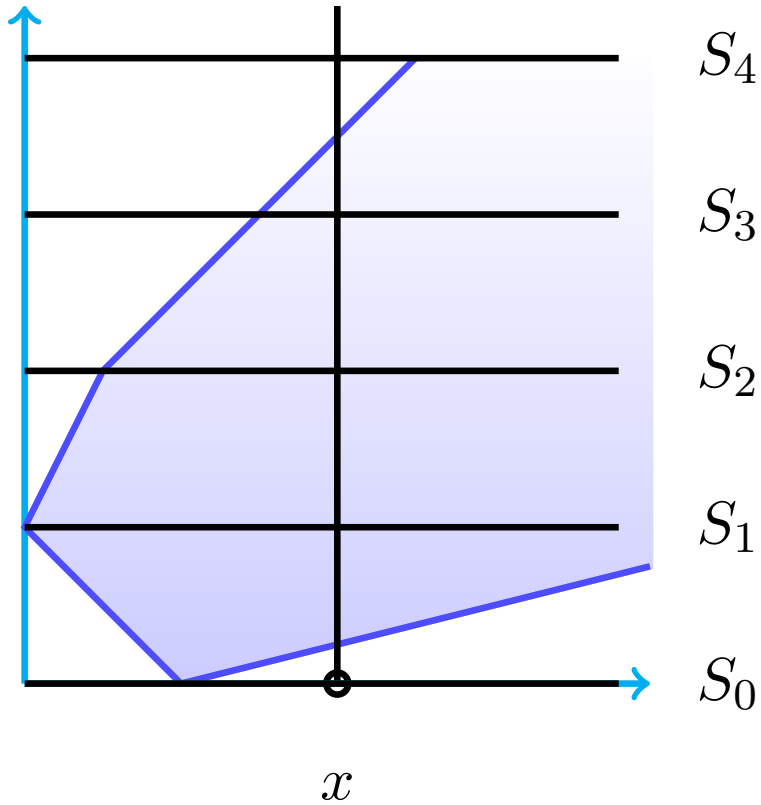
$$\exists\, x \boxed{\forall\, y\ \exists\, z :\ \ A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \boldsymbol{c}}$$

$L(C_0, Q_0) \subseteq \mathbb{N}^3$

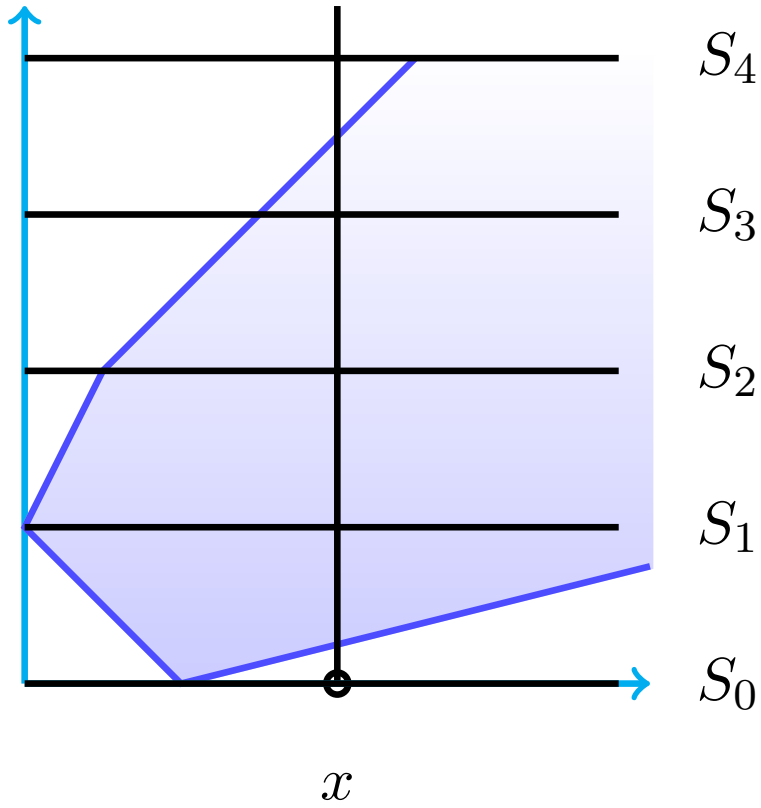
$L(C_1, Q_1) \subseteq \mathbb{N}^2$: projection of $L(C_0, Q_0)$

$L(C_2, Q_2) \subseteq \mathbb{N}^1$: universal projection of $L(C_1, Q_1)$

Quantified IP: Universal projection



Quantified IP: Universal projection



x belongs to the universal projection $\iff$ $x \in \bigcap_{i \in \mathbb{N}} S_i$

Quantified IP: Universal projection

$$S = L(C, Q) \subseteq \mathbb{N}^2 \quad \rightsquigarrow$$

$$\pi^*(S) = \{x \colon \forall\, y.\ (x, y) \in S\} = \bigcap_{i \in \mathbb{N}} \pi\left(S \cap \{(x, y) \colon y = i\}\right)$$

Quantified IP: Universal projection

$$S = L(C, Q) \subseteq \mathbb{N}^2 \quad \rightsquigarrow$$

$$\pi^*(S) = \{x \colon \forall\, y.\ (x, y) \in S\} = \bigcap_{i \in \mathbb{N}} \pi\left(S \cap \{(x, y) \colon y = i\}\right)$$

1. Universality = bounded universality: $\bigcap_{i \in \mathbb{N}} = \bigcap_{i < N}$

Quantified IP: Universal projection

$$S = L(C, Q) \subseteq \mathbb{N}^2 \quad \rightsquigarrow$$

$$\pi^*(S) = \{x \colon \forall\, y.\ (x, y) \in S\} = \bigcap_{i \in \mathbb{N}} \pi\left(S \cap \{(x, y) \colon y = i\}\right)$$

1. Universality = bounded universality: $\displaystyle\bigcap_{i \in \mathbb{N}} = \bigcap_{i < N}$

2. Parallel cross-sections (i) are hybrid linear sets
   and (ii) share directions: $S \cap \{(x, y) \colon y = i\} = L(B_i, P)$

Quantified IP: Universal projection

$$S = L(C, Q) \subseteq \mathbb{N}^2 \quad \rightsquigarrow$$

$$\pi^*(S) = \{x \colon \forall\, y.\ (x, y) \in S\} = \bigcap_{i \in \mathbb{N}} \pi\left(S \cap \{(x, y) \colon y = i\}\right)$$

1. Universality = bounded universality: $\bigcap_{i \in \mathbb{N}} = \bigcap_{i < N}$

2. Parallel cross-sections (i) are hybrid linear sets and (ii) share directions: $S \cap \{(x, y) \colon y = i\} = L(B_i, P)$

3. Long intersections have small size: $\bigcap_{i < N} L(B_i, P) = L(B, P)$, size of numbers in B small

Quantified IP: Universal projection

$S = L(C, Q) \subseteq \mathbb{N}^2 \quad \rightsquigarrow$

$\pi^*(S) = \{x \colon \forall\, y.\ (x, y) \in S\} = \bigcap_{i \in \mathbb{N}} \pi\left(S \cap \{(x, y) \colon y = i\}\right)$

1. Universality = bounded universality: $\bigcap_{i \in \mathbb{N}} = \bigcap_{i < N}$

2. Parallel cross-sections (i) are hybrid linear sets
   and (ii) share directions: $S \cap \{(x, y) \colon y = i\} = L(B_i, P)$

3. Long intersections have small size:
   $\bigcap_{i < N} L(B_i, P) = L(B, P)$, size of numbers in B small

Conclusion: Universal projection $\pi^*(S) = L(B, P)$,
size of numbers in $\pi^*(S)$ does not blow up relative to S.

Quantified IP: Quantifier elimination

$$\exists\, x \boxed{\forall\, y\ \exists\, z :\ \ A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \boldsymbol{c}}$$

$L(C_0, Q_0) \subseteq \mathbb{N}^3$

$L(C_1, Q_1) \subseteq \mathbb{N}^2$: projection of $L(C_0, Q_0)$

$L(C_2, Q_2) \subseteq \mathbb{N}^1$: universal projection of $L(C_1, Q_1)$

Quantified IP: Quantifier elimination

$$\exists\, x \boxed{\forall\, y\ \exists\, z :\ \ A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \boldsymbol{c}}$$

$L(C_0, Q_0) \subseteq \mathbb{N}^3$

$L(C_1, Q_1) \subseteq \mathbb{N}^2$: projection of $L(C_0, Q_0)$

$L(C_2, Q_2) \subseteq \mathbb{N}^1$: universal projection of $L(C_1, Q_1)$

$x \leq M_1$: small model property

Quantified IP: Quantifier elimination

$$\exists\, x\ \forall\, y\ \exists\, z:\ \ A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \boldsymbol{c}$$

$L(C_0, Q_0) \subseteq \mathbb{N}^3$

$L(C_1, Q_1) \subseteq \mathbb{N}^2$: projection of $L(C_0, Q_0)$

$L(C_2, Q_2) \subseteq \mathbb{N}^1$: universal projection of $L(C_1, Q_1)$

$x \leq M_1$: small model property

$y \leq M_2$, $z \leq M_3$: relativization property

Quantified IP: Summary

$$\psi = \mathrm{Q}_k \boldsymbol{x}_k \ \ldots \ \forall \boldsymbol{x}_2 . \ \exists \boldsymbol{x}_1 : A \cdot \boldsymbol{x} \leq \boldsymbol{c}$$

Lemma (small model property):
The validity of ψ does not change if the variables $\boldsymbol{x}_k$ are interpreted over $[0, M]$ instead of $\mathbb{N}$, where $\log M = |\psi|^{O(k)}$.

Lemma (relativization property):
The validity of ψ does not change if, for each $i \in [1, k]$, the variables $\boldsymbol{x}_i$ are interpreted over $[0, M_i]$ instead of $\mathbb{N}$, where $\log M_i = |\psi|^{O(2k-i)}$.

Conclusion:
QIP with k quantifier blocks is complete for k^{th} level of $\mathbf{PH}$.

Very big numbers

Presburger arithmetic with bounded constants

We typically use arbitrary integers in Presburger formulae.
Does the expressive power of Presburger arithmetic change if only bounded numbers are allowed?

Presburger arithmetic with bounded constants

We typically use arbitrary integers in Presburger formulae. Does the expressive power of Presburger arithmetic change if only bounded numbers are allowed?

No. The formula $y = 2^n \cdot x$ is equivalent to

$$\begin{aligned}
&\exists y_0\, \exists y_1\, \ldots \exists y_n \colon y_0 = x \wedge \\
&y_1 = 2y_0 \wedge \\
&y_2 = 2y_1 \wedge \\
&\ldots \\
&y_n = 2y_{n-1} \wedge \\
&y = y_n
\end{aligned}$$

What numbers can we express in Presburger arithmetic?

The formula we've just seen shows that

> numbers up to $2^{\mathrm{poly}(n)}$ can be expressed by formulae of size $\mathrm{poly}(n)$ where only -2, -1, 0, 1, and 2 are used.

Can we express even larger numbers using small formulae?

Very big numbers in Presburger arithmetic

Theorem (Fischer and Rabin, 1974)

There is a sequence of formulae $F_n(x, y)$ of size $O(n)$ such that

$$F_n(x, y) \text{ holds} \iff x = 2^{2^n} \cdot y.$$

Very big numbers in Presburger arithmetic

Theorem (Fischer and Rabin, 1974)
There is a sequence of formulae $F_n(x, y)$ of size $O(n)$ such that

$$F_n(x, y) \text{ holds} \iff x = 2^{2^n} \cdot y.$$

Theorem (Haase, 2014)
There exists a constant $c > 0$, a sequence of integers $(r_n)_{n \geq 1}$, and a sequence of $\forall\exists^*$-formulae $H_n(z)$ of size $O(n)$ such that

$$H_n(z) \text{ holds} \iff r_n \text{ divides } z$$

and $r_n \geq 2^{2^{cn}}$ for all n.

What we have learned in this course

Geometry in $\mathbb{R}^d$

- Convex sets and cones
- Convex polyhedra
- Minkowski—Weyl theorem
- Faces of polyhedra
- Dimension of polyhedra
- Farkas' lemma

Linear programming

- Fourier—Motzkin elimination
- Simplex method
- Ellipsoid method
- Linear programming is in $\mathbf{P}$

Integer programming

- Linear, hybrid linear, and semi-linear sets
- Integer linear inequalities and discrete Minkowski—Weyl
- Integer programming is $\mathbf{NP}$-complete
- Branch and bound
- Randomized rounding

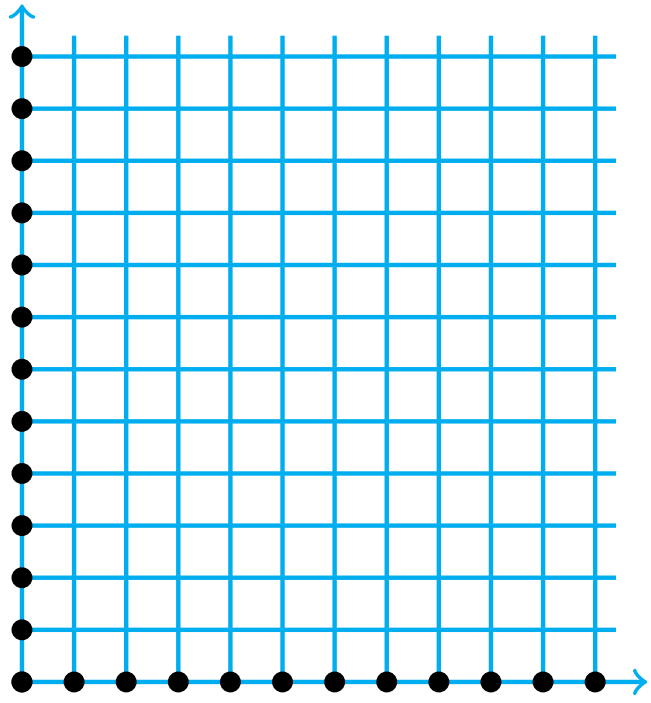
Presburger arithmetic

- Quantifier elimination
- Automata-based decision procedures
- Generator-based decision procedures
- Encodings of hard problems and very big numbers
- Subset sum and quantified integer programming

Logical theories of linear arithmetic

- Syntax and semantics
- Geometry of definable sets
- Expressive power
- Existential conjunctive fragments
- Algorithms (decision procedures)

Our groups are looking for new interns,
PhD students, postdocs, and faculty!



Thank you!

d.chistikov@warwick.ac.uk
christoph.haase@cs.ox.ac.uk