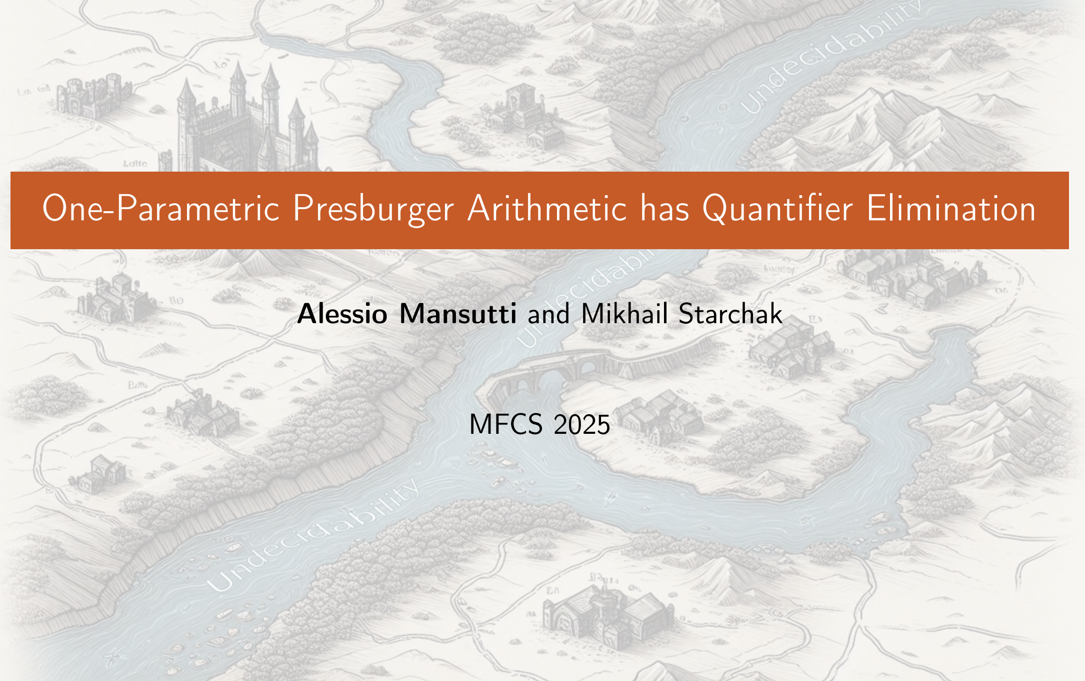


One-Parametric Presburger Arithmetic has Quantifier Elimination

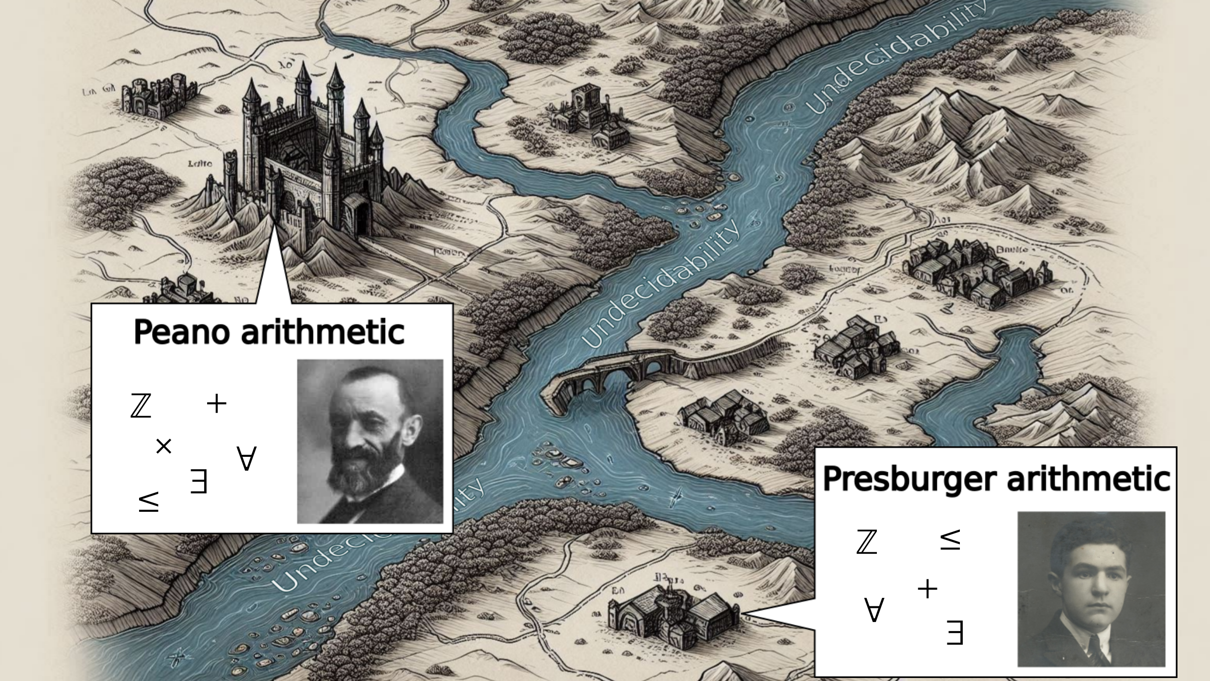
Alessio Mansutti and Mikhail Starchak

MFCS 2025

Undecidability
Undecidability
Undecidability
Undecidability

Peano arithmetic

ℤ + × ∀ ∃ ≤

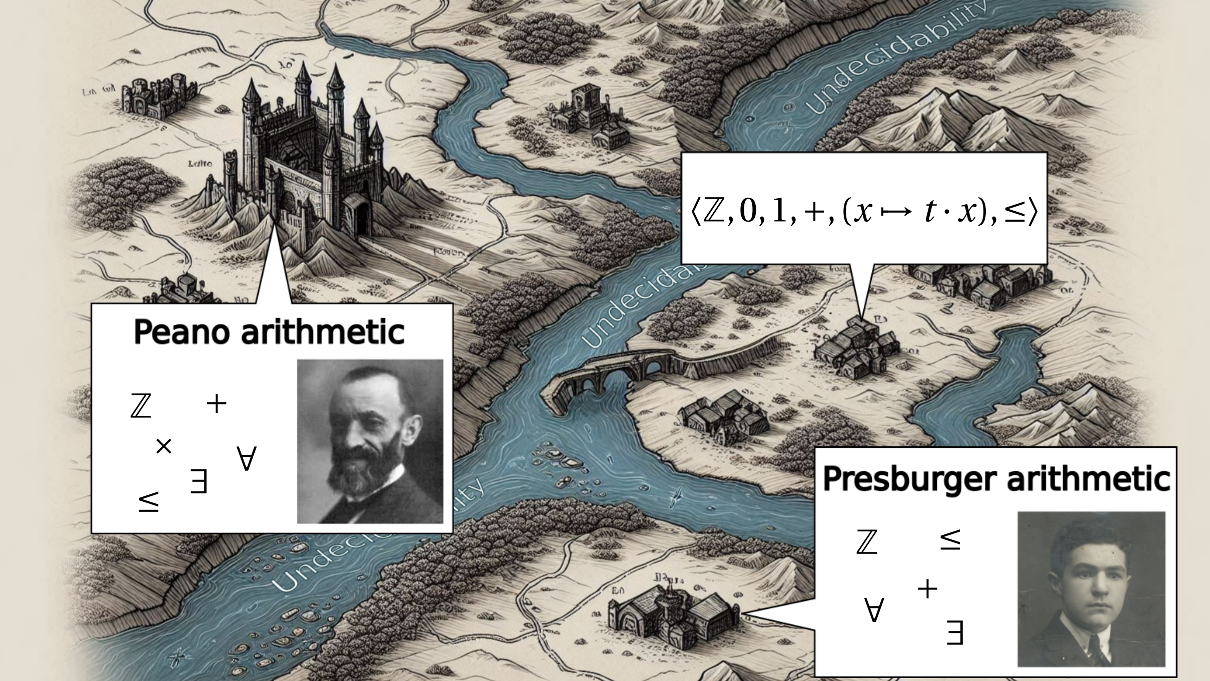


Peano arithmetic

$\mathbb{Z}$ $+$ $\times$ $\forall$ $\exists$ $\leq$

Presburger arithmetic

$\mathbb{Z}$ $\leq$ $+$ $\forall$ $\exists$



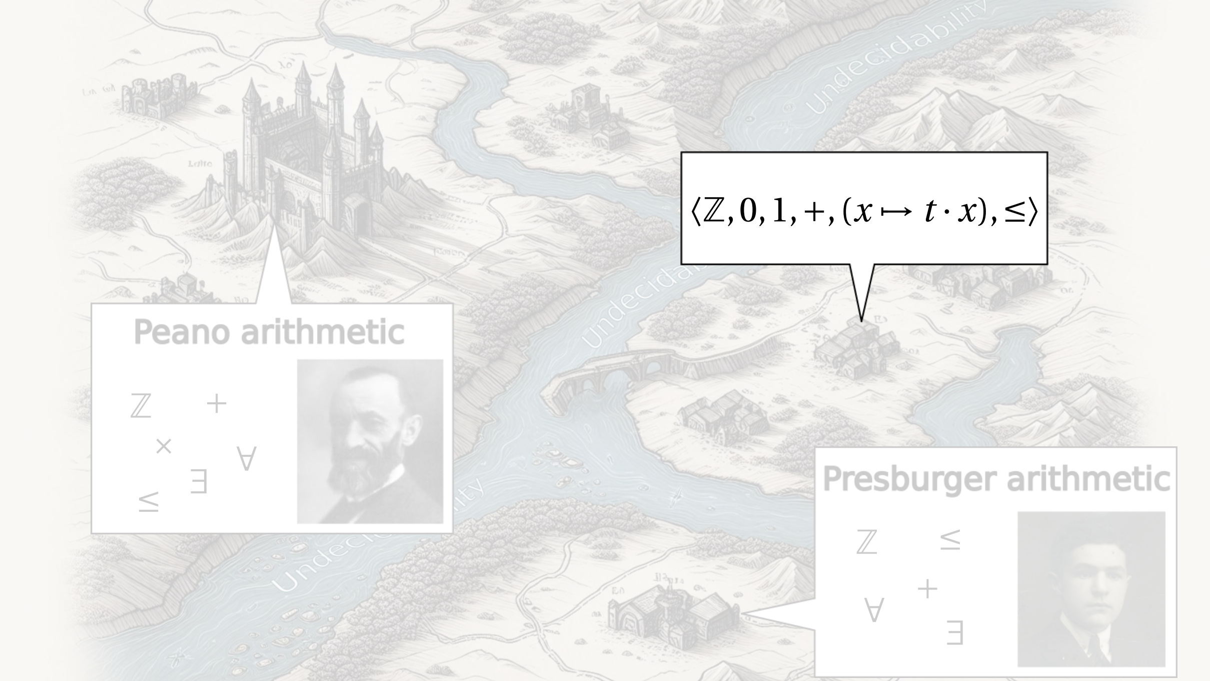
$\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$

Peano arithmetic

$\mathbb{Z}$ $+$ $\times$ $\forall$ $\exists$ $\leq$

Presburger arithmetic

$\mathbb{Z}$ $\leq$ $+$ $\forall$ $\exists$



$$\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$$

Peano arithmetic

$\mathbb{Z}$ $+$ $\times$ $\forall$ $\exists$ $\leq$

Presburger arithmetic

$\mathbb{Z}$ $\leq$ $+$ $\forall$ $\exists$

One-parametric Presburger arithmetic (1PPA)

First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

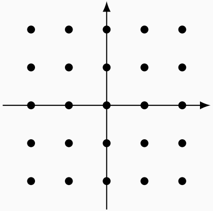
One-parametric Presburger arithmetic (1PPA)

First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

Twisting squares (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

$$|2x + (2t-2)y| \leq t^2 - 2t + 2 \ \wedge \ |(2-2t)x + 2y| \leq t^2 - 2t + 2$$



One-parametric Presburger arithmetic (1PPA)

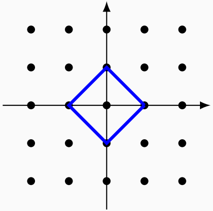
First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

Twisting squares (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

$$|2x + (2t-2)y| \leq t^2 - 2t + 2 \;\wedge\; |(2-2t)x + 2y| \leq t^2 - 2t + 2$$

$t = 0$: $|2x - 2y| \leq 2 \;\wedge\; |2x + 2y| \leq 2$ 5 solutions



One-parametric Presburger arithmetic (1PPA)

First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

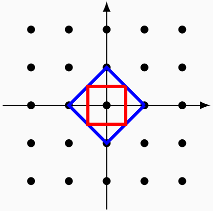
In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

Twisting squares (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

$$|2x + (2t-2)y| \leq t^2 - 2t + 2 \ \wedge \ |(2-2t)x + 2y| \leq t^2 - 2t + 2$$

$t = 0$: $|2x - 2y| \leq 2 \ \wedge \ |2x + 2y| \leq 2$ 5 solutions

$t = 1$: $|2x| \leq 1 \ \wedge \ |2y| \leq 1$ 1 solution



One-parametric Presburger arithmetic (1PPA)

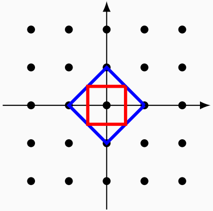
First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

Twisting squares (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

$$|2x + (2t-2)y| \leq t^2 - 2t + 2 \ \wedge \ |(2-2t)x + 2y| \leq t^2 - 2t + 2$$

$t = 0$:	$\lvert 2x - 2y \rvert \leq 2 \ \wedge \ \lvert 2x + 2y \rvert \leq 2$	5 solutions
$t = 1$:	$\lvert 2x \rvert \leq 1 \ \wedge \ \lvert 2y \rvert \leq 1$	1 solution
$t = 2$:	$\lvert 2x + 2y \rvert \leq 2 \ \wedge \ \lvert -2x + 2y \rvert \leq 2$	same as $t = 0$



One-parametric Presburger arithmetic (1PPA)

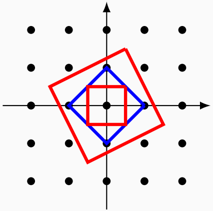
First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

Twisting squares (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

$$|2x + (2t-2)y| \leq t^2 - 2t + 2 \ \wedge \ |(2-2t)x + 2y| \leq t^2 - 2t + 2$$

$t = 0$:	$\lvert 2x - 2y \rvert \leq 2 \ \wedge \ \lvert 2x + 2y \rvert \leq 2$	5 solutions
$t = 1$:	$\lvert 2x \rvert \leq 1 \ \wedge \ \lvert 2y \rvert \leq 1$	1 solution
$t = 2$:	$\lvert 2x + 2y \rvert \leq 2 \ \wedge \ \lvert -2x + 2y \rvert \leq 2$	same as $t = 0$
$t = 3$:	$\lvert 2x + 4y \rvert \leq 5 \ \wedge \ \lvert -4x + 2y \rvert \leq 5$	5 solutions



One-parametric Presburger arithmetic (1PPA)

First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

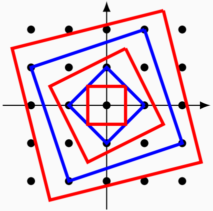
In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

Twisting squares (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

$$|2x + (2t-2)y| \leq t^2 - 2t + 2 \ \wedge \ |(2-2t)x + 2y| \leq t^2 - 2t + 2$$

For a fixed $t \geq 0$, this formula:

- has $t^2 - 2t + 2$ solutions when t is odd
- has $t^2 - 2t + 5$ solutions when t is even



One-parametric Presburger arithmetic (1PPA)

First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

"Chinese Remainder Theorem"

The following formula is valid:

$$t \geq 1 \implies \forall a \forall b \exists x : \quad \begin{aligned} & 0 \leq x < t(t+1) \\ \wedge \quad & t \mid x - a \\ \wedge\ & t+1 \mid x - b \end{aligned}$$

where $(p(t) \mid \tau) := \exists w\, (w \cdot p(t) = \tau)$.

"For every positive integer t, and for all integers a and b, there is an integer x in the interval $[0..t(t+1)-1]$ that is congruent to a modulo t, and to b modulo $t+1$."

One-parametric Presburger arithmetic (1PPA)

First-order theory of the structure $\langle \mathbb{Z}, 0, 1, +, (x \mapsto t \cdot x), \leq \rangle$.

In the multiplication function $x \mapsto t \cdot x$, the parameter t is a fixed free variable.

A formula $\varphi(\boldsymbol{x})$ of 1PPA defines a parametric Presburger family $\{[\![\varphi]\!]_k : k \in \mathbb{Z}\}$, where

$$[\![\varphi]\!]_k : \text{set of solution to } \varphi \text{ after replacing } t \text{ with } k$$

We can ask several questions about φ:

- satisfiability: is $[\![\varphi]\!]_k$ non-empty for some k?
- validity: is $[\![\varphi]\!]_k$ non-empty for every k?
- finiteness: is $[\![\varphi]\!]_k$ non-empty only for finitely many k?

Eventual quasi-polynomials and 1PPA

A function $f : \mathbb{N} \to \mathbb{N}$ is an eventual quasi-polynomial (EQP) whenever there are

- a threshold T and a period P, and
- a family of univariate polynomials $f_0, \dots, f_{P-1}$

such that for every $n \geq T$, $f(n) = f_{(n \bmod P)}(n)$.

Eventual quasi-polynomials and 1PPA

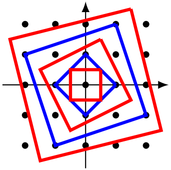
A function $f : \mathbb{N} \to \mathbb{N}$ is an eventual quasi-polynomial (EQP) whenever there are

- a threshold T and a period P, and
- a family of univariate polynomials $f_0, \ldots, f_{P-1}$

such that for every $n \geq T$, $f(n) = f_{(n \bmod P)}(n)$.

Examples:

$$\left\lfloor \frac{x}{2} \right\rfloor = \begin{cases} \frac{x}{2} & \text{if } x \text{ is even} \\ \frac{x-1}{2} & \text{if } x \text{ is odd} \end{cases}$$



Eventual quasi-polynomials and 1PPA

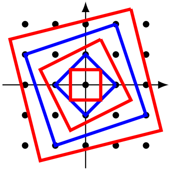
A function $f\colon \mathbb{N} \to \mathbb{N}$ is an eventual quasi-polynomial (EQP) whenever there are

- a threshold T and a period P, and
- a family of univariate polynomials $f_0, \dots, f_{P-1}$

such that for every $n \geq T$, $f(n) = f_{(n \bmod P)}(n)$.

Examples:

$$\left\lfloor \frac{x}{2} \right\rfloor = \begin{cases} \frac{x}{2} & \text{if } x \text{ is even} \\ \frac{x-1}{2} & \text{if } x \text{ is odd} \end{cases}$$



Theorem (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

Let φ be a 1PPA *formula. The counting function $f(k) := \#[\![\varphi]\!]_k$ is an EQP.*

Eventual quasi-polynomials and 1PPA

Theorem (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

Let φ *be a* 1PPA *formula. The counting function* $f(k) := \#[\![\varphi]\!]_k$ *is an EQP.*

Eventual quasi-polynomials and 1PPA

Theorem (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

Let φ *be a* 1PPA *formula. The counting function* $f(k) := \#[\![\varphi]\!]_k$ *is an EQP.*

Proof idea: Show the result for quantifier-free formulae. Then,

$$\varphi \;=\; \exists x_1 \; \forall x_2 \; \ldots : \psi$$

Eventual quasi-polynomials and 1PPA

Theorem (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

Let φ be a 1PPA *formula. The counting function $f(k) := \#[\![\varphi]\!]_k$ is an EQP.*

Proof idea: Show the result for quantifier-free formulae. Then,

$$\begin{tikzcd} \varphi \;=\; \exists x_1 \; \forall x_2 \; \dots : \psi \arrow[d, "\begin{array}{l}\textit{bounded quantifier elimination}\text{ (Weispfenning. }\textit{ISSAC 1997}\text{)} \\ \text{“}\exists y \leq p(t)\text{” constrains } y \text{ in } [0..p(t)]\end{array}"] \\ \varphi \;\equiv\; \exists y_1 \leq p_1(t) \; \forall y_2 \leq p_2(t) \; \dots : \gamma \end{tikzcd}$$

Eventual quasi-polynomials and 1PPA

Theorem (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

Let φ be a 1PPA *formula. The counting function $f(k) := \#[\![\varphi]\!]_k$ is an EQP.*

Proof idea: Show the result for quantifier-free formulae. Then,

$$\varphi \;=\; \exists x_1 \; \forall x_2 \; \dots : \psi$$

↓

bounded quantifier elimination (Weispfenning. *ISSAC 1997*)

"$\exists y \leq p(t)$" constrains y in $[0..p(t)]$

$$\varphi \;\equiv\; \exists y_1 \leq p_1(t) \; \forall y_2 \leq p_2(t) \; \dots : \gamma$$

↓

parsimonious transformation (Chen, Li, Sam. *Trans. Amer. Math. Soc. 2012*)

$\#[\![\varphi]\!]_k = \#[\![\varphi']\!]_k$ for every k

φ' quantifier-free

Eventual quasi-polynomials and 1PPA

Theorem (Bogart, Goodrick, Woods. *Discrete Analysis 2017*)

In *Discrete Analysis 2017*, Bogart, Goodrick and Woods ask whether the parsimonious transformation can be replaced with quantifier elimination.

Let φ

Proof

$$\varphi \;=\; \exists x_1 \, \forall x_2 \, \dots : \psi$$

$\downarrow$ *bounded quantifier elimination* (Weispfenning. *ISSAC 1997*)

"$\exists y \leq p(t)$" constrains y in $[0..p(t)]$

$$\varphi \;\equiv\; \exists y_1 \leq p_1(t) \; \forall y_2 \leq p_2(t) \; \dots : \gamma$$

$\downarrow$ *parsimonious transformation* (Chen, Li, Sam. *Trans. Amer. Math. Soc. 2012*)

$\#[\![\varphi]\!]_k = \#[\![\varphi']\!]_k$ for every k

φ' quantifier-free

Eventual quasi-polynomials and 1PPA

Theorem (Bogart, Goodrick, Woods, *Discrete Analysis 2017*)

In *Discrete Analysis 2017*, Bogart, Goodrick and Woods ask whether the parsimonious transformation can be replaced with quantifier elimination.

Let φ

Proof

In *Arch. Math. Logic 2018*, Goodrick conjectures that extending 1PPA with a function $x \mapsto \left\lfloor \frac{x}{p(t)} \right\rfloor$ for every polynomial p suffices.

$$\varphi \;\equiv\; \exists y_1 \le p_1(t)\ \forall y_2 \le p_2(t)\ \ldots : \gamma$$

parsimonious transformation (Chen, Li, Sam. *Trans. Amer. Math. Soc. 2012*)

$\#[\![\varphi]\!]_k = \#[\![\varphi']\!]_k$ for every k

φ' quantifier-free

Eventual quasi-polynomials and 1PPA

Theorem (Bogart, Goodrick, Woods, *Discrete Analysis 2017*)

Let φ

In *Discrete Analysis 2017*, Bogart, Goodrick and Woods ask whether the parsimonious transformation can be replaced with quantifier elimination.

Proof

In *Arch. Math. Logic 2018*, Goodrick conjectures that extending 1PPA with a function $x \mapsto \left\lfloor \frac{x}{p(t)} \right\rfloor$ for every polynomial p suffices.

$$\varphi \;\equiv\; \exists y_1 \leq p_1(t) \; \forall y_2 \leq p_2(t) \;\dots : \gamma$$

parsi *ns. Amer. Math. Soc. 2012)*

We prove Goodrick's conjecture.

$\#[\![\varphi$

φ' quantifier-free

Our results

Theorem

There is a quantifier elimination procedure for the extension of 1PPA *with the functions:*

- *integer division:* $x \mapsto \left\lfloor \frac{x}{t^d} \right\rfloor$ *one function for each* $d \in \mathbb{N}$, *assumes* $t \neq 0$
- *integer remainder function:* $x \mapsto (x \bmod p)$ *for each* $p \in \mathbb{Z}[t]$
- *divisibility relation:* $p \mid x$ *for each* $p \in \mathbb{Z}[t]$

(The functions $x \mapsto \left\lfloor \frac{x}{p(t)} \right\rfloor$ capture all these functions and relations.)

Our results

Theorem

There is a quantifier elimination procedure for the extension of 1PPA *with the functions:*

- *integer division:* $x \mapsto \left\lfloor \frac{x}{t^d} \right\rfloor$ *one function for each* $d \in \mathbb{N}$, *assumes* $t \neq 0$
- *integer remainder function:* $x \mapsto (x \bmod p)$ *for each* $p \in \mathbb{Z}[t]$
- *divisibility relation:* $p \mid x$ *for each* $p \in \mathbb{Z}[t]$

(The functions $x \mapsto \left\lfloor \frac{x}{p(t)} \right\rfloor$ capture all these functions and relations.)

Theorem

For the class of all existential formulae of 1PPA, *the following holds:*

Satisfiability:	*Universality:*	*Finiteness:*
NP-*complete*	**coNEXP**-*complete*	**coNP**-*complete*

Overview of our procedure

Input: A quantifier-free formula $\varphi(\boldsymbol{x}, \boldsymbol{z})$ from the extended language of 1PPA (1PPA$^+$).
Output: A quantifier-free formula ψ from 1PPA$^+$ that is equivalent to $\exists \boldsymbol{x}\, \varphi$.

Overview of our procedure

Input: A quantifier-free formula $\varphi(\boldsymbol{x}, \boldsymbol{z})$ from the extended language of 1PPA (1PPA$^+$).
Output: A quantifier-free formula ψ from 1PPA$^+$ that is equivalent to $\exists \boldsymbol{x} \varphi$.

Step I. Preprocessing: Remove divisions and remainder functions.

$$\cdots + \left\lfloor \frac{\tau}{t^d} \right\rfloor + \cdots \leq 0 \quad \rightarrow \quad \exists x \left(\cdots + x + \cdots \leq 0 \land \left(t^d x \leq \tau < t^d (x+1) \right) \right)$$

$$\cdots + (\tau \bmod f) + \cdots \leq 0 \quad \rightarrow \quad \exists x \left(\cdots + x + \cdots \leq 0 \land \left(0 \leq x < f - 1 \right) \land \left(f \mid \tau - x \right) \right)$$

Overview of our procedure

Input: A quantifier-free formula $\varphi(\boldsymbol{x}, \boldsymbol{z})$ from the extended language of 1PPA (1PPA$^+$).
Output: A quantifier-free formula ψ from 1PPA$^+$ that is equivalent to $\exists \boldsymbol{x}\,\varphi$.

Step I. Preprocessing: Remove divisions and remainder functions.

$$\cdots + \left\lfloor \frac{\tau}{t^d} \right\rfloor + \cdots \leq 0 \quad \to \quad \exists x \left(\cdots + x + \cdots \leq 0 \land \left(t^d x \leq \tau < t^d (x+1) \right) \right)$$

$$\cdots + (\tau \bmod f) + \cdots \leq 0 \quad \to \quad \exists x \left(\cdots + x + \cdots \leq 0 \land \left(0 \leq x < f - 1 \right) \land \left(f \mid \tau - x \right) \right)$$

Step II. Bounded quantifier elimination:

$$\exists \boldsymbol{x}' : \varphi'(\boldsymbol{x}', \boldsymbol{z}) \quad \to_\beta \quad \exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$$

such that $\exists \boldsymbol{z} : \gamma(\boldsymbol{z}, \boldsymbol{z})$ is equivalent to $\bigvee_\beta \exists \boldsymbol{w}_\beta \leq B_\beta : \gamma_\beta(\boldsymbol{w}_\beta, \boldsymbol{z})$

Step II: Bounded quantifier elimination in $\mathbf{NP}$ (simplified)

Naïve bounded quantifier elimination

Input: $\exists \boldsymbol{x} \colon \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \le B \colon \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
for $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ do
    $(a \cdot x + \tau \sim 0) \leftarrow$ guess an (in)equality in $\varphi$ featuring $x$, or $x \le 0$
    $\tau \leftarrow \tau + w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \le$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} \,/\, x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \wedge (a \mid \tau)$
    $\ell \leftarrow a$
return $Q\varphi$
```

Step II: Bounded quantifier elimination in $\mathbf{NP}$ (simplified)

Naïve bounded quantifier elimination

Input: $\exists \boldsymbol{x} \colon \varphi(\boldsymbol{x}, \boldsymbol{z})$
Output: $\exists \boldsymbol{w} \leq B \colon \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
for $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ do
    $(a \cdot x + \tau \sim 0) \leftarrow$ guess an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} / x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \land (a \mid \tau)$
    $\ell \leftarrow a$
return $Q\varphi$
```

Consider $\tau_1 \leq a \cdot x \leq \tau_2$ with $a > 0$.

"between τ_1 and τ_2 there is a multiple of a"

Step II: Bounded quantifier elimination in $\mathbf{NP}$ (simplified)

Naïve bounded quantifier elimination

Input: $\exists \boldsymbol{x} \colon \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \leq B \colon \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
**for** $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ **do**
    $(a \cdot x + \tau \sim 0) \leftarrow$ **guess** an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot \mathit{mod}(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} \,/\, x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \land (a \mid \tau)$
    $\ell \leftarrow a$
**return** $Q\varphi$
```

"$a \cdot \mathit{mod}(\varphi)$" is a positive polynomial in $\mathbb{Z}[t]$ that upper bounds the product between a and all the divisors appearing in φ.

Step II: Bounded quantifier elimination in $\mathbf{NP}$ (simplified)

Naïve bounded quantifier elimination

Input: $\exists \boldsymbol{x} : \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
**for** $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ **do**
    $(a \cdot x + \tau \sim 0) \leftarrow$ **guess** an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} / x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \wedge (a \mid \tau)$
    $\ell \leftarrow a$
**return** $Q\varphi$
```

$\varphi[\frac{-\tau}{a} / x]$: $\quad b \cdot x + \quad \rho = 0$

Step II: Bounded quantifier elimination in **NP** (simplified)

Naïve bounded quantifier elimination

Input: $\exists \boldsymbol{x} : \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
for $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ do
    $(a \cdot x + \tau \sim 0) \leftarrow$ guess an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} / x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \wedge (a \mid \tau)$
    $\ell \leftarrow a$
return $Q\varphi$
```

$\varphi[\frac{-\tau}{a} / x]$: $\quad a \cdot b \cdot x + a \cdot \rho = 0$

Step II: Bounded quantifier elimination in **NP** (simplified)

Naïve bounded quantifier elimination

Input: $\exists \boldsymbol{x} : \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
for $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ do
    $(a \cdot x + \tau \sim 0) \leftarrow$ guess an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} \,/\, x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \wedge (a \mid \tau)$
    $\ell \leftarrow a$
return $Q\varphi$
```

$\varphi[\frac{-\tau}{a} \,/\, x]$: $\qquad -b \cdot \tau + a \cdot \rho = 0$

Step II: Bounded quantifier elimination in **NP** (simplified)

Naïve bounded quantifier elimination

Input: $\exists \boldsymbol{x} : \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
for $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ do
    $(a \cdot x + \tau \sim 0) \leftarrow$ guess an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} / x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \land (a \mid \tau)$
    $\ell \leftarrow a$
return $Q\varphi$
```

$\varphi[\frac{-\tau}{a} / x]$: $-b \cdot \tau + a \cdot \rho = 0$

Note: $(-b \cdot \tau + a \cdot \rho) = \det \begin{bmatrix} a & \tau \\ b & \rho \end{bmatrix}$

Step II: Bounded quantifier elimination in **NP** (simplified)

Naïve bounded quantifier elimination

Input: $\exists \boldsymbol{x} \colon \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \leq B \colon \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
for $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ do
    $(a \cdot x + \tau \sim 0) \leftarrow$ guess an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} / x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \wedge (a \mid \tau)$
    $\ell \leftarrow a$
return $Q\varphi$
```

Step II: Bounded quantifier elimination in $\mathbf{NP}$ (simplified)

Bounded quantifier elimination meets Bareiss's algorithm

Input: $\exists \boldsymbol{x} \colon \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \leq B \colon \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
for $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ do
    $(a \cdot x + \tau \sim 0) \leftarrow$ guess an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + \ell \cdot w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} / x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \wedge (a \mid \tau)$
    $\ell \leftarrow a$
return $Q\varphi$
```

Step II: Bounded quantifier elimination in **NP** (simplified)

Bounded quantifier elimination meets Bareiss's algorithm

Input: $\exists \boldsymbol{x} \colon \varphi(\boldsymbol{x}, \boldsymbol{z})$

Output: $\exists \boldsymbol{w} \leq B \colon \gamma(\boldsymbol{w}, \boldsymbol{z})$

```
$Q \leftarrow$ empty sequence of bounded quantifiers
$\ell \leftarrow 1$
for $x$ in $\boldsymbol{x}$ and occurring in $\varphi$ do
    $(a \cdot x + \tau \sim 0) \leftarrow$ guess an (in)equality in $\varphi$ featuring $x$, or $x \leq 0$
    $\tau \leftarrow \tau + \ell \cdot w$ with $w$ fresh free variable
    append to $Q$ the quantifier $\exists w \leq$ "$a \cdot mod(\varphi)$"
    $\varphi \leftarrow \varphi[\frac{-\tau}{a} \,/\, x]$
    divide each (in)equality in $\varphi$ by $\ell$
    $\varphi \leftarrow \varphi \wedge (a \mid \tau)$
    $\ell \leftarrow a$
return $Q\varphi$
```

Desnanot–Jacobi identity:

$$\begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} \end{vmatrix} = a_{11} \cdot \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

Overview of our procedure

Input: A quantifier-free formula $\varphi(\boldsymbol{x}, \boldsymbol{z})$ from the extended language of 1PPA (1PPA$^+$).
Output: A quantifier-free formula ψ from 1PPA$^+$ that is equivalent to $\exists \boldsymbol{x}\, \varphi$.

Step I. Preprocessing: Remove divisions and remainder functions.

Step II. Bounded quantifier elimination: compute $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

Overview of our procedure

Input: A quantifier-free formula $\varphi(\boldsymbol{x}, \boldsymbol{z})$ from the extended language of 1PPA (1PPA$^+$).
Output: A quantifier-free formula ψ from 1PPA$^+$ that is equivalent to $\exists \boldsymbol{x}\, \varphi$.

Step I. Preprocessing: Remove divisions and remainder functions.

Step II. Bounded quantifier elimination: compute $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

Step III. Remove the divisibility relations.

$$f \mid \tau(\boldsymbol{w}) + \sigma(\boldsymbol{z}) \quad \rightarrow \quad f \mid \tau(\boldsymbol{w}) + (\sigma(\boldsymbol{z}) \bmod f)$$

Overview of our procedure

Input: A quantifier-free formula $\varphi(\boldsymbol{x}, \boldsymbol{z})$ from the extended language of 1PPA (1PPA$^+$).
Output: A quantifier-free formula ψ from 1PPA$^+$ that is equivalent to $\exists \boldsymbol{x}\, \varphi$.

Step I. Preprocessing: Remove divisions and remainder functions.

Step II. Bounded quantifier elimination: compute $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

Step III. Remove the divisibility relations.

$$f \mid \tau(\boldsymbol{w}) + \sigma(\boldsymbol{z}) \quad \rightarrow \quad f \mid \tau(\boldsymbol{w}) + (\sigma(\boldsymbol{z}) \bmod f)$$

Bounded!

Overview of our procedure

Input: A quantifier-free formula $\varphi(\boldsymbol{x}, \boldsymbol{z})$ from the extended language of 1PPA (1PPA$^+$).
Output: A quantifier-free formula ψ from 1PPA$^+$ that is equivalent to $\exists \boldsymbol{x} \varphi$.

Step I. Preprocessing: Remove divisions and remainder functions.

Step II. Bounded quantifier elimination: compute $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

Step III. Remove the divisibility relations.

$$\begin{aligned} f \mid \tau(\boldsymbol{w}) + \sigma(\boldsymbol{z}) \quad &\to \quad f \mid \tau(\boldsymbol{w}) + (\sigma(\boldsymbol{z}) \bmod f) \\ &\to \quad \exists w \leq p(t) \colon f \cdot w = \tau(\boldsymbol{w}) + (\sigma(\boldsymbol{z}) \bmod f) \end{aligned}$$

Overview of our procedure

Input: A quantifier-free formula $\varphi(\boldsymbol{x}, \boldsymbol{z})$ from the extended language of 1PPA (1PPA$^+$).
Output: A quantifier-free formula ψ from 1PPA$^+$ that is equivalent to $\exists \boldsymbol{x}\, \varphi$.

Step I. Preprocessing: Remove divisions and remainder functions.

Step II. Bounded quantifier elimination: compute $\exists \boldsymbol{w} \leq B : \gamma(\boldsymbol{w}, \boldsymbol{z})$

Step III. Remove the divisibility relations.

$$\begin{aligned} f \mid \tau(\boldsymbol{w}) + \sigma(\boldsymbol{z}) \quad &\rightarrow \quad f \mid \tau(\boldsymbol{w}) + (\sigma(\boldsymbol{z}) \bmod f) \\ &\rightarrow \quad \exists w \leq p(t) : f \cdot w = \tau(\boldsymbol{w}) + (\sigma(\boldsymbol{z}) \bmod f) \end{aligned}$$

Step IV. Elimination of bounded quantifiers by "bit blasting".

Step IV: Elimination of bounded quantifiers

$$\exists x \leq t^2 + t - 1 \ \exists z \leq t + 2 \ : \ (t+1) \cdot z = x + (-b \bmod t+1)$$

Assume $t \geq 2$.

Step IV: Elimination of bounded quantifiers

$$\exists x \leq t^2 + t - 1 \; \exists z \leq t + 2 \; : \; (t+1) \cdot z = x + (-b \bmod t + 1)$$

Assume $t \geq 2$. Bit blast:

$$\exists z \leq t + 2 \colon \varphi \quad \rightarrow \quad \exists z_0, z_1, z_2 \leq t - 1 \colon \quad 0 \leq z_2 \cdot t^2 + z_1 \cdot t + z_0 \leq t + 2$$
$$\wedge \, \varphi[z_2 \cdot t^2 + z_1 \cdot t + z_0 \,/\, z]$$

Step IV: Elimination of bounded quantifiers

$$\exists x \leq t^2 + t - 1 \; \exists z \leq t + 2 \; : \; (t+1) \cdot z = x + (-b \bmod t+1)$$

Assume $t \geq 2$. Bit blast:

$$\exists z \leq t+2 \colon \varphi \quad \rightarrow \quad \exists z_0, z_1, z_2 \leq t-1 \colon \quad 0 \leq z_2 \cdot t^2 + z_1 \cdot t + z_0 \leq t+2$$
$$\wedge\, \varphi[z_2 \cdot t^2 + z_1 \cdot t + z_0 \,/\, z]$$

The equality $(t+1) \cdot z = x - (b \bmod t+1)$ becomes:

$$(t+1) \cdot (z_2 \cdot t^2 + z_1 \cdot t + z_0) = (x_2 \cdot t^2 + x_1 \cdot t + x_0) + (-b \bmod t+1).$$

Step IV: Elimination of bounded quantifiers

$$\exists x \leq t^2 + t - 1 \; \exists z \leq t + 2 \; : \; (t+1) \cdot z = x + (-b \bmod t+1)$$

Assume $t \geq 2$. Bit blast:

$$\exists z \leq t + 2 \colon \varphi \quad \rightarrow \quad \exists z_0, z_1, z_2 \leq t - 1 \colon \quad 0 \leq z_2 \cdot t^2 + z_1 \cdot t + z_0 \leq t + 2$$
$$\land \varphi[z_2 \cdot t^2 + z_1 \cdot t + z_0 \,/\, z]$$

The equality $(t+1) \cdot z = x - (b \bmod t+1)$ becomes:

$$-z_2 \cdot t^3 + (x_2 - z_1 - z_2) \cdot t^2 + (x_1 - z_0 - z_1) \cdot t + (x_0 - z_0) + (-b \bmod t+1) = 0.$$

Step IV: Elimination of bounded quantifiers

$$\exists x \leq t^2 + t - 1 \; \exists z \leq t + 2 \; : \; (t+1) \cdot z = x + (-b \bmod t+1)$$

Assume $t \geq 2$. Bit blast:

$$\exists z \leq t + 2 \colon \varphi \quad \to \quad \exists z_0, z_1, z_2 \leq t - 1 \colon \quad 0 \leq z_2 \cdot t^2 + z_1 \cdot t + z_0 \leq t + 2$$
$$\land \, \varphi[z_2 \cdot t^2 + z_1 \cdot t + z_0 \, / \, z]$$

The equality $(t+1) \cdot z = x - (b \bmod t+1)$ becomes:

$$-z_2 \cdot t^3 + (x_2 - z_1 - z_2) \cdot t^2 + (x_1 - z_0 - z_1) \cdot t + (x_0 - z_0) + (-b \bmod t+1) = 0.$$

Divide by t the maximal subterm with no quantified variables:

$$(-b \bmod t+1) \quad \to \quad \left\lfloor \tfrac{-b \bmod t+1}{t} \right\rfloor \cdot t + \big((-b \bmod t+1) \bmod t\big)$$

Step IV: Elimination of bounded quantifiers

$$\begin{aligned}-z_2 \cdot t^3 + (x_2 - z_1 - z_2) \cdot t^2 + \left(x_1 - z_0 - z_1 + \left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor\right) \cdot t \\ + (x_0 - z_0) + \big((-b \bmod t + 1) \bmod t\big) = 0\end{aligned}$$

Step IV: Elimination of bounded quantifiers

$$-z_2 \cdot t^3 + (x_2 - z_1 - z_2) \cdot t^2 + \left(x_1 - z_0 - z_1 + \left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor\right) \cdot t \\ + (x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big) = 0$$

- $(x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big)$ belongs to $[-t..2 \cdot t]$...
- ...and must be divisibile by t. (This only applies to equalities.)

Step IV: Elimination of bounded quantifiers

$$-z_2 \cdot t^3 + (x_2 - z_1 - z_2) \cdot t^2 + \left(x_1 - z_0 - z_1 + \left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor\right) \cdot t \\ + (x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big) = 0$$

- $(x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big)$ belongs to $[-t..2 \cdot t]$...
- ...and must be divisibile by t. (This only applies to equalities.)

Guess $r_0 \in \{-1, 0, 1, 2\}$ and rewrite the equality as

$$-z_2 \cdot t^2 + (x_2 - z_1 - z_2) \cdot t + \left(x_1 - z_0 - z_1 + \left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor\right) + r_0 = 0$$

$$\wedge \ (x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big) = r_0 \cdot t$$

Important: x_0 has only integer coefficients!

Step IV: Elimination of bounded quantifiers

Let's do another iteration:

$$
\begin{aligned}
&-z_2 \cdot t^2 + (x_2 - z_1 - z_2) \cdot t + \left(x_1 - z_0 - z_1 + \left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor\right) + r_0 = 0 \\
\wedge\ &(x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big) = r_0 \cdot t
\end{aligned}
$$

Step IV: Elimination of bounded quantifiers

Let's do another iteration:

$$-z_2 \cdot t^2 + (x_2 - z_1 - z_2) \cdot t + \left(x_1 - z_0 - z_1 + \left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor\right) + r_0 = 0$$

$$\wedge\ (x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big) = r_0 \cdot t$$

Divide by t!

Step IV: Elimination of bounded quantifiers

Let's do another iteration:

$$-z_2 \cdot t^2 + (x_2 - z_1 - z_2) \cdot t + \left(x_1 - z_0 - z_1 + \left\lfloor \frac{\left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor + r_0}{t} \right\rfloor \cdot t + \left(\left(\left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor + r_0 \right) \bmod t \right) \right) = 0$$

$$\wedge \ (x_0 - z_0) + \big((-b \bmod t+1) \bmod t \big) = r_0 \cdot t$$

Step IV: Elimination of bounded quantifiers

Let's do another iteration:

$$-z_2 \cdot t^2 + (x_2 - z_1 - z_2) \cdot t + \left(x_1 - z_0 - z_1 + \left\lfloor \frac{\left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor + r_0}{t} \right\rfloor \cdot t + \left(\left(\left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor + r_0 \right) \bmod t \right) \right) = 0$$

$$\wedge\ (x_0 - z_0) + \big((-b \bmod t + 1) \bmod t \big) = r_0 \cdot t$$

belongs to $[-2 \cdot t..2 \cdot t]$
so guess $r_1 \in [-2..2]$

Step IV: Elimination of bounded quantifiers

Let's do another iteration:

$$\begin{aligned} & -z_2 \cdot t + (x_2 - z_1 - z_2) + \left\lfloor \frac{\left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor + r_0}{t} \right\rfloor + r_1 = 0 \\ \wedge\ & (x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big) = r_0 \cdot t \\ \wedge\ & (x_1 - z_0 - z_1) + \left(\left(\left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor + r_0\right) \bmod t\right) = r_1 \cdot t \end{aligned}$$

Step IV: Elimination of bounded quantifiers

Let's do another iteration:

$$
\begin{aligned}
& - z_2 \cdot t + (x_2 - z_1 - z_2) + \left\lfloor \frac{\left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor + r_0}{t} \right\rfloor + r_1 = 0 \\
\land\ & (x_0 - z_0) + \big((-b \bmod t+1) \bmod t\big) = r_0 \cdot t \\
\land\ & (x_1 - z_0 - z_1) + \left(\left(\left\lfloor \frac{-b \bmod t+1}{t} \right\rfloor + r_0\right) \bmod t\right) = r_1 \cdot t
\end{aligned}
$$

Now all variables but z_2 have only integer coefficients!

- Repeat until all quantified variables only occur with integer coefficients.
- Afterwards, call a quantifier-elimination procedure for Presburger arithmetic.

Our results

Theorem

There is a quantifier elimination procedure for the extension of 1PPA *with the functions:*

- *integer division:* $x \mapsto \left\lfloor \frac{x}{t^d} \right\rfloor$ *one function for each* $d \in \mathbb{N}$, *assumes* $t \neq 0$
- *integer remainder function:* $x \mapsto (x \bmod p)$ *for each* $p \in \mathbb{Z}[t]$
- *divisibility relation:* $p \mid x$ *for each* $p \in \mathbb{Z}[t]$

Theorem

For the class of all existential formulae of 1PPA, *the following holds:*

Satisfiability:	*Universality:*	*Finiteness:*
NP-*complete*	**coNEXP**-*complete*	**coNP**-*complete*