Computing over Integer Linear-Exponential Straight-Line Programs (**ILESLP**)

S Hitarth

?

Alessio Mansutti

IMDEA Software Institute, Madrid

Gurupreran Shabadi

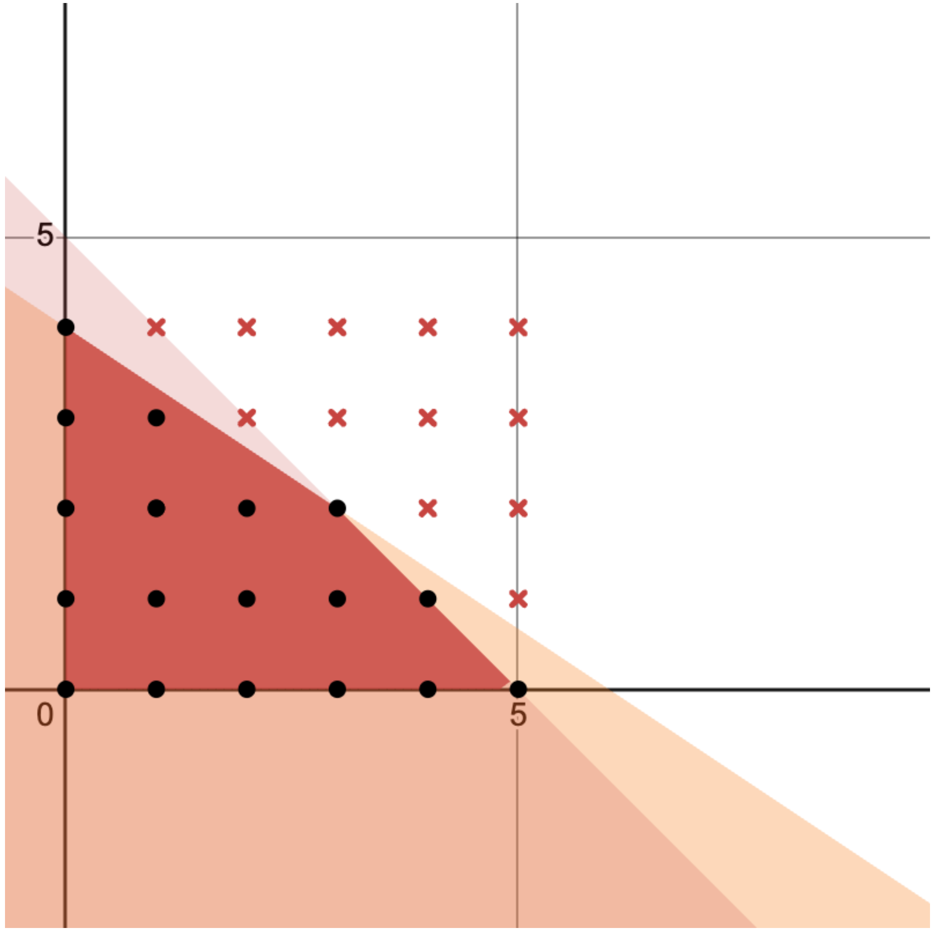
University of Pennsylvania

Integer Linear Programming (ILP)

$$\max\ (x - y)$$

$$2x + 3y \leq 12$$
$$x + y \leq 5$$
$$x, y \geq 0$$
$$x, y \in \mathbb{Z}$$

Feasibility is NP-Complete



Integer Linear-Exponential Programming (ILEP)

ILP = Set of Linear Constraints $\sum_i a_i x_i \leq b$

\+ Exponential Constraints $x = 2^y$

\+ Remainder Constraints $x = y \bmod 2^z$ $(7 \bmod 2^2) = 3$

$\max\ (x - y)$

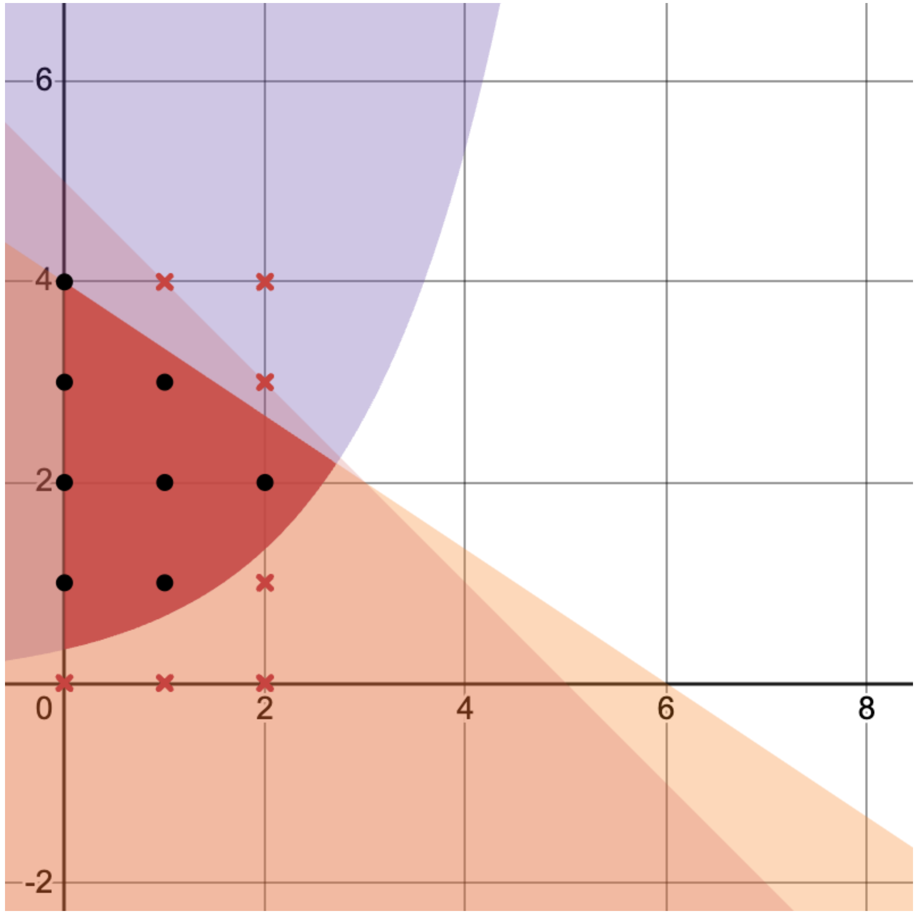
$3y \geq 2^x$

$2x + 3y \leq 12$

$x + y \leq 5$

$x, y \geq 0$

$x, y \in \mathbb{Z}$



Feasibility of Integer Linear-Exponential Programs

What is a solution to this?

$$x_4 \geq 2^{x_3}$$
$$x_3 \geq 2^{x_2}$$
$$x_2 \geq 2^{x_1}$$
$$x_1 \geq 2^{x_0}$$
$$x_0 \geq 2$$
$$x_i \in \mathbb{Z}$$

$$x_4 = 2^{2^{2^{2}}}$$

$$x_k = \underbrace{2^{2^{2^{\cdot^{\cdot^{2}}}}}}_{k}$$

Do ILEP have *nice* and *small* representation for their solutions?

~~If a solution exists,~~

~~there is *small* one, too.~~

Chistikov, Mansutti, Starchak [CMS, ICALP 2024] showed that

Feasibility of ILEP is in NP!

i.e., there is a small PTIME checkable certificate for a feasible ILEP

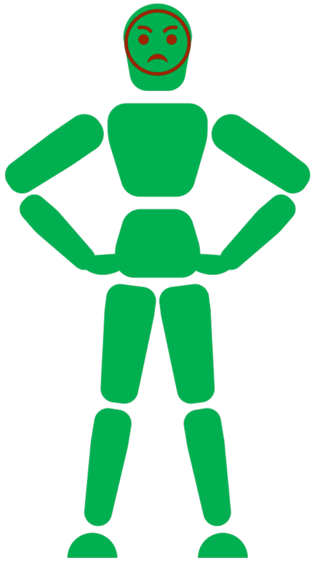
Philosophical Detour: What is a *nice* representation?

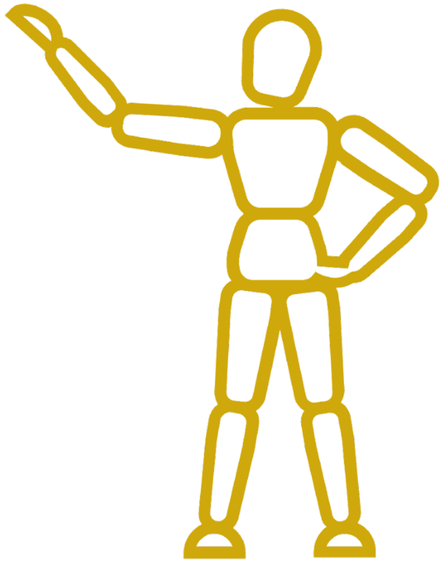
I want a solution v
for the problem:

$$A\vec{x} \le \boldsymbol{b}$$
$$\vec{x} \in \mathbb{Z}^n$$

An **explicit** solution in binary…

A **solution** is
“Choose a vector $\vec{x}$ such that
$$A\vec{x} \le \boldsymbol{b}$$
$$\vec{x} \in \mathbb{Z}^n$$
”

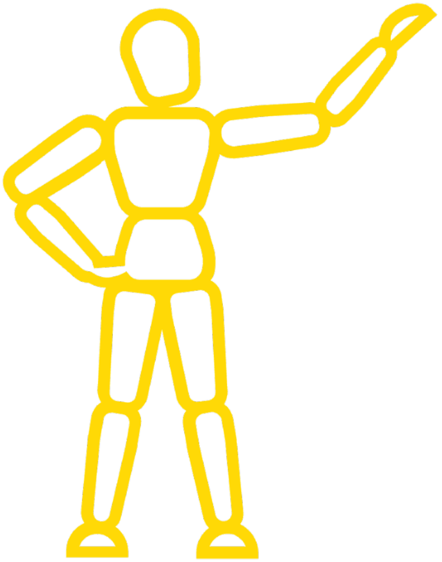




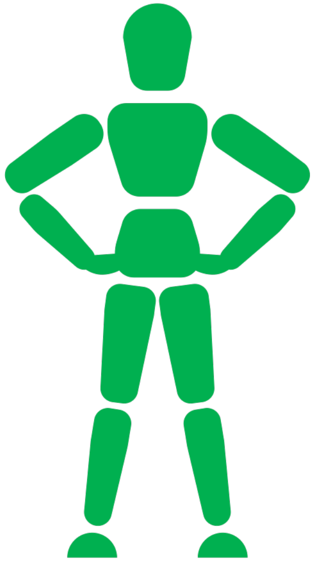
Philosophical Detour: What is a *nice* representation?

Here's a solution $\vec{x} = (1,100,101)$.

But why did you want a binary representation?



...as I can check if this is indeed a solution and perform ***computations*** on it.



What is a *nice* representation for solutions to ILEP?

1. Can we *perform arithmetic operations* (addition, subtraction) on this representation?
2. Can we *compare* two given values in this representation?
3. Can we *plugin* the given solution and *efficiently check* if it correct?

Optimal solutions of ILEP have succinct representations

If an instance of ILEP has **an (optimal) solution**, then it has one **representable** with a **polynomial-size ILESLP**.

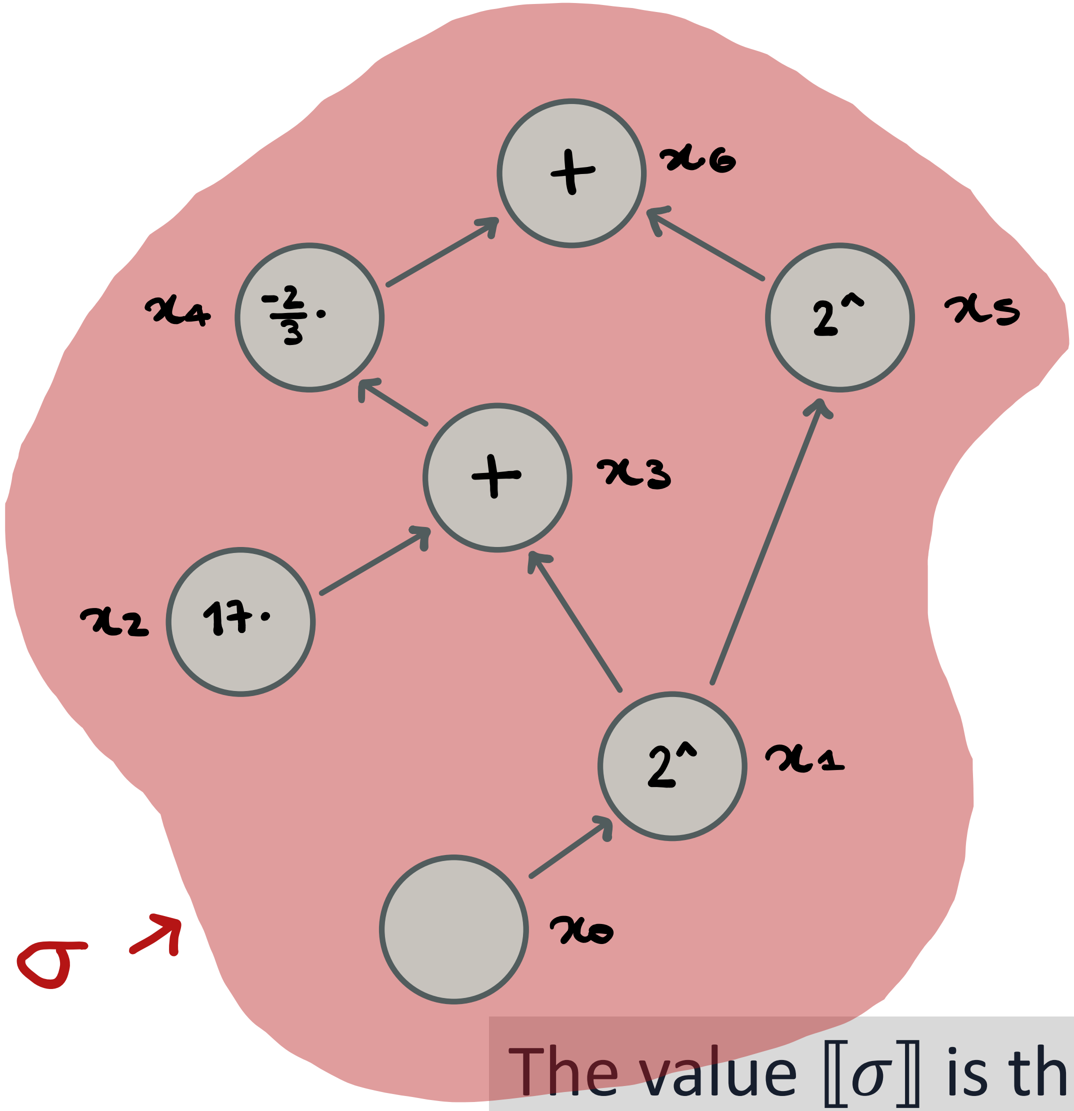
… and this representation is *nice* for computations.

The New Representation

Integer Linear-Exponential Straight-Line Programs (**ILESLP**)

$$x_6 \leftarrow x_4 + x_5$$
$$x_5 \leftarrow 2^{x_1}$$
$$x_4 \leftarrow \frac{-2}{3} \cdot x_3$$
$$x_3 \leftarrow x_1 + x_2$$
$$x_2 \leftarrow 17 \cdot x_1$$
$$x_1 \leftarrow 2^{x_0}$$
$$x_0 \leftarrow 0$$

$x_i \in \mathbb{Z}$



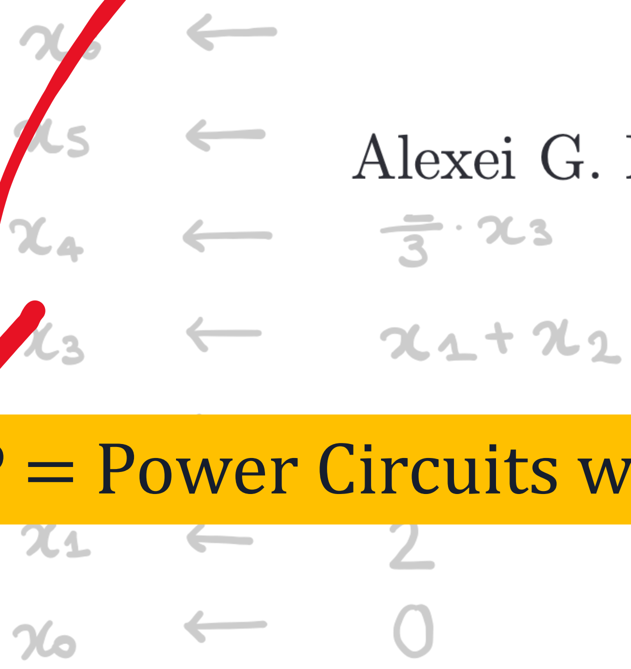
The value $[\![\sigma]\!]$ is the value assigned to the top variable

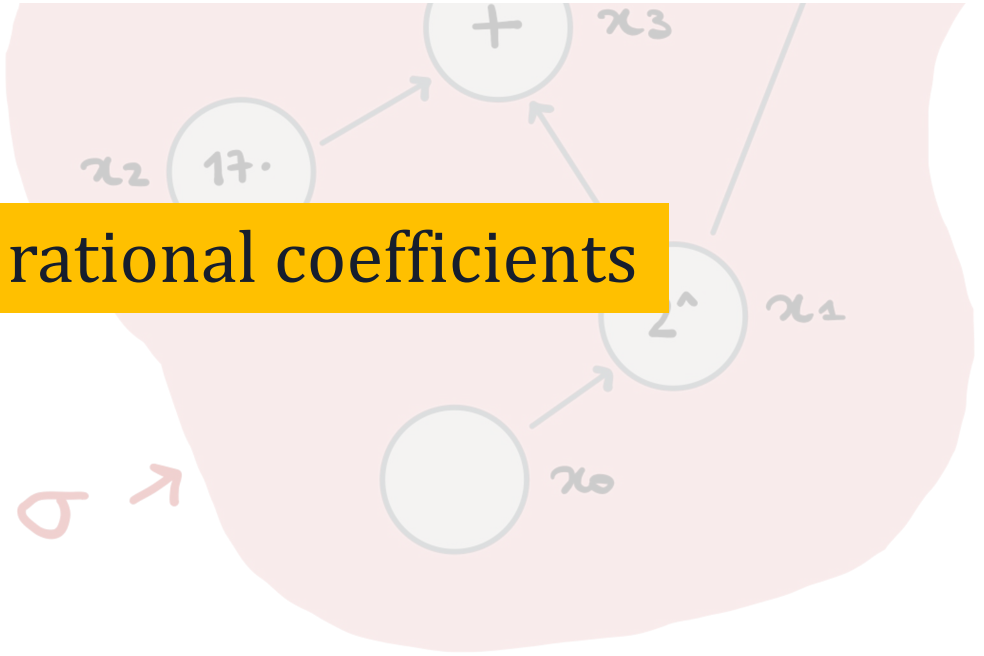
The New Representation

Integer Linear-Exponential Straight-Line Programs (**ILESLP**)

Power Circuits, Exponential Algebra, and Time Complexity

Alexei G. Myasnikov, Alexander Ushakov, and Dong Wook Won





ILESLP = Power Circuits with rational coefficients

Is ILESLP a *nice* representation?

ILESLP is well-defined?

Integrality: Does every variable of σ evaluate to an integer?

Perform arithmetic operations?

Addition/Subtraction: Compute σ such that $[\![\sigma]\!] = [\![\sigma_1]\!] \pm [\![\sigma_2]\!]$? Trivially P

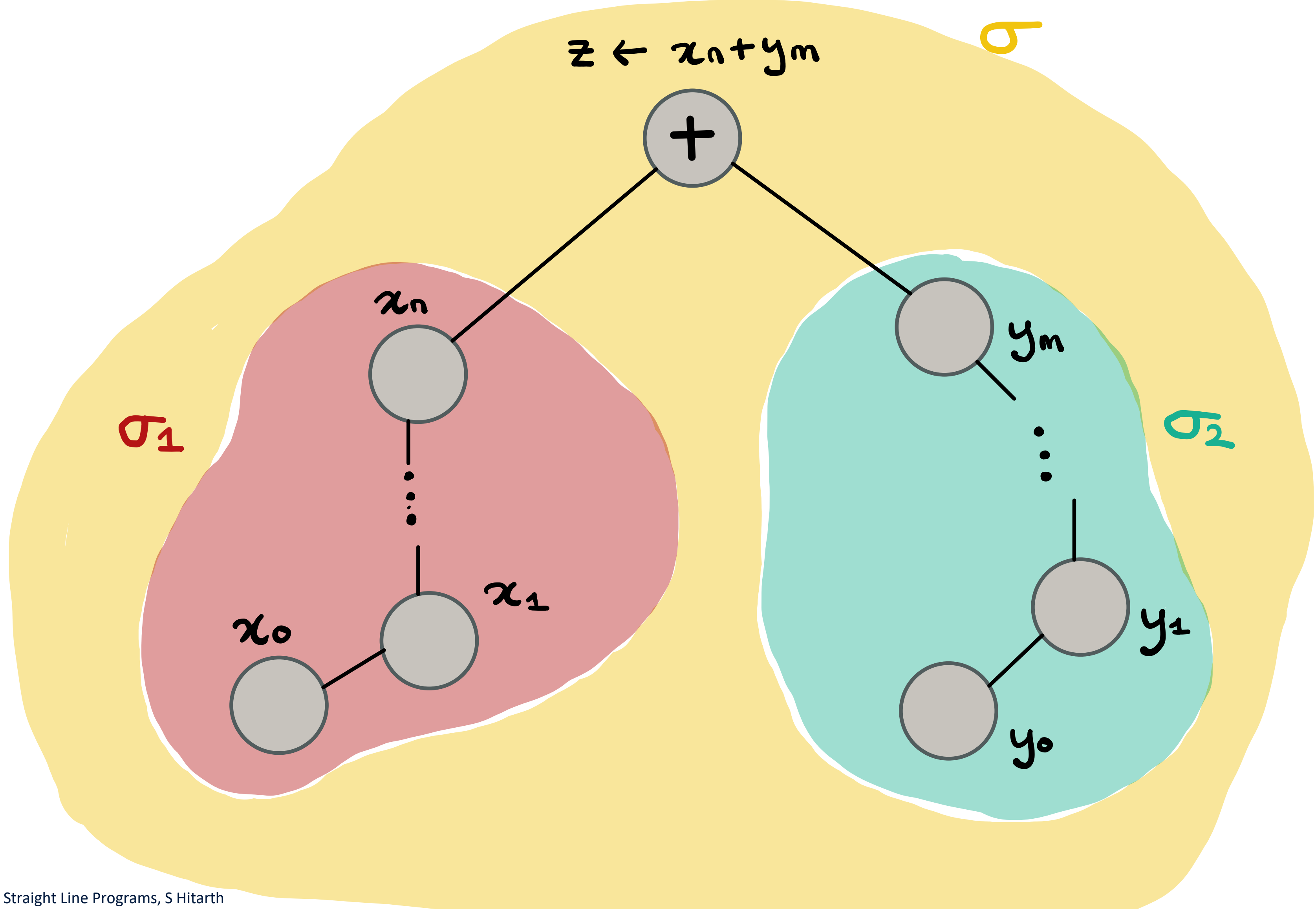
Mod: Compute σ such that $[\![\sigma]\!] = [\![\sigma_1]\!]\ mod\ 2^{[\![\sigma_2]\!]}$?

Compare two ILESLPs?

Positivity: $[\![\sigma]\!] \geq 0$? Equivalent to $[\![\sigma_1]\!] \geq [\![\sigma_2]\!]$?

Addition/Substraction: ILESLP

Addition/Subtraction: Compute σ such that $[\![\sigma]\!] = [\![\sigma_1]\!] \pm [\![\sigma_2]\!]$?



Is ILESLP a *good* representation?

ILESLP is well-defined?

Integrality: Does every variable of σ evaluate to an integer? $P^{FACTORING}$

Perform arithmetic operations?

Addition/Subtraction: Compute σ such that $[\![\sigma]\!] = [\![\sigma_1]\!] \pm [\![\sigma_2]\!]$? P

Mod: Compute σ such that $[\![\sigma]\!] = [\![\sigma_1]\!] \; mod \; 2^{[\![\sigma_2]\!]}$? $P^{FACTORING}$

Compare two ILESLPs?

Positivity: $[\![\sigma]\!] \geq 0$? Equivalent to $[\![\sigma_1]\!] \geq [\![\sigma_2]\!]$? P

Positivity: $[\![\sigma]\!] \geq 0$? Equivalent to $[\![\sigma_1]\!] \geq [\![\sigma_2]\!]$?

$x_8 \leftarrow x_7 + x_6$
$x_7 \leftarrow 2^{x_3}$
$x_6 \leftarrow x_5 + x_4$
$x_5 \leftarrow -\frac{5}{6} \cdot 2^{x_2}$
$x_4 \leftarrow -\frac{1}{6} \cdot 2^{x_1}$
...

$6 \cdot x_8 \quad \leftarrow 6 \cdot 2^{x_3} - 5 \cdot 2^{x_2} - 2^{x_1}$
...

Positivity: $[\![\sigma]\!] \geq 0$? Equivalent to $[\![\sigma_1]\!] \geq [\![\sigma_2]\!]$?

$$\begin{aligned}
x_n &\leftarrow \rho_n \\
x_{n-1} &\leftarrow \rho_{n-1} \\
x_{n-2} &\leftarrow \rho_{n-2} \\
&\vdots \\
x_1 &\leftarrow \rho_1 \\
x_0 &\leftarrow 0
\end{aligned}$$

Simplify →

$$Lx_i \leftarrow \sum_j a_{ij} 2^{x_j}$$

$$a_{ij} \in \mathbb{Z}$$

$$L \in \mathbb{Z}$$

L is LCM of all denominators appearing in the SLP

Close-Far Trick

$x \geq y \geq z$ $\qquad \tau := a \cdot 2^x + b \cdot 2^y + c \cdot 2^z$ $\qquad \lceil \log_2(|b| + |c|) \rceil = C$

Case 1. x and y are far: $x > y + C$

Lower Bound if $a \geq 1$

$$\tau \geq a \cdot 2^x - |b| \cdot 2^y - |c| \cdot 2^y$$
$$\geq 2^x - (|b| + |c|) \cdot 2^y$$
$$\geq 2^x - 2^{y+C}$$
$$> \mathbf{0}$$

$\tau > 0$

Upper Bound if $a \leq -1$

$$\tau \leq a \cdot 2^x + |b| \cdot 2^y + |c| \cdot 2^y$$
$$\leq -2^x + (|b| + |c|) \cdot 2^y$$
$$\geq -2^x + 2^{y+C}$$
$$< \mathbf{0}$$

$\tau < 0$

Close-Far Trick

$x \geq y \geq z$ $\tau := a \cdot 2^x + b \cdot 2^y + c \cdot 2^z$ $\lceil \log_2(|b| + |c|) \rceil = C$

Case 2. x and y are close: $x = y + d_{xy}$, where $d_{xy} \in [0, C]$

Reduce to Simpler Problem

$$\tau := \left(a \cdot 2^{d_{xy}} + b\right) \cdot 2^y + c \cdot 2^z$$

Iterate over the remaining variables…

$\tau := some\ small\ number$ …

Close-Far Trick (an example)

$x \geq y \geq z$ $\quad\quad \tau := a \cdot 2^x - 5 \cdot 2^y - 3 \cdot 2^z \quad\quad a \neq 0$

$$\tau := a \cdot 2^x - 5 \cdot 2^y - 3 \cdot 2^z$$
$$\geq a \cdot 2^x - (5+3) \cdot 2^y$$
$$\geq a \cdot 2^x - 2^{y+3}$$

Case 1. x and y are far if $x > y + 3$ then $sign(\tau) = sign(a)$

Case 2. x and y are close if $x = y + c$ then $\tau = 2^y \cdot (2^c - 2^3)$

Positivity: $[\![\sigma]\!] \geq 0$? Equivalent to $[\![\sigma_1]\!] \geq [\![\sigma_2]\!]$?

	x_0	x_1	x_2	x_3	…	x_{n-1}	x_n
x_0	0						
x_1	d_{10}	0					
x_2	d_{20}	$< C$	0				
x_3	$> C$	d_{31}	d_{32}	0			
⋮	…	…	…	…	0		
x_{n-1}						0	
x_n							0

$$Lx_i \leftarrow \sum_j a_{ij} 2^{x_j}$$

$$Lx_k \leftarrow \sum_j a_{kj} 2^{x_j}$$

$$L(x_k - x_i) = \sum_j (a_{kj} - a_{ij}) 2^{x_j}$$

$$\tau := a \cdot 2^x + b \cdot 2^y + c \cdot 2^z$$

Positivity: $[\![\sigma]\!] \geq 0$? Equivalent to $[\![\sigma_1]\!] \geq [\![\sigma_2]\!]$?

	x_0	x_1	x_2	x_3	…	x_{n-1}	x_n
x_0	0						
x_1	d_{10}	0					
x_2	d_{20}	$< C$	0				
x_3	$> C$	d_{31}	d_{32}	0			
⋮	…	…	…	…	0		
x_{n-1}	…	…	…	…	…	0	
x_n	?	…	…	…	…	…	0

Check the entry $(n, 0)$!

ILP vs ILEP: Solutions encoded in binary

	Integer Linear Programs	Integer Linear-Exponential Programs
Feasible?	NP	*NP
Exists a *small* binary solution?	Yes	No
Exists a *small* binary optimal?	Yes	No

*[CMS,2024]

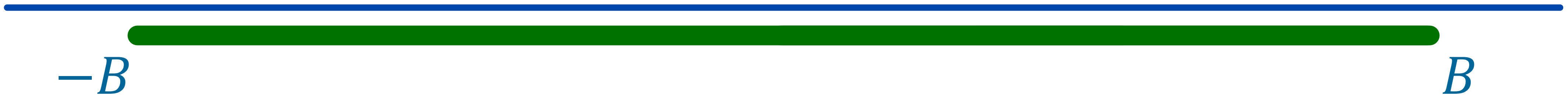
ILP vs ILEP: Solutions encoded in *compressed circuit*

	Integer Linear Programs	Integer Linear-Exponential Programs
Feasible?	NP	*NP
Exists a *small* circuit solution?	Yes	Yes
Exists a *small* circuit optimal?	Yes	Yes

Open Problem #1. Binary Search over ILESLP

$\max z$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $z \in [-B, B]$

$$\varphi := \boldsymbol{A}\vec{x} \leq \boldsymbol{b}$$
$$\vec{x} \in \mathbb{Z}^n$$



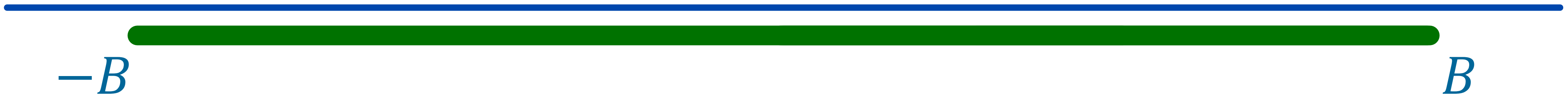
In ILP, we need polynomially many calls to a feasibility oracle to find an optimal.

Open Problem #1. Binary Search over ILESLP

$\max z$

$ILEP\ Program$

$z \in [-B, B]$

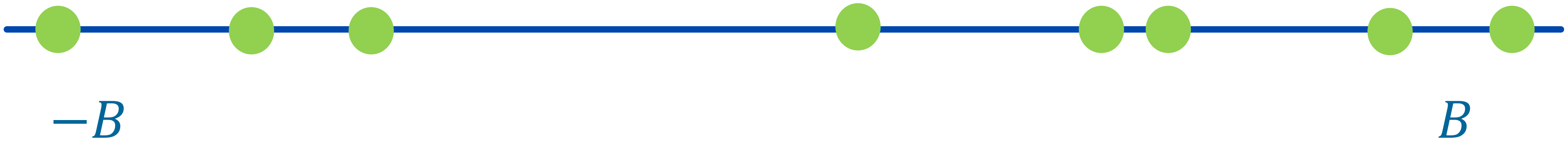


But B could be non-elementary in size of input!

Can we perform ***binary search*** for optimization in Integer Linear-Exponential Programs?

Open Problem #1. Binary Search over ILESLP

Restrict to a fixed class C of ILESLP [less than k gates]



Is there an efficient algorithm that

"Given as input $\sigma_1, \sigma_3 \in C$, finds an $\sigma_2 \in C$ such that σ_3 *is roughly between* σ_1 and σ_3."?

i.e., $|\{\sigma \in C \mid \sigma_1 \leq \sigma \leq \sigma_2\}| \approx |\{\sigma \in C \mid \sigma_2 \leq \sigma \leq \sigma_3\}|$

Open Problem #2. Ranking ILESLP

Restrict to a fixed class C of ILESLP [less than k gates]

Find an efficiently computable function

$$f : C \to \mathbb{R}$$

such that $f(\sigma_1) \geq f(\sigma_2)$ iff $[\![\sigma_1]\!] \geq [\![\sigma_2]\!]$.

i.e., f assigns a rank/order to each ILESLP.

Open Problem #3. Normalization of ILESLP

Myasnikov et al. give a normalization algorithm for power circuits…

Is there an efficient normalization procedure for ILESLP?

Open Problem #4. ILESLP with 2 bases over integers

$$x_i \leftarrow \sum_j a_{ij} 2^{x_j} + b_{ij} 3^{x_j}$$

On the Decidability of
Presburger Arithmetic Expanded with Powers

Toghrul Karimov* Florian Luca† Joris Nieuwveld‡ Joël Ouaknine§ James Worrell¶

The existential theory of $\langle \mathbb{N}; , 0,1, <, +, x \mapsto \alpha^x, x \mapsto \beta^x \rangle$
is hard to decide!

Can we efficiently solve the positivity problem over ILESLP with 2 bases?

Conclusion

ILESLP are *nice* arithmetic circuits with gates $+, r\ \cdot$, and 2 .

Open Problems

- Can we perform ***binary search*** over a given class of ILESLP?
- Find an **efficiently computable** function $f: C \to \mathbb{R}$ such that $f(\sigma_1) \geq f(\sigma_2)$ iff $[\![\sigma_1]\!] \geq [\![\sigma_2]\!]$.
- Is there an efficient **normalization procedure** for ILESLP?
- Can we efficiently solve the positivity problem over **ILESLP with 2 bases?**

Thanks!

Backup

Mod: Compute $[\![\sigma_1]\!] \bmod 2^{[\![\sigma_2]\!]}$?

$[\![\sigma_1]\!]$ $\quad x_n \leftarrow \frac{1}{L} \sum_j a_{nj} 2^{x_j}$

$[\![\sigma_2]\!]$ $\quad y_n \leftarrow \cdots$

$$x_n \leftarrow \frac{1}{L}(a_{n(n-1)} 2^{x_{n-1}} + a_{n(n-2)} 2^{x_{n-2}} + a_{n(n-3)} 2^{x_{n-3}} + \cdots + a_{n0} 2^{x_0}$$

Let $J = \{i \mid [\![\sigma_2]\!](y_n) \geq [\![\sigma_1]\!](x_{n-i})]\}$

$$U = \sum_{i \in [0, n-1] \setminus J} a_{nj} 2^{x_j} \qquad B = \sum_{i \in J} a_{nj} 2^{x_j}$$

Mod: Compute $[\![\sigma_1]\!] \bmod 2^{[\![\sigma_2]\!]}$?

$$U = \sum_{i \in [0,n-1] \setminus J} a_{nj} 2^{x_j} \qquad B = \sum_{i \in J} a_{nj} 2^{x_j}$$

$$x_n \leftarrow \frac{1}{L} U + \frac{1}{L} B$$

How big can B be? $$B \leq \left(\sum_{i \in J} a_{nj} \right) 2^{y_n} = A \cdot 2^{y_n}$$

Find a *small* q such that $0 \leq B - q \cdot 2^{y_n} \leq 2^{y_n}$

Mod: Compute $[\![\sigma_1]\!]$ mod $2^{[\![\sigma_2]\!]}$?

$$U = \sum_{i \in [0, n-1] \setminus J} a_{n_j} 2^{x_j} \qquad B = \sum_{i \in J} a_{n_j} 2^{x_j}$$

$$x_n = \frac{1}{L}(U + q \cdot 2^{y_n}) + \frac{1}{L}(B - q \cdot 2^{y_n})$$

$$x_n = \frac{1}{L}(U' + q) \cdot 2^{y_n} + \frac{1}{L}(B - q \cdot 2^{y_n})$$

Compute $r = (U' + q) \; mod \; L$

$$x_n = \frac{1}{L}(U + q \cdot 2^{y_n} - r \cdot 2^{y_n}) + \frac{1}{L}(B - q \cdot 2^{y_n} + r \cdot 2^{y_n})$$

Mod: Compute $[\![\sigma_1]\!]$ mod $2^{[\![\sigma_2]\!]}$?

$$x_n = \frac{1}{L}(U + q \cdot 2^{y_n} \; - r \cdot 2^{y_n}) + \frac{1}{L}(B \; - q \cdot 2^{y_n} + r \cdot 2^{y_n})$$

$$x_n = \left(\frac{U' + q \; - r}{L}\right) \cdot 2^{y_n} + \frac{1}{L}(B \; - q \cdot 2^{y_n} + r \cdot 2^{y_n})$$

$$x_n \bmod 2^{y_n} \equiv \frac{1}{L}(B \; - q \cdot 2^{y_n} + r \cdot 2^{y_n}) \bmod 2^{y_n}$$

$$x_n \bmod 2^{y_n} \equiv \frac{1}{L}(B \; - q \cdot 2^{y_n} + r \cdot 2^{y_n})$$