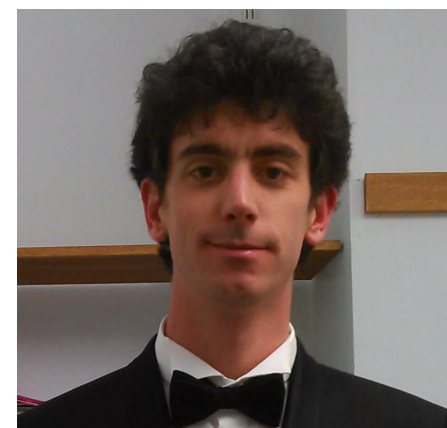


# Orbit and Group Closure



**Mahsa Shirmohammadi**  
IFIP WG 2.2, September 2025

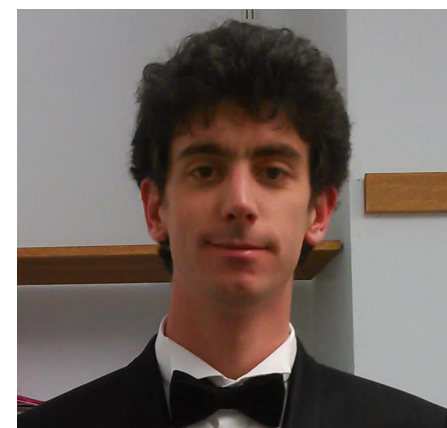


# Orbit and Group Closure

(Geometric) Complexity Theory  
Quantum Computation  
Automata and Series  
Program Analysis



**Mahsa Shirmohammadi**  
IFIP WG 2.2, September 2025



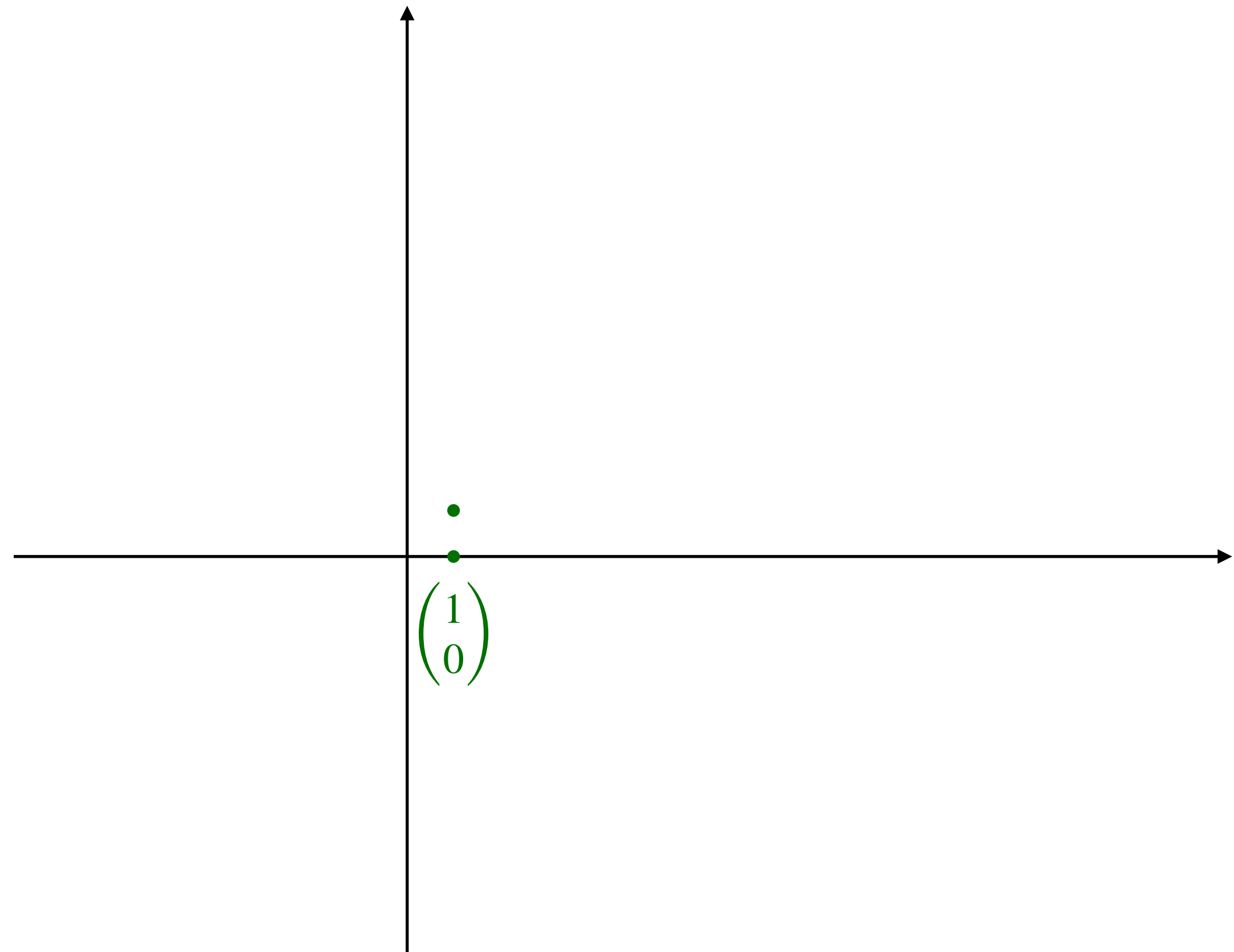
# Orbit

$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$

# Orbit

$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



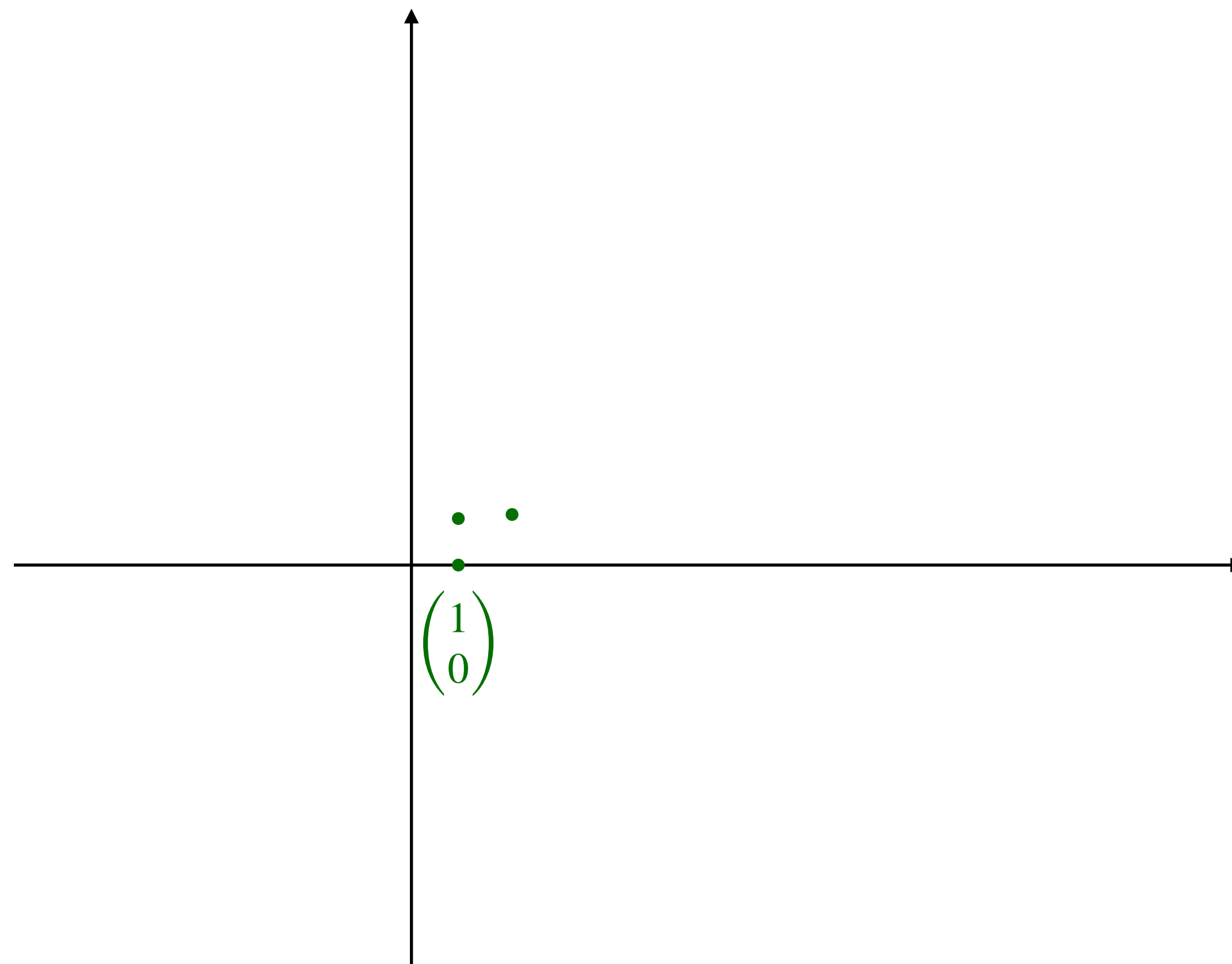


# Orbit

$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



# Orbit

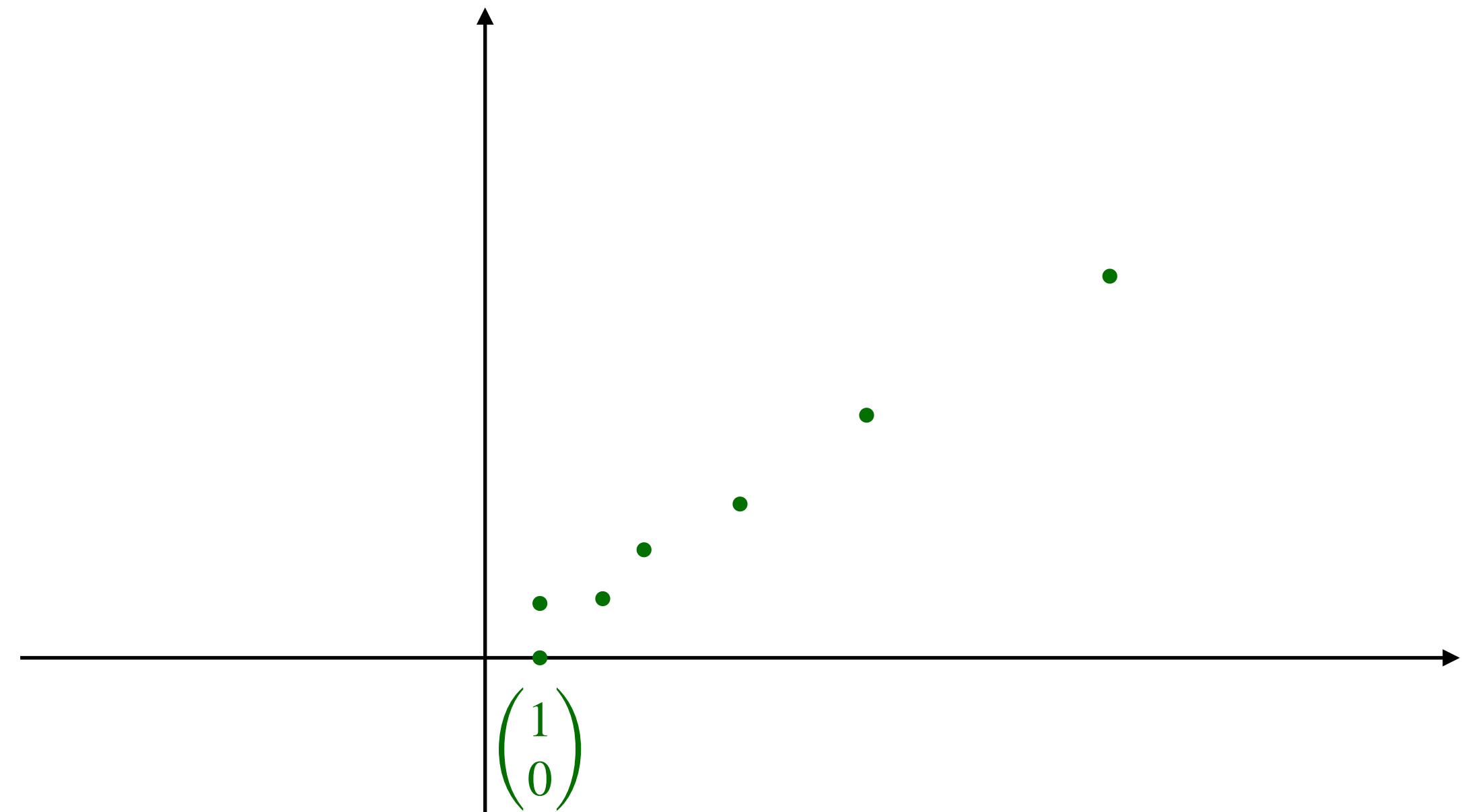
$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

⋮

$$\begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



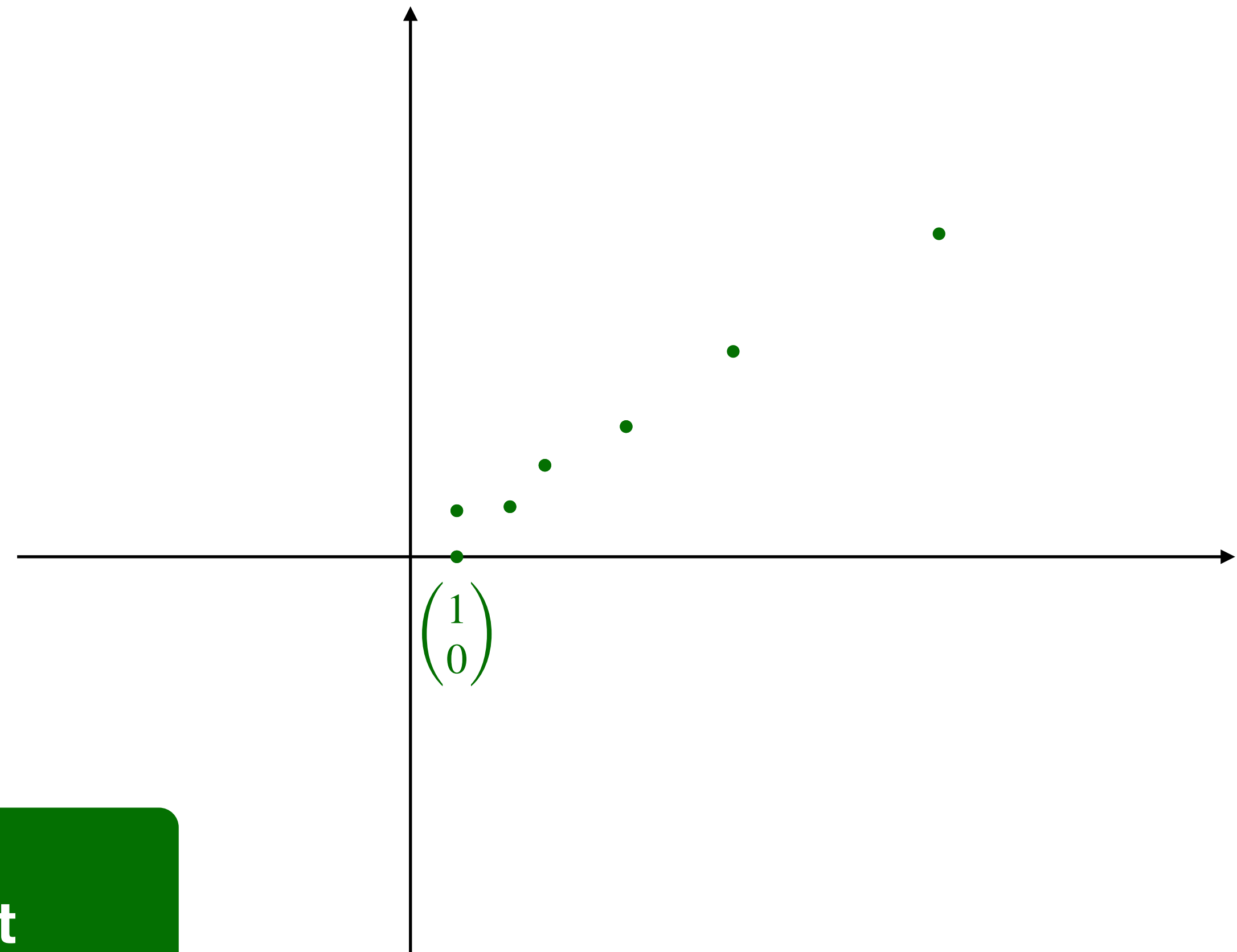
Fibonacci numbers

$$\begin{pmatrix} x+y \\ x \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



# Orbit-Closure

$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$



The orbit closure is the smallest set that contains the orbit and is closed in topology

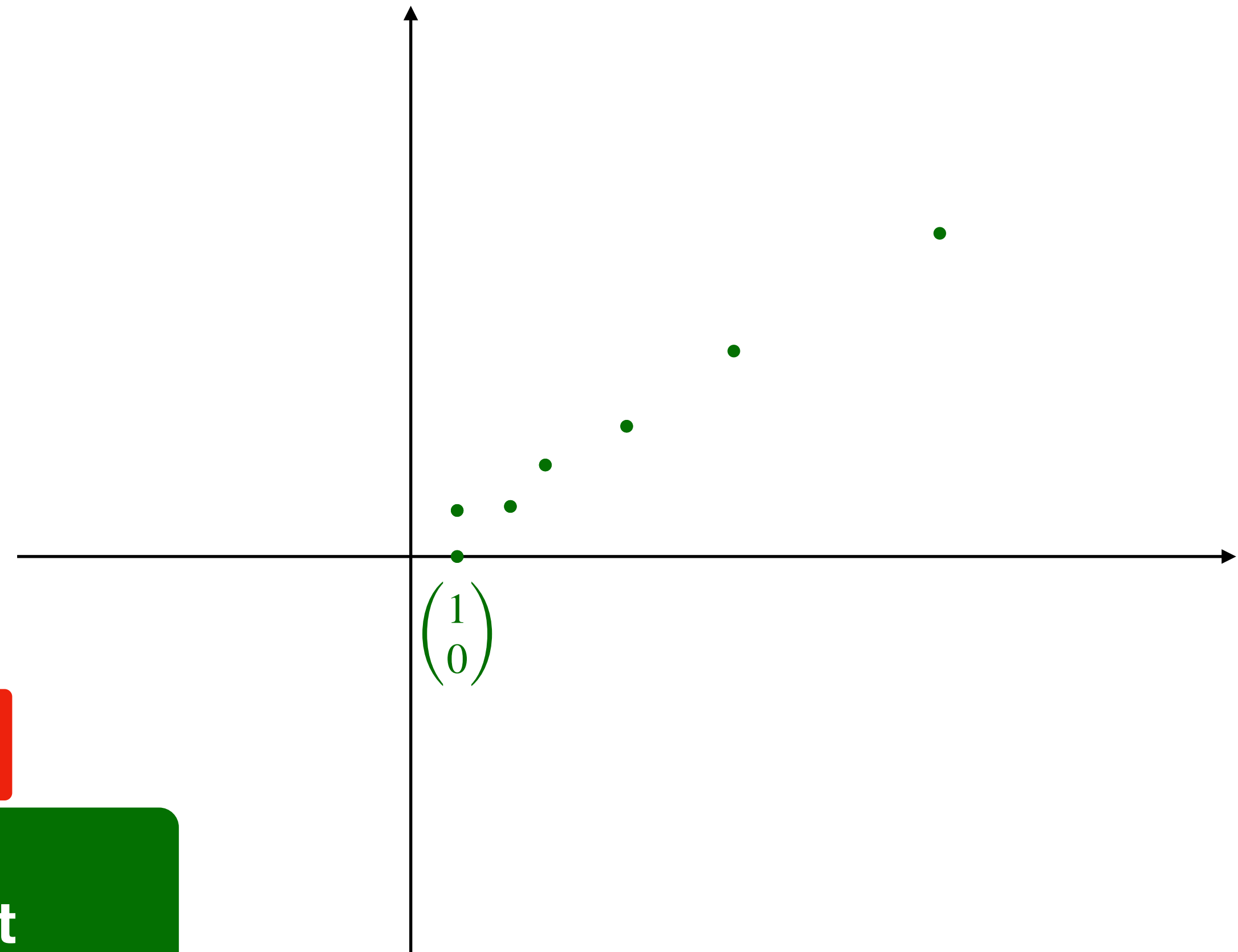
# Orbit-Closure

$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$

Zariski Topology:

Closed sets are algebraic!

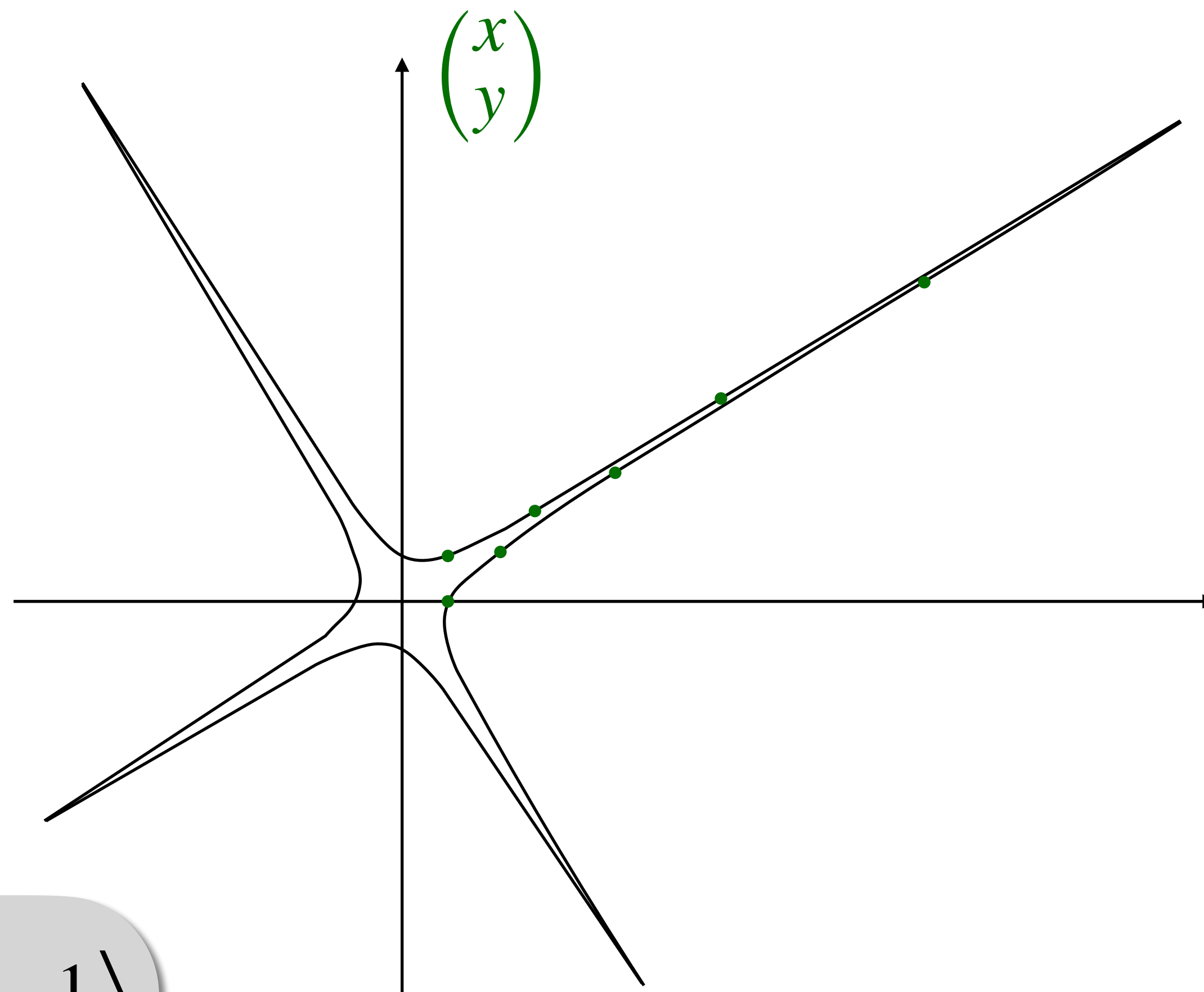
The orbit closure is the smallest set that contains the orbit and is closed in topology





# Orbit-Closure

$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$



$\overline{Gv}$

$$\langle x^4 + y^4 - 2x^3y - x^2y^2 + 2xy^3 - 1 \rangle$$

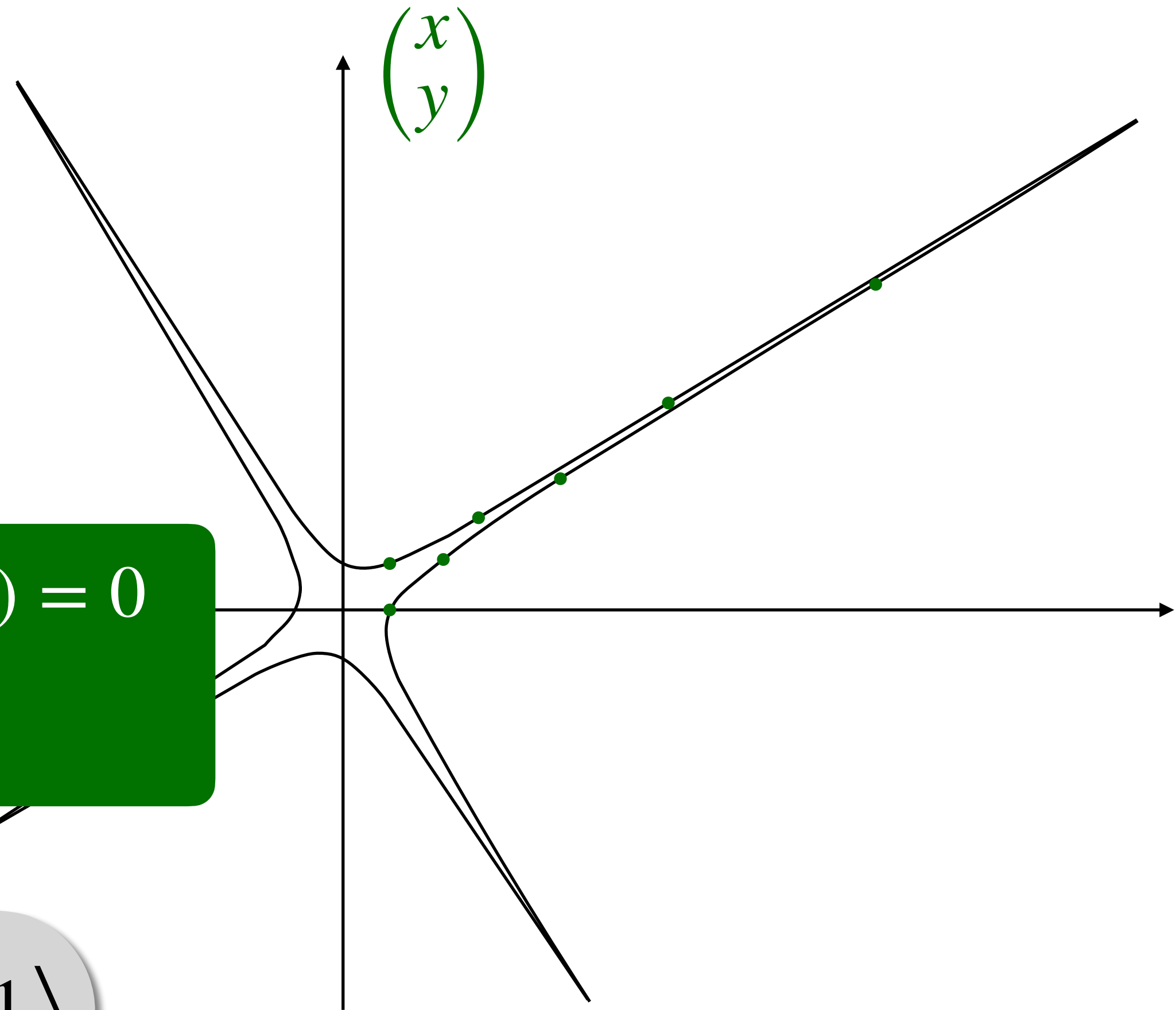
# Orbit-Closure

$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$

$P(x, y) = 0$  and  $Q(x, y) = 0$  then  $P + Q(x, y) = 0$   
 $P(x, y) = 0$  then  $PQ(x, y) = 0$

$\overline{Gv}$

$$\langle x^4 + y^4 - 2x^3y - x^2y^2 + 2xy^3 - 1 \rangle$$





# Orbit and Group-Closure

(Geometric) Complexity Theory

## Leslie Valiant

23 languages

Article Talk

Read Edit View history Tools

From Wikipedia, the free encyclopedia

"Les Valiant" redirects here. Not to be confused with *Valiant (disambiguation)*.

**Leslie Gabriel Valiant** <sup>FRS</sup><sup>[4][5]</sup> (born 28 March 1949) is a British American<sup>[6]</sup> computer scientist and *computational theorist*.<sup>[7][8]</sup> He was born to a chemical engineer father and a translator mother.<sup>[9]</sup> He is currently the T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics at *Harvard University*.<sup>[10][11][12][13]</sup> Valiant was awarded the *Turing Award* in 2010, having been described by the *A.C.M.* as a heroic figure in theoretical computer science and a role model for his courage and creativity in addressing some of the deepest unsolved problems in science; in particular for his "striking combination of depth and breadth".<sup>[6]</sup>

Leslie Valiant  
FRS



▶  $VP = VNP ?$  [STOC'79]

### The complexity of computing the permanent

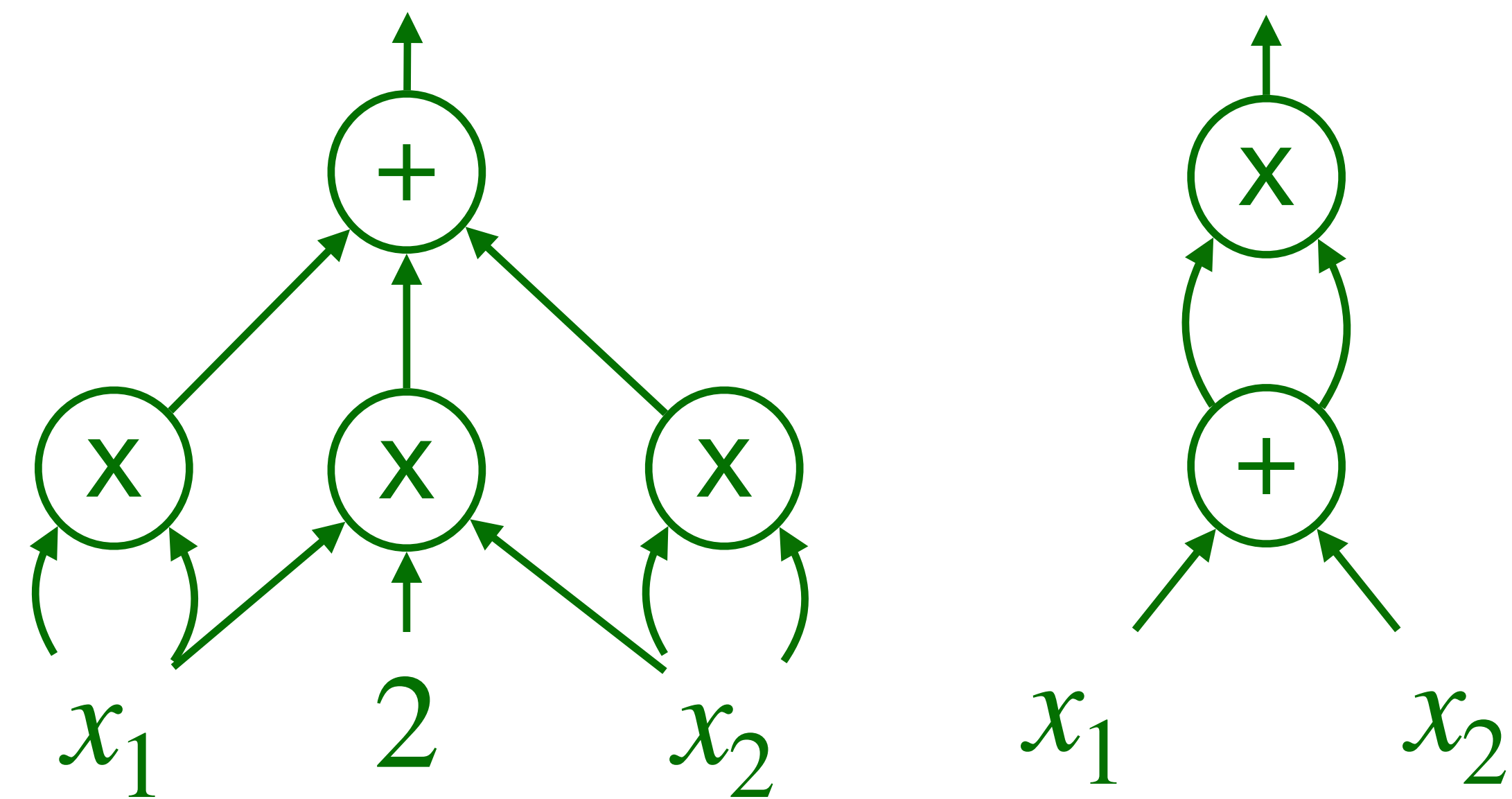
LG Valiant - Theoretical computer science, 1979 - Elsevier

It is shown that the permanent function of (0, 1)-matrices is a complete problem for the class of counting problems associated with nondeterministic polynomial time computations. Related counting problems are also considered. The reductions used are characterized by their nontrivial use of arithmetic.

☆ Save Cite Cited by 3486 Related articles All 12 versions



$\text{comp}(f) = \min(\text{size}(C) \mid \text{circuit } C \text{ computing } f)$



$f(x_1, x_2) = x_1^2 + 2x_1x_2 + x_2^2$

# Leslie Valiant

[Article](#) [Talk](#)

[Read](#) [Edit](#) [View history](#) [Tools](#)


From Wikipedia, the free encyclopedia

*"Les Valiant" redirects here. Not to be confused with Valiant (disambiguation).*

**Leslie Gabriel Valiant** <sup>FRS</sup><sup>[4][5]</sup> (born 28 March 1949) is a British American<sup>[6]</sup> computer scientist and computational theorist.<sup>[7][8]</sup> He was born to a chemical engineer father and a translator mother.<sup>[9]</sup> He is currently the T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics at Harvard University.<sup>[10][11][12][13]</sup> Valiant was awarded the Turing Award in 2010, having been described by the A.C.M. as a heroic figure in theoretical computer science and a role model for his courage and creativity in addressing some of the deepest unsolved problems in science; in particular for his "striking combination of depth and breadth".<sup>[6]</sup>

Leslie Valiant

FRS



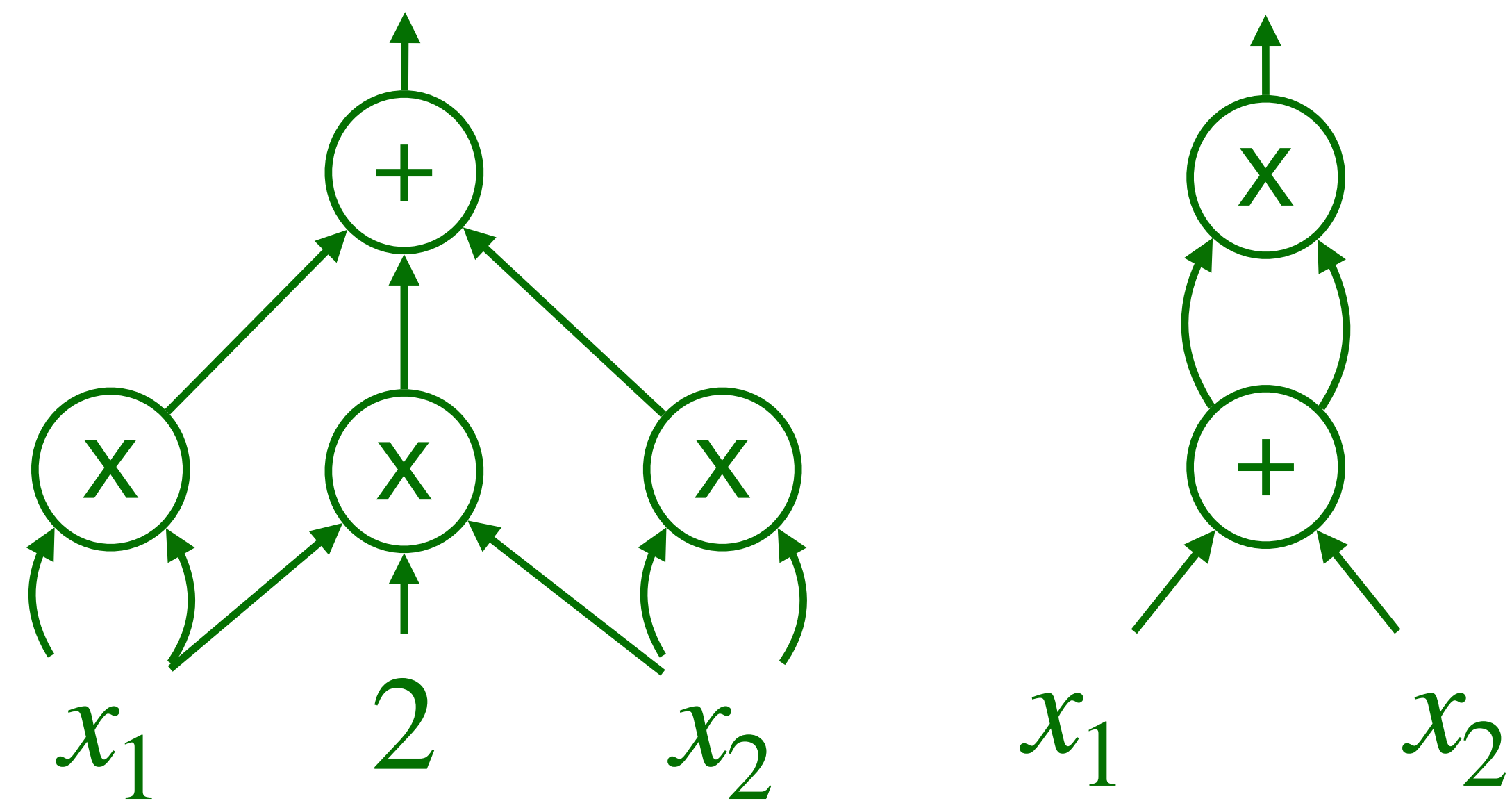
▶ VP = VNP ? [STOC'79]



$\text{comp}(f) = \min(\text{size}(C) \mid \text{circuit } C \text{ computing } f)$

VP: all  $f(x_1, \dots, x_k)$  such that

$\deg(f) \leq \text{poly}(k) \qquad \text{comp}(f) \leq \text{poly}(k)$



$f(x_1, x_2) = x_1^2 + 2x_1x_2 + x_2^2$

# Leslie Valiant

[Article](#) [Talk](#)

[Read](#) [Edit](#) [View history](#) [Tools](#)


From Wikipedia, the free encyclopedia

"Les Valiant" redirects here. Not to be confused with [Valiant \(disambiguation\)](#).

**Leslie Gabriel Valiant** <sup>FRS</sup><sup>[4][5]</sup> (born 28 March 1949) is a British American<sup>[6]</sup> computer scientist and computational theorist.<sup>[7][8]</sup> He was born to a chemical engineer father and a translator mother.<sup>[9]</sup> He is currently the T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics at [Harvard University](#).<sup>[10][11][12][13]</sup> Valiant was awarded the [Turing Award](#) in 2010, having been described by the [A.C.M.](#) as a heroic figure in theoretical computer science and a role model for his courage and creativity in addressing some of the deepest unsolved problems in science; in particular for his "striking combination of depth and breadth".<sup>[6]</sup>

Leslie Valiant

FRS



▶ VP = VNP ? [STOC'79]

$\text{comp}(f) = \min(\text{size}(C) \mid \text{circuit } C \text{ computing } f)$

VP: all  $f(x_1, \dots, x_k)$  such that  
 $\deg(f) \leq \text{poly}(k)$        $\text{comp}(f) \leq \text{poly}(k)$

VNP: all  $f(x_1, \dots, x_k)$  of the form

$$\sum_{\bar{y} \in \{0,1\}^{p(\bar{x})}} g(x_1, \dots, x_k, y_1, \dots, y_{p(k)})$$

where  $g$  is in VP

## Leslie Valiant

23 languages

Article Talk

Read Edit View history Tools

From Wikipedia, the free encyclopedia

*"Les Valiant" redirects here. Not to be confused with Valiant (disambiguation).*

**Leslie Gabriel Valiant** FRS<sup>[4][5]</sup> (born 28 March 1949) is a British American<sup>[6]</sup> computer scientist and computational theorist.<sup>[7][8]</sup> He was born to a chemical engineer father and a translator mother.<sup>[9]</sup> He is currently the T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics at Harvard University.<sup>[10][11][12][13]</sup> Valiant was awarded the Turing Award in 2010, having been described by the A.C.M. as a heroic figure in theoretical computer science and a role model for his courage and creativity in addressing some of the deepest unsolved problems in science; in particular for his "striking combination of depth and breadth".<sup>[6]</sup>

Leslie Valiant  
FRS



▶ VP = VNP ? [STOC'79]



VP: all  $f(x_1, \dots, x_k)$  such that

$$\deg(f) \leq \text{poly}(k) \quad \text{comp}(f) \leq \text{poly}(k)$$

VNP: all  $f(x_1, \dots, x_k)$  of the form

$$\sum_{\bar{y} \in \{0,1\}^{p(\bar{x})}} g(x_1, \dots, x_k, y_1, \dots, y_{p(k)})$$

where  $g$  is in VP

## Leslie Valiant

23 languages

Article Talk

Read Edit View history Tools

From Wikipedia, the free encyclopedia

*"Les Valiant" redirects here. Not to be confused with Valiant (disambiguation).*

**Leslie Gabriel Valiant** <sup>FRS</sup><sup>[4][5]</sup> (born 28 March 1949) is a British American<sup>[6]</sup> computer scientist and computational theorist.<sup>[7][8]</sup> He was born to a chemical engineer father and a translator mother.<sup>[9]</sup> He is currently the T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics at Harvard University.<sup>[10][11][12][13]</sup> Valiant was awarded the Turing Award in 2010, having been described by the A.C.M. as a heroic figure in theoretical computer science and a role model for his courage and creativity in addressing some of the deepest unsolved problems in science; in particular for his "striking combination of depth and breadth".<sup>[6]</sup>

Leslie Valiant  
FRS

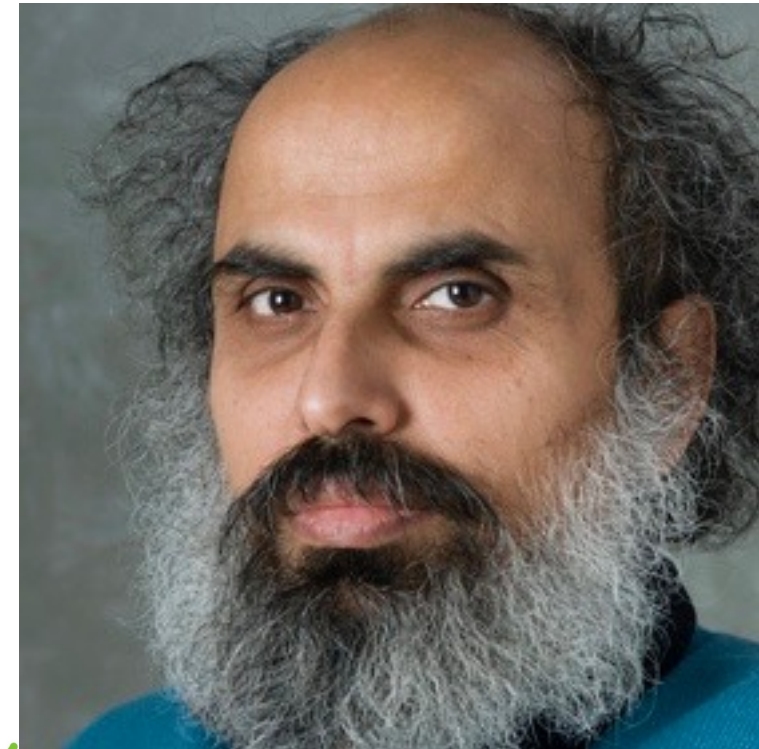


$$\text{Det}(M) = \sum_{\sigma \in S_n} \text{Sgn}(\sigma) \prod_{i=1}^n M_{i,\sigma(i)}$$

▶ VP = VNP ? [STOC'79]

$$\text{Perm}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^n M_{i,\sigma(i)}$$

# (Geometric) Complexity Theory



K Mulmuley

M Sohoni

[FOCS'12]



VP = VNP if and only if  $\text{Perm}(M) \in \overline{\text{SL}_{n^2} \text{Det}(M)}$

$$\text{Det}(M) = \sum_{\sigma \in S_n} \text{Sgn}(\sigma) \prod_{i=1}^n M_{i,\sigma(i)}$$



▶ VP = VNP ? [STOC'79]

$$\text{Perm}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^n M_{i,\sigma(i)}$$



# Orbit and Group-Closure

Quantum Computation

C Moore



JP Crutchfield



## Quantum automata and quantum grammars

[C Moore](#), [JP Crutchfield](#) - Theoretical Computer Science, 2000 - Elsevier

... a function that assigns **quantum** probabilities to words. We also define **quantum grammars**, in ... , generated by **quantum grammars** and recognized by **quantum automata**, have pleasing ...

☆ Save  Cite Cited by 556 Related articles All 17 versions



# Quantum Computing Model

►  $\mathbf{v} \in \mathbb{Q}^{n \times 1}$  of unit norm

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \dots + v_n^2} = 1$$

►  $\{M_\sigma\}_{\sigma \in \Sigma}$  unitary matrix

length preserving  $\|\mathbf{v}\| = \|M_\sigma \mathbf{v}\|$

►  $P \in \mathbb{Q}^{n \times n}$  orthogonal projection matrix

Every word  $w = \sigma_1 \cdots \sigma_n$  will have a matrix  $M_w = M_{\sigma_n} \cdots M_{\sigma_1}$

# Quantum Computing Model

▶  $\mathbf{v} \in \mathbb{Q}^{n \times 1}$  of unit norm

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \dots + v_n^2} = 1$$

▶  $\{M_\sigma\}_{\sigma \in \Sigma}$  unitary matrix

length preserving  $\|\mathbf{v}\| = \|M_\sigma \mathbf{v}\|$

▶  $P \in \mathbb{Q}^{n \times n}$  orthogonal projection matrix

Every word  $w = \sigma_1 \cdots \sigma_n$  will have a matrix  $M_w = M_{\sigma_n} \cdots M_{\sigma_1}$

Quantum automaton computes a function  $\text{Val} : \Sigma^* \rightarrow \mathbb{Q}$  where

$$\text{Val}(w) = \|P M_w \mathbf{v}\|^2$$

# Quantum Computing Model

▶  $\mathbf{v} \in \mathbb{Q}^{n \times 1}$  of unit norm

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \dots + v_n^2} = 1$$

▶  $\{M_\sigma\}_{\sigma \in \Sigma}$  unitary matrix

length preserving  $\|\mathbf{v}\| = \|M_\sigma \mathbf{v}\|$

▶  $P \in \mathbb{Q}^{n \times n}$  orthogonal projection matrix

Every word  $w = \sigma_1 \cdots \sigma_n$  will have a matrix  $M_w = M_{\sigma_n} \cdots M_{\sigma_1}$

Quantum automaton computes a function  $\text{Val} : \Sigma^* \rightarrow \mathbb{Q}$  where

$$\text{Val}(w) = \|P M_w \mathbf{v}\|^2$$



$\lambda$ -threshold problem asks, given  $(\mathbf{v}, \{M_\sigma\}_{\sigma \in \Sigma}, P)$ , whether

$$\exists w \in \Sigma^* \text{ such that } \|P M_w \mathbf{v}\|^2 > \lambda$$

# Quantum Computing Model

▶  $\mathbf{v} \in \mathbb{Q}^{n \times 1}$  of unit norm

▶  $\{M_\sigma\}_{\sigma \in \Sigma}$  unitary matrix

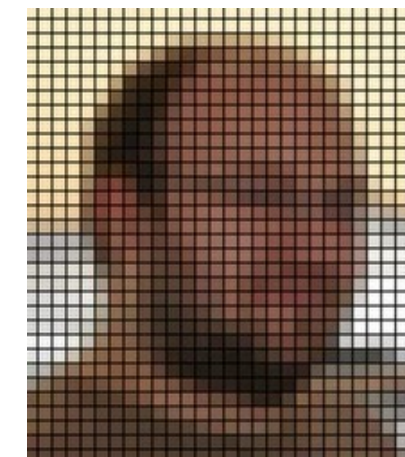
▶  $P \in \mathbb{Q}^{n \times n}$  orthogonal projection matrix

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \dots + v_n^2} = 1$$

length preserving  $\|\mathbf{v}\| = \|M_\sigma \mathbf{v}\|$



H Derksen



E. Jeandel



P. Koiran

Decidable!

[SIAM JC'05]



$\lambda$ -threshold problem asks, given  $(\mathbf{v}, \{M_\sigma\}_{\sigma \in \Sigma}, P)$ , whether

$$\exists w \in \Sigma^* \text{ such that } \|P M_w \mathbf{v}\|^2 > \lambda$$

# Quantum Computing Model

▶  $\mathbf{v} \in \mathbb{Q}^{n \times 1}$  of unit norm

▶  $\{M_\sigma\}_{\sigma \in \Sigma}$  unitary matrix

▶  $P \in \mathbb{Q}^{n \times n}$  orthogonal projection matrix

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \dots + v_n^2} = 1$$

length preserving  $\|\mathbf{v}\| = \|M_\sigma \mathbf{v}\|$

$$\|P M_w \mathbf{v}\|^2 \leq \lambda \quad \text{for all } w \in \Sigma^*$$

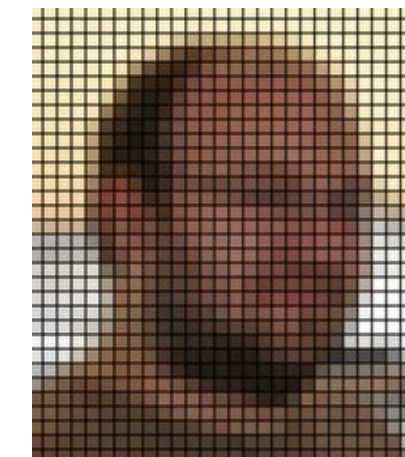


$$\|P \underbrace{\langle M_\sigma \mid \sigma \in \Sigma \rangle \mathbf{v}}_{\text{orbit-closure}}\|^2 \leq \lambda$$

orbit-closure



H Derksen



E. Jeandel



P. Koiran

Decidable!  
[SIAM JC'05]



$\lambda$ -threshold problem asks, given  $(\mathbf{v}, \{M_\sigma\}_{\sigma \in \Sigma}, P)$ , whether

$$\exists w \in \Sigma^* \text{ such that } \|P M_w \mathbf{v}\|^2 > \lambda$$



# Algorithms



No complexity bounds!

This Euclidian closure  $\overline{\langle M_\sigma \mid \sigma \in \Sigma \rangle v}$  is algebraic!

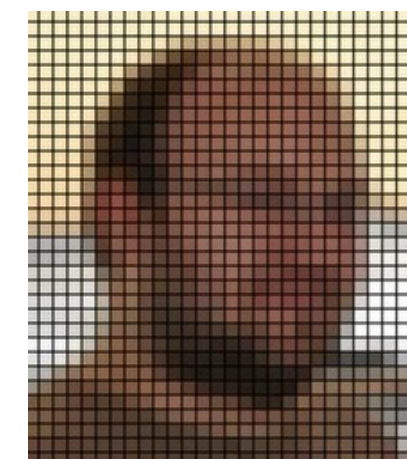
The vanishing ideal  $I \subseteq \mathbb{Q}[x_1, \dots, x_n]$  of the orbit-closure can be finitely presented:

$$I = \langle P_1, \dots, P_k \mid P_i \in \mathbb{Q}_n \rangle$$

$\underbrace{\overline{\langle M_\sigma \mid \sigma \in \Sigma \rangle v}}_{\text{orbit-closure}}$



H Derksen



E. Jeandel



P. Koiran

Decidable!  
[SIAM JC'05]

# The Orbit and Group-Closure Problems

**Rational Series and Automata**

## NONCOMMUTATIVE RATIONAL PÓLYA SERIES

JASON BELL AND DANIEL SMERTNIG

ABSTRACT. A (noncommutative) Pólya series over a field  $K$  is a formal power series whose nonzero coefficients are contained in a finitely generated subgroup of  $K^\times$ . We show that rational Pólya series are unambiguous rational series, proving a 40 year old conjecture of Reutenauer. The proof combines methods from noncommutative algebra, automata theory, and number theory (specifically, unit equations). As a corollary, a rational series is a Pólya series if and only if it is Hadamard sub-invertible. Phrased differently, we show that every weighted finite automaton taking values in a finitely generated subgroup of a field (and zero) is equivalent to an unambiguous weighted finite automaton.

# Orbit and Group-Closure



**Program Analysis**

# Does this program terminate?

$$G = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \right\rangle$$

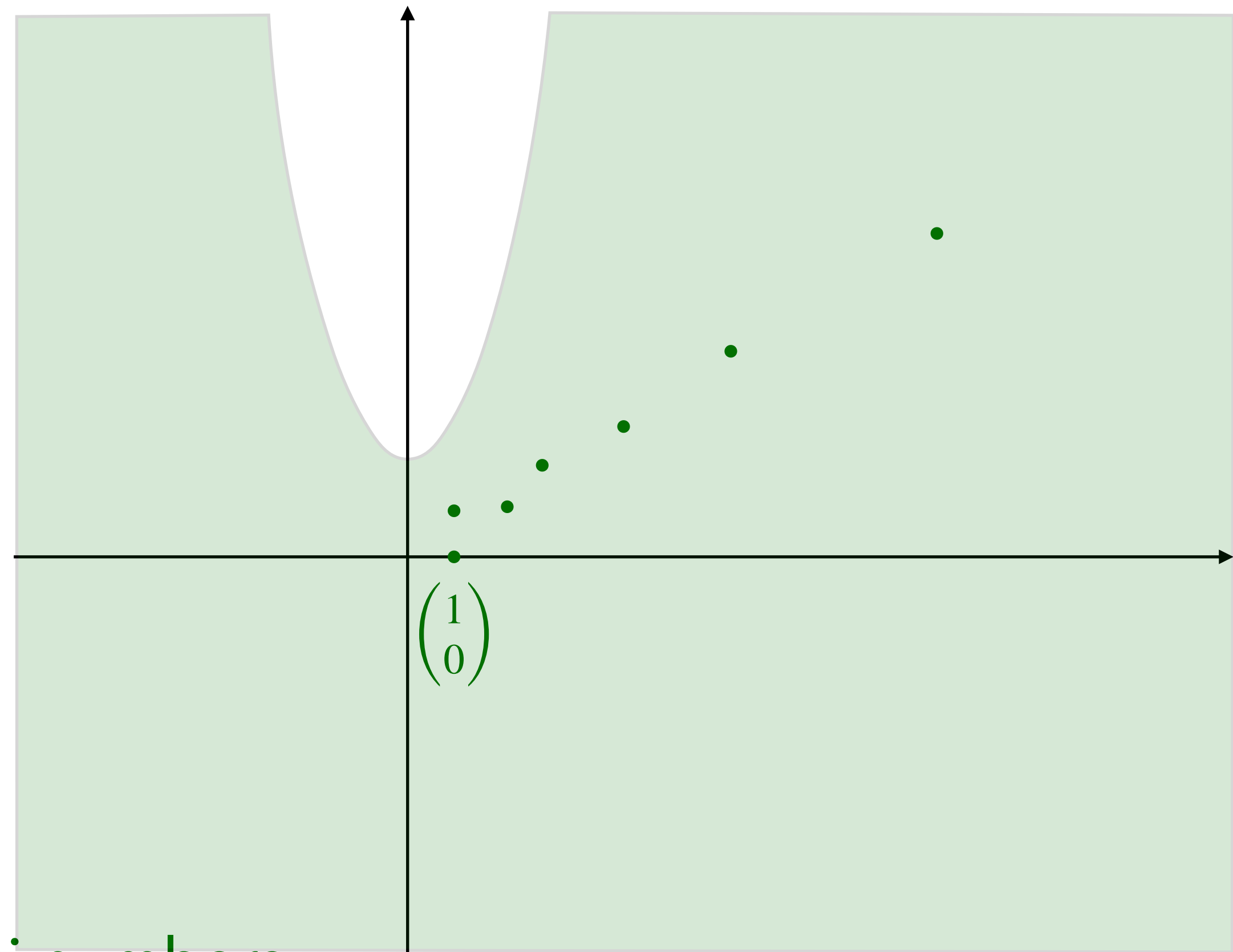
$x := 1;$

$y := 0;$

while  $y - x^2 \leq 2$  do

$$\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

done



Fibonacci numbers



# Does this program terminate?

Deciding termination of simple linear loops is open!

Terence Tao

Article

Talk

Read

Edit

View history

Tools

From Wikipedia, the free encyclopedia

Terence Chi-Shen Tao


FAA FRS

(Chinese: 陶哲軒; born 17 July 1975) is an Australian mathematician. He is a professor of mathematics at the University of California, Los Angeles (UCLA), where he holds the James and Carol Collins chair. His research includes topics in harmonic analysis, partial differential equations, algebraic combinatorics, arithmetic combinatorics, geometric combinatorics, probability theory, compressed sensing and analytic number theory.<sup>[4]</sup>

Tao was born to Chinese immigrant parents and raised in Adelaide. Tao won the Fields Medal in 2006 and won the Royal Medal and Breakthrough Prize in Mathematics in 2014. He is also a 2006 MacArthur Fellow. Tao has been the author or co-author of over three hundred research papers.<sup>[5]</sup> He is widely regarded as one of the greatest living mathematicians.<sup>[6][7][8][9][10]</sup>

Terence Tao

FAA FRS



Tao in 2021

$$\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

done

Fibonacci numbers

The figure shows a 2D coordinate system with a light green shaded region. A white curve, representing the growth of Fibonacci numbers, starts at the origin and curves upwards. Several green dots are plotted along the curve, showing the sequence's growth. A green speech bubble points to the curve with the text: "It is faintly outrageous that this problem is still open...".



# Does this program terminate?

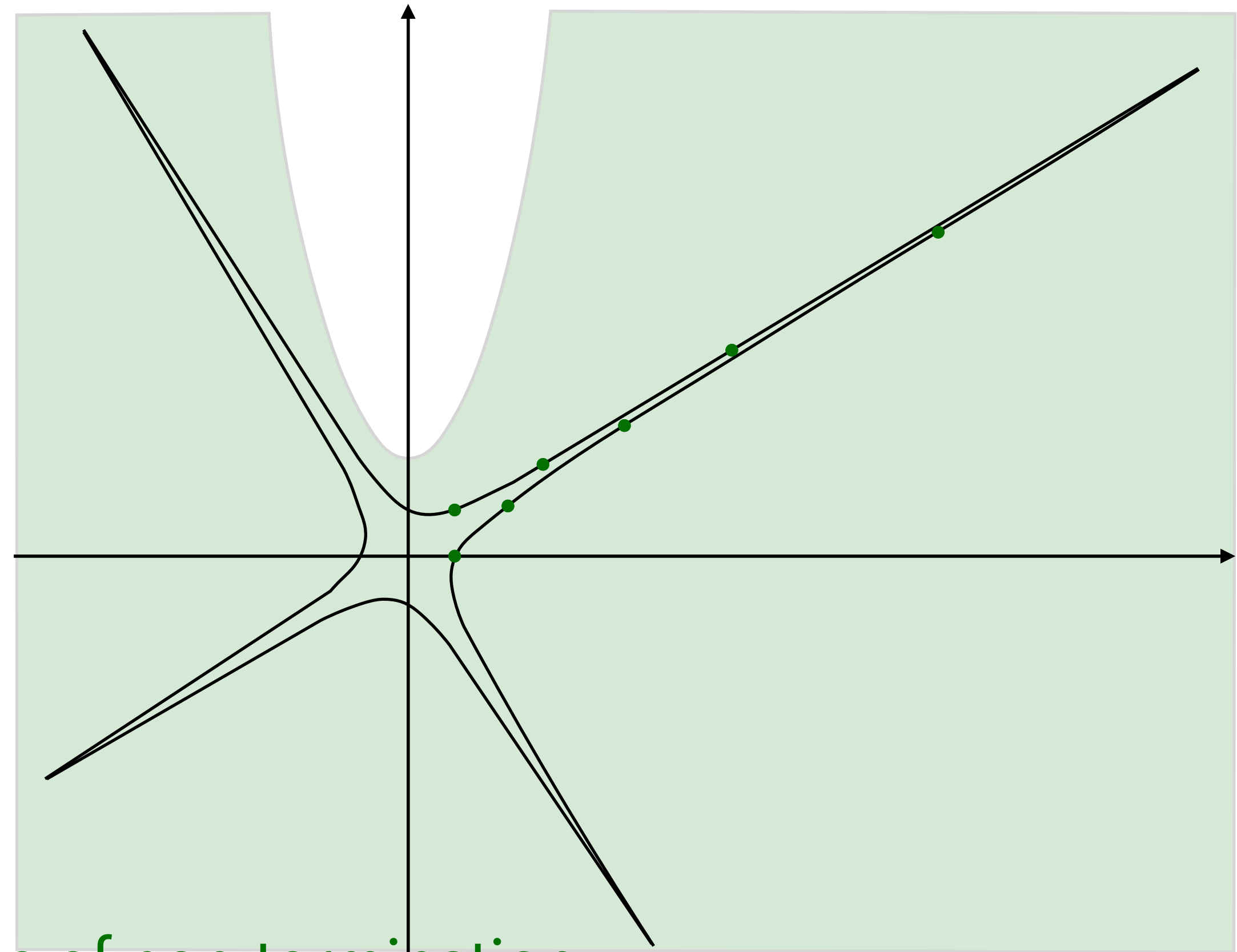
$x := 1;$

$y := 0;$

while  $y - x^2 \leq 2$  do

$$\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

done



Certificate of non-termination

$\overline{Gv}$  defined by  $\langle x^4 + y^4 - 2x^3y - x^2y^2 + 2xy^3 - 1 \rangle$

# Does this program terminate?

► It is an invariant: it holds at every step

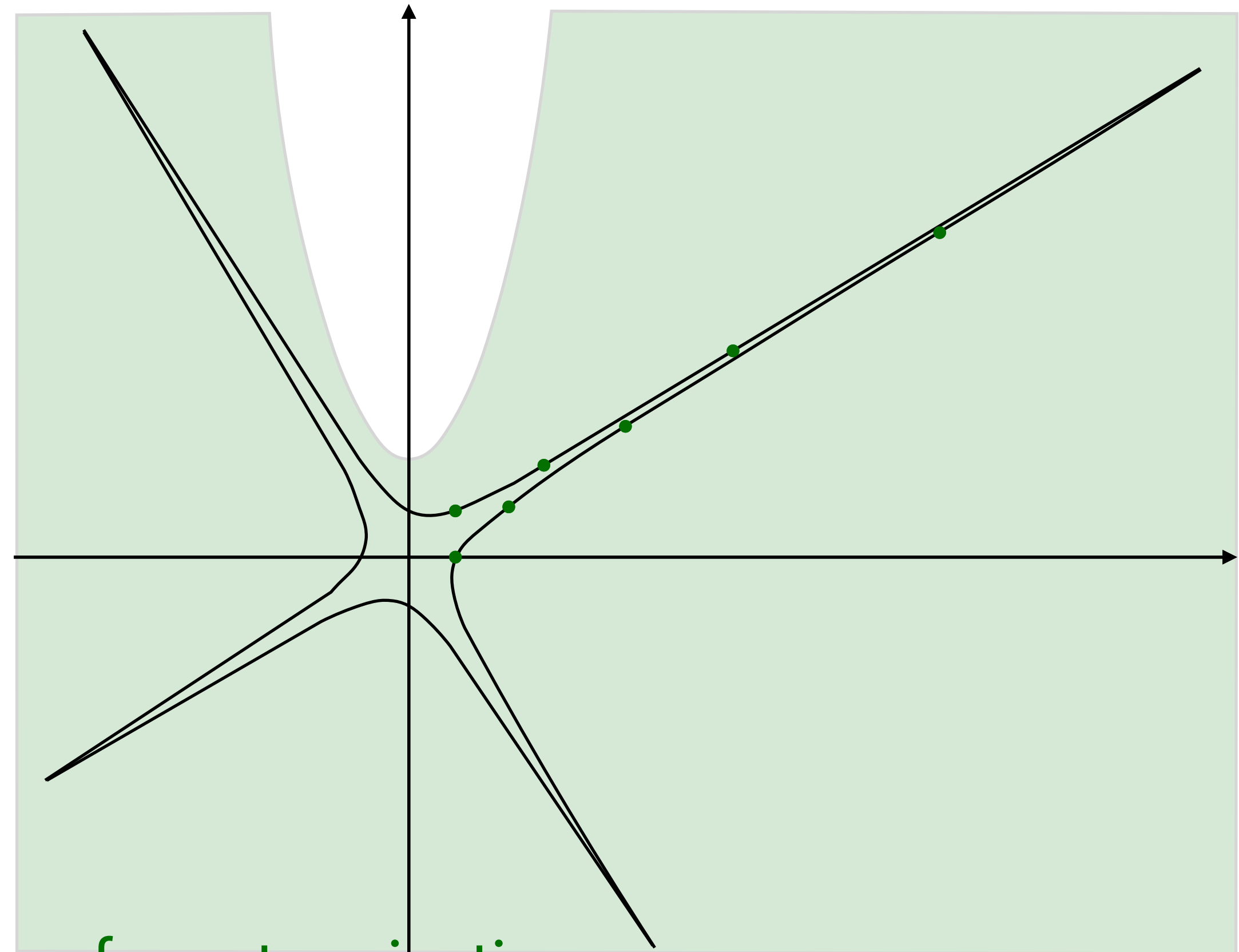
```
x := 1;
```

```
y := 0;
```

```
while  $y - x^2 \leq 2$  do
```

$$\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

```
done
```



Certificate of non-termination

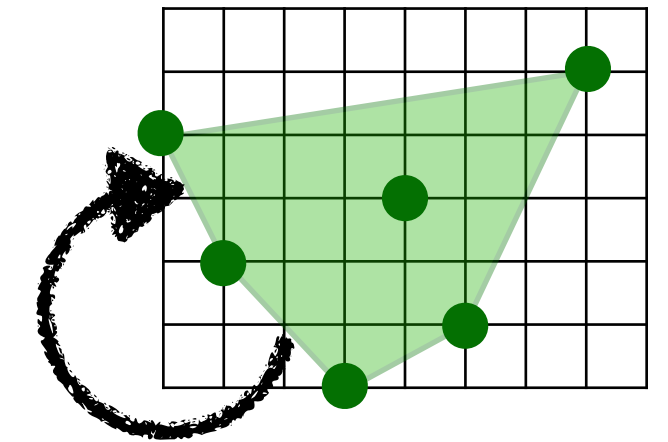
$$\overline{Gv} = \langle x^4 + y^4 - 2x^3y - x^2y^2 + 2xy^3 - 1 \rangle$$

# Invariants



Inductive invariant = invariant preserved by the transition relation

preserved by  
transition



*The classical approach to the verification of temporal safety properties of programs requires the construction of inductive invariants [...]. Automation of this construction is the main challenge in program verification.*

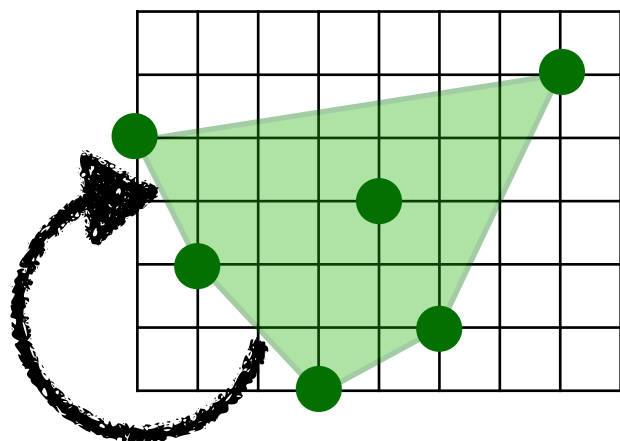
D. Beyer, T. Henzinger, R. Majumdar, and A. Rybalchenko  
Invariant Synthesis for Combined Theories, 2007

# Invariants

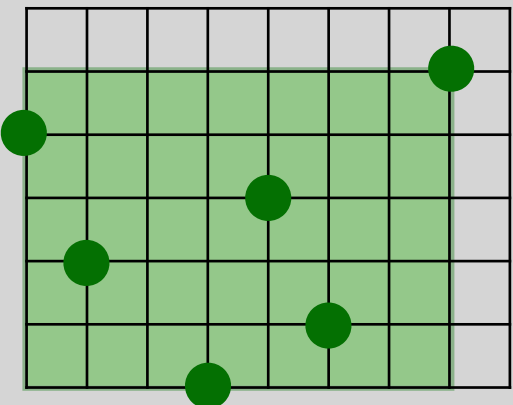


Inductive invariant = invariant preserved by the transition relation

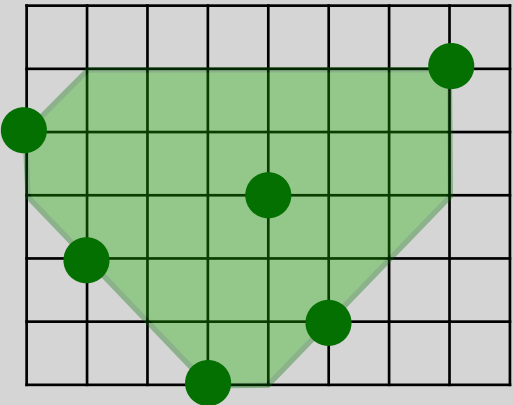
preserved by transition



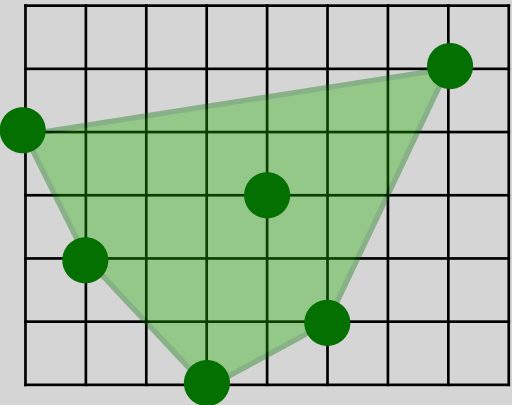
Which invariants?



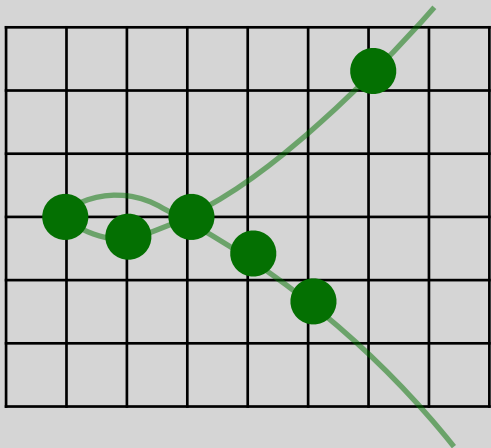
Intervals



Octagons



Polyhedrons



Algebraic Sets



# Affine invariants



## Theorem (Karr 76)

There is an algorithm computing the strongest affine inductive invariant for affine programs.

orbit-closure

degree one

reachable states are an orbit under a semigroup

### Affine relationships among variables of a program

M Karr - Acta Informatica, 1976 - Springer

Several optimizations of programs can be performed when in certain regions of a program equality relationships hold between a linear combination of the variables of the program and a constant. This paper presents a practical approach to detecting these relationships by considering the problem from the viewpoint of linear algebra. Key to the practicality of this approach is an algorithm for the calculation of the "sum" of linear subspaces.

☆ Save Cite Cited by 577 Related articles All 8 versions

# Affine invariants



## Theorem (Karr 76)

There is an algorithm computing the strongest affine inductive invariant for affine programs.

### Affine relationships among variables of a program

M Karr - Acta Informatica, 1976 - Springer

Several optimizations of programs can be performed when in certain regions of a program equality relationships hold between a linear combination of the variables of the program and a constant. This paper presents a practical approach to detecting these relationships by considering the problem from the viewpoint of linear algebra. Key to the practicality of this approach is an algorithm for the calculation of the "sum" of linear subspaces.

☆ Save 📄 Cite Cited by 577 Related articles All 8 versions

### A Note on Karr's Algorithm

Markus Müller-Olm<sup>1\*</sup> and Helmut Seidl<sup>2</sup>

<sup>1</sup> FernUniversität Hagen, FB Informatik, LG PI 5, Universitätsstr. 1, 58097 Hagen, Germany

[mmo@ls5.informatik.uni-dortmund.de](mailto:mmo@ls5.informatik.uni-dortmund.de)

<sup>2</sup> TU München, Informatik, I2, 85748 München, Germany

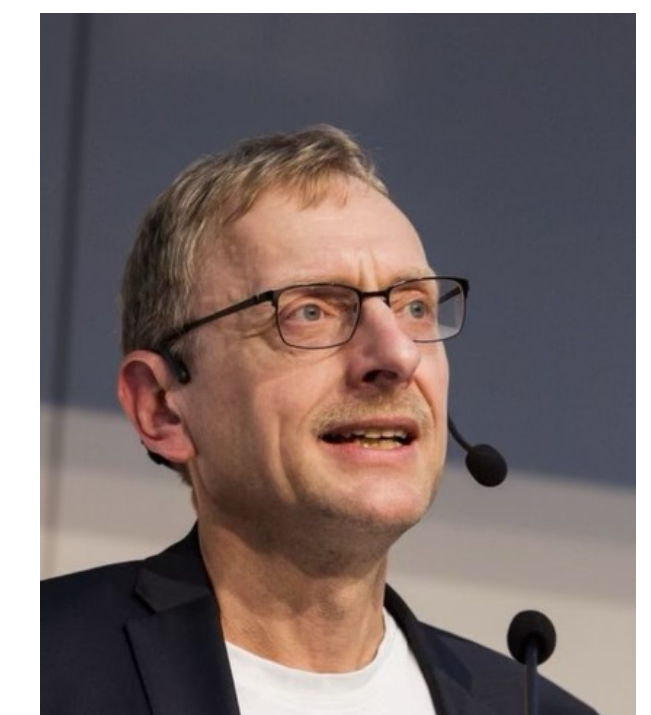
[seidl@informatik.tu-muenchen.de](mailto:seidl@informatik.tu-muenchen.de)

**Abstract.** We give a simple formulation of Karr's algorithm for computing all affine relationships in affine programs. This simplified algorithm runs in time  $\mathcal{O}(nk^3)$  where  $n$  is the program size and  $k$  is the number of program variables assuming unit cost for arithmetic operations. This improves upon the original formulation by a factor of  $k$ . Moreover, our re-formulation avoids exponential growth of the lengths of intermediately occurring numbers (in binary representation) and uses less complicated elementary operations. We also describe a generalization that determines all polynomial relations up to degree  $d$  in time  $\mathcal{O}(nk^{3d})$ .



M. Müller-Olm

H. Seidl



[ICALP'04]



# Polynomial invariants

... it is a challenging open problem whether or not the set of all valid polynomial relations can be computed not just the ones of some given form..

Computing polynomial program invariants

M Müller-Olm, [H Seidl](#) - Information Processing Letters, 2004 - Elsevier

... the **polynomial** relation in question is valid at the target **program** point if and only if the ... , we compute the weakest precondition of a generic **polynomial** relation at the target **program** point...

☆ Save Cite Cited by 120 Related articles All 16 versions

A Note on Karr’s Algorithm

Markus Müller-Olm<sup>1\*</sup> and Helmut Seidl<sup>2</sup>

<sup>1</sup> FernUniversität Hagen, FB Informatik, LG PI 5, Universitätsstr. 1, 58097 Hagen, Germany  
mmo@ls5.informatik.uni-dortmund.de

<sup>2</sup> TU München, Informatik, I2, 85748 München, Germany  
seidl@informatik.tu-muenchen.de

**Abstract.** We give a simple formulation of Karr’s algorithm for computing all affine relationships in affine programs. This simplified algorithm runs in time  $\mathcal{O}(nk^3)$  where  $n$  is the program size and  $k$  is the number of program variables assuming unit cost for arithmetic operations. This improves upon the original formulation by a factor of  $k$ . Moreover, our re-formulation avoids exponential growth of the lengths of intermediately occurring numbers (in binary representation) and uses less complicated elementary operations. We also describe a generalization that determines all polynomial relations up to degree  $d$  in time  $\mathcal{O}(nk^{3d})$ .



M. Müller-Olm



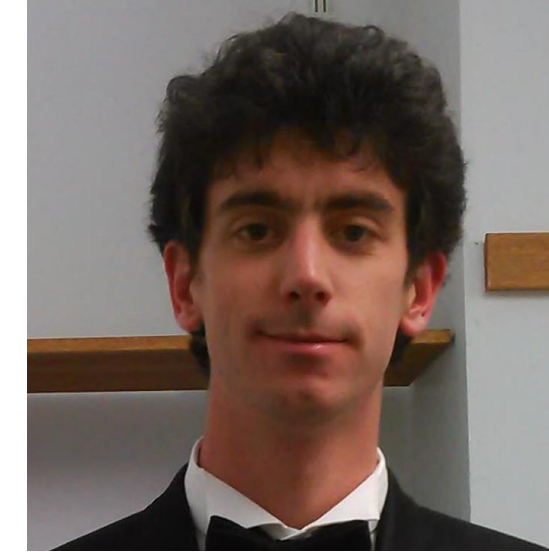
H. Seidl

[ICALP’04]

# Polynomial invariants



E. Hrushovski



A. Pouly



J. Ouaknine



J. Worrell



Theorem [LICS'18]

There is an algorithm computing strongest polynomial inductive invariant for affine programs.

orbit-closure

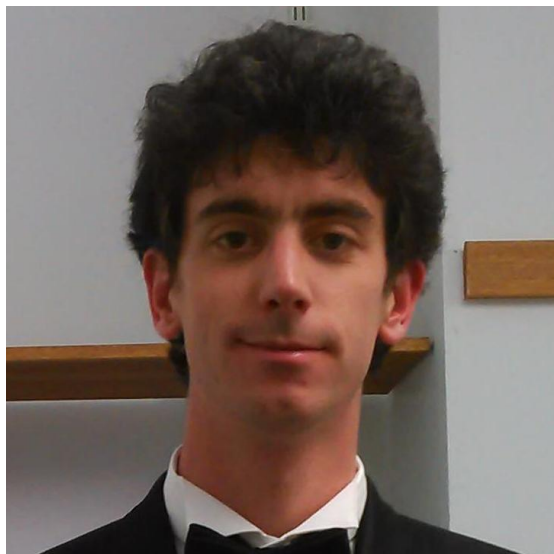
reachable states are an  
orbit under a semigroup



# Polynomial invariants



E. Hrushovski



A. Pouly



J. Ouaknine

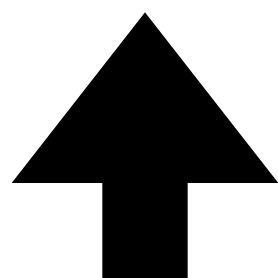


J. Worrell



Theorem [LICS'18]

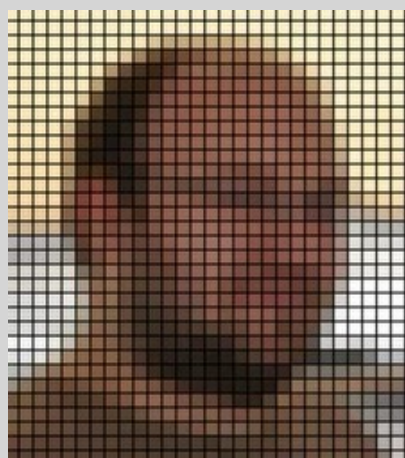
There is an algorithm computing strongest polynomial inductive invariant for affine programs.



No complexity bounds!



H Derksen



E. Jeandel



P. Koiran

$\overline{\langle M_\sigma \mid \sigma \in \Sigma \rangle v}$   
orbit-closure

Decidable!  
[SIAM JC'05]

# Group Closure



Let  $S \subseteq GL_n(\mathbb{Q})$ . What is the complexity of computing  $\overline{\langle S \rangle}$  ?

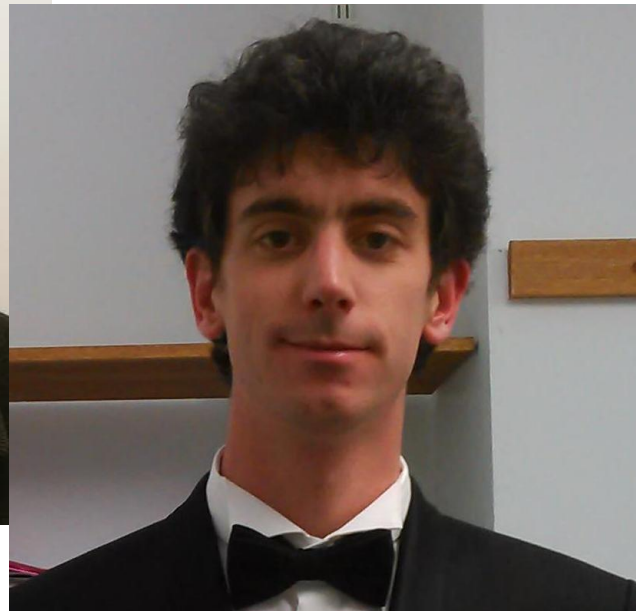
J. Worrell



S. Schmitz



K. Nosan



A. Pouly



M. Shirmohammadi



# Polynomial invariants

... it is a challenging open problem whether or not the set of all valid polynomial relations can be computed not just the ones of some given form..

## A Note on Karr's Algorithm

Markus Müller-Olm<sup>1\*</sup> and Helmut Seidl<sup>2</sup>

<sup>1</sup> FernUniversität Hagen, FB Informatik, LG PI 5, Universitätsstr. 1, 58097 Hagen, Germany

`mmo@ls5.informatik.uni-dortmund.de`

<sup>2</sup> TU München, Informatik, I2, 85748 München, Germany

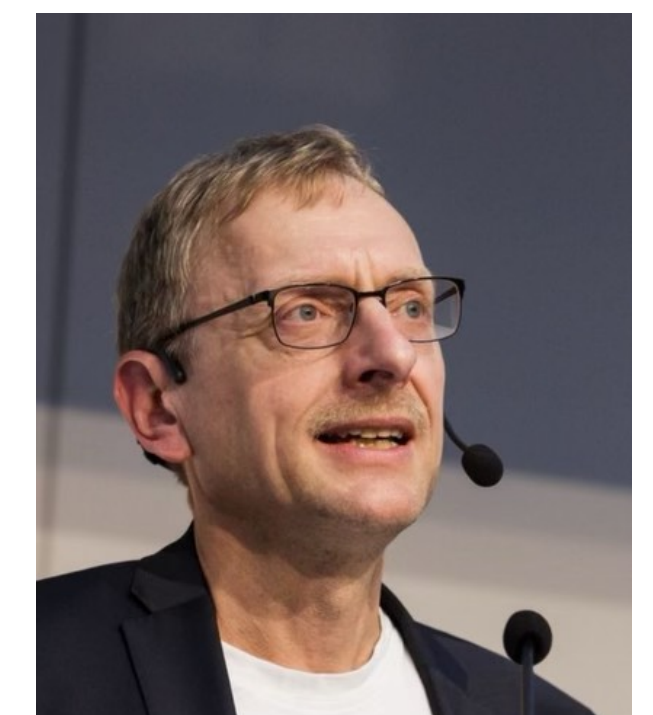
`seidl@informatik.tu-muenchen.de`

**Abstract.** We give a simple formulation of Karr's algorithm for computing all affine relationships in affine programs. This simplified algorithm runs in time  $\mathcal{O}(nk^3)$  where  $n$  is the program size and  $k$  is the number of program variables assuming unit cost for arithmetic operations. This improves upon the original formulation by a factor of  $k$ . Moreover, our re-formulation avoids exponential growth of the lengths of intermediate occurring numbers (in binary representation) and uses less complicated elementary operations. We also describe a generalization that determines all polynomial relations up to degree  $d$  in time  $\mathcal{O}(nk^{3d})$ .

[ICALP'04]



M. Müller-Olm



H. Seidl

# Group Closure

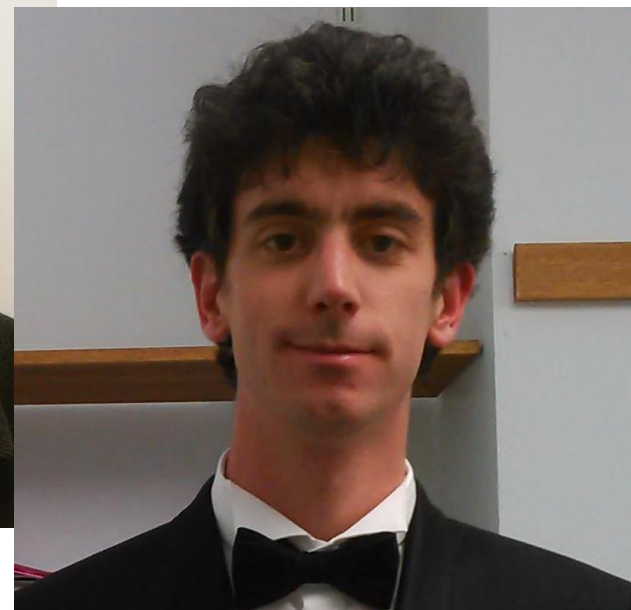


Let  $S \subseteq GL_n(\mathbb{Q})$ . What is the complexity of computing  $\overline{\langle S \rangle}$  ?

Bound the degree

Theorem [ISSAC'22]

The closure  $\overline{\langle S \rangle}$  can be computed in  $10 \text{ EXPTIME}(\text{size}(S), n)$ .



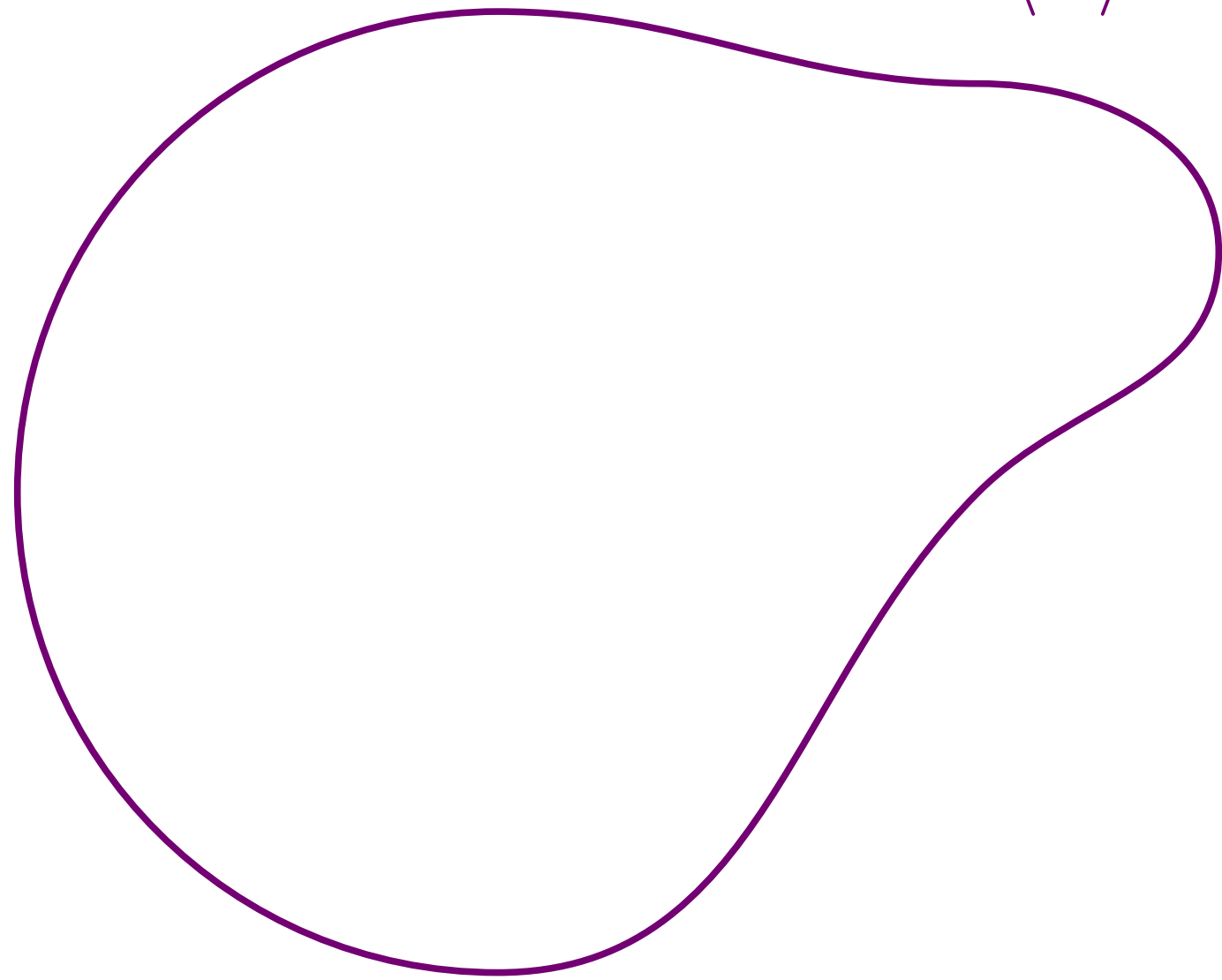
M. Shirmohammadi



# Group Closure

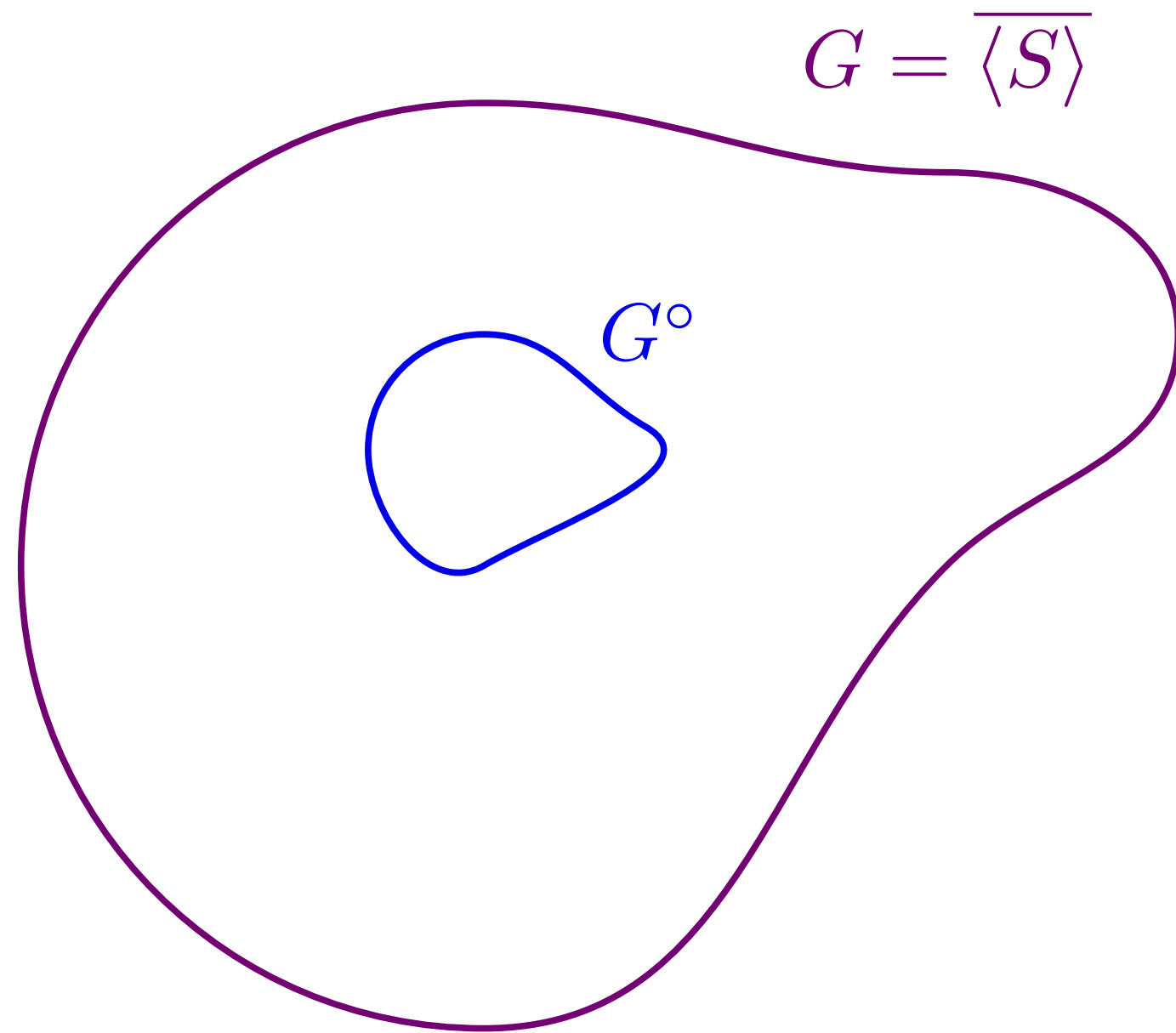
$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$

$$G = \overline{\langle S \rangle}$$



# Identity Component

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$

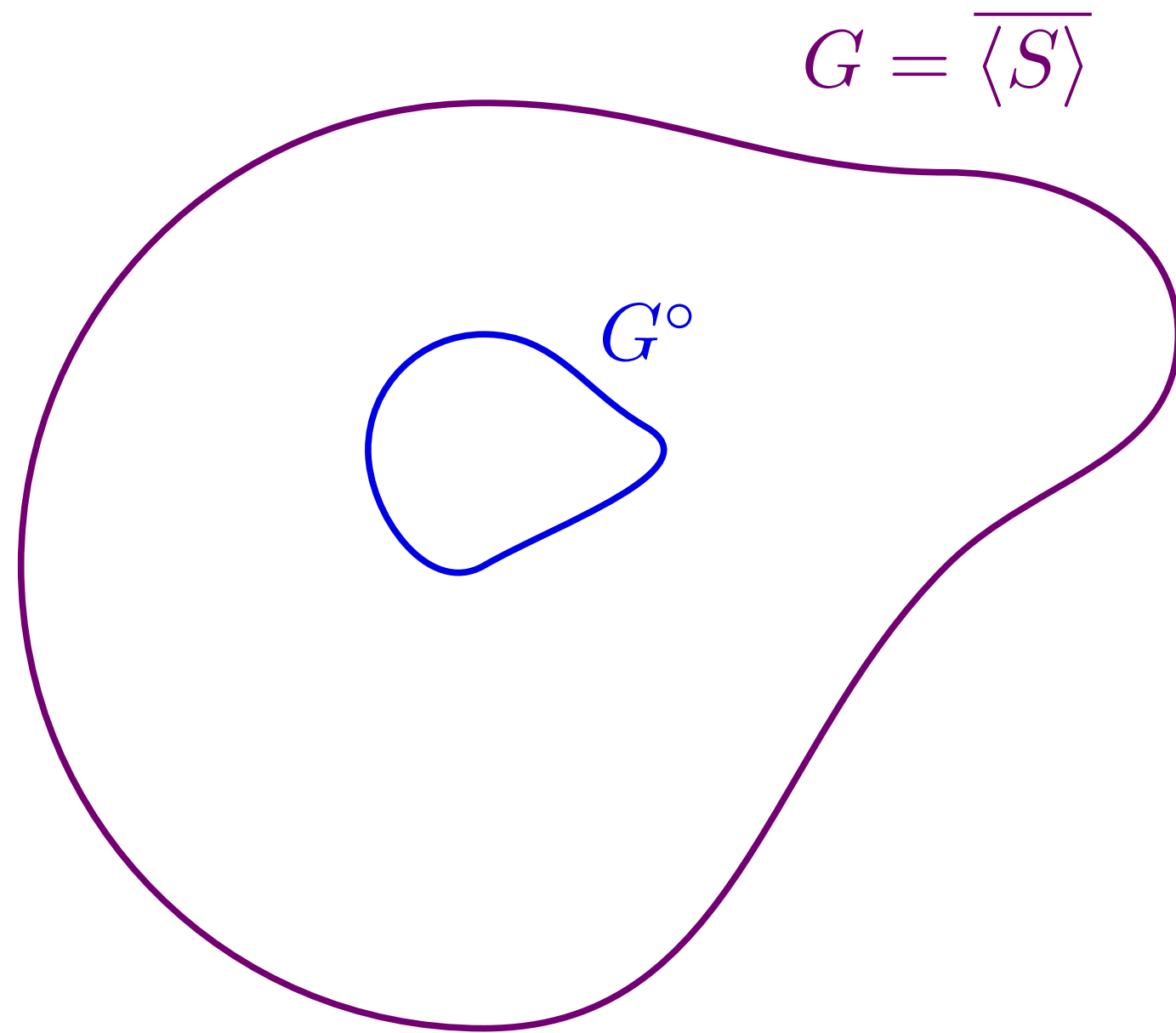


Identity Component  $G^\circ$

►  $G^\circ \trianglelefteq G$

# Identity Component

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



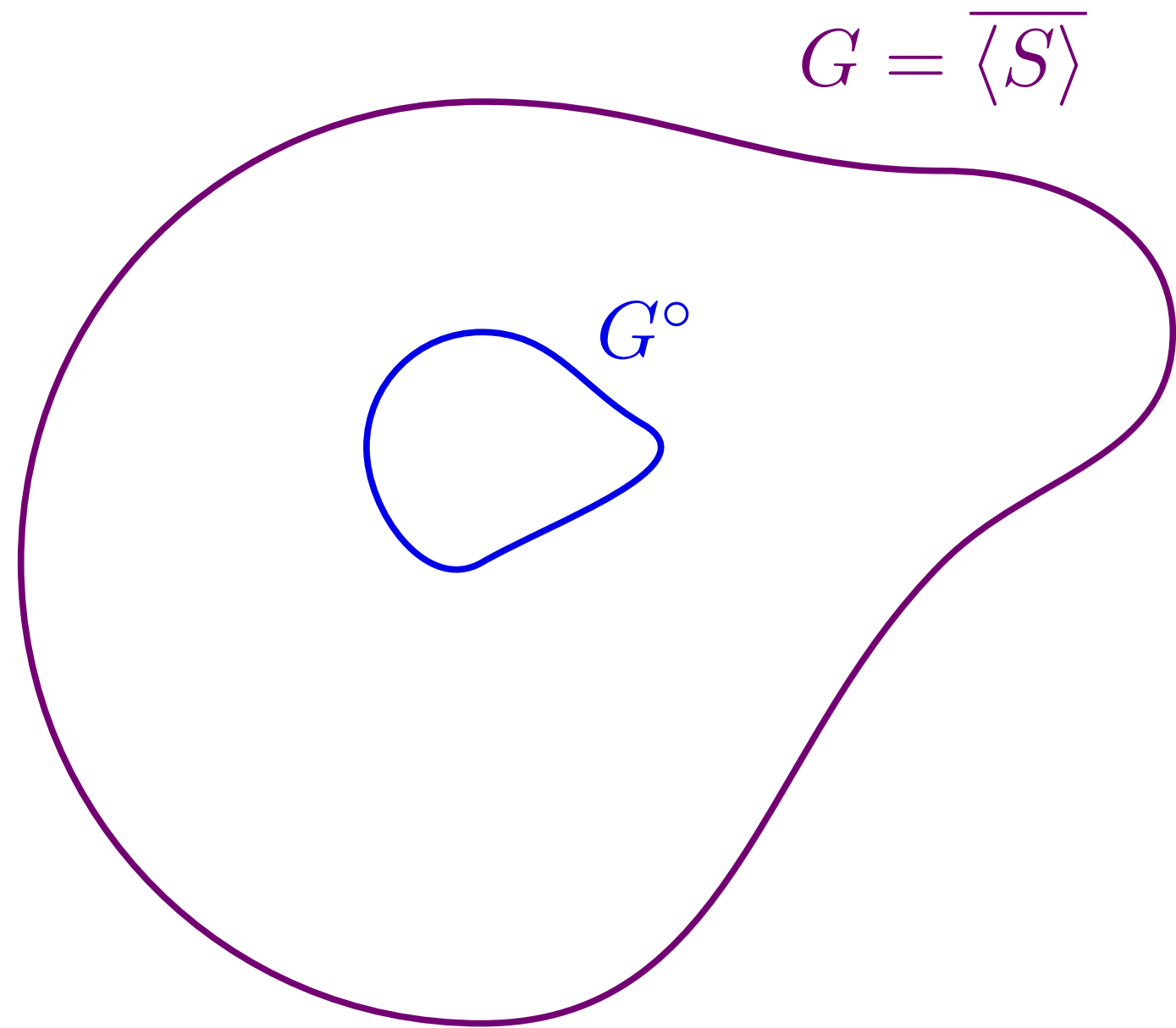
## Identity Component $G^\circ$

►  $G^\circ \trianglelefteq G$

►  $G/G^\circ \cong$  finite rational group  $\mathrm{GL}_p(\mathbb{Q})$

# Identity Component

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



Size of finite rational groups  $\mathrm{GL}_p(\mathbb{Q})$

is at most  $(2p)!$

## Identity Component $G^\circ$

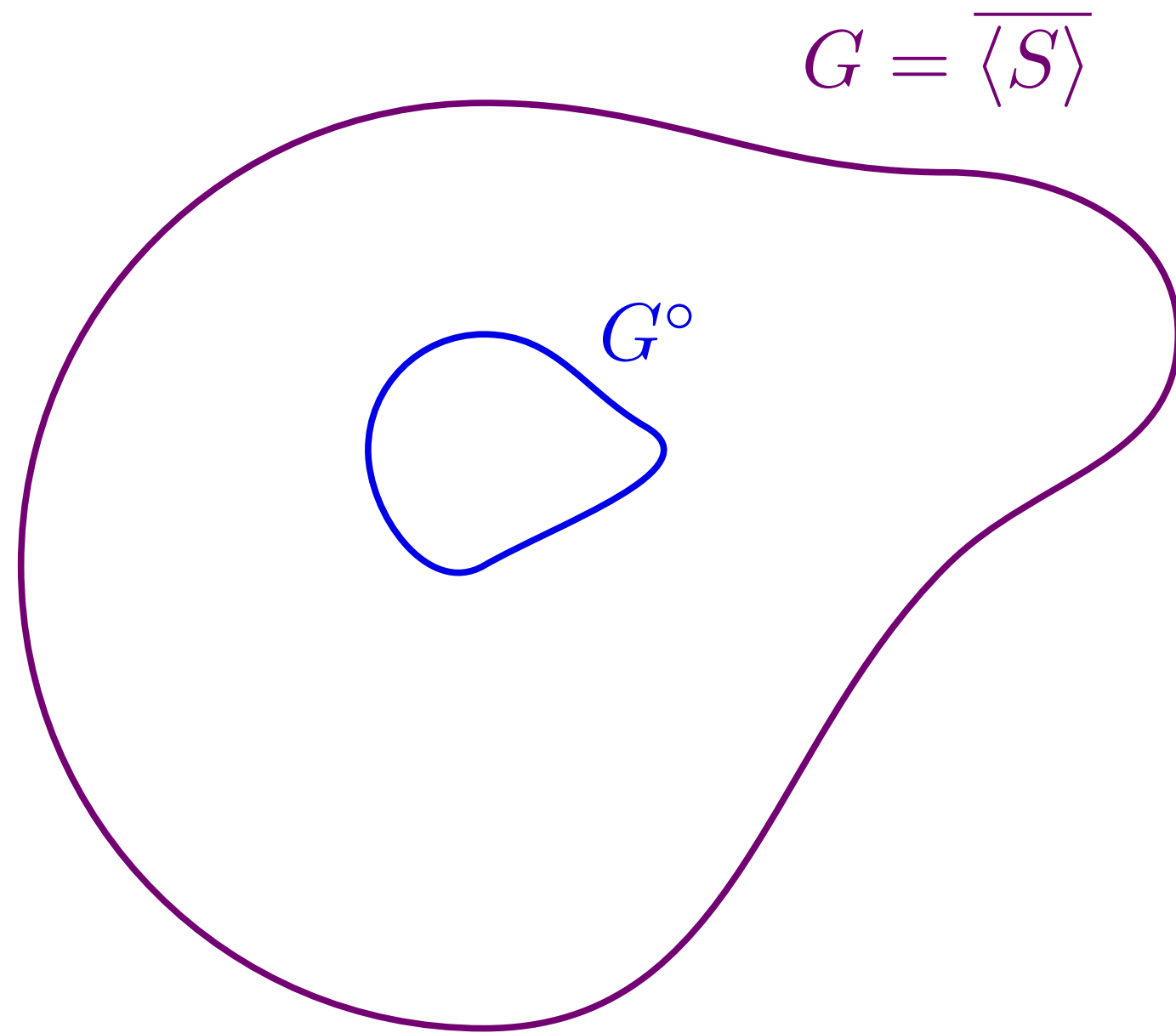
►  $G^\circ \trianglelefteq G$

►  $G/G^\circ \cong$  finite rational group  $\mathrm{GL}_p(\mathbb{Q})$



# Identity Component

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



Can we degree bound  $G^\circ$ ?



Size of finite rational groups  $\mathrm{GL}_p(\mathbb{Q})$

is at most  $(2p)!$

## Identity Component $G^\circ$

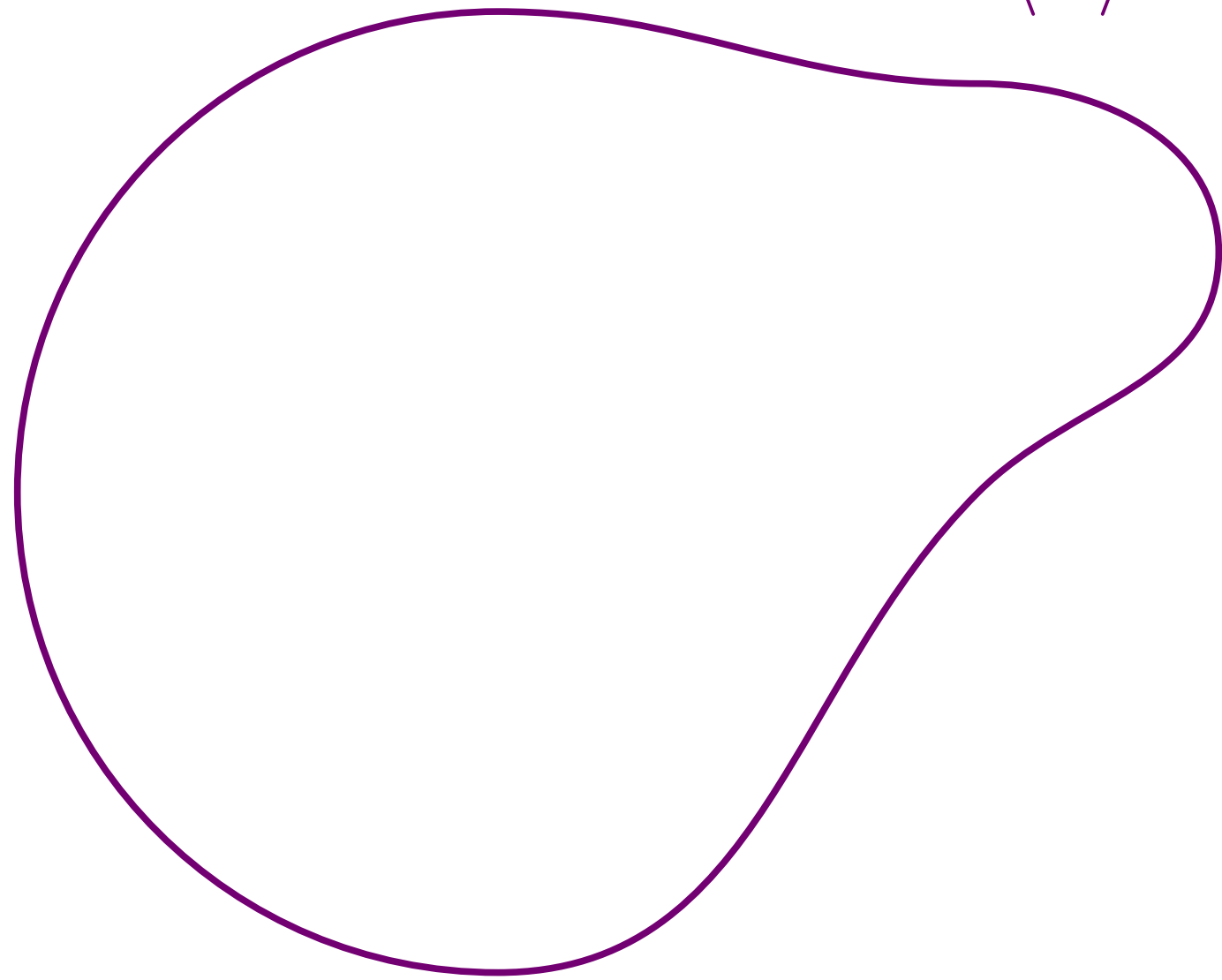
►  $G^\circ \trianglelefteq G$

►  $G/G^\circ \cong$  finite rational group  $\mathrm{GL}_p(\mathbb{Q})$

# Group Closure

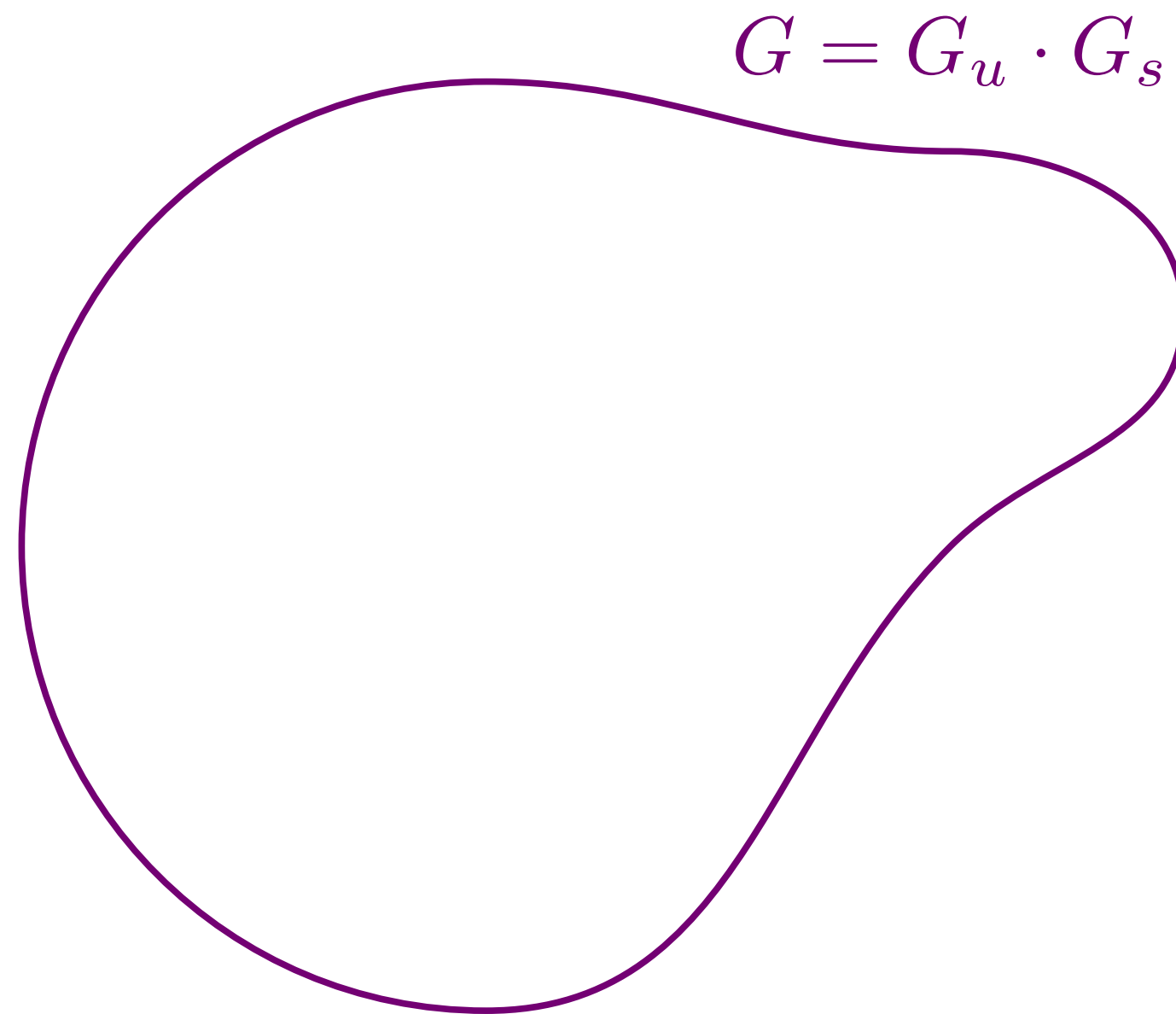
$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$

$$G = \overline{\langle S \rangle}$$



# Semisimple and Unipotent

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



## Jordan-Chevalley decomposition

► Each  $g \in G$  can be written as

$$g = g_u g_s$$

$(g_u - I)^n = 0$   
unipotent

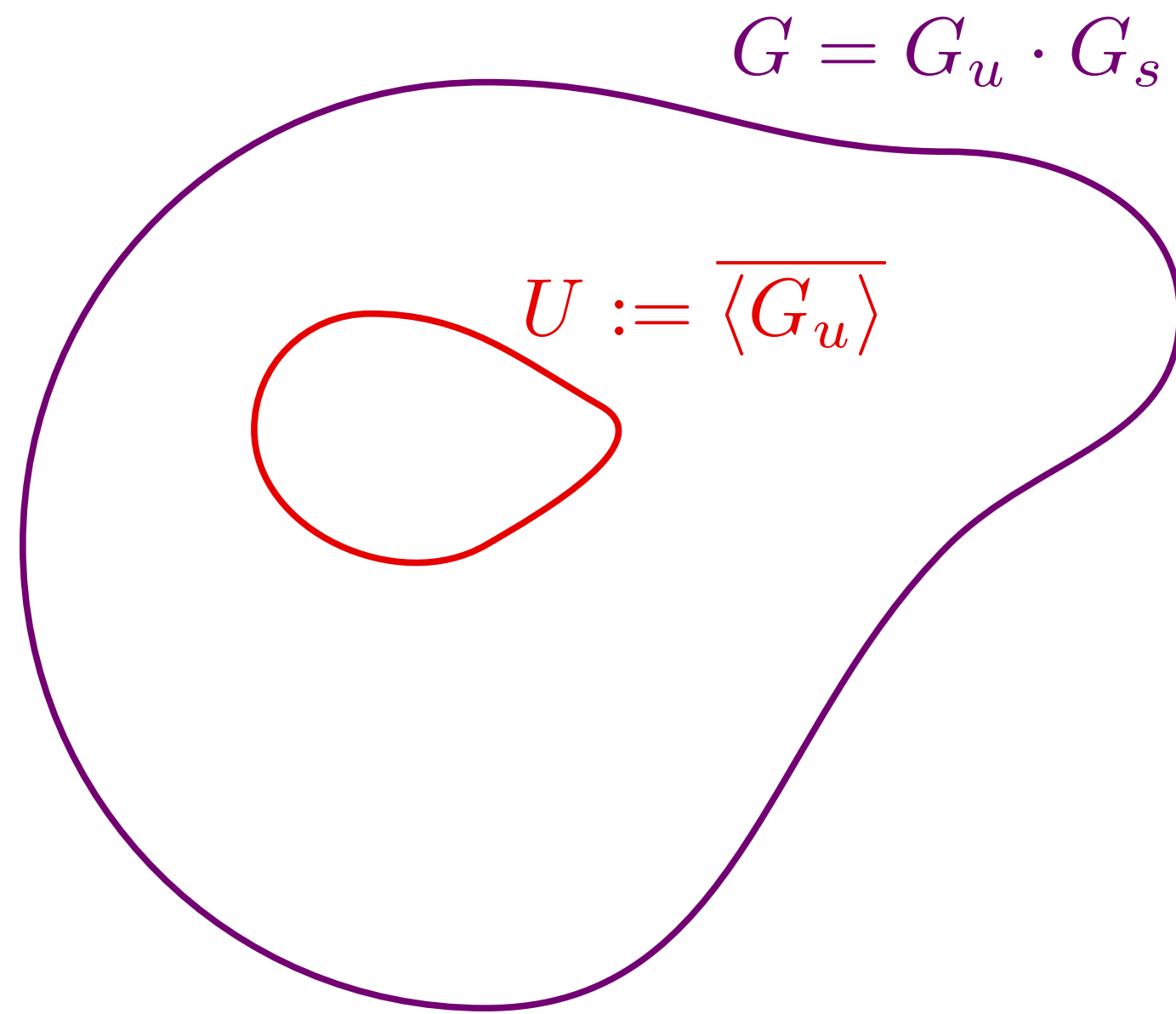
semisimple  
diagonalisable over  $\overline{Q}$

►  $G_u := \{g_u : g \in G\}$  and  $G_s := \{g_s : g \in G\}$  are algebraic sets in  $G$

$$G = G_u \cdot G_s$$

# Unipotent Closure

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$

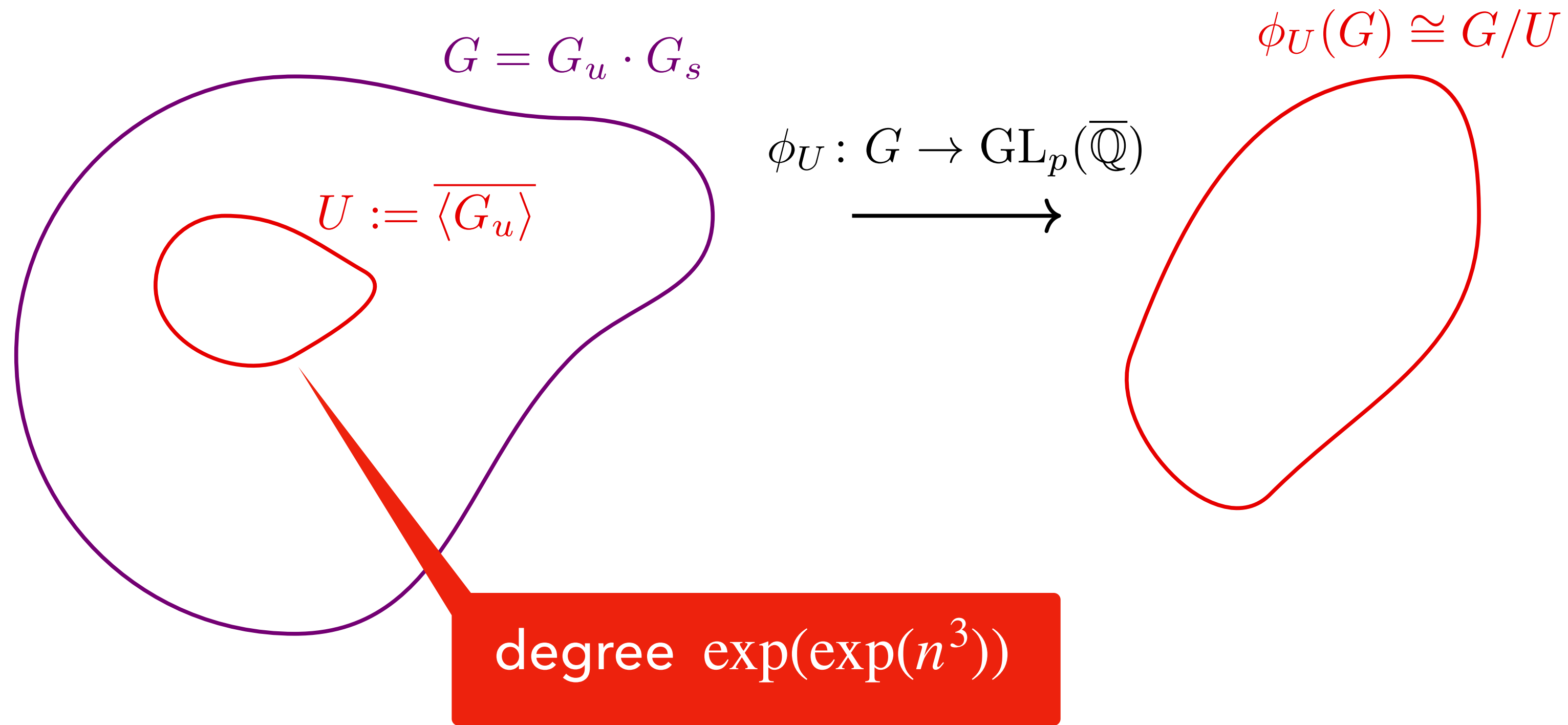


►  $G_u := \{g_u : g \in G\}$  and  $U \trianglelefteq G$



# Unipotent Closure

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$

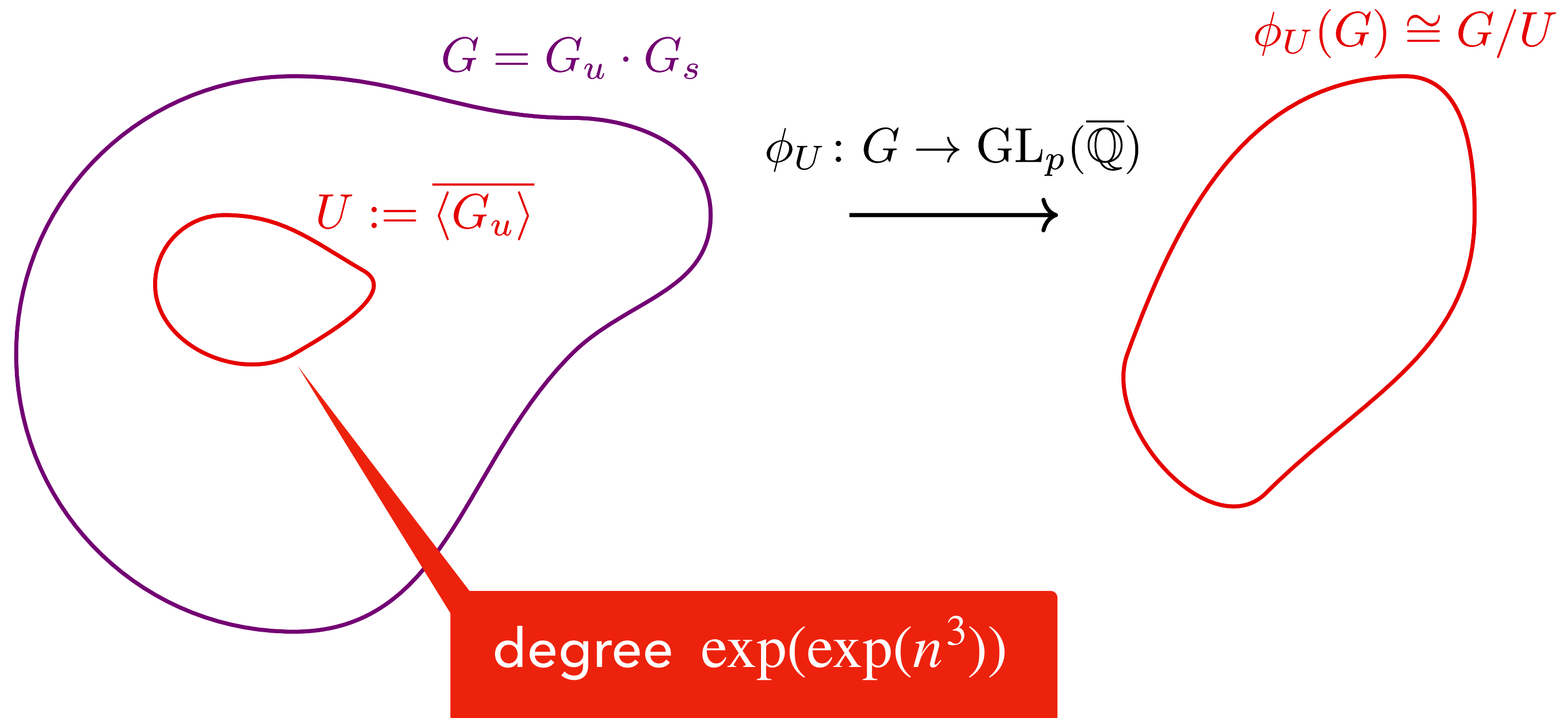


►  $G_u := \{g_u : g \in G\} \text{ and } U \trianglelefteq G$

# Unipotent Closure

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$

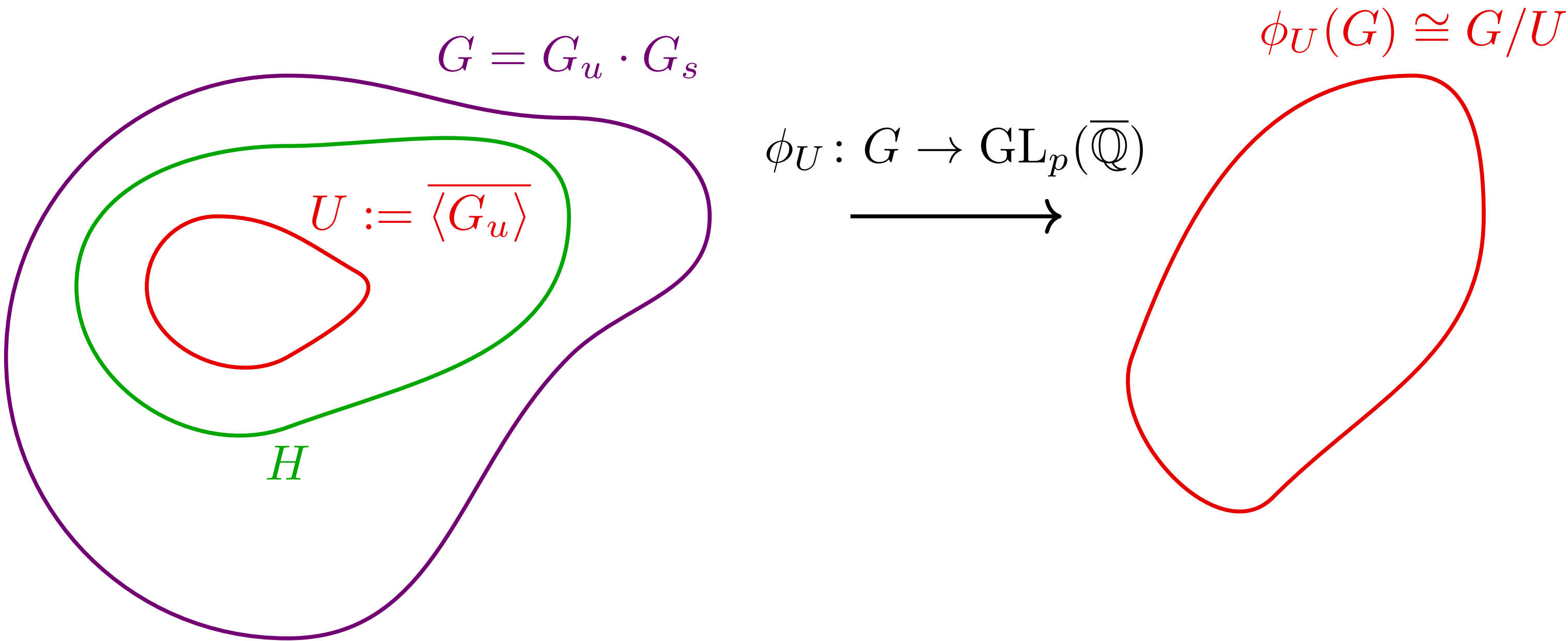
But  $G/U$  is not finite!



►  $G_u := \{g_u : g \in G\}$  and  $U \trianglelefteq G$

# Fictional H

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



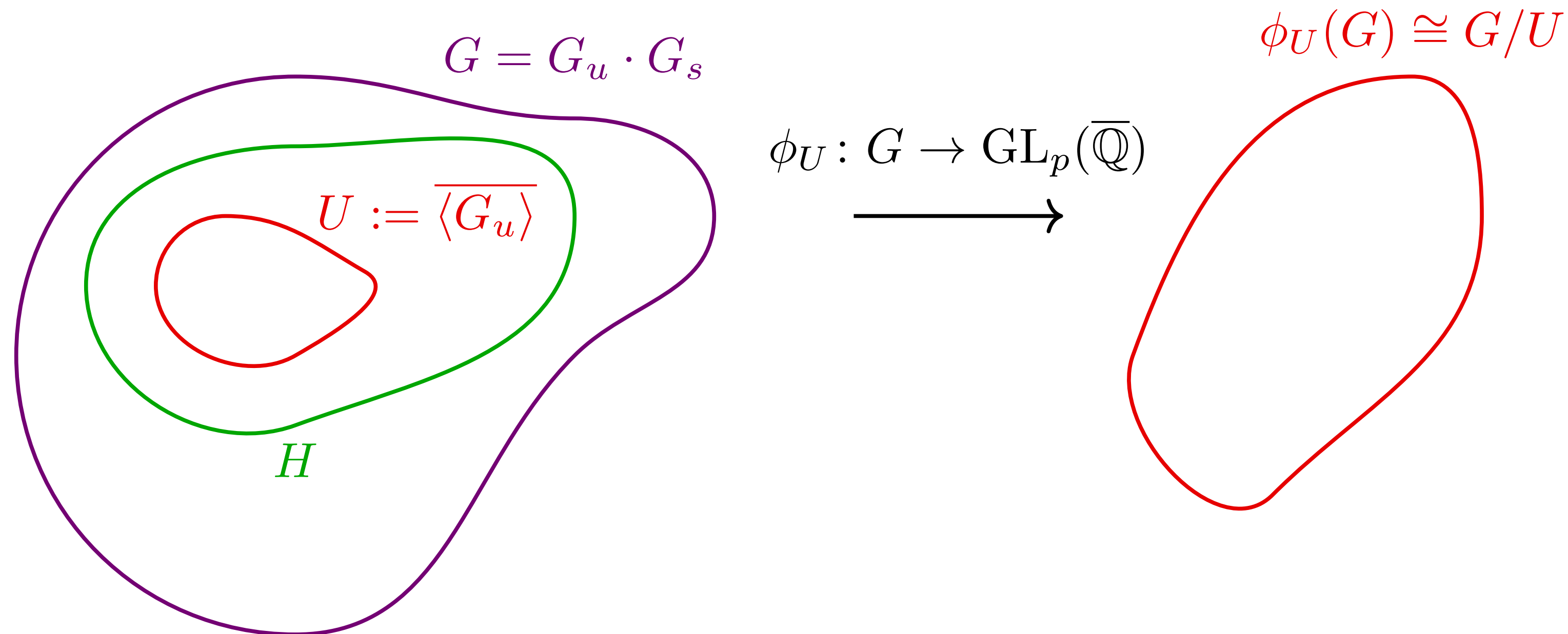
$$U \trianglelefteq H \trianglelefteq G$$

$G^\circ \trianglelefteq H \trianglelefteq G$  so that  
 $G/H$  is also a finite  
rational group!

$H/U$  commutative!

# Fictional H

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



$$U \trianglelefteq H \trianglelefteq G$$

$G^\circ \trianglelefteq H \trianglelefteq G$  so that  $G/H$  is also a finite rational group!

$H/U$  commutative!

Given the index of  $H$  in  $G$ , a variant of Schreier's Lemma gives  $S'$  such that  $H = \overline{\langle S' \rangle}$

Given degree of  $U$ , we can compute degree of  $H$ :

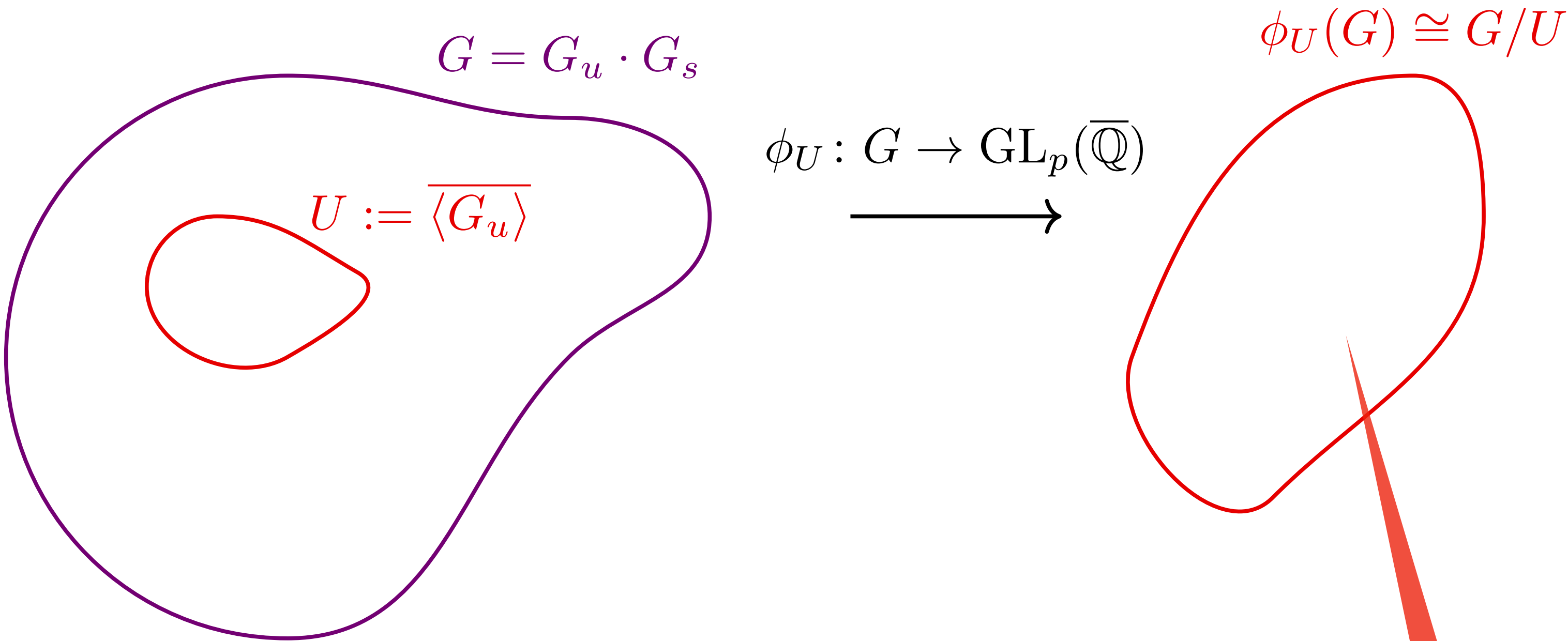
$$H = \overline{\left( \prod_{g \in S'} \langle g \rangle \right)} \cdot U$$

Given the degree of  $H$ , and its index in  $G$  we can compute a degree bound on  $G$ !!!



# Define H

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



$$U \trianglelefteq H \trianglelefteq G$$

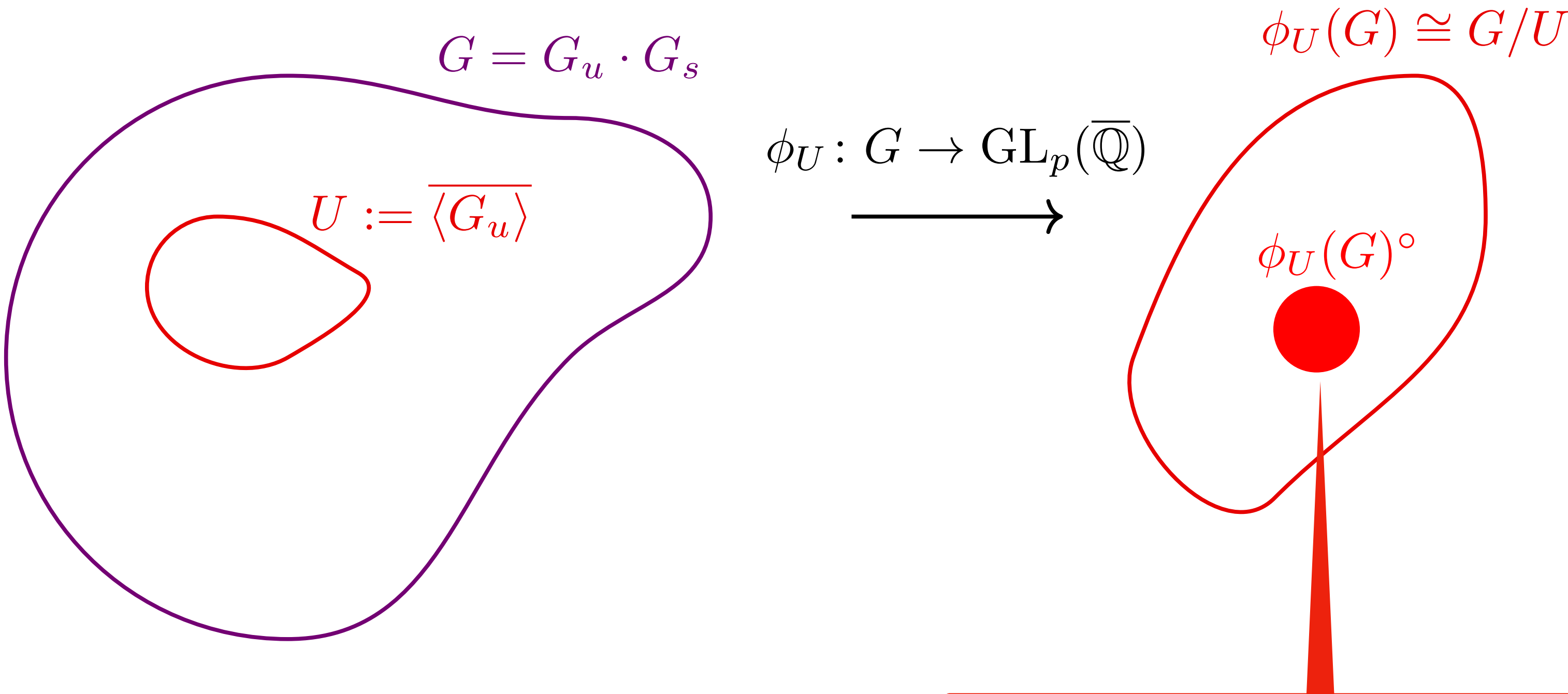
$G^\circ \trianglelefteq H \trianglelefteq G$  so that  
 $G/H$  is also a finite  
rational group!

$H/U$  commutative!

exclusively  
semisimple matrices

# Define H

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



$$U \trianglelefteq H \trianglelefteq G$$

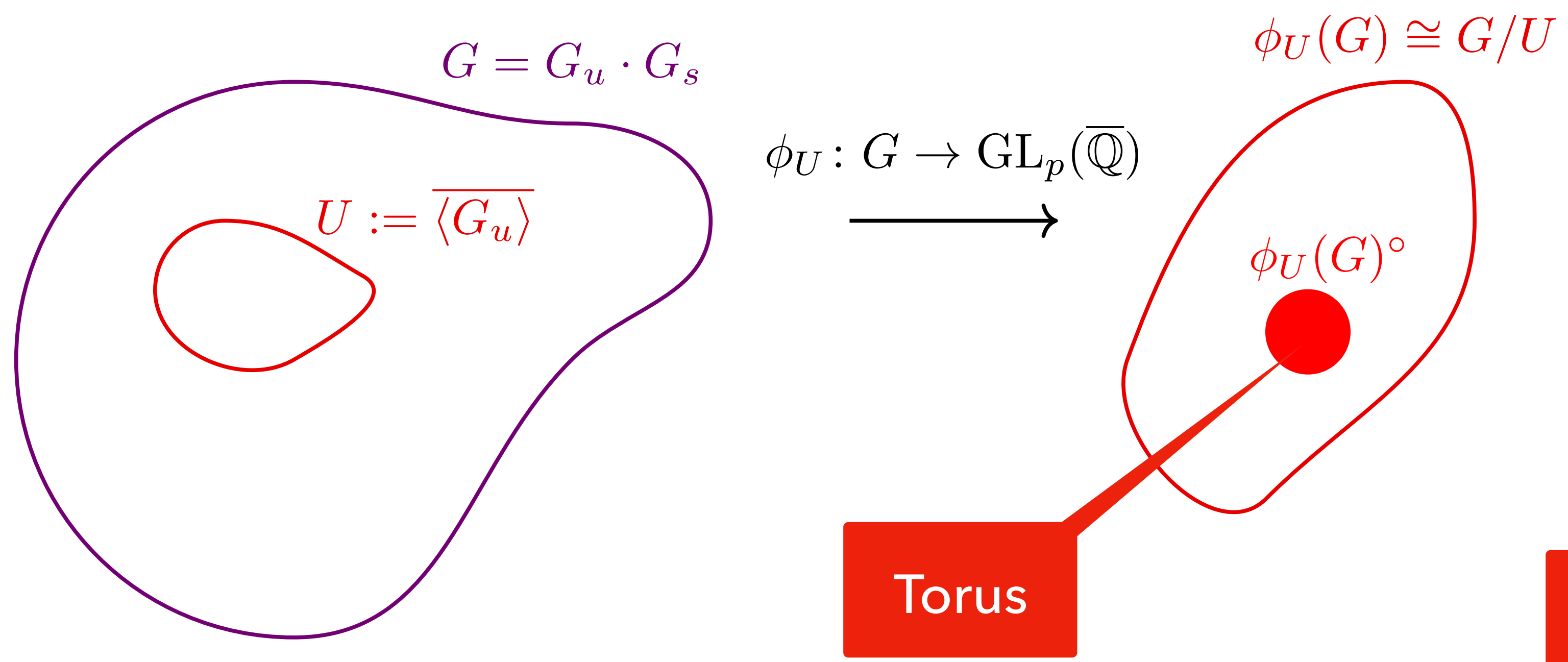
$G^\circ \trianglelefteq H \trianglelefteq G$  so that  
 $G/H$  is also a finite  
rational group!

$H/U$  commutative!

connected  
exclusively semisimple matrices

# Define H

$S \subseteq \mathrm{GL}_n(\mathbb{Q})$



$U \trianglelefteq H \trianglelefteq G$

$G^\circ \trianglelefteq H \trianglelefteq G$  so that  $G/H$  is also a finite rational group!

$H/U$  commutative!

A torus  $T \leq \mathrm{GL}_n(K)$  can be made diagonal

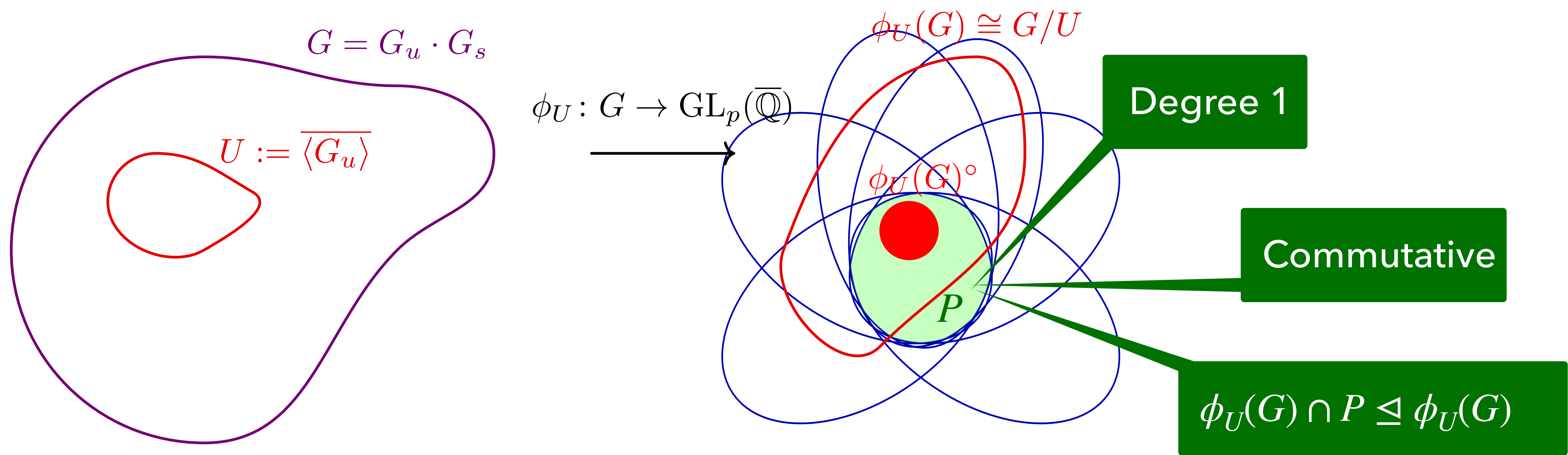
The maximal Tori in  $\mathrm{GL}_n(K)$  are conjugate to the group of diagonal matrices

Degree 1



# Define H

$S \subseteq \mathrm{GL}_n(\mathbb{Q})$

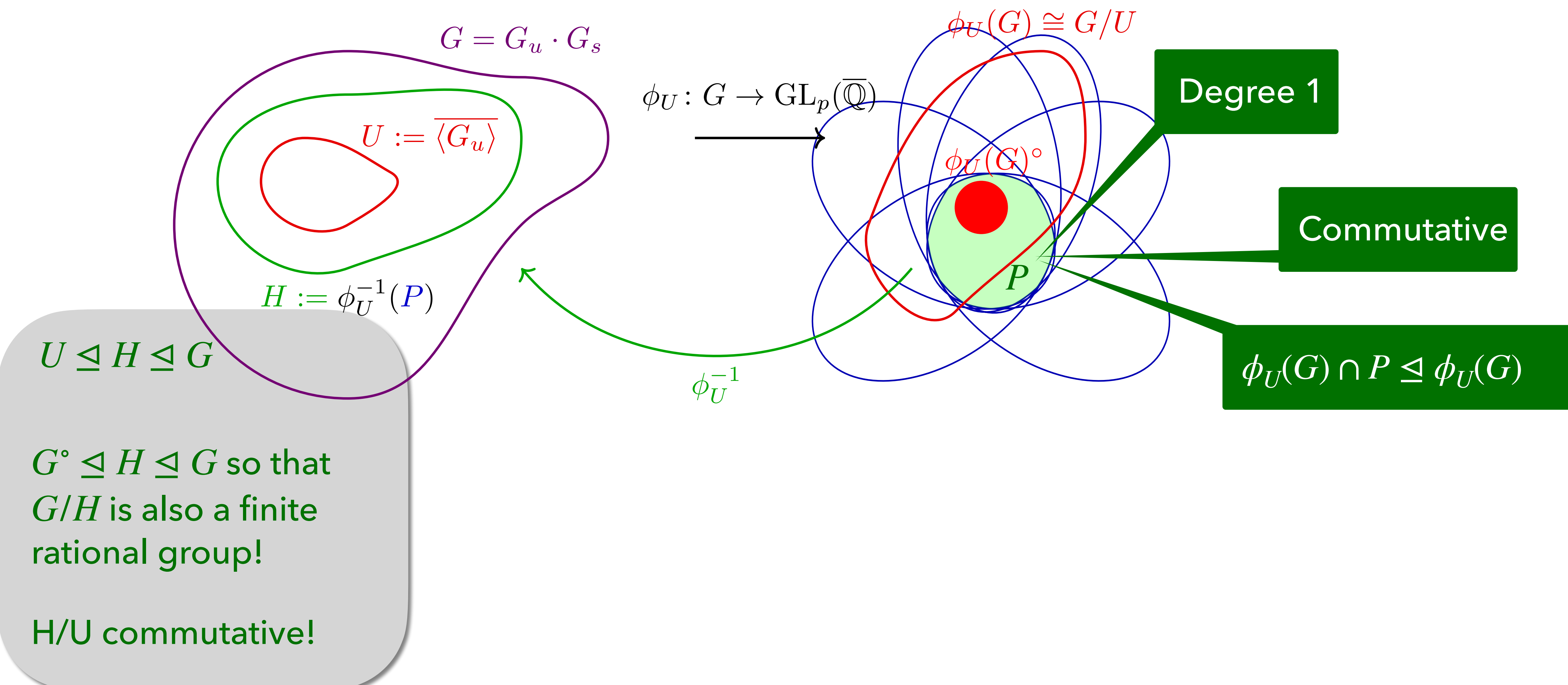


A torus  $T \leq \mathrm{GL}_n(K)$  can be made diagonal

The maximal Tori in  $\mathrm{GL}_n(K)$  are conjugate to the group of diagonal matrices

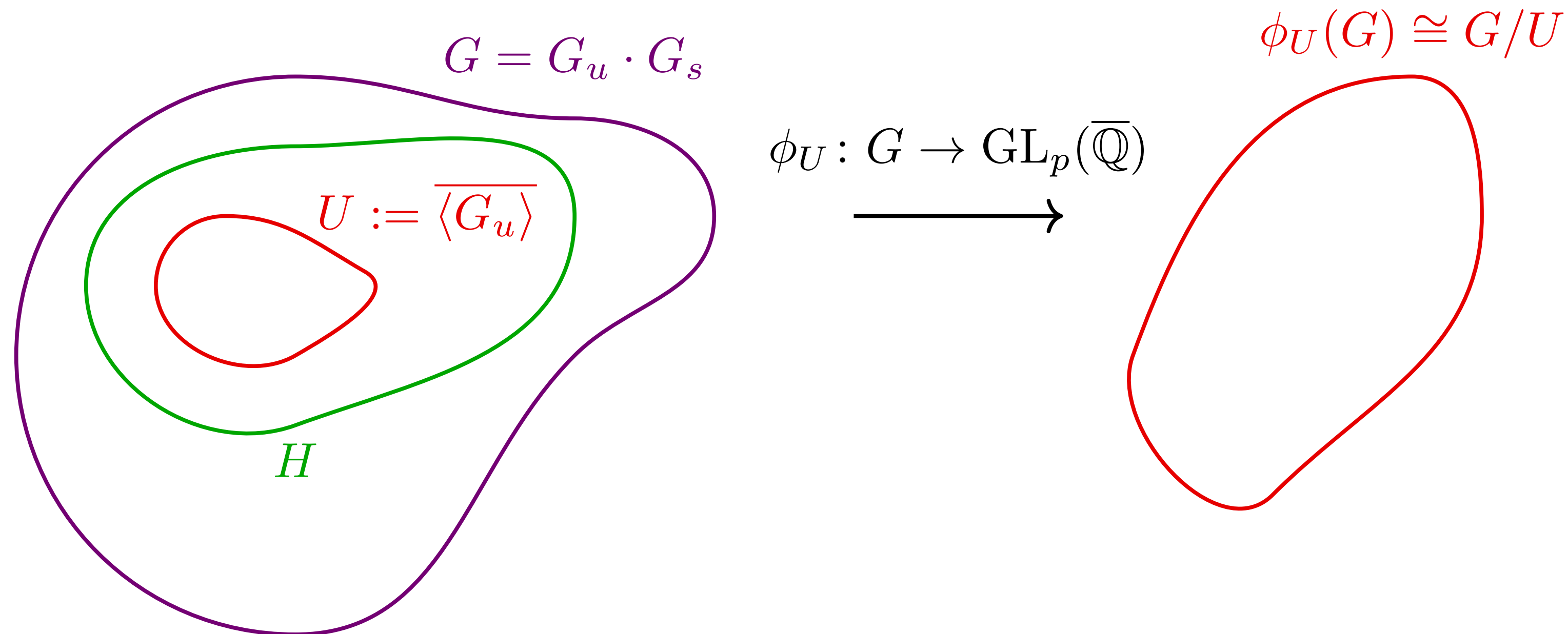
# Define H

$S \subseteq \mathrm{GL}_n(\mathbb{Q})$



# Not Fictional Anymore!

$$S \subseteq \mathrm{GL}_n(\mathbb{Q})$$



$$U \trianglelefteq H \trianglelefteq G$$

$G^\circ \trianglelefteq H \trianglelefteq G$  so that  $G/H$  is also a finite rational group!

$H/U$  commutative!

Given the index of  $H$  in  $G$ , a variant of Schreier's Lemma gives  $S'$  such that  $H = \overline{\langle S' \rangle}$

Given degree of  $U$ , we can compute degree of  $H$ :

$$H = \overline{\left( \prod_{g \in S'} \langle g \rangle \right)} \cdot U$$

Given the degree of  $H$ , and its index in  $G$  we can compute a degree bound on  $G$ !!!



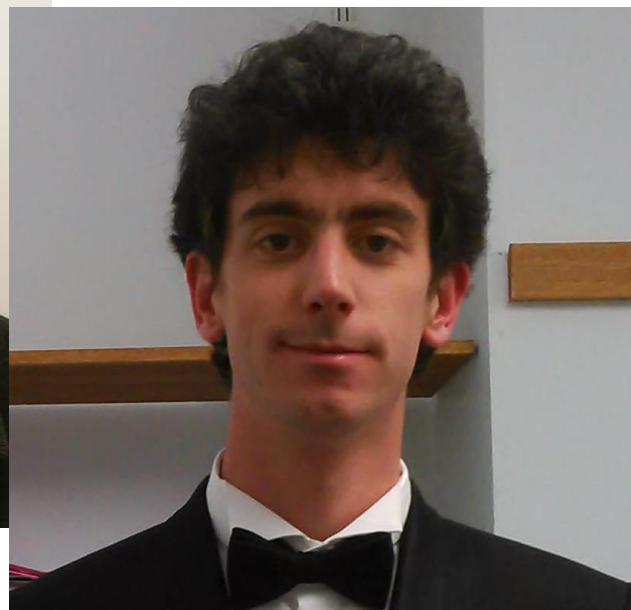
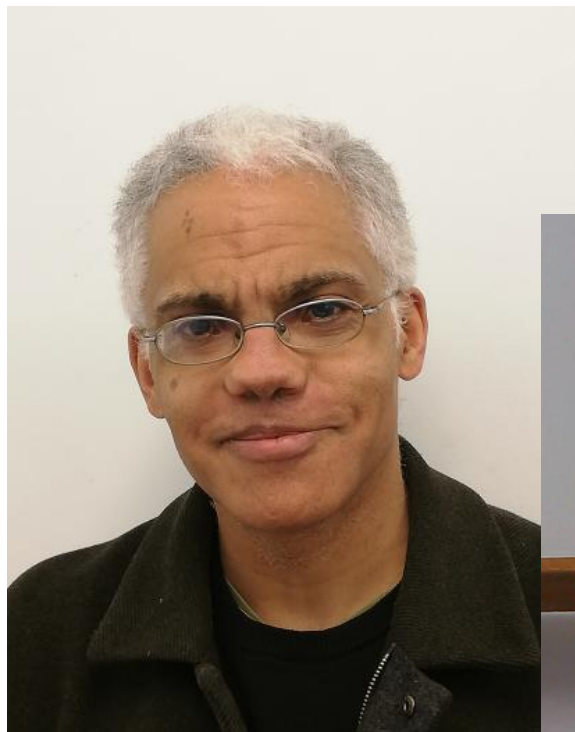
# Group Closure

$$S \subseteq \text{GL}_n(\mathbb{Q})$$

Theorem [ISSAC'22]

The closure  $\overline{\langle S \rangle}$  can be computed in  $10 \text{ EXPTIME}(\text{size}(S), n)$ .

J. Worrell



A. Pouly



S. Schmitz



K. Nosan



M. Shirmohammadi

# Cyclic Groups/Semigroups

Theorem [POPL'25]

Computation of  $\overline{\langle M \rangle v}$  is in PTIME (previously 2EXPSPACE).

$w \in \overline{\langle M \rangle v}$  reduces in PTIME to ACIT

R. Ait El Manssour



G. Kenison

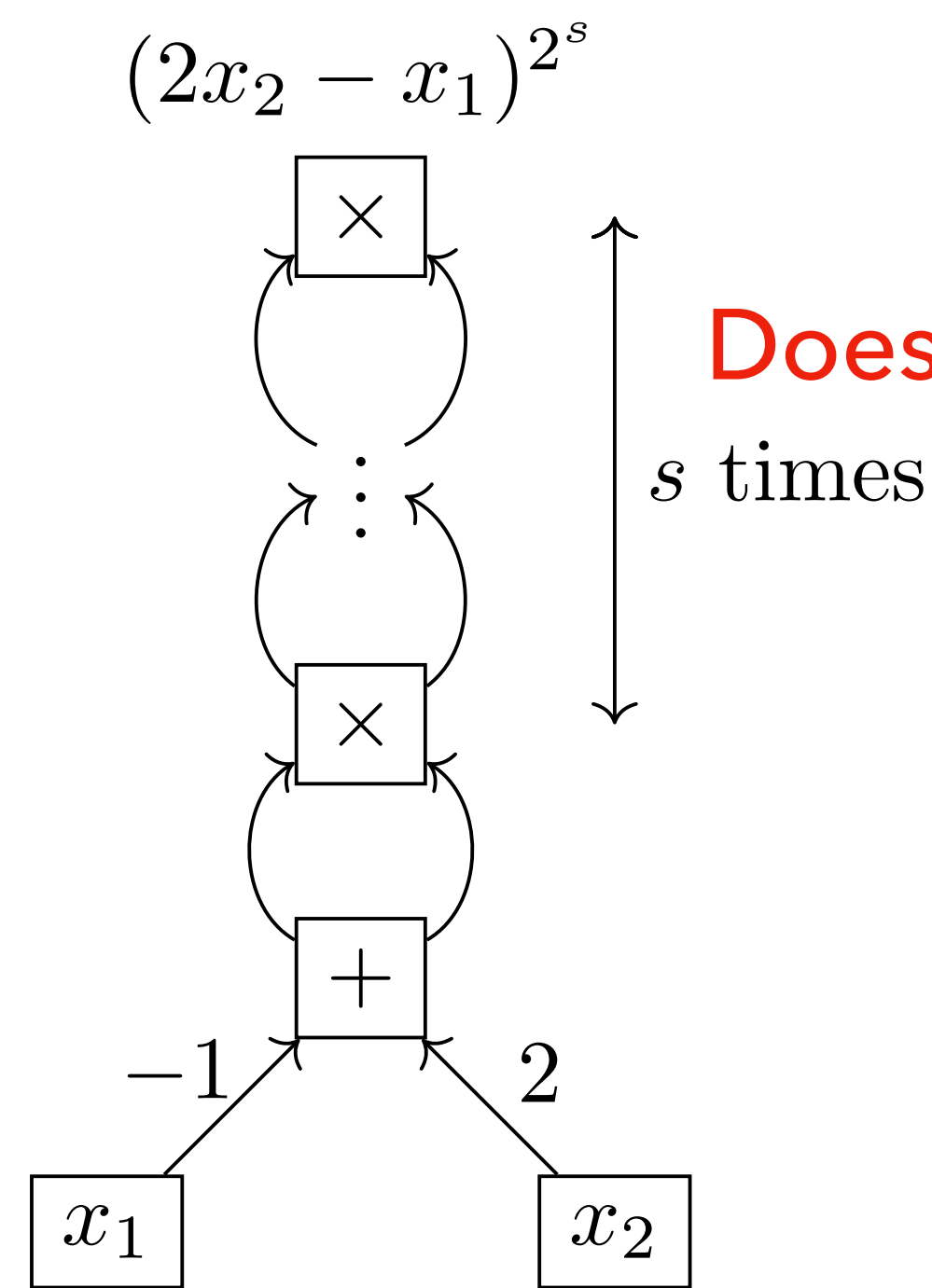


A. Varonka



M. Shirmohammadi

# Cyclic Groups/Semigroups



Theorem [POPL'25]  
Computation of  $\overline{\langle M \rangle} v$  is in PTIME (previously 2EXPSpace).  
 $w \in \overline{\langle M \rangle} v$  reduces in PTIME to ACIT



A. Varonka



G. Kenison



R. Ait El Manssour



M. Shirmohammadi



# Identity Problems over number fields

$(2 \times 3 - 4)^{2^s} = 0?$ <p>A circuit diagram for the identity problem <math>(2 \times 3 - 4)^{2^s} = 0?</math>. It features two input boxes labeled <math>x_1</math> and <math>x_2</math>. <math>x_1</math> has an upward arrow from the constant 4, and <math>x_2</math> has an upward arrow from the constant 3. Both inputs feed into an addition node (+). The output of the addition node feeds into a multiplication node (<math>\times</math>), which is part of a stack of <math>s</math> identical multiplication nodes. Each multiplication node also receives a constant input of -1 from the left. The final output of the top multiplication node is the result of the expression <math>(2 \times 3 - 4)^{2^s}</math>.</p> <p>randomised ptime ACIT</p>	$(2\sqrt{2} - \sqrt{7})^{2^s} = 0?$ <p>A circuit diagram for the identity problem <math>(2\sqrt{2} - \sqrt{7})^{2^s} = 0?</math>. It features two input boxes labeled <math>x_1</math> and <math>x_2</math>. <math>x_1</math> has an upward arrow from the constant <math>\sqrt{7}</math>, and <math>x_2</math> has an upward arrow from the constant <math>\sqrt{2}</math>. Both inputs feed into an addition node (+). The output of the addition node feeds into a multiplication node (<math>\times</math>), which is part of a stack of <math>s</math> identical multiplication nodes. Each multiplication node also receives a constant input of -1 from the left. The final output of the top multiplication node is the result of the expression <math>(2\sqrt{2} - \sqrt{7})^{2^s}</math>.</p> <p>randomised ptime [LICS'22]</p>	$(2\sqrt[2]{2} - \sqrt[2]{7})^{2^s} = 0?$ <p>A circuit diagram for the identity problem <math>(2\sqrt[2]{2} - \sqrt[2]{7})^{2^s} = 0?</math>. It features two input boxes labeled <math>x_1</math> and <math>x_2</math>. <math>x_1</math> has an upward arrow from the constant <math>\sqrt[2]{7}</math>, and <math>x_2</math> has an upward arrow from the constant <math>\sqrt[2]{2}</math>. Both inputs feed into an addition node (+). The output of the addition node feeds into a multiplication node (<math>\times</math>), which is part of a stack of <math>s</math> identical multiplication nodes. Each multiplication node also receives a constant input of -1 from the left. The final output of the top multiplication node is the result of the expression <math>(2\sqrt[2]{2} - \sqrt[2]{7})^{2^s}</math>.</p> <p>coNP [LICS'22]</p>	$(2e^{\frac{2\pi i}{5}} - e^{\frac{2\pi i}{3}})^{2^s} = 0?$ <p>A circuit diagram for the identity problem <math>(2e^{\frac{2\pi i}{5}} - e^{\frac{2\pi i}{3}})^{2^s} = 0?</math>. It features two input boxes labeled <math>x_1</math> and <math>x_2</math>. <math>x_1</math> has an upward arrow from the constant <math>e^{\frac{2\pi i}{3}}</math>, and <math>x_2</math> has an upward arrow from the constant <math>e^{\frac{2\pi i}{5}}</math>. Both inputs feed into an addition node (+). The output of the addition node feeds into a multiplication node (<math>\times</math>), which is part of a stack of <math>s</math> identical multiplication nodes. Each multiplication node also receives a constant input of -1 from the left. The final output of the top multiplication node is the result of the expression <math>(2e^{\frac{2\pi i}{5}} - e^{\frac{2\pi i}{3}})^{2^s}</math>.</p> <p>randomised ptime [ISASC'21]</p>
---	--	--	---

J.Worrell



K.Nosan



S.Perifel



N.Balaji





# Program Synthesis



Given a safe invariant, and a partial program can you complete it properly?

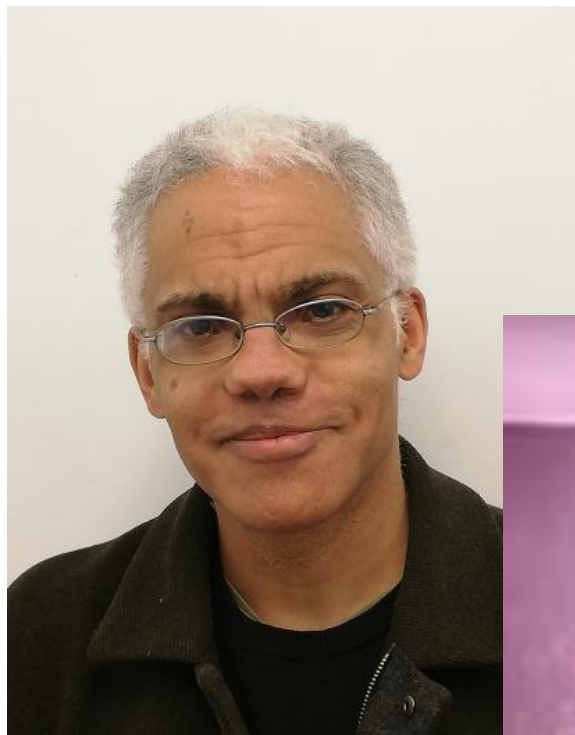
$$\overline{Gv} = \langle x^4 + y^4 - 2x^3y - x^2y^2 + 2xy^3 - 1 \rangle$$

```
x := ?;  y := 0;
while TRUE do
  (x y) := (1 ?) (x y)
            (? 0)
done
```

J. Worrell



A. Varonka



R. Ait El Manssour



G. Kenison



M. Shirmohammadi

# Commutative Groups/Semigroups

Theorem [Preprint]

$w \in \underbrace{\langle M_\sigma \mid \sigma \in \Sigma \rangle}_{\text{Commutative}} v$  is in PH.

Commutative

Reduce the problem to Radical Ideal Membership

J. Worrell



A. Varonka



R. Ait El Manssour



G. Kenison



M. Shirmohammadi



# Interprocedural Programs

```
procedure  $P()$ 
begin
  if  $(*)$  then
     $x := Ax$ 
    call  $P()$ 
     $x := Bx$ 
  else
    skip
  endif
end
```

```
procedure  $Q()$ 
begin
  if  $(*)$  then
     $x := Ax$ 
    call  $Q()$ 
     $x := Bx$ 
    call  $Q()$ 
  else
    skip
  endif
end
```

[Submitted]

What about orbit-closure in recursive programs?

J. Worrell



M. Naraghi



R. Ait El Manssour



M. Shirmohammadi

# WHAT?!

## Theorem [Preprint]

Given a finite semigroup  $S = \langle A_1, \dots, A_r \rangle \subseteq Q^{n \times n}$ , every matrix  $A \in S$  has polynomial bitsize in size of generators  $A_i$ .



N. Lhote



R. Guttenburg

J. Worrell



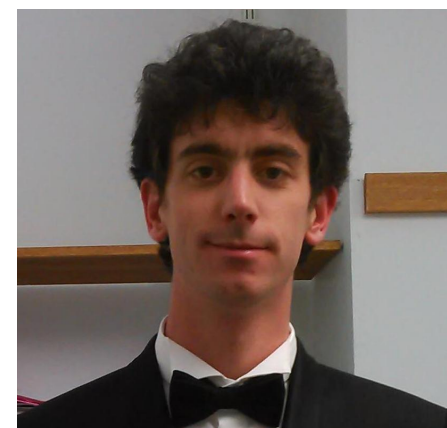
R. Ait El Manssour



M. Shirmohammadi



# Orbit and Group-Closure



M. Shirmohammadi