Condition Numbers & Probability for explaining Algorithms in Computational Geometry

Josué Tonelli-Cueto
UT San Antonio

MINDS & CIS
Seminar Series
6/sep/2022

Fun Fact of the Day (6/SEP/2022)

Source: Donna Haraway — Storytelling for Earthly Survival

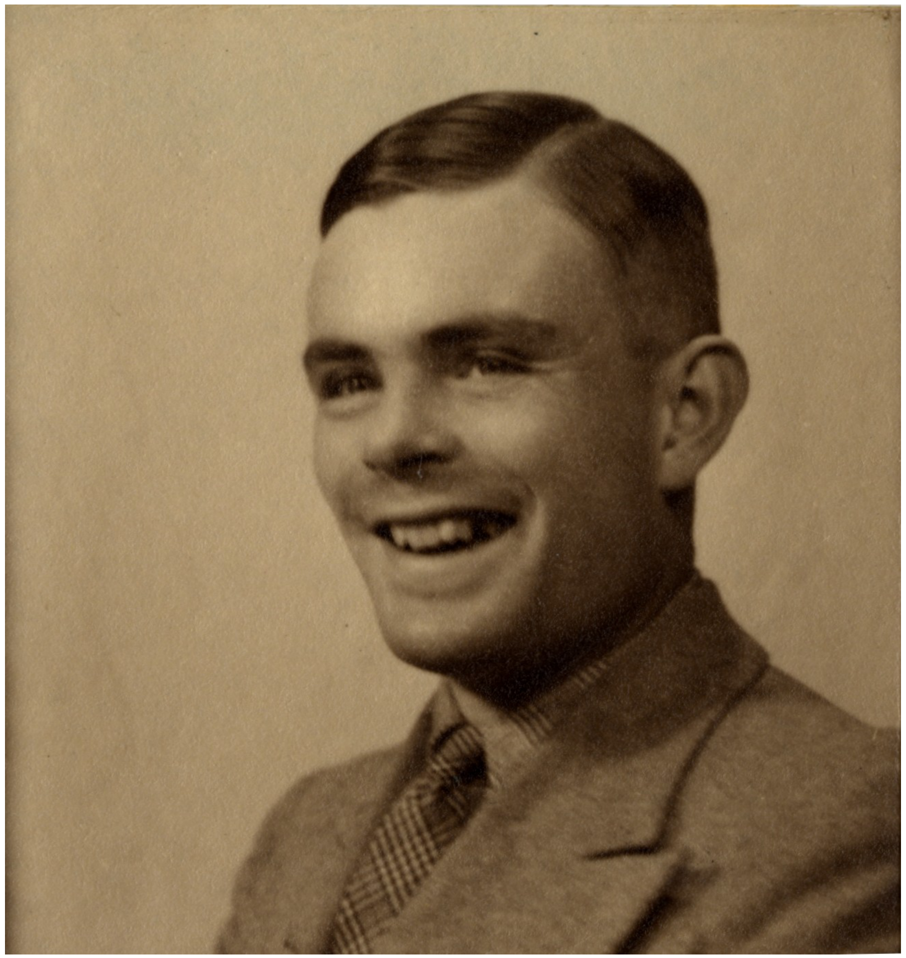
Birthday of Donna Haraway (79 years old)

— among other things, known by the Cyborg Manifesto

Why some algorithms
work a lot better
than predicted?

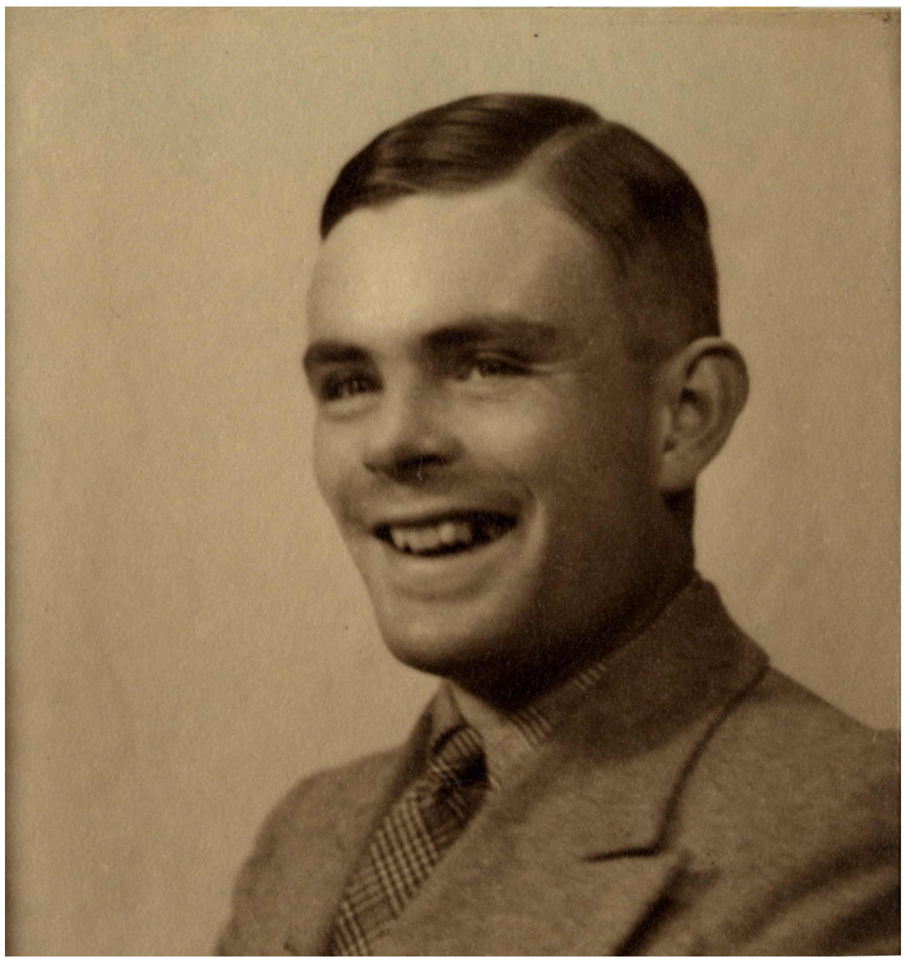
The Foundational Myth:

The Foundational Myth: Turing

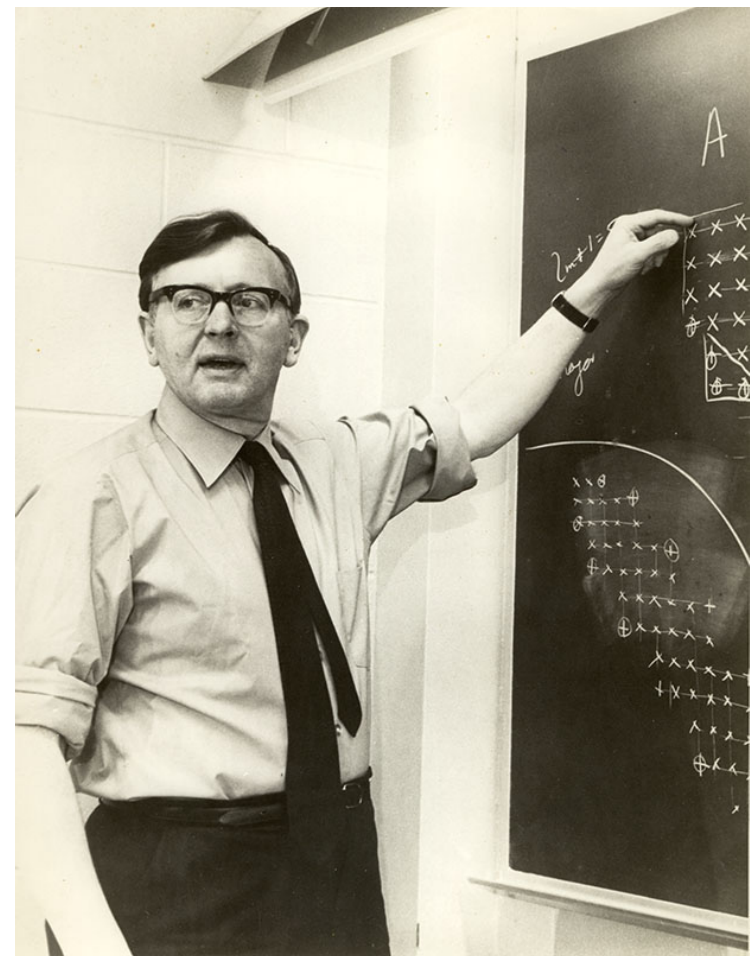


Source: King's College [ATM/K/7/11]

The Foundational Myth: Turing vs. Wilkinson



Source: King's College [ATM/K/7/11]



Source: Higham's web

The Foundational Myth:
Turing vs. Wilkinson

However, it happened that some time after my arrival, a system of 18 equations arrived in Mathematics Division and after talking around it for some time we finally decided to abandon theorizing and to solve it. A system of 18 is surprisingly formidable, even when one has had previous experience with 12, and we accordingly de-

Wilkinson, 1970 Turing Lecture

THE FOUNDATIONAL MYTH:
Turing vs. Wilkinson

However, it happened that some time after my arrival, a system of 18 equations arrived in Mathematics Division and after talking around it for some time we finally decided to abandon theorizing and to solve it. A system of 18 is surprisingly formidable, even when one has had previous experience with 12, and we accordingly decided on a joint effort. The operation was manned by Fox, Goodwin, Turing, and me, and we decided on Gaussian elimination with complete pivoting. Turing was not particularly enthusiastic, partly because he was not an experienced performer on a desk machine and partly because he was convinced that it would be a failure. History repeated itself remarkably closely. Again the system was mildly ill-conditioned, the last equation had a coefficient of order 10^{-4} (the original coefficients being of order unity) and the residuals were again of order 10^{-10}, that is of the size corresponding to the exact solution rounded to ten decimals. It is interesting that in connection with this example we subsequently performed one or two steps of what would now be called "iterative refinement," and this convinced us that the first solution had had almost six correct figures.

Wilkinson, 1970 Turing Lecture

The Foundational Myth:

Turing vs. Wilkinson

I suppose this must be regarded as a defeat for Turing since he, at that time, was a keener adherent than any of the rest of us to the pessimistic school. However, I'm sure that this experience made quite an impression on him and set him thinking afresh on the problem of rounding errors in elimination processes. About a year later he produced his famous paper "Rounding-off errors in matrix processes" [1] which together with the paper of J. von Neumann and H. Goldstine [4] did a great deal to dispel the gloom. The second round undoubtedly went to Turing!

Wilkinson, 1970 Turing Lecture

THE FOUNDATIONAL MYTH:
Turing vs. Wilkinson

I suppose this must be regarded as a defeat for Turing since he, at that time, was a keener adherent than any of the rest of us to the pessimistic school. However, I'm sure that this experience made quite an impression on him and set him thinking afresh on the problem of rounding errors in elimination processes. About a year later he produced his famous paper "Rounding-off errors in matrix processes" [1] which together with the paper of J. von Neumann and H. Goldstine [4] did a great deal to dispel the gloom. The second round undoubtedly went to Turing!

ROUNDING-OFF ERRORS IN MATRIX PROCESSES

By A. M. TURING

(*National Physical Laboratory, Teddington, Middlesex*)

[Received 4 November 1947]

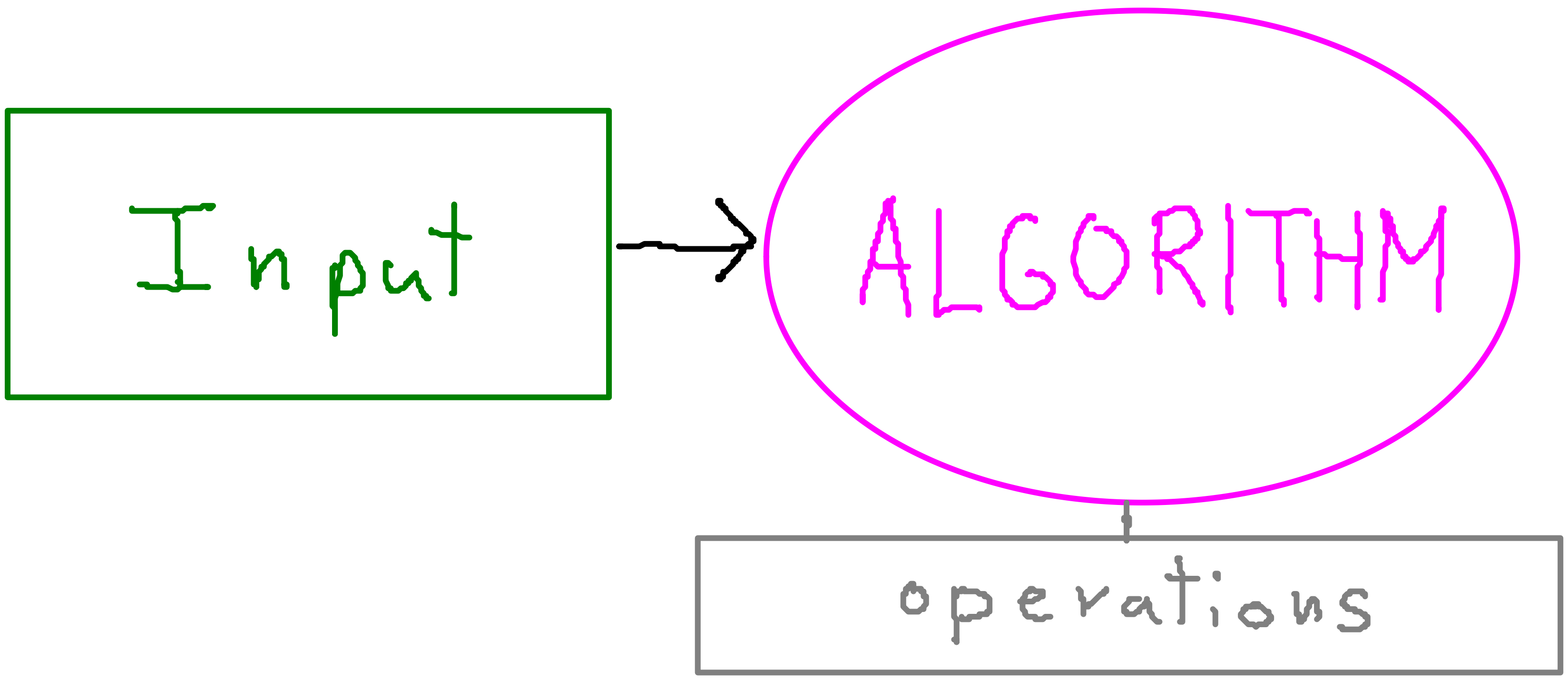
Wilkinson, 1970 Turing Lecture

Complexity of (Traditional) Algorithms

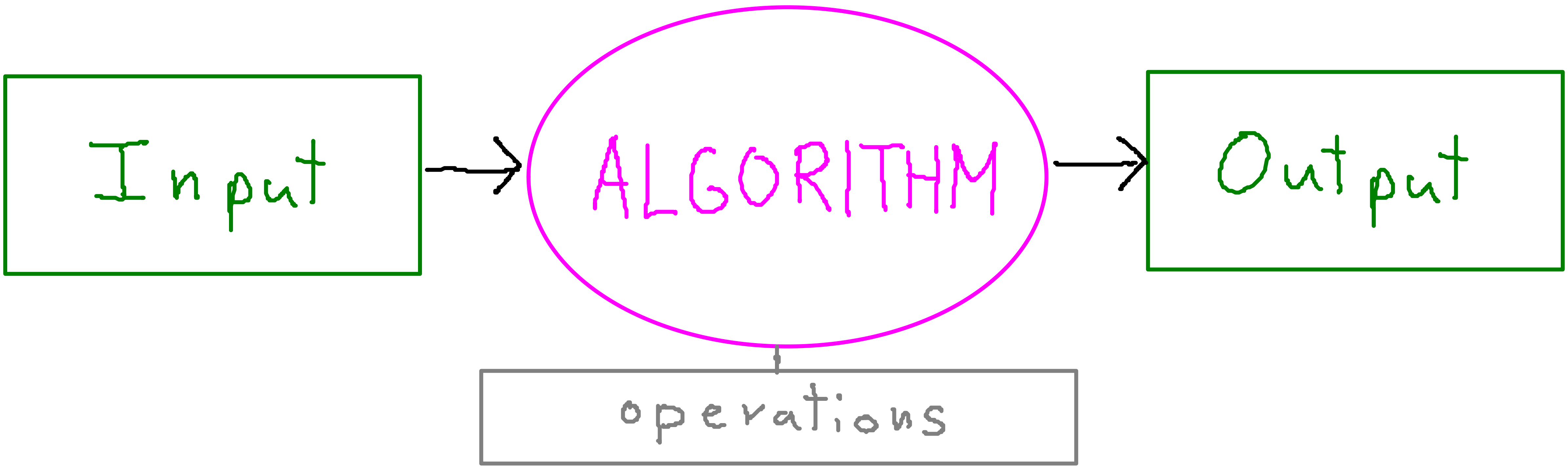
Complexity of (Traditional) Algorithms

Input

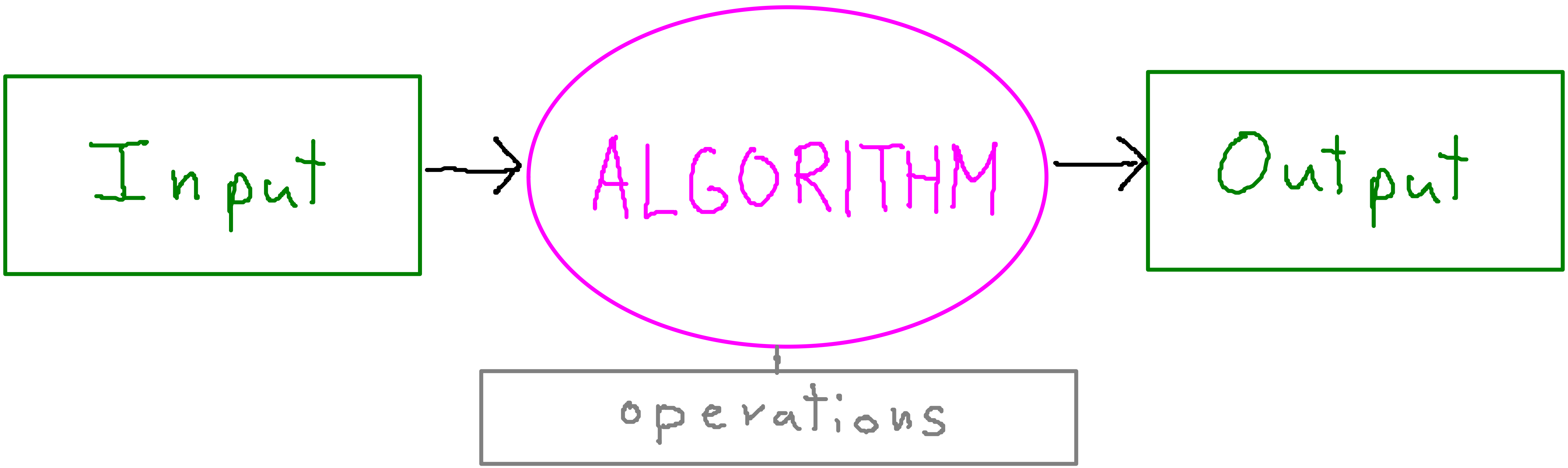
Complexity of (Traditional) Algorithms



Complexity of (Traditional) Algorithms



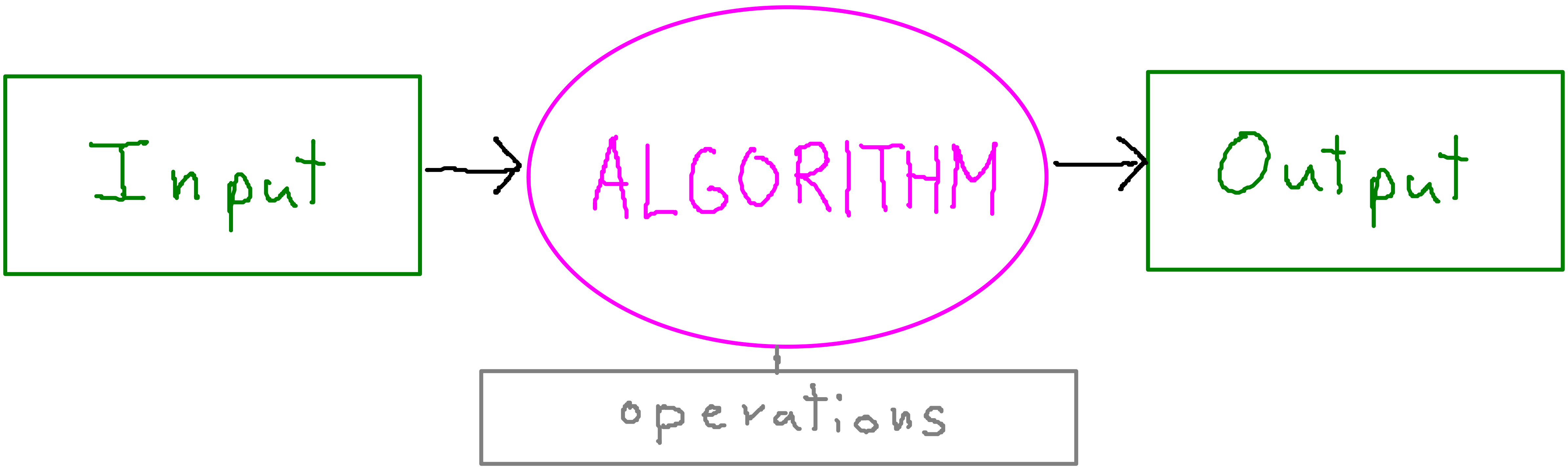
Complexity of (Traditional) Algorithms



Worst-case form of complexity estimate:

$$\text{run-time}(\text{ALGORITHM}, \text{Input}) \leq f(\text{size}(\text{Input}))$$

Complexity of (Traditional) Algorithms

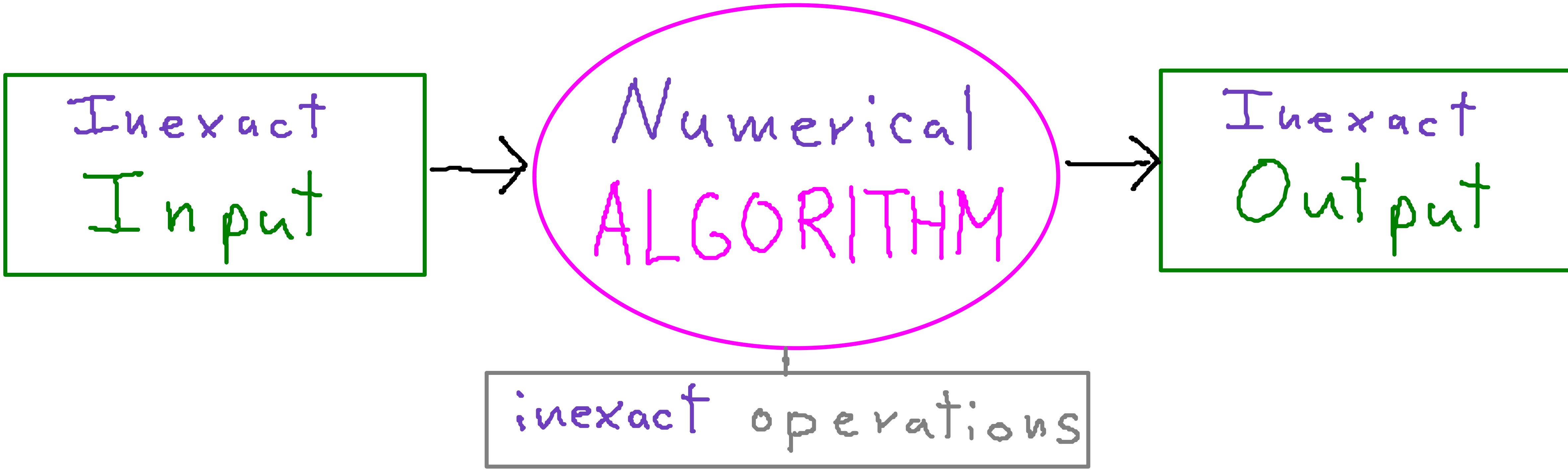


Worst-case form of complexity estimate:

$$\text{run-time}(\text{ALGORITHM}, \text{Input}) \leq f(\text{size}(\text{Input}))$$

⚠ sometimes size has several parameters
(e.g. #variables, degree...)

Complexity of Numerical Algorithms I



⚠ usual form of complexity fails!

ALL INPUTS OF THE SAME SIZE ARE EQUAL, BUT SOME INPUTS ARE MORE EQUAL THAN OTHERS

Complexity of Numerical Algorithms II

Condition-based complexity { (Turing)
(Goldstine, von Neumann)

Complexity of Numerical Algorithms II

Condition-based complexity (Turing) (Goldstine, von Neumann)

cond(Input) : measures numerical sensitivity of Input

Complexity of Numerical Algorithms II

Condition-based complexity | (Turing)
(Goldstine, von Neumann)

cond(Input) : measures numerical sensitivity of Input

cond big $\Rightarrow$ $\boxed{\text{small variations of Input} \longrightarrow \text{big variations of Output}}$

Complexity of Numerical Algorithms II

Condition-based complexity (Turing) (Goldstine, von Neumann)

cond(Input) : measures numerical sensitivity of Input

cond big ⇒ $\boxed{\text{small variations of Input} \longrightarrow \text{big variations of Output}}$

cond small ⇒ $\boxed{\text{'big' variations of Input} \longrightarrow \text{small variations of Output}}$

Complexity of Numerical Algorithms II

Condition-based complexity! (Turing) (Goldstine, von Neumann)

cond(Input) : measures numerical sensitivity of Input

cond big $\Rightarrow$ $\boxed{\text{small variations of Input} \longrightarrow \text{big variations of Output}}$

cond small $\Rightarrow$ $\boxed{\text{'big' variations of Input} \longrightarrow \text{small variations of Output}}$

⚠ cond is a property of the computational problem, not of the algorithm!

Complexity of Numerical Algorithms II

Condition-based complexity II (Turing)
(Goldstine, von Neumann)

Complexity of Numerical Algorithms III

Condition-based complexity II (Turing) (Goldstine, von Neumann)

Condition-based form of complexity estimates

$$\text{run-time}(\text{ALGORITHM}, \text{Input}) \leq g(\text{size}(\text{Input}), \text{cond}(\text{Input}))$$

Complexity of Numerical Algorithms II

Condition-based complexity II (Turing)
(Goldstine, von Neumann)

Condition-based form of complexity estimates

run-time(ALGORITHM, Input) $\leq$ g(size(Input), cond(Input))

Can we have a complexity estimate
of a numerical algorithm only depending on size?

Complexity of Numerical Algorithms IV

Probabilistic complexity I (Goldstine, von Neumann)
(Smale) (Demmel)

Complexity of Numerical Algorithms IV

Probabilistic complexity I (Goldstine, von Neumann) (Smale) (Demmel)

Statistical complexity might have been a better name

Complexity of Numerical Algorithms IV

Probabilistic complexity I (Goldstine, von Neumann) (Smale) (Demmel)

Can we have a complexity estimate of a numerical algorithm only depending on size?

Statistical complexity might have been a better name

Complexity of Numerical Algorithms IV

Probabilistic complexity I (Goldstine, von Neumann) (Smale) (Demmel)

Can we have a complexity estimate of a numerical algorithm only depending on size?

Yes, if we randomize the Input

Statistical complexity might have been a better name

Complexity of Numerical Algorithms IV

Probabilistic complexity I (Goldstine, von Neumann)
(Smale) (Demmel)

Can we have a complexity estimate
of a numerical algorithm only depending on size?

Yes, if we randomize the Input

How do we randomize the Input?

Statistical complexity might have been a better name

Complexity of Numerical Algorithms IV

Probabilistic complexity I (Goldstine, von Neumann) (Smale) (Demmel)

Can we have a complexity estimate of a numerical algorithm only depending on size?

Yes, if we randomize the Input

How do we randomize the Input?

We choose the probability distribution depending on the context!

Statistical complexity might have been a better name

Complexity of (Numerical) Algorithms V

Probabilistic complexity II (Goldstine, von Neumann)
(Smale) (Demmel)

Complexity of (Numerical) Algorithms V

Probabilistic complexity II (Goldstine, von Neumann) (Smale) (Demmel)

Probabilistic form of complexity estimates

$$\mathbb{P}_{\text{input}}\left[\text{run-time}(\text{ALGORITHM}, \text{input}) \geq t\right] \leq f(s,t)$$

where $\text{size}(\text{input}) \leq s$

Complexity of (Numerical) Algorithms V

Probabilistic complexity II (Goldstine, von Neumann) (Smale) (Demmel)

Probabilistic form of complexity estimates

$$\mathbb{P}_{\mathit{input}}\left[\text{run-time}(\text{ALGORITHM}, \mathit{input}) \geq t\right] \leq f(s, t)$$

where $\text{size}(\mathit{input}) \leq s$

... and if we are lucky

$$\mathbb{E}_{\mathit{input}}\left[\text{run-time}(\text{ALGORITHM}, \mathit{input}) \geq t\right] \leq f(s)$$

Complexity of (Numerical) Algorithms V

Smoothed complexity] (Spielman, Teng)

Complexity of (Numerical) Algorithms VI

Smoothed complexity] (Spielman, Teng)

Smoothed form of complexity estimates

$$\sup_{\substack{\text{Input} \\ \text{size(Input)}=s}} \mathbb{P}_{\text{noise}}\left[\text{runtime}(\text{ALGORITHM}, \text{Input} + \sigma\,\text{noise}) \geq t\right] \leq f(s,t)/\sigma$$

Complexity of (Numerical) Algorithms V

Smoothed complexity I (Spielman, Teng)

Smoothed form of complexity estimates

$$\sup_{\substack{\text{Input}\\ \text{size}(\text{Input})=s}} \mathbb{P}_{\text{noise}}\left[\text{run-time}(\text{ALGORITHM}, \text{Input}+\sigma\,\text{noise}) \geq t\right] \leq f(s,t)/\sigma$$

... and if we are lucky

$$\sup_{\substack{\text{Input}\\ \text{size}(\text{Input})=s}} \mathbb{E}_{\text{noise}}\left[\text{run-time}(\text{ALGORITHM}, \text{Input}+\sigma\,\text{noise})\right] \leq f(s)/\sigma$$

Complexity of (Numerical) Algorithms VII

Smoothed complexity II (Spielman, Teng)

Complexity of (Numerical) Algorithms VIII

Smoothed complexity II (Spielman, Teng)

Smoothed form of complexity estimates

$$\sup_{\substack{\text{Input} \\ \text{size}(\text{Input})=s}} \mathbb{P}_{\text{noise}}\left[\text{run-time}(\text{ALGORITHM}, \text{Input}+\sigma\,\text{noise}) \geq t\right] \leq f(s, t/\sigma)$$

Complexity of (Numerical) Algorithms VII

Smoothed complexity I (Spielman, Teng)

Worst-case form of complexity estimate

$$\text{run-time}(\text{ALGORITHM}, \text{Input}) \leq f(\text{size}(\text{Input}))$$

$$\uparrow \quad \sigma \to 0$$

Smoothed form of complexity estimates

$$\sup_{\substack{\text{Input} \\ \text{size}(\text{Input})=s}} \mathbb{P}_{\text{noise}}\left[\text{run-time}(\text{ALGORITHM}, \text{Input} + \sigma\,\text{noise}) \geq t\right] \leq f(s, t/\sigma)$$

Complexity of (Numerical) Algorithms VII

Smoothed complexity II (Spielman, Teng)

Worst-case form of complexity estimate

$$\text{run-time}(\text{ALGORITHM}, \text{Input}) \leq f(\text{size}(\text{Input}))$$

$$\uparrow \quad \sigma \to 0$$

Smoothed form of complexity estimates

$$\sup_{\substack{\text{Input} \\ \text{size}(\text{Input})=s}} \mathbb{P}_{\text{noise}}\left[\text{run-time}(\text{ALGORITHM}, \text{Input} + \sigma\,\text{noise}) \geq t\right] \leq f(s, t/\sigma)$$

$$\downarrow \quad \sigma \to \infty$$

Probabilistic form of complexity estimates

$$\mathbb{P}_{\text{input}}\left[\text{run-time}(\text{ALGORITHM}, \text{input}) \geq t\right] \leq f(s, t)$$

Complexity of (Numerical) Algorithms VIII

Success stories

Complexity of (Numerical) Algorithms VIII

Success stories

Solving Linear Equations

See any intro to numerical analysis/random matrix theory

Complexity of (Numerical) Algorithms VIII

Success stories

Solving Linear Equations

See any intro to numerical analysis/random matrix theory

Linear Programming

Complexity of (Numerical) Algorithms VIII

Success stories

Solving Linear Equations

See any intro to numerical analysis/random matrix theory

Linear Programming

Condition Goffin, Renegar, Cheung, Cucker, Peña, ...

Complexity of (Numerical) Algorithms VIII

Success stories

Solving Linear Equations

See any intro to numerical analysis/random matrix theory

Linear Programming

Condition — Goffin, Renegar, Cheng, Cucker, Peña, ...

Prob./Smoothed — Bürgisser, Cucker, Lotz, Dunagan, Spielman, Teng...

Complexity of (Numerical) Algorithms VIII

Success stories

Solving Linear Equations

See any intro to numerical analysis/random matrix theory

Linear Programming

Condition Goffin, Renegar, Cheng, Cucker, Peña, ...

Prob./Smoothed Bürgisser, Cucker, Lotz, Dunagan, Spielman, Teng...

Solving Polynomial Systems

Smale's 17th Problem

Beltrán, Pardo, Bürgisser, Cucker, Lairez...

Complexity of (Numerical) Algorithms VIII

Success stories

Solving Linear Equations

See any intro to numerical analysis/random matrix theory

Linear Programming

Condition Goffin, Renegar, Cheng, Cucker, Peña, ...

Prob./Smoothed Bürgisser, Cucker, Lotz, Dunagan, Spielman, Teng...

Solving Polynomial Systems

Smale's 17th Problem

Beltrán, Pardo, Bürgisser, Cucker, Lairez ...

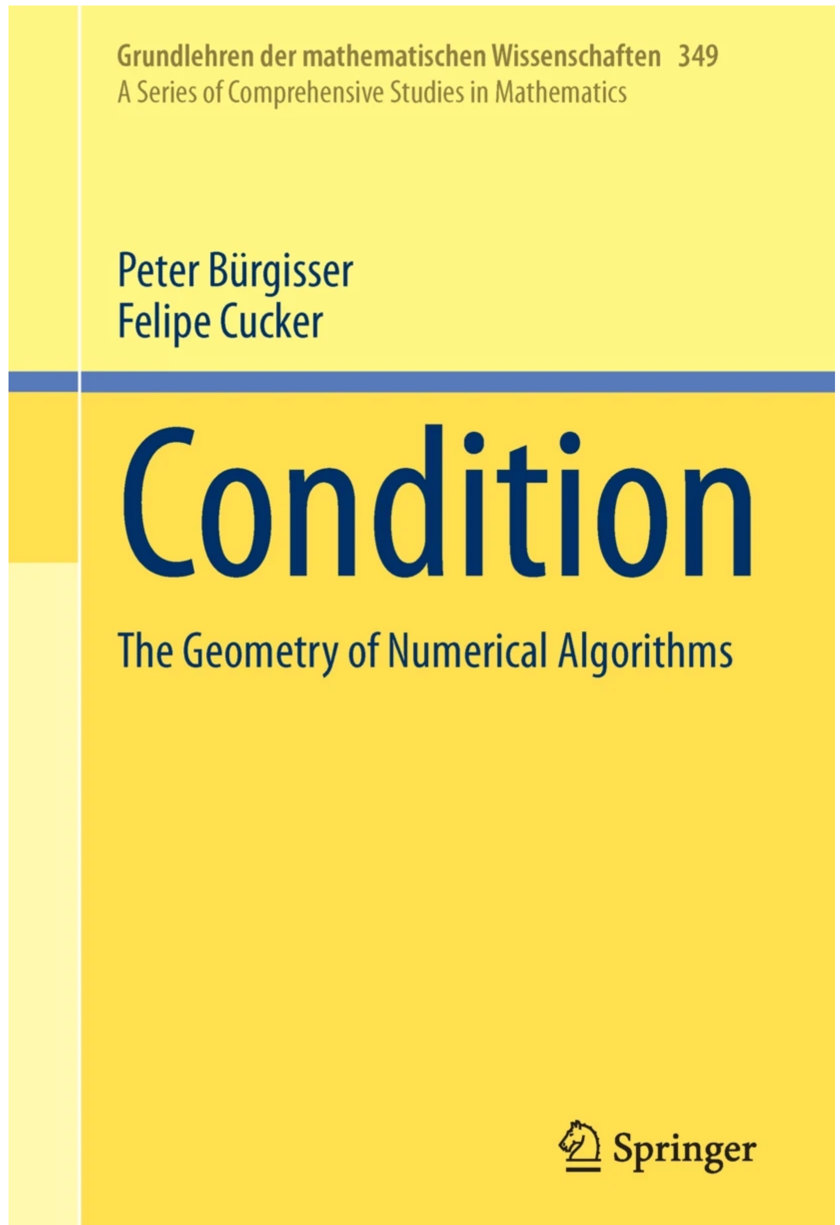
...

Complexity of (Numerical) Algorithms IX

A good introduction...

Complexity of (Numerical) Algorithms IX

A good introduction...



THE FRAMEWORK IN ACTION I

the PLANTINGA-VEGTER algorithm

Joint work with F. Cucker & A. A. Ergür

Plus extra work with E. Tsigaridas

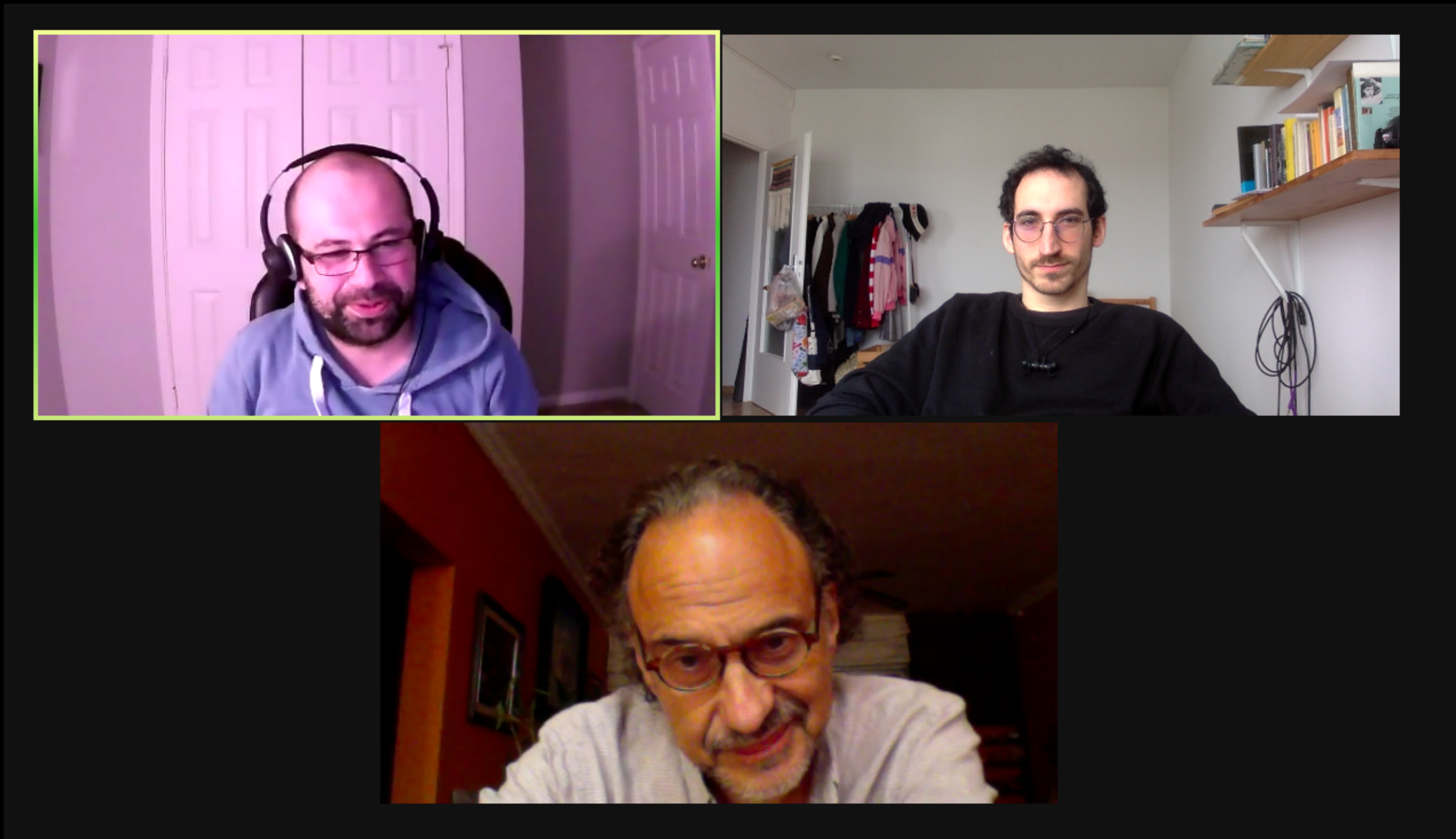


Photo while working on another project

An algorithm for Visualizing Implicit Curves & Surfaces

An algorithm for Visualizing Implicit Curves & Surfaces

Eurographics Symposium on Geometry Processing (2004)
R. Scopigno, D. Zorin, (Editors)

Isotopic Approximation of Implicit Curves and Surfaces

Simon Plantinga and Gert Vegter

Institute for Mathematics and Computing Science
University of Groningen
simon@cs.rug.nl gert@cs.rug.nl

An algorithm for Visualizing Implicit Curves & Surfaces

Eurographics Symposium on Geometry Processing (2004)
R. Scopigno, D. Zorin, (Editors)

Isotopic Approximation of Implicit Curves and Surfaces

Simon Plantinga and Gert Vegter

Institute for Mathematics and Computing Science
University of Groningen
simon@cs.rug.nl gert@cs.rug.nl

C^1-Function

$f: \mathbb{R}^n \to \mathbb{R}$

An algorithm for Visualizing Implicit Curves & Surfaces

Eurographics Symposium on Geometry Processing (2004)
R. Scopigno, D. Zorin, (Editors)

Isotopic Approximation of Implicit Curves and Surfaces

Simon Plantinga and Gert Vegter

Institute for Mathematics and Computing Science
University of Groningen
simon@cs.rug.nl gert@cs.rug.nl

C^1-Function

$$f : \mathbb{R}^n \to \mathbb{R} \xrightarrow{\text{PV Algorithm}}$$

An algorithm for Visualizing Implicit Curves & Surfaces

Eurographics Symposium on Geometry Processing (2004)
R. Scopigno, D. Zorin, (Editors)

Isotopic Approximation of Implicit Curves and Surfaces

Simon Plantinga and Gert Vegter

Institute for Mathematics and Computing Science
University of Groningen
simon@cs.rug.nl gert@cs.rug.nl

C^1-Function $\xrightarrow{\text{PV Algorithm}}$ PL Approximation

$f: \mathbb{R}^n \to \mathbb{R}$ of $Z(f) \cap [-a, a]^n$

An algorithm for Visualizing Implicit Curves & Surfaces

Eurographics Symposium on Geometry Processing (2004)
R. Scopigno, D. Zorin, (Editors)

Isotopic Approximation of Implicit Curves and Surfaces

Simon Plantinga and Gert Vegter

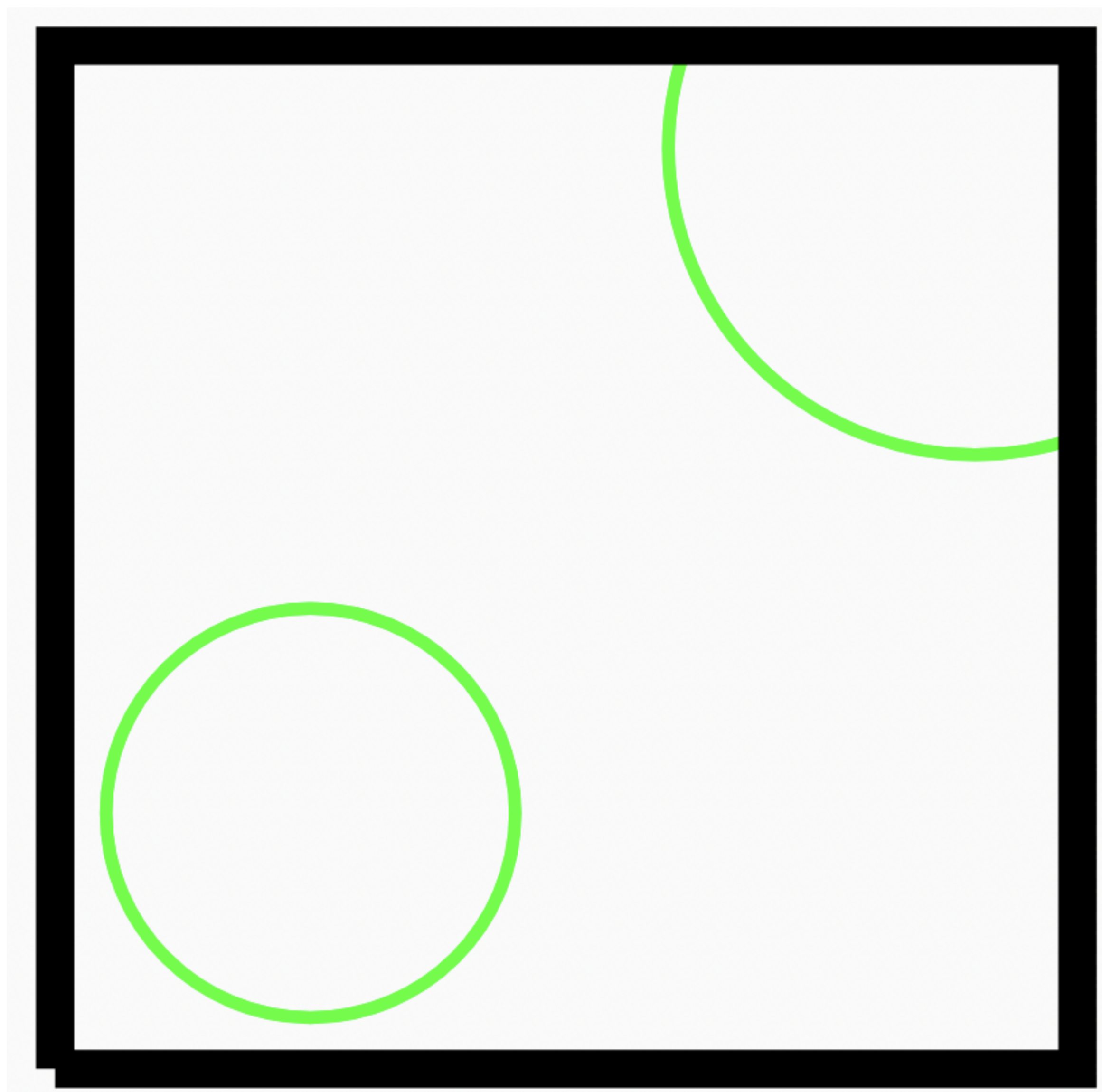
Institute for Mathematics and Computing Science
University of Groningen
simon@cs.rug.nl gert@cs.rug.nl

C^1-Function $f : \mathbb{R}^n \to \mathbb{R}$ $\xrightarrow{\text{PV Algorithm}}$ PL Approximation of $Z(f) \cap [-a,a]^n$

Extended to hypersurfaces by Galehouse

PV Algorithm in Action I

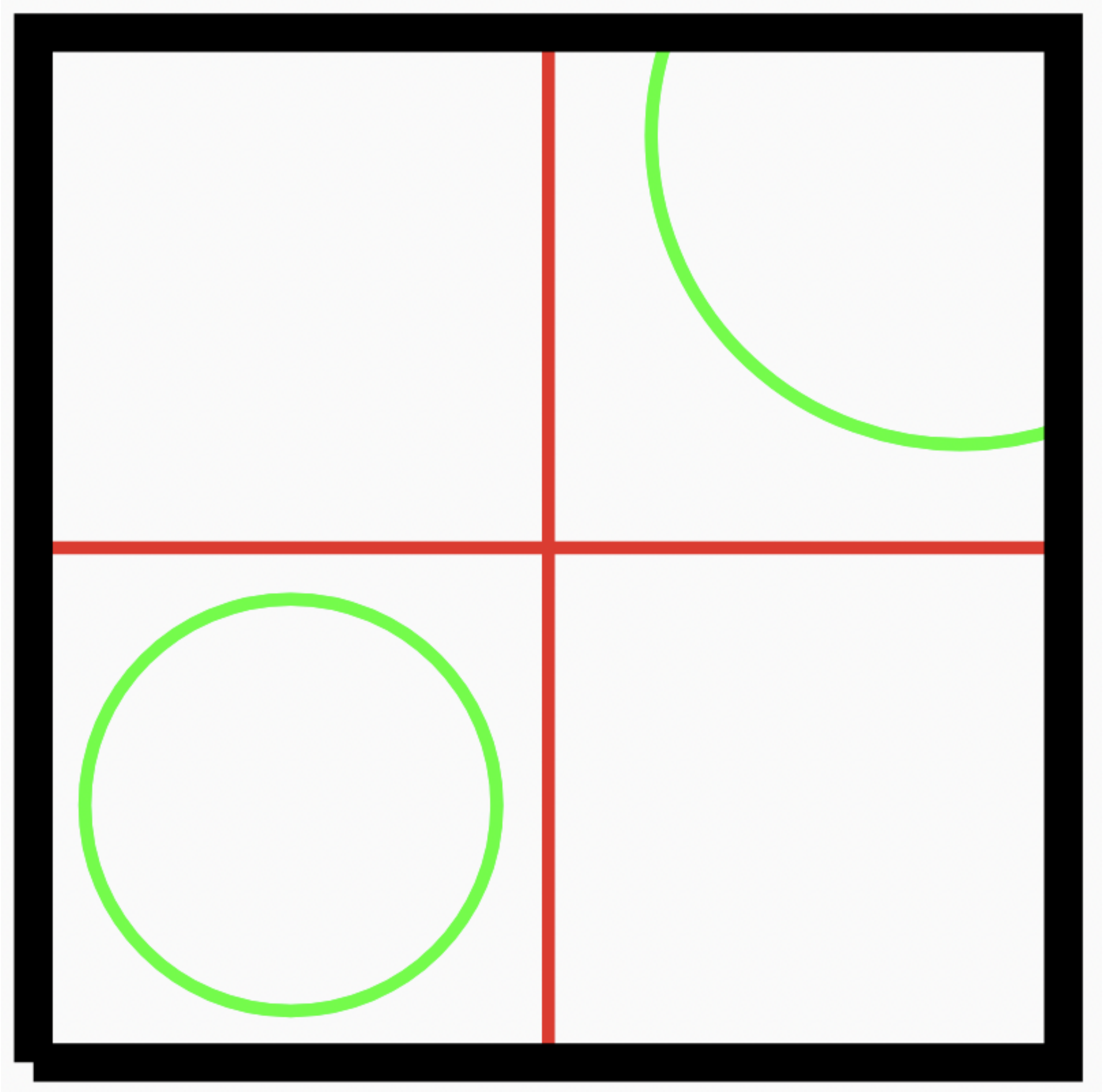
$$g = (x^2+y^2)^2 - 6(x^3+x^2y+xy^2+y^3) - 34(x^2+y^2) - 320xy + 376(x+y) + 3128$$



$[-10, 10]$

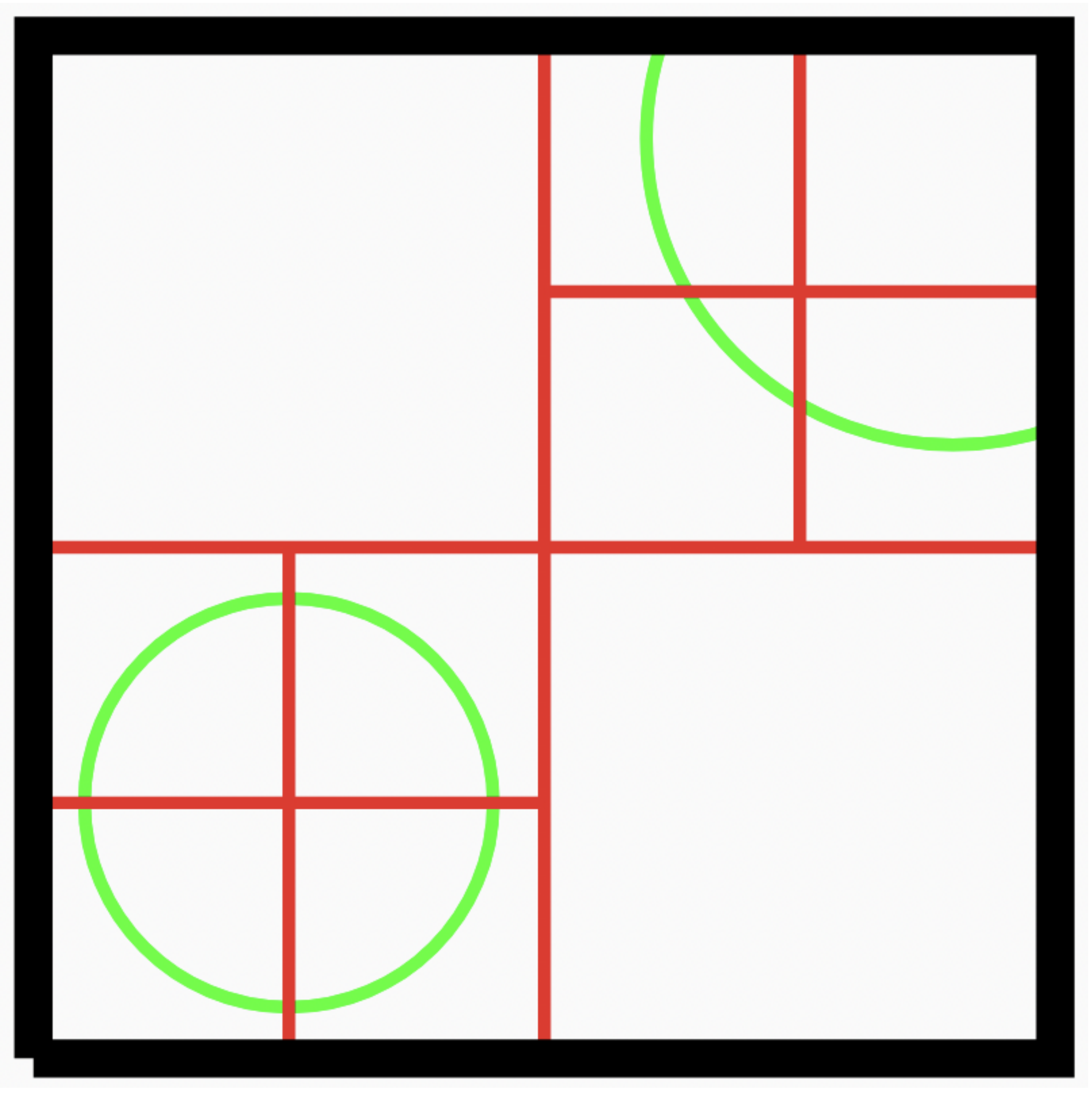
PV Algorithm in Action I

Subdivision Step I



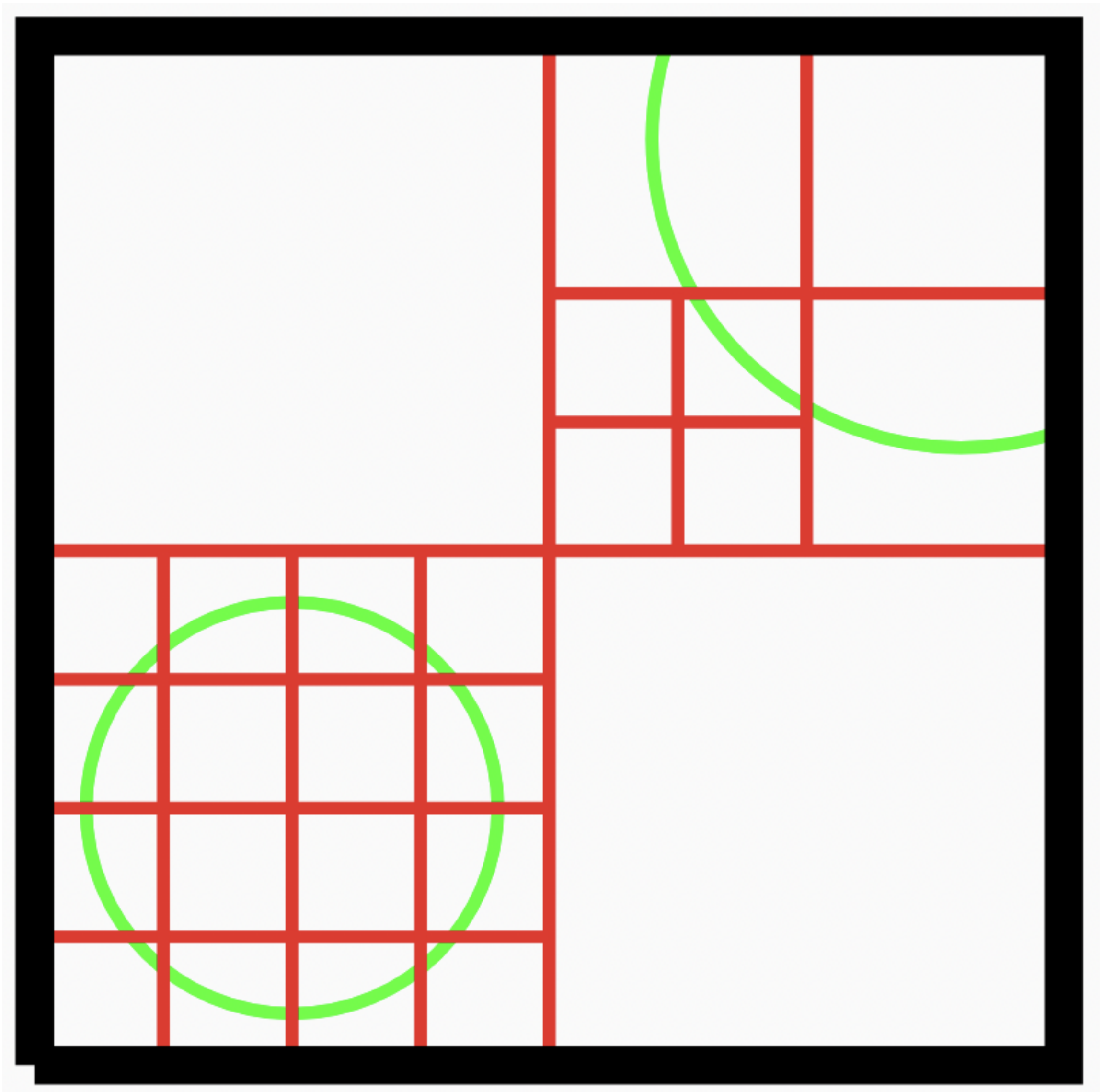
PV Algorithm in Action I

Subdivision Step II



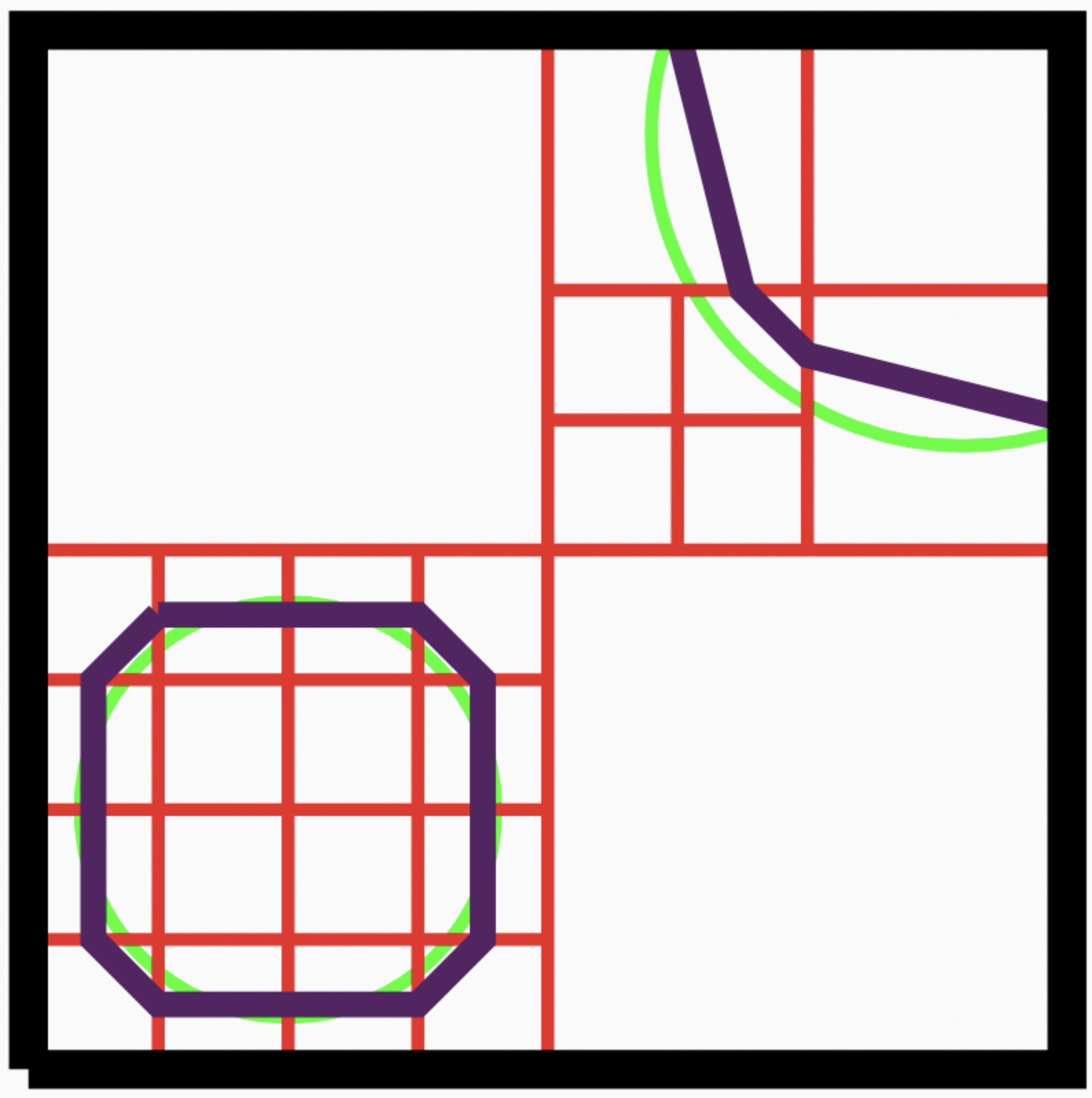
PV Algorithm in Action I

Subdivision Step III

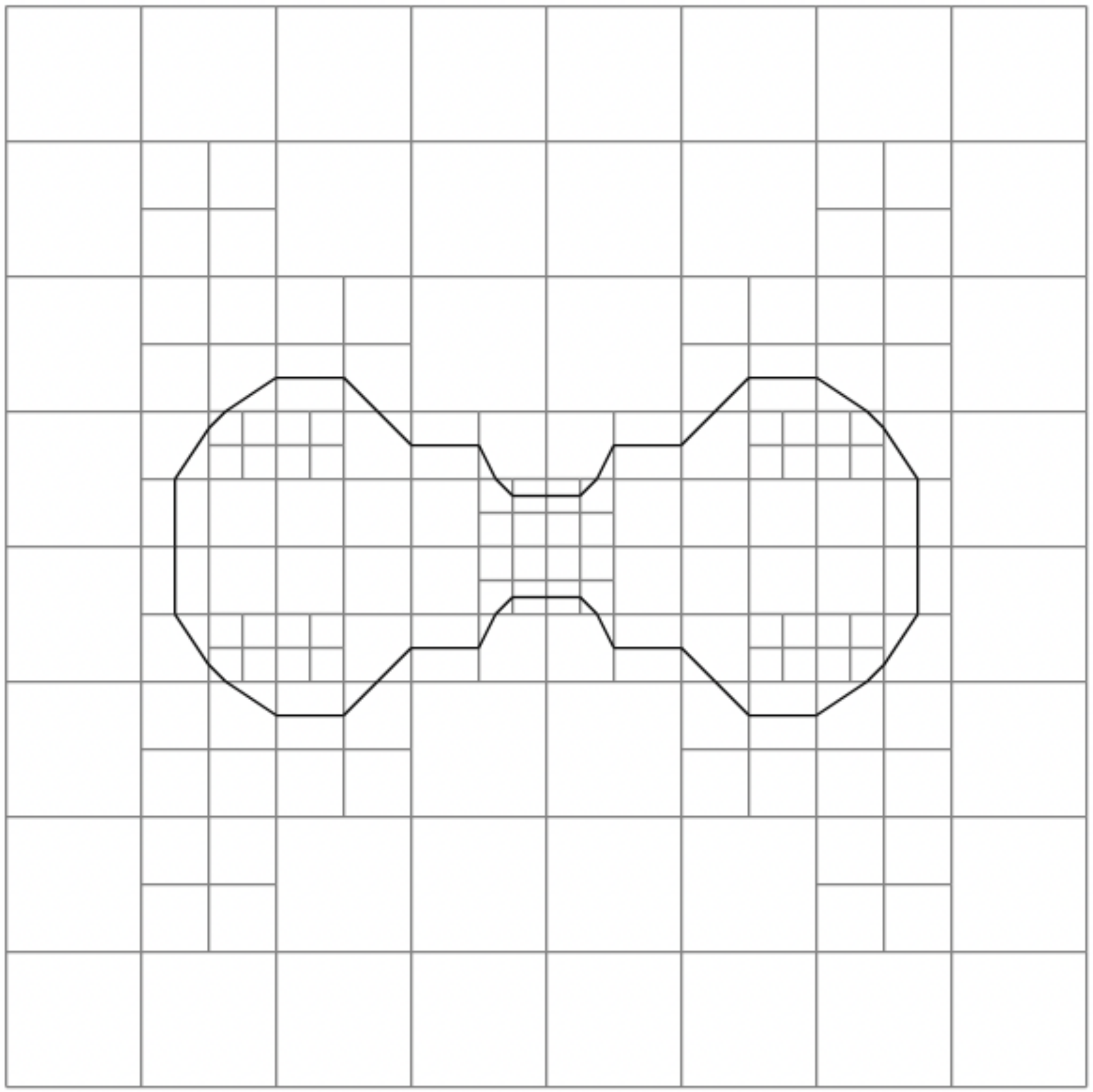


PV Algorithm in Action I

Postprocessing Step

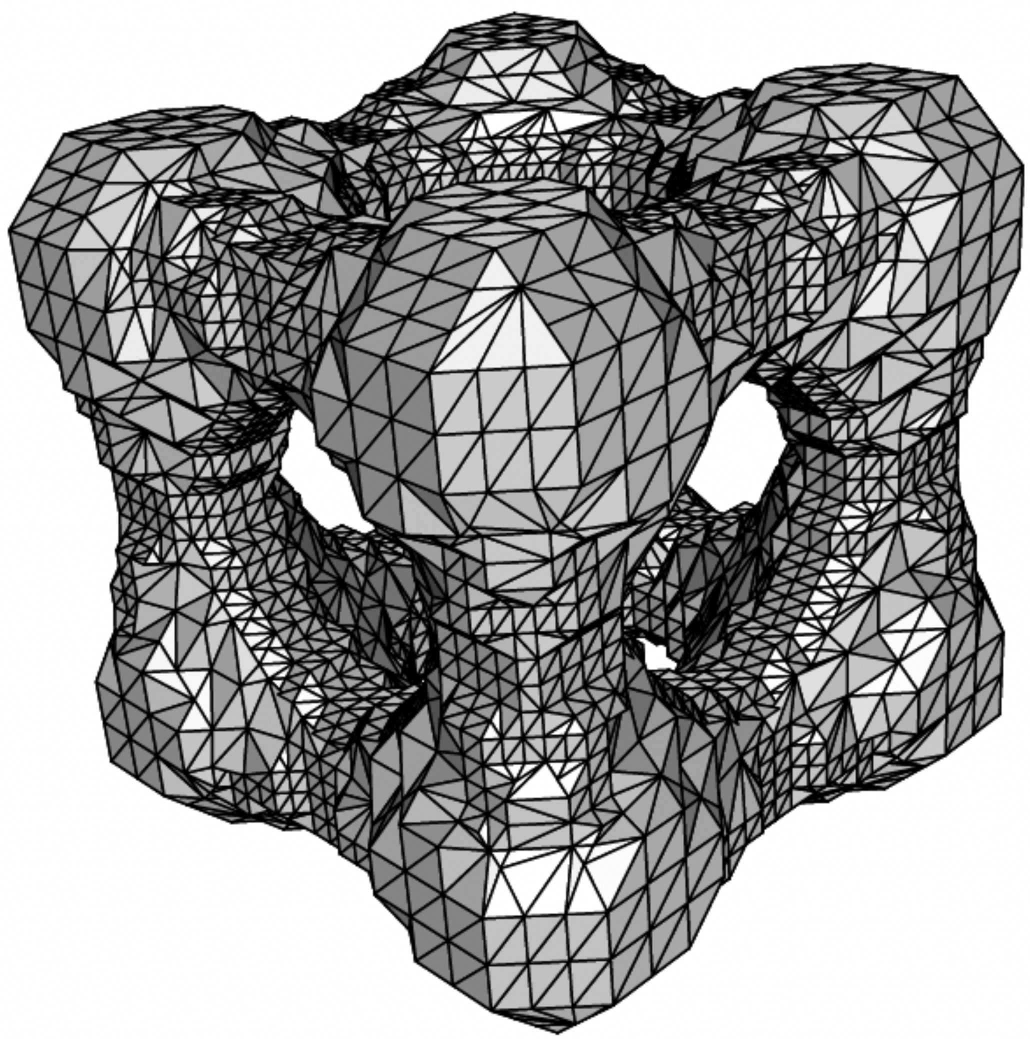


PV Algorithm in Action III



$$g = x^2(1-x)(1+x) - y^2 + 0.01$$

PV Algorithm in Action III



$$8 = X^4 - 5X^2 + Y^4 - 5Y^2 + Z^4 - 5Z^2 + 10$$

PV Algorithm works in practice,

but worst-case bounds* were

too pessimistic!

* by Burr, Gao, Tsigaridas

Plantinga-Vegter Criterion I

PLANTINGA-VEGTER CRITERION I

$$C_{\&}(B): \left\{ \begin{array}{c} \\ \text{OR} \\ \\ \end{array} \right.$$

Plantinga-Vegter Criterion I

$$C_{g}(B): \begin{cases} 0 \notin g(B) \\ \text{OR} \\ \end{cases}$$

Plantinga-Vegter Criterion I

$$C_g(B) : \begin{cases} 0 \notin g(B) \\ \text{OR} \\ \forall x, y \in B, \ \langle \nabla_x g, \nabla_y g \rangle \neq 0 \end{cases}$$

Plantinga-Vegter Criterion I

$$C_g(B): \begin{cases} 0 \notin g(B) \\ \text{OR} \\ \forall x, y \in B, \ \langle \nabla_x g, \nabla_y g \rangle \neq 0 \end{cases}$$

THM. If S is a subdivision of $[-a, a]$ s.t. for all $B \in S$, $C_g(B)$ holds.

Plantinga-Vegter Criterion I

$$C_g(B): \begin{cases} 0 \notin g(B) \\ \text{OR} \\ \forall x,y \in B, \ \langle \nabla_x g, \nabla_y g \rangle \neq 0 \end{cases}$$

THM. If S is a subdivision of $[-a,a]$ s.t. for all $B \in S$, $C_g(B)$ holds. Then we can produce a PL approx of $Z(g) \cap [-a,a]$ that is isotopically equiv.

PLANTINGA-VEGTER ALGORITHM
(Abstract level)

PLANTINGA-VEGTER ALGORITHM

(Abstract level)

$\mathcal{I} \neq \emptyset$

PLANTINGA-VEGTER ALGORITHM

(Abstract level)

$\mathcal{I} \neq \emptyset$

$$\boxed{\begin{array}{l} \text{Take } B \in \mathcal{I} \\ \mathcal{I} \leftarrow \mathcal{I} \setminus \{B\} \end{array}}$$

Plantinga-Vegter Algorithm

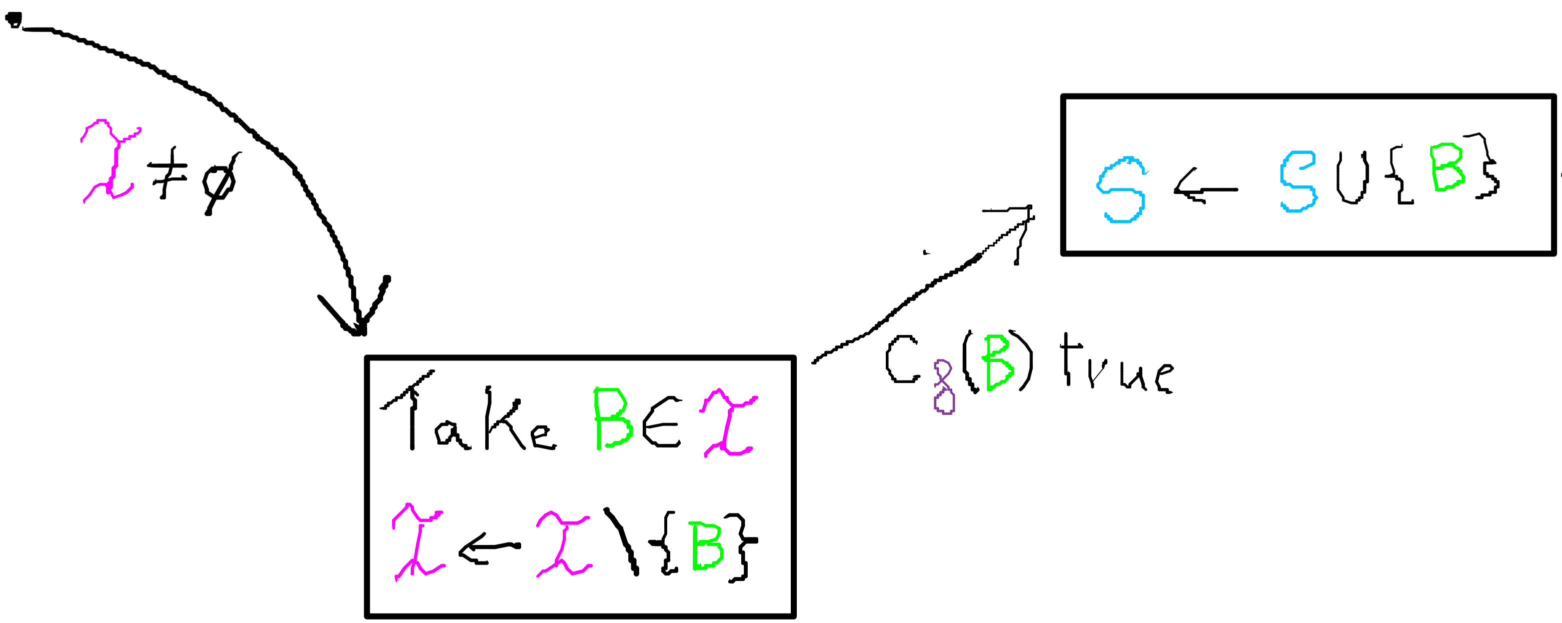
(Abstract level)

$\mathcal{I} \neq \emptyset$

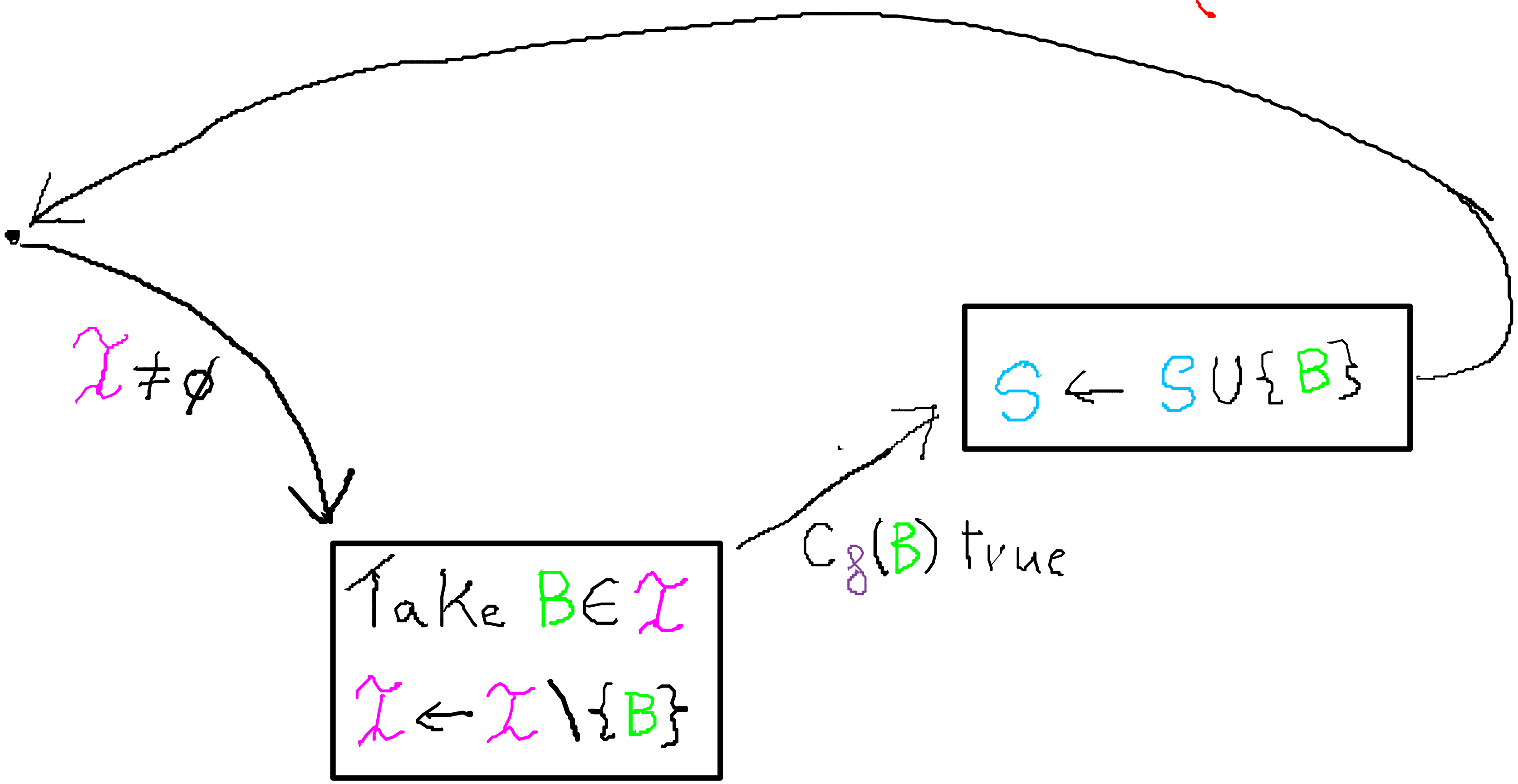
$$\boxed{\begin{array}{l} \text{Take } B \in \mathcal{I} \\ \mathcal{I} \leftarrow \mathcal{I} \setminus \{B\} \end{array}}$$

$C_\delta(B)$ true

PLANTINGA-VEGTER ALGORITHM
(Abstract level)

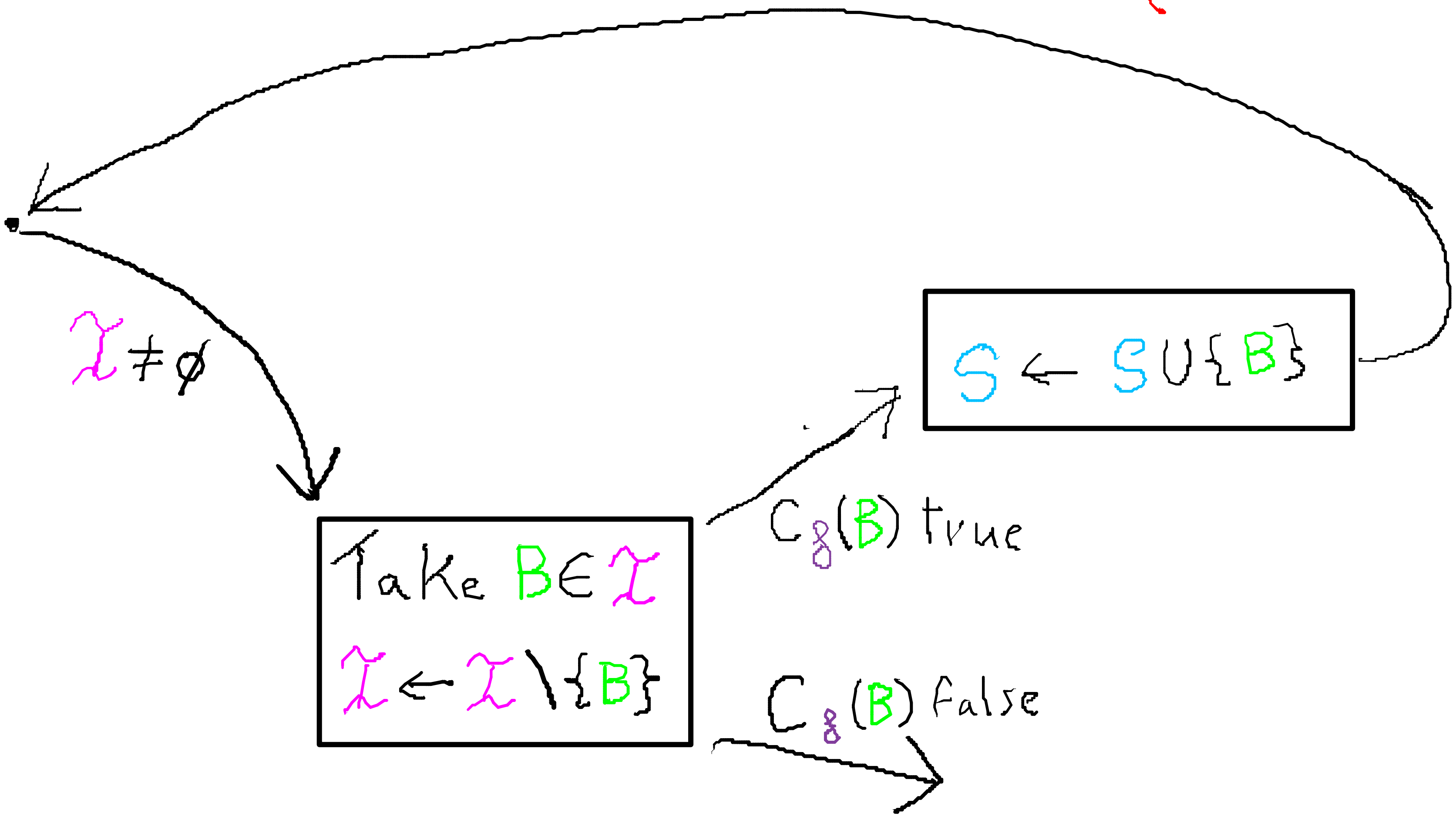


Plantinga-Vegter Algorithm
(Abstract level)



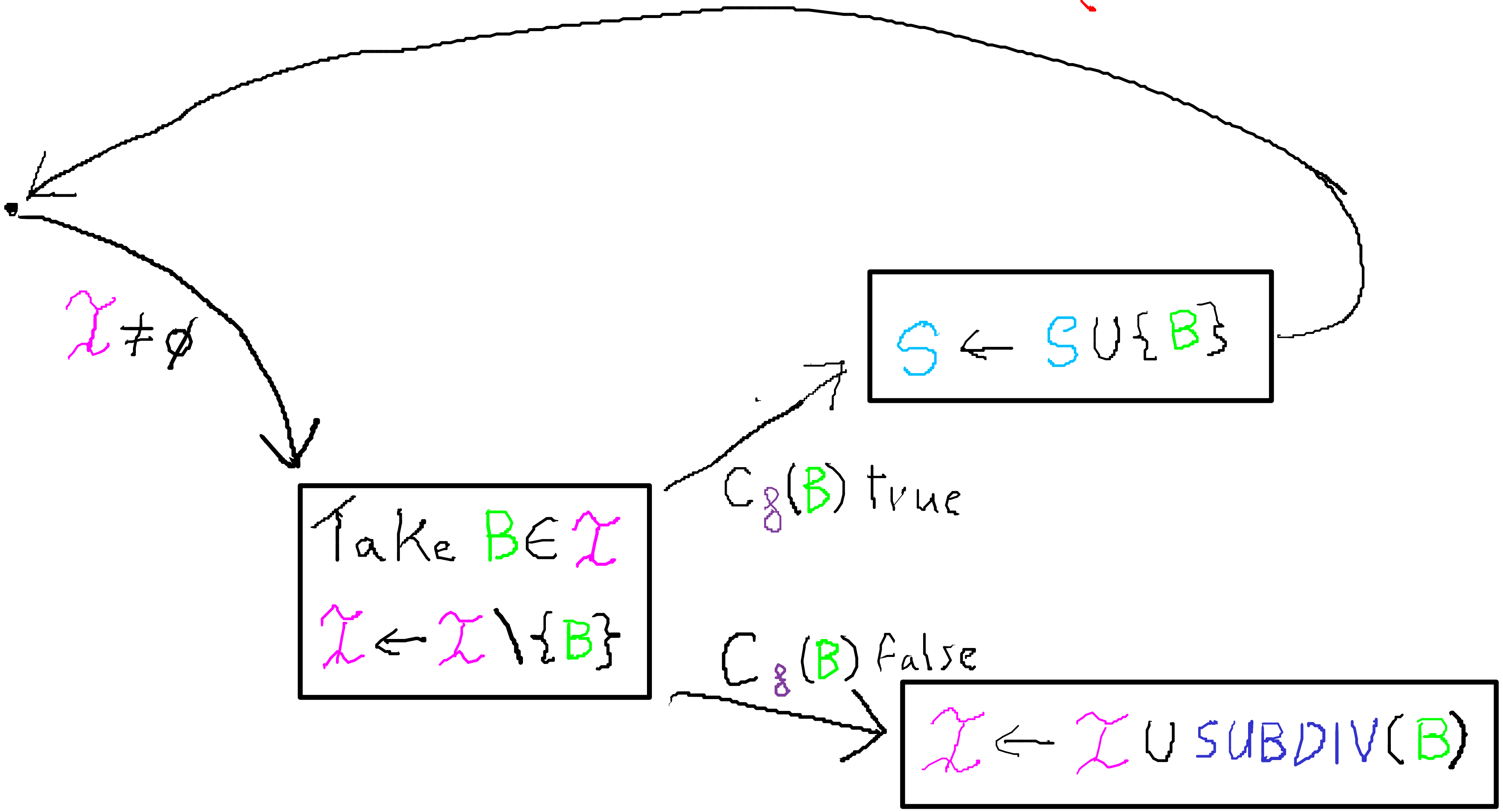
PLANTINGA-VEGTER ALGORITHM

(Abstract level)

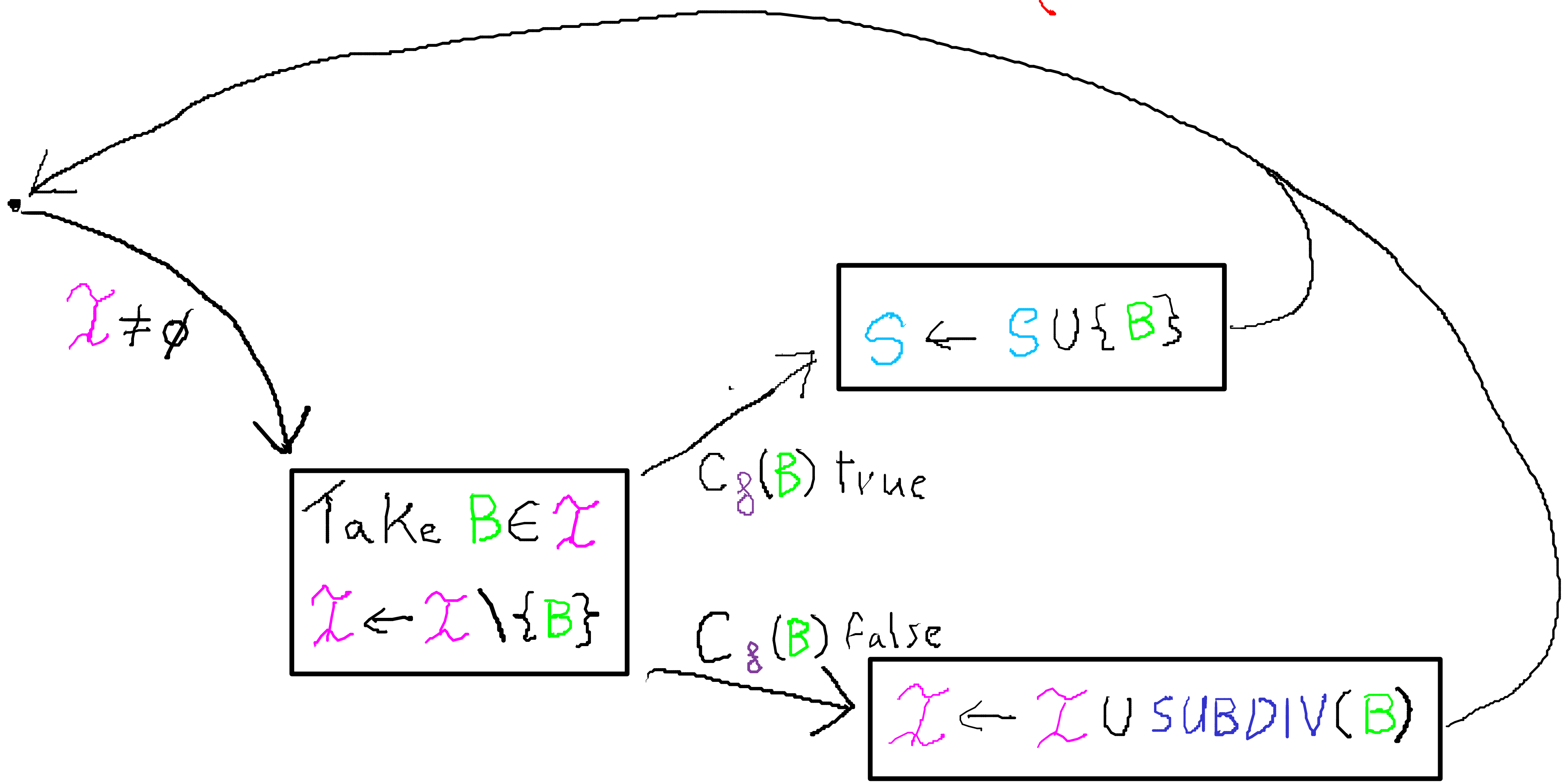


PLANTINGA-VEGTER ALGORITHM

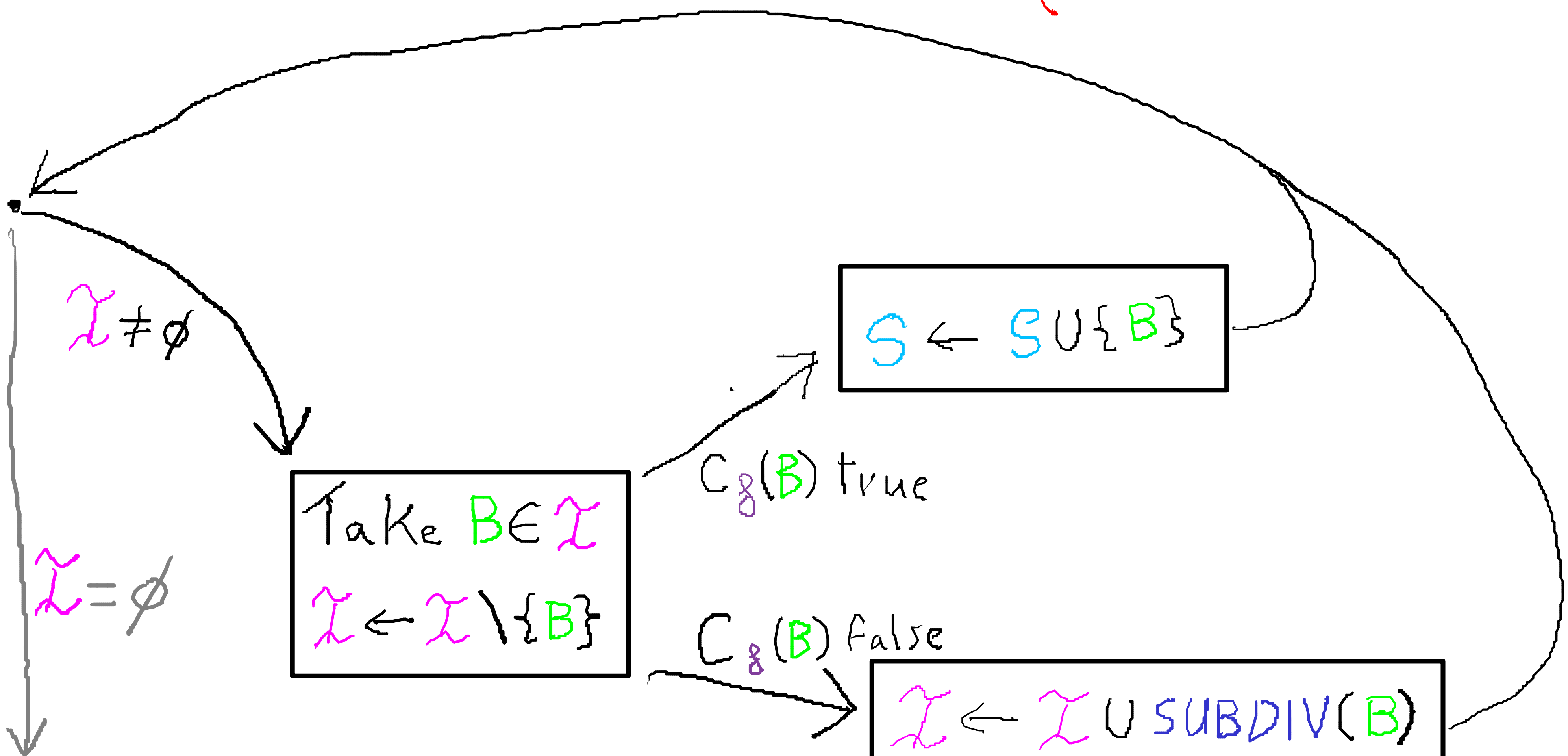
(Abstract level)



Plantinga-Vegter Algorithm
(Abstract level)

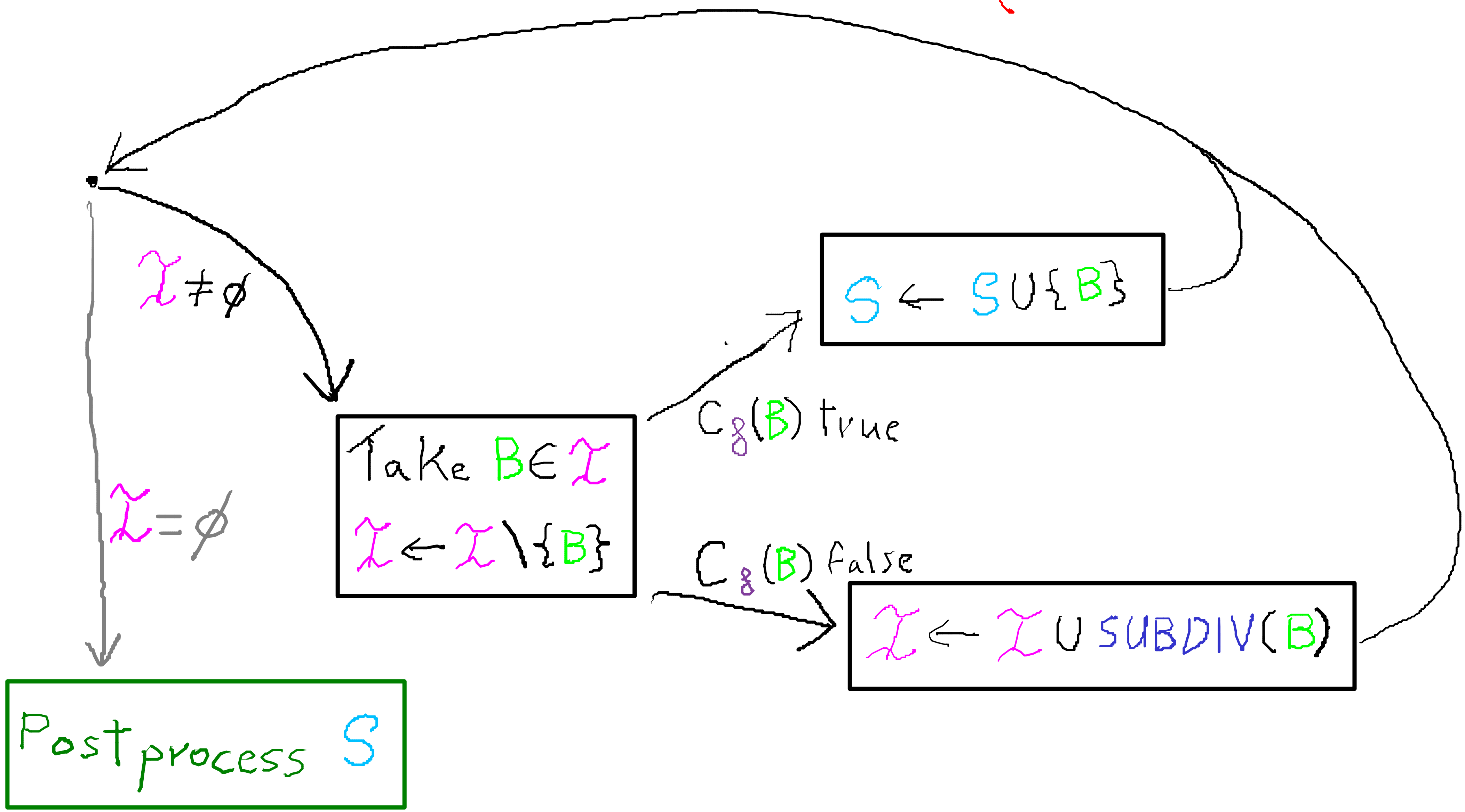


PLANTINGA-VEGTER ALGORITHM
(Abstract level)

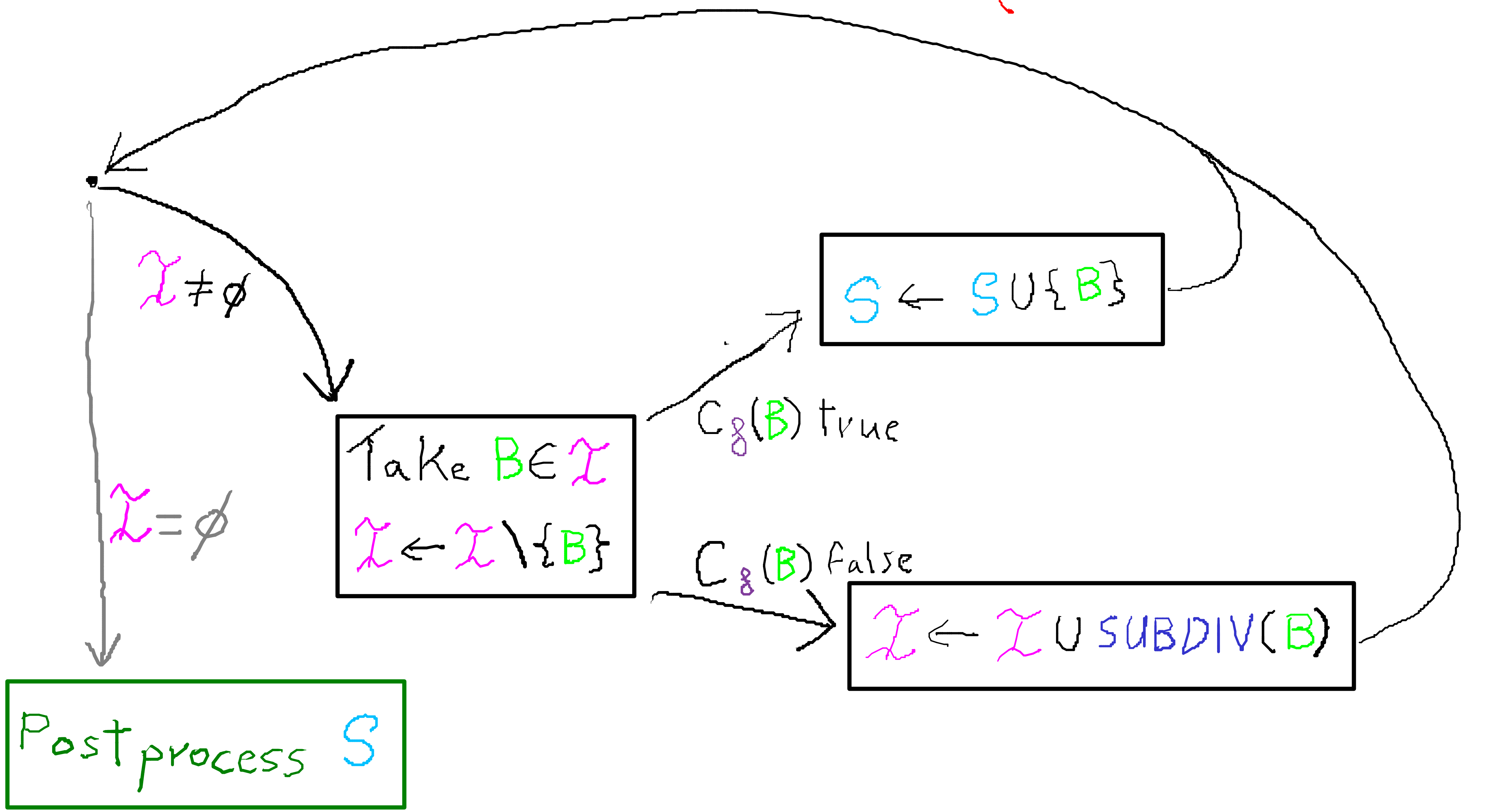


PLANTINGA-VEGTER ALGORITHM

(Abstract level)



PLANTINGA-VEGTER ALGORITHM
(Abstract level)



Q: How do we check $C_8(B)$?

Interval Approximations I

INTERVAL APPROXIMATIONS I

DEF. An <u>interval approximation</u> of

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

INTERVAL APPROXIMATIONS I

DEF. An <u>interval approximation</u> of

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

is a map

$$\square F : \square \mathbb{R}^n \longrightarrow \square \mathbb{R}^m$$

where $\square \mathbb{R}^k := \{ B \subseteq \mathbb{R}^k \mid B \text{ is a box} \}$

Interval Approximations I

Def. An <u>interval approximation</u> of

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

is a map

$$\square F : \square \mathbb{R}^n \longrightarrow \square \mathbb{R}^m$$

where $\square \mathbb{R}^k := \{ B \subseteq \mathbb{R}^k \mid B \text{ is a box} \}$

such that for all $B \in \square \mathbb{R}^n$,

$$F(B) \subseteq \square F(B)$$

Complexity Context for PV

Complexity Context for PV

Input

Complexity Context for PV

Input

$g \in \mathbb{R}[x_1, \dots, x_n]$

Complexity Context for PV

Input

$g \in \mathbb{R}[x_1, \ldots, x_n]$

Complexity Context for PV

Input

$\delta \in \mathbb{R}[x_1, \dots, x_n]$

Output

Complexity Context for PV

INPUT

$g \in \mathbb{R}[x_1, \ldots, x_n]$

OUTPUT

PL-approximation of $\mathcal{Z}(g) \cap [-1,1]$

Complexity Context for PV

Input

$g \in \mathbb{R}[x_1, \ldots, x_n]$

Output

PL-approximation of $\mathcal{Z}(g) \cap [-1,1]$

Complexity Parameters

Complexity Context for PV

Input

$g \in \mathbb{R}[x_1, \ldots, x_n]$

Output

PL-approximation of $\mathcal{Z}(g) \cap [-1,1]$

Complexity Parameters

d : degree of g

Complexity Context for PV

Input

$g \in \mathbb{R}[X_1, \ldots, X_n]$

Output

PL-approximation of $\mathcal{Z}(g) \cap [-1,1]$

Complexity Parameters

d : degree of g

n : number variables

COMPLEXITY CONTEXT for PV

INPUT

$g \in \mathbb{R}[X_1, \ldots, X_n]$

OUTPUT

PL-approximation of $\mathcal{Z}(g) \cap [-1,1]$

COMPLEXITY PARAMETERS

d : degree of g

n : number variables

M : number of monomials in g

Complexity Context for PV

Input

$g \in \mathbb{R}[x_1, \dots, x_n]$

Output

PL-approximation of $\mathcal{Z}(g) \cap [-1,1]$

Complexity Parameters

d : degree of g

n : number variables

M : number of monomials in g

Measure of Complexity

COMPLEXITY CONTEXT for PV

INPUT

$g \in \mathbb{R}[x_1, \ldots, x_n]$

OUTPUT

PL-approximation of $\mathcal{Z}(g) \cap [-1,1]$

COMPLEXITY PARAMETERS

d : degree of g

n : number variables

M : number of monomials in g

MEASURE OF COMPLEXITY

Size of subdivision

Interval Approximations II

INTERVAL APPROXIMATIONS II

$\|\delta\|_1 := \sum_{\alpha} |\delta_\alpha|$ (1-norm of δ)

INTERVAL APPROXIMATIONS II

$$\|g\|_1 := \sum_\alpha |g_\alpha| \quad \text{(1-norm of } g\text{)}$$

$$\square g(B) := g(m(B)) + d\,\|g\|_1 \left[-\frac{w(B)}{2}, \frac{w(B)}{2}\right]$$

INTERVAL APPROXIMATIONS II

$\|g\|_1 := \sum_\alpha |g_\alpha|$ (1-norm of g)

$\Box\, g(B) := g(m(B)) + d\,\|g\|_1 \left[-\frac{w(B)}{2}, \frac{w(B)}{2}\right]$

$\Box\, \|\nabla g\|_1(B) := \nabla_{m(B)} g + \sqrt{2n}\, d^2\, \|g\|_1 \left[-\frac{w(B)}{2}, \frac{w(B)}{2}\right]$

Interval Approximations II

$$\|g\|_1 := \sum_\alpha |g_\alpha| \quad \text{(1-norm of } g\text{)}$$

$$\square g(B) := g(m(B)) + d\,\|g\|_1 \left[-\frac{w(B)}{2}, \frac{w(B)}{2}\right]$$

$$\square \|\nabla g\|_1(B) := \nabla_{m(B)} g + \sqrt{2n}\, d^2 \|g\|_1 \left[-\frac{w(B)}{2}, \frac{w(B)}{2}\right]$$

THM. $\square g$ is an interval approx of g

$\square \|\nabla g\|_1$ is an interval approx of $\|\nabla g\|_1$

Plantinga-Vegter Criterion II

Plantinga-Vegter Criterion II

$$C_g^{\square}(B): \begin{cases} 0 \notin \square g(B) \\ \text{OR} \\ 0 \notin \square \|\nabla g\|_1(B) \end{cases}$$

Plantinga-Vegter Criterion II

$$C^{\square}_{\gamma}(B): \begin{cases} 0 \notin \square\, \gamma(B) \\ \text{OR} \\ 0 \notin \square\, \|\nabla\gamma\|_1(B) \end{cases}$$

THM.

$$C^{\square}_{\gamma}(B) \Rightarrow C_{\gamma}(B)$$

Condition-based Estimate I

Condition number

CONDITION-BASED ESTIMATE I

Condition number

$$C(g, x) := \frac{\|g\|_1}{\max\{|g(x)|, \|\nabla_x g\|_1\}} \in [1, \infty]$$

CONDITION-BASED ESTIMATE I

Condition number

$$C(g, x) := \frac{\|g\|_1}{\max\{|g(x)|, \|\nabla_x g\|_1\}} \in [1, \infty]$$

PROP. $C(g, x) = \infty \iff$

CONDITION-BASED ESTIMATE I

Condition number

$$C(g, x) := \frac{\| g \|_1}{\max\{ |g(x)|, \| \nabla_x g \|_1 \}} \in [1, \infty]$$

PROP. $C(g, x) = \infty \iff g$ singular at x

CONDITION-BASED ESTIMATE I

Condition number

$$C(g,x) := \frac{\|g\|_1}{\max\{|g(x)|, \|\nabla_x g\|_1\}} \in [1,\infty]$$

PROP. $C(g,x)=\infty \iff g$ singular at x

PROP. If $x \in B$ and $C(g,x)\, d\sqrt{2n}\, w(B) < 1$,

CONDITION-BASED ESTIMATE I

Condition number

$$C(g, x) := \frac{\|g\|_1}{\max\{|g(x)|, \|\nabla_x g\|_1\}} \in [1, \infty]$$

PROP. $C(g, x) = \infty \iff g$ singular at x

PROP. If $x \in B$ and $C(g, x)\, d\sqrt{2n}\, w(B) < 1$,
then $C^D_g(B)$ holds

Condition-based Estimate II

THM. The PV algorithm produces a subdivision with at most

$$2^{\frac{5}{2}n} n^{n/2} d^n \, \mathbb{E}_{x\in I^n} C(g,x)^n$$

boxes on g

CONDITION-BASED ESTIMATE II

THM. The PV algorithm produces a subdivision with at most

$$2^{5/2\,n}\, n^{n/2}\, d^n\, \mathbb{E}_{x\in I^n}\, C(g,x)^n$$

boxes on g

Proof relies on continuous amortization by Burr, Krahmer & Yap

PROBABILISTIC ESTIMATE I

UNIFORM CASE

Probabilistic Estimate I

Uniform case

$$f := \sum_{\alpha \in A} f_\alpha X^\alpha \qquad \#A =: M$$

where f_α independent uniformly dist. on intervals of length 2

Probabilistic Estimate I

Uniform case

$$f := \sum_{\alpha \in A} f_\alpha X^\alpha \qquad \#A =: M$$

where f_α independent uniformly dist. on intervals of length 2

THM. On f, the PV algorithm produces a subdivision with

$$2n\, 32^{n+1} d^{2n} M^{n+2}$$

boxes on average.

PROBABILISTIC ESTIMATE II

GAUSSIAN CASE

Probabilistic Estimate II

Gaussian case

$$f := \sum_{\alpha \in A} f_\alpha X^\alpha \qquad \#A =: M$$

where f_α indepent Gaussians of variance 1

Probabilistic Estimate II

Gaussian case

$$f := \sum_{\alpha \in A} f_\alpha X^\alpha \qquad \# A =: M$$

where f_α indepent Gaussians of variance 1

THM. On f, the PV algorithm produces a subdivision with

$$2\, n^{\frac{3}{2}} \left(10(n+1)\right)^{n+1} d^{2n} M^{n+2}$$

boxes on average.

PROBABILISTIC ESTIMATE II

GENERAL CASE: ZINTZO POLYNOMIALS

PROBABILISTIC ESTIMATE III

GENERAL CASE: ZINTZO POLYNOMIALS

$f := \sum_{\alpha \in A} f_\alpha X^\alpha$ where f_α independent s.t.

PROBABILISTIC ESTIMATE III
GENERAL CASE: ZINTZO POLYNOMIALS

$f := \sum_{\alpha \in A} f_\alpha X^\alpha$ where f_α independent s.t.

(Anti-concentration)

$$\forall t \in \mathbb{R},\ \forall \varepsilon > 0,\ \mathbb{P}(|f_\alpha - t| < \varepsilon) < \rho_\alpha \varepsilon$$

Probabilistic Estimate II

General case: Zintzo Polynomials

$f := \sum_{\alpha \in A} f_\alpha X^\alpha$ where f_α independent s.t.

(Anti-concentration)

$$\forall t \in \mathbb{R}, \forall \varepsilon > 0, \ \mathbb{P}(|f_\alpha - t| < \varepsilon) < \rho_\alpha \varepsilon$$

(Tail bounds)

$$\forall t \in \mathbb{R}_+, \ \mathbb{P}(|f_\alpha| > t) \leq 2 \exp\left(-\left(t/L_\alpha\right)^p\right)$$

PROBABILISTIC ESTIMATE III
GENERAL CASE: ZINTZO POLYNOMIALS

$f := \sum_{\alpha \in A} f_\alpha X^\alpha$ where f_α independent s.t.

(Anti-concentration)

$$\forall t \in \mathbb{R}, \forall \varepsilon > 0, \ \mathbb{P}(|f_\alpha - t| < \varepsilon) < \rho_\alpha \varepsilon$$

(Tail bounds)

$$\forall t \in \mathbb{R}_+, \ \mathbb{P}(|f_\alpha| \geq t) \leq 2\exp\left(-\left(t/L_\alpha\right)^p\right)$$

→ Allow us to use Geometric Functional Analysis

PROBABILISTIC ESTIMATE III

GENERAL CASE: ZINTZO POLYNOMIALS

$f := \sum_{\alpha \in A} f_\alpha X^\alpha$ where f_α independent s.t.

(Anti-concentration)

$\forall t \in \mathbb{R}, \forall \varepsilon > 0, \ \mathbb{P}(|f_\alpha - t| < \varepsilon) < \rho_\alpha \varepsilon$

(Tail bounds)

$\forall t \in \mathbb{R}_+, \ \mathbb{P}(|f_\alpha| \geq t) \leq 2 \exp\left(-\left(t/L_\alpha\right)^p\right)$

$\rightarrow$ Allow us to use Geometric Functional Analysis

THE MODEL IS ROBUST

So

PV algorithm

is also efficient in theory!

The Framework in Action II

the Descartes solver

Joint work with A. A. Ergür & E. Tsigaridas

Photo while working on this project

Real Root Isolation I: The Problem

Real Root Isolation I: The Problem

INPUT:

Real Root Isolation I: The Problem

INPUT:

$g \in \mathbb{Z}[X]$

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \ldots, J_k$ s.t.

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \ldots, J_k$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \dots, J_k$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(f) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{k} J_i$

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \ldots, J_k$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(f) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{k} J_i$

2) $\forall i,\ \# Z(f) \cap J_i = 1$

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \dots, J_k$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(f) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{k} J_i$

2) $\forall i, \# Z(f) \cap J_i = 1$

INPUT SIZE PARAMETERS:

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \dots, J_k$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(f) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{k} J_i$

2) $\forall i, \# Z(f) \cap J_i = 1$

INPUT SIZE PARAMETERS:

d : degree of f

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \dots, J_K$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(f) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{K} J_i$

2) $\forall i, \# Z(f) \cap J_i = 1$

INPUT SIZE PARAMETERS:

d : degree of f

τ : bit-size of coefficients of f

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \dots, J_K$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(f) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{K} J_i$

2) $\forall i, \# Z(f) \cap J_i = 1$

INPUT SIZE PARAMETERS:

d : degree of f

τ : bit-size of coefficients of f

MEASURE OF RUN-TIME

Real Root Isolation I: The Problem

INPUT:

$f \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \dots, J_k$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(f) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{k} J_i$

2) $\forall i, \# Z(f) \cap J_i = 1$

INPUT SIZE PARAMETERS:

d : degree of f

τ : bit-size of coefficients of f

MEASURE OF RUN-TIME

Bit complexity

Real Root Isolation I: The Problem

INPUT:

$$f \in \mathbb{Z}[X]$$

OUTPUT:

Intervals $J_1, \ldots, J_k$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(f) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{k} J_i$

2) $\forall i, \ \# Z(f) \cap J_i = 1$

We can also handle continuous inputs!

INPUT SIZE PARAMETERS:

d : degree of f

τ : bit-size of coefficients of f

MEASURE OF RUN-TIME

Bit complexity

Real Root Isolation II: The State of the Art

Real Root Isolation II: The State of the Art

STURM SOLVER $\qquad \tilde{O}_B(d^4\tau^2)$

Real Root Isolation II: The State of the Art

Sturm Solver $\tilde{O}_B(d^4\tau^2)$

Descartes Solver $\tilde{O}_B(d^4\tau^2)$

Real Root Isolation II: The State of the Art

Sturm Solver	$\tilde{O}_B(d^4\tau^2)$
Descartes Solver	$\tilde{O}_B(d^4\tau^2)$
ANewDsc	$\tilde{O}_B(d^3+d^2\tau)$

(Sagraloff & Mehlhorn; 2016)

Real Root Isolation II: The State of the Art

Sturm Solver $\tilde{O}_B(d^4\tau^2)$

Descartes Solver $\tilde{O}_B(d^4\tau^2)$

ANewDsc $\tilde{O}_B(d^3 + d^2\tau)$

(Sagraloff & Mehlhorn; 2016)

Pan's Algorithm $\tilde{O}_B(d^2\tau)$

(Pan; 2002)

Real Root Isolation II: The State of the Art

Sturm Solver	$\tilde{O}_B(d^4\tau^2)$
Descartes Solver	$\tilde{O}_B(d^4\tau^2)$
ANewDsc (Sagraloff & Mehlhorn; 2016)	$\tilde{O}_B(d^3+d^2\tau)$
Pan's Algorithm (Pan; 2002)	$\tilde{O}_B(d^2\tau)$

Q: Can we beat the champion?

Real Root Isolation III: What do we wish?

$$\tilde{O}_B(d\tau)$$

Real Root Isolation III: What do we wish?

$$\tilde{O}_B(d\tau)$$

We wish to find real roots almost as fast as we read the polynomial!

Descartes Solver I: Rule of Signs

Descartes Solver I: Rule of Signs

$V(g) :=$ # sign variations of $g_0, g_1, \ldots$

Descartes Solver I: Rule of Signs

$V(g) :=$ # sign variations of $g_0, g_1, \ldots$

Portrait by Frans Hal
Source: WikiMedia Commons

Descartes Solver I: Rule of Signs

$V(g) :=$ # sign variations of $g_0, g_1, \ldots$

THM (Descartes' rule of signs)

$$\#Z(g, \mathbb{R}_>) \leq V(g)$$

Portrait by Frans Hal
Source: WikiMedia Commons

Descartes Solver I: Rule of Signs

$V(g) :=$ # sign variations of $g_0, g_1, \ldots$

THM (Descartes' rule of signs)

$$\#Z(g, \mathbb{R}_>) \le V(g)$$

Moreover,

$V(g) \le 1 \Rightarrow$ Equality

Portrait by Frans Hal
Source: WikiMedia Commons

Descartes Solver I: Rule of Signs

$V(g) :=$ # sign variations of $g_0, g_1, \dots$

THM (Descartes' rule of signs)

$$\#Z(g, \mathbb{R}_>) \leq V(g)$$

Moreover,

$$V(g) \leq 1 \Rightarrow \text{Equality}$$

Portrait by Frans Hal
Source: WikiMedia Commons

COR

$$\#Z(g,(a,b)) \leq V(g,(a,b)) := V\left((X+1)^d \, g\left(\frac{bX+a}{X+1}\right)\right)$$

$(0,\infty) \to (a,b)$
bijection

Descartes Solver II:

The Descartes' Oracle

Descartes Solver II: The Descartes' Oracle

1) Overcounting: $\#Z(g, J) \leq V(g, J)$

Descartes Solver II:

The Descartes' Oracle

1) Overcounting: $\#Z(g, J) \leq V(g, J)$
2) Exactness I: $V(g, J) < 1 \Rightarrow$ Equality

Descartes Solver II:

The Descartes' Oracle

1) Overcounting: $\#Z(g,J) \leq V(g,J)$

2) Exactness I: $V(g,J) \leq 1 \Rightarrow$ Equality

3) Exactness II:

$$\#Z(g, D(m(J), cw(J))) \leq K \Rightarrow V(g,J) \leq K$$

Descartes Solver II: The Descartes' Oracle

1) Overcounting: $\#Z(g,J) \leq V(g,J)$

2) Exactness I: $V(g,J) \leq 1 \Rightarrow$ Equality

3) Exactness II:

$$\#Z(g, D(m(J), cw(J))) \leq K \Rightarrow V(g,J) \leq K$$

Obreshkoff's Thm: Descartes sees the complex roots around!

DESCARTES SOLVER II:
The Descartes' Oracle

1) Overcounting: $\#Z(g, J) \leq V(g, J)$

2) Exactness I: $V(g,J) < 1 \Rightarrow$ Equality

3) Exactness II:

$$\#Z(g, \mathbb{D}(m(J), cw(J))) \leq K \Rightarrow V(g, J) \leq K$$

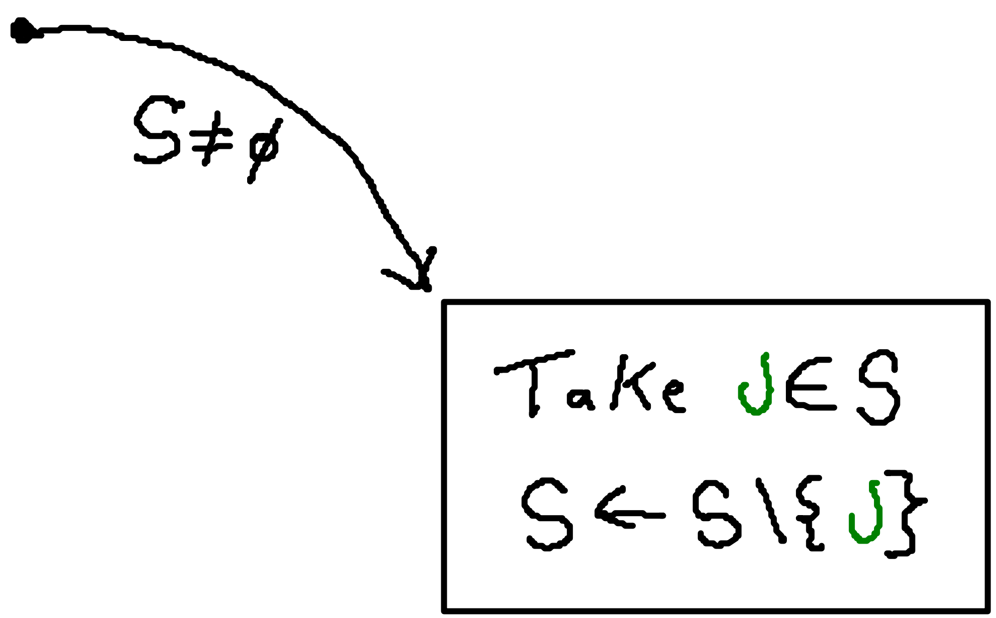
Obreshkoff's Thm: Descartes sees the complex roots around!

4) Subadditivity:

$$\dot{\bigcup} J_i \subseteq J \Rightarrow \sum V(g, J_i) \leq V(g, J)$$

Descartes Solver III: The Algorithm

Descartes Solver III: The Algorithm



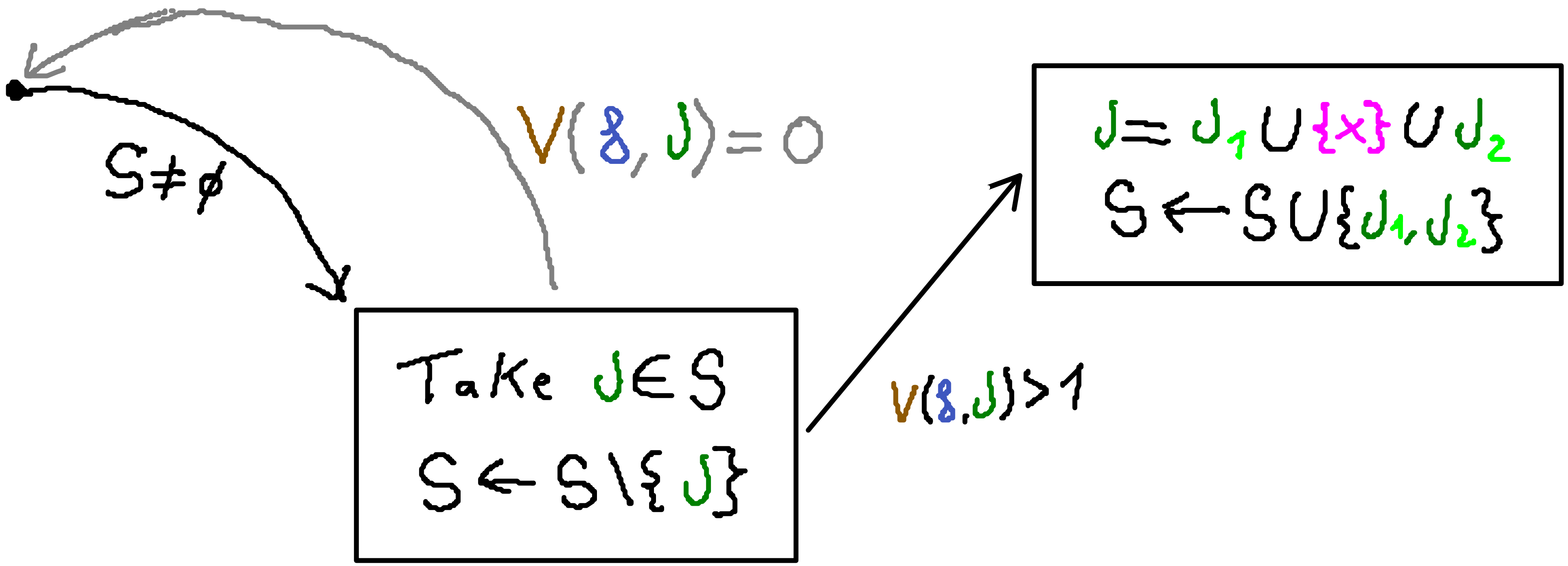
Descartes Solver III: The Algorithm

$S \neq \emptyset$

$V(\gamma, J) = 0$

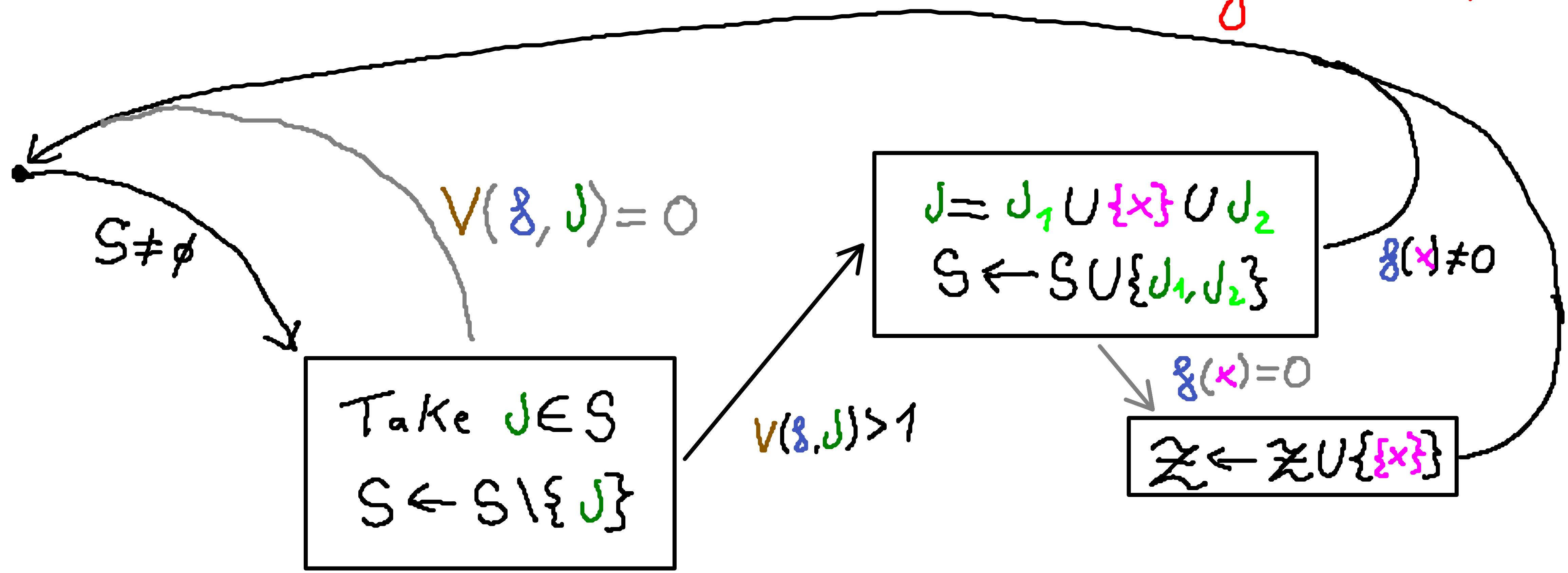
$$\boxed{\begin{array}{l} \text{Take } J \in S \\ S \leftarrow S \setminus \{J\} \end{array}}$$

Descartes Solver III: The Algorithm



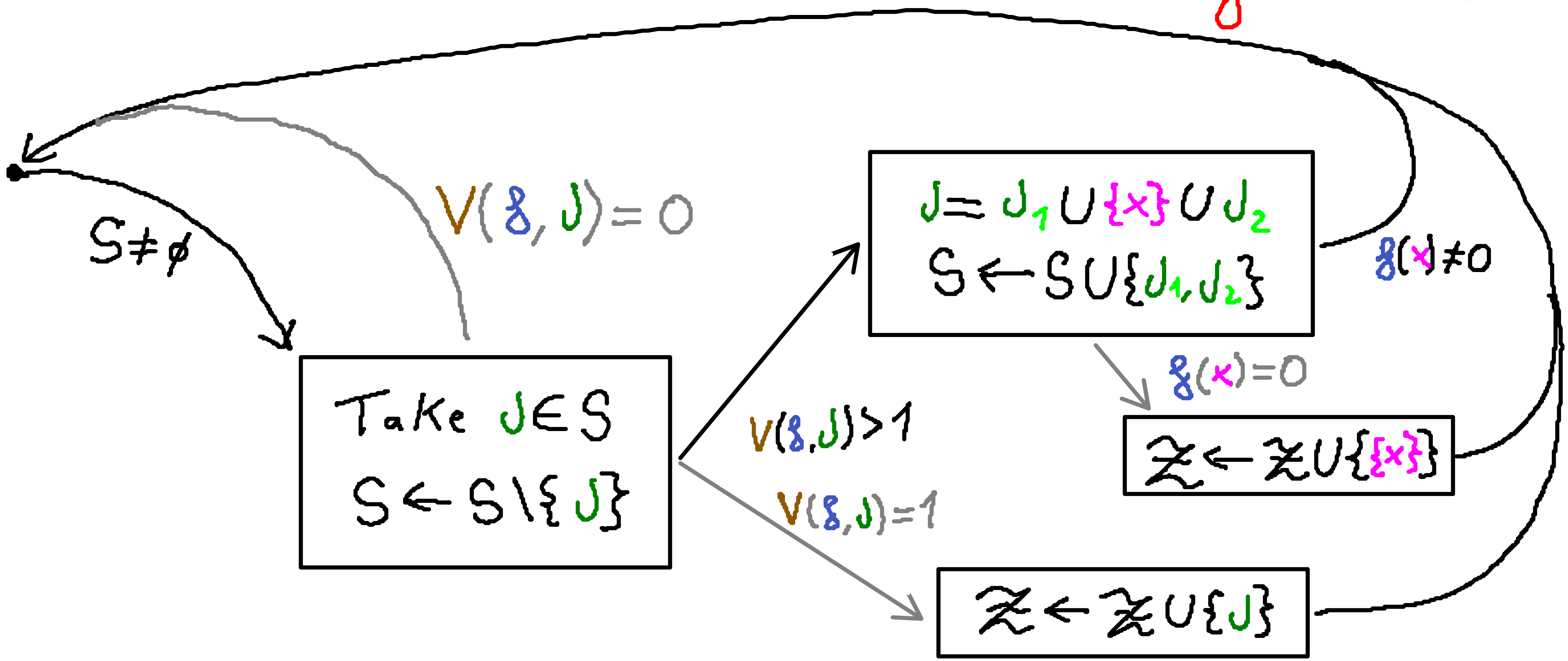
Descartes Solver III:

The Algorithm

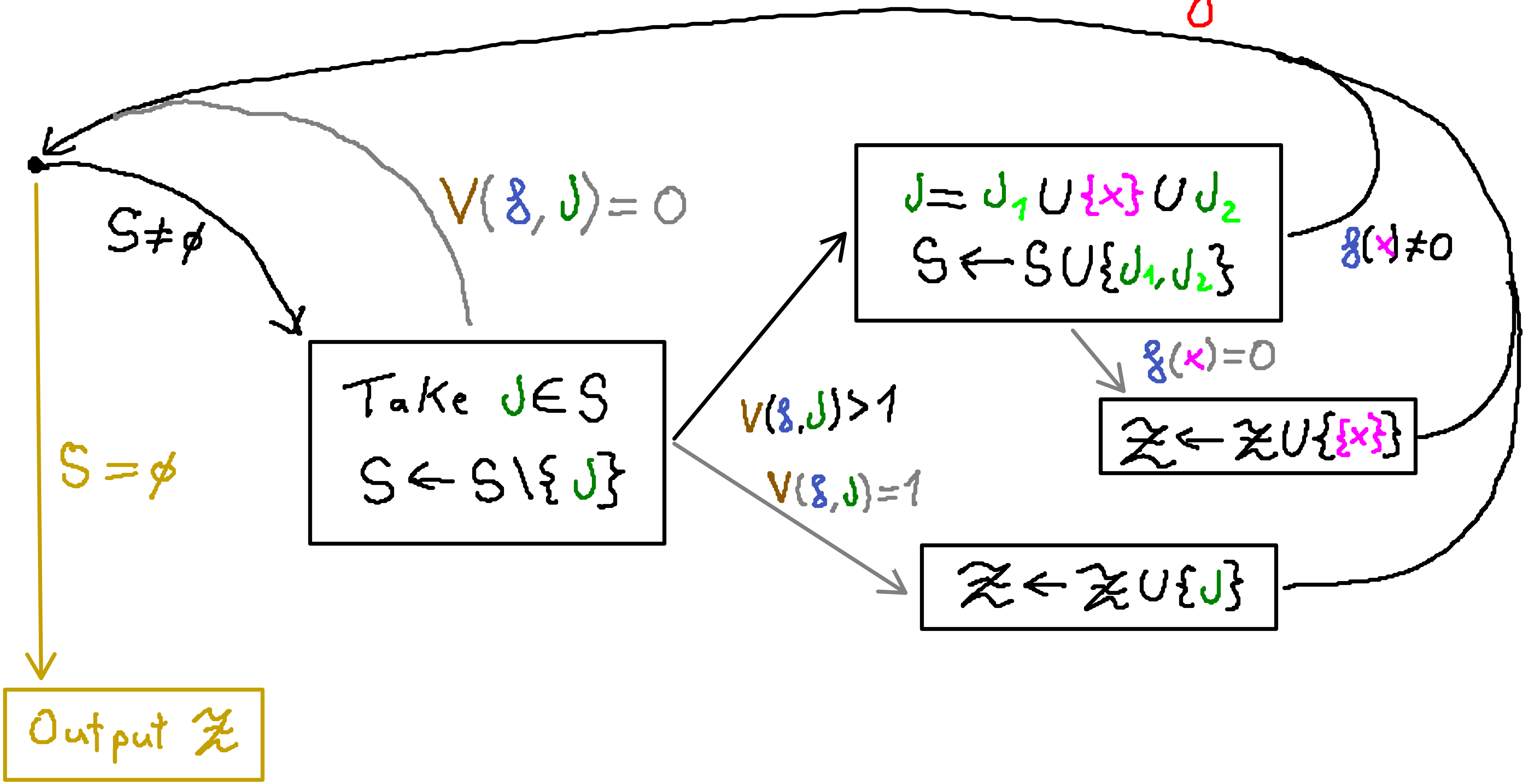


Descartes Solver III:

The Algorithm

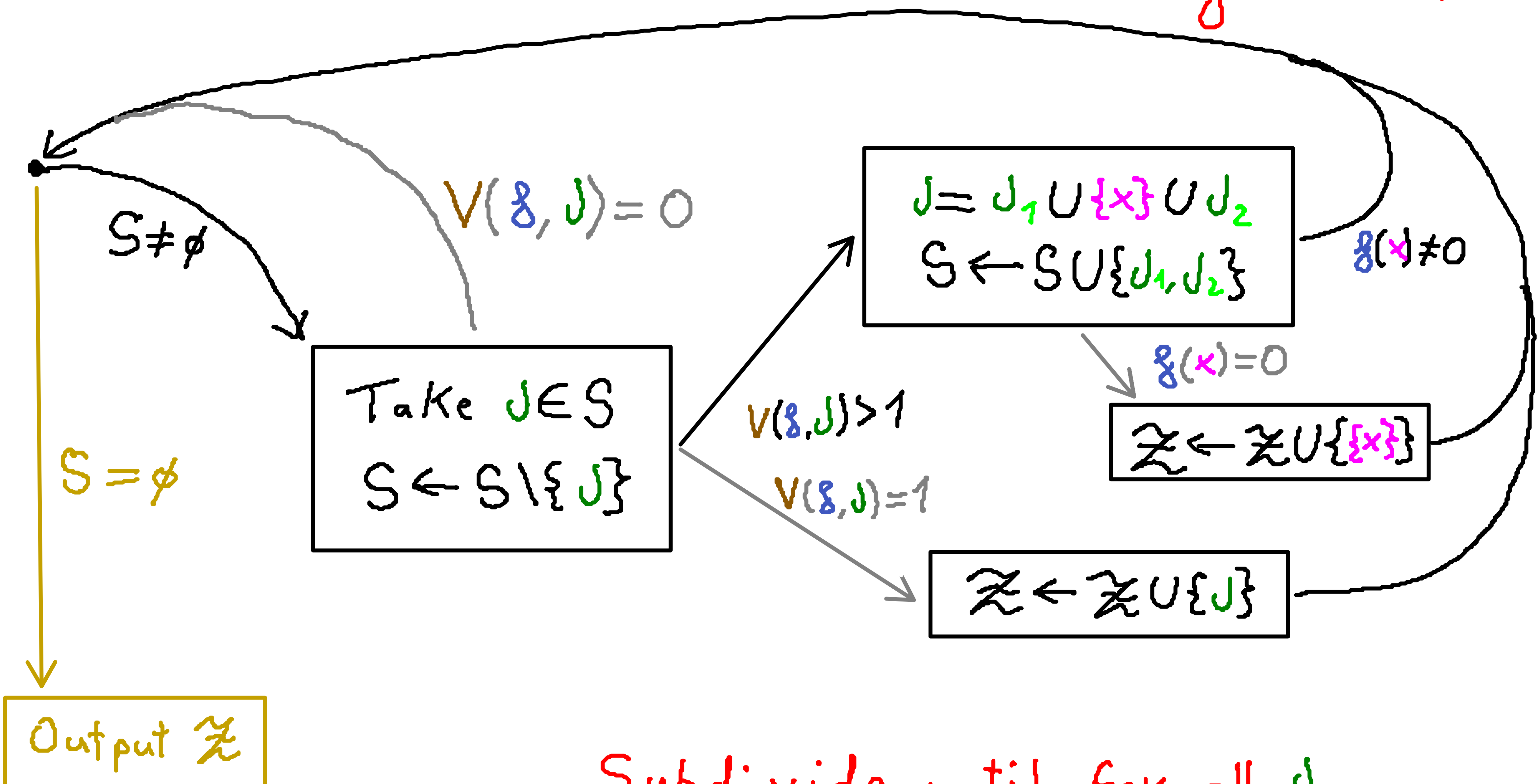


Descartes Solver III: The Algorithm



Descartes Solver III:

The Algorithm



Subdivide until for all J, $V(g, J) \leq 1$!

DESCARTES SOLVER IV:

Descartes' tree

DESCARTES SOLVER IV:

Descartes' tree

$$\Upsilon(\delta, I)$$

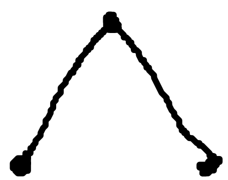
Descartes Solver IV: Descartes' tree

$\Upsilon(\delta, \mathrm{I})$

.

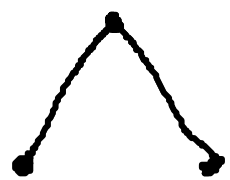
Descartes Solver IV: Descartes' tree

$\Upsilon(\gamma, I)$



Descartes Solver IV: Descartes' tree

$\Upsilon(g, I)$

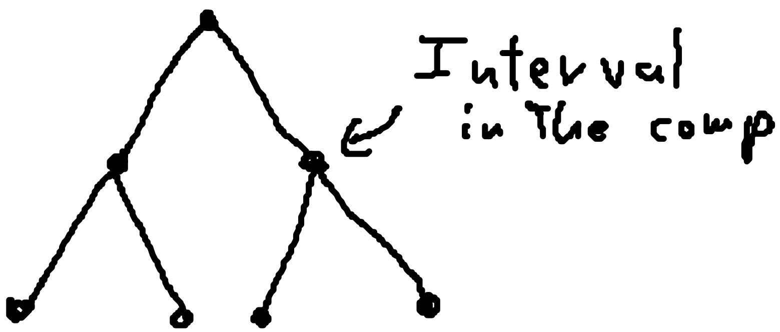


Interval in the comp

Descartes Solver IV:

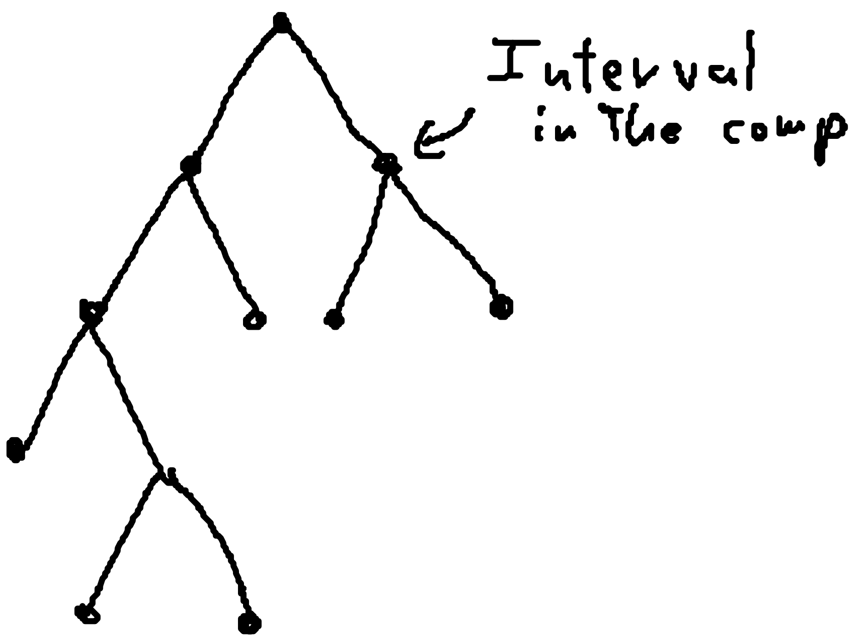
Descartes' tree

$\gamma(g, I)$



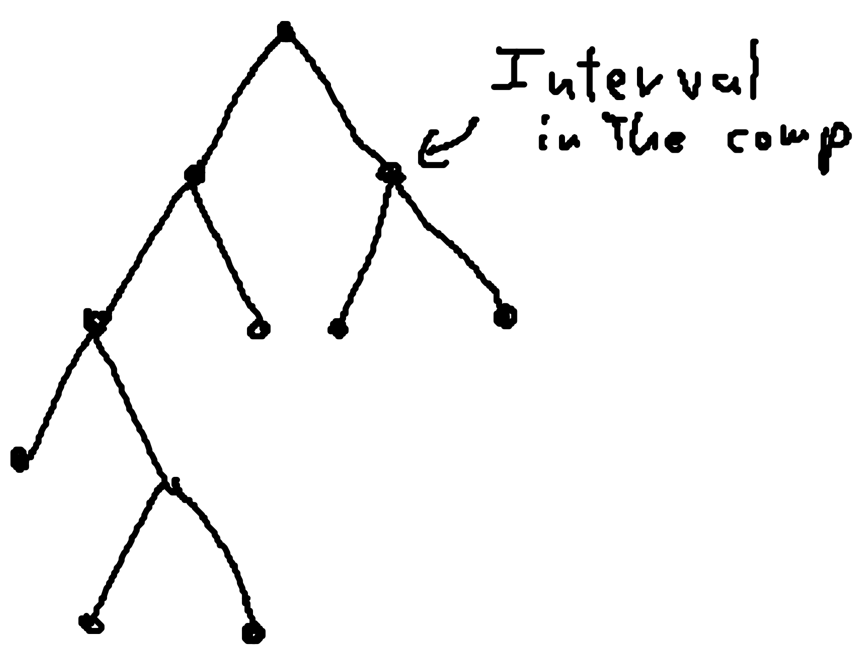
Descartes Solver IV: Descartes' tree

$\tau(g, I)$



Descartes Solver IV: Descartes' tree

$\Upsilon(g, I)$



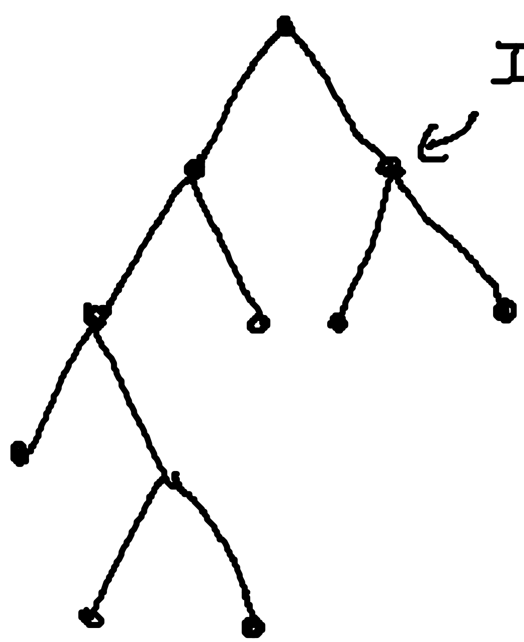
size of $\Upsilon(g, I)$

$\updownarrow$

run-time of Descartes(g, I)

Descartes Solver IV: Descartes' tree

$\Upsilon(g, I)$



Interval in the comp

size of $\Upsilon(g, I)$

$\Updownarrow$

run-time of DESCARTES(g, I)

We only need to control the size of subdiv. tree!

Real Root Isolation IV: Are we being pessimistic?

¿Can we explain this?

Real Root Isolation IV:

Are we being pessimistic?

DESCARTES SOLVER
seems to behave faster in practice!

¿Can we explain this?

Real Root Isolation V: Beyond pessimism

Yes,

Real Root Isolation V: Beyond pessimism

Yes, bounding

$$\mathbb{E}\left\{\mathrm{cost}(\mathrm{SOLVER}, \mathfrak{f})^{\ell} \mid \text{bit-size}(\mathfrak{f}) \leq \tau, \deg(\mathfrak{f}) \leq d\right\}$$

Real Root Isolation V:
Beyond pessimism

Yes, bounding

$$\mathbb{E}\{\mathrm{cost}(\mathrm{SOLVER}, \mathfrak{f})^{\ell} \mid \text{bit-size}(\mathfrak{f}) \leq \tau, \deg(\mathfrak{f}) \leq d\}$$

What's a 'good' random model for $\mathfrak{f}$?

Real Root Isolation V:

Beyond pessimism

Yes, bounding

$$\mathbb{E}\{\mathrm{cost}(\mathrm{SOLVER}, \mathfrak{g})^{\ell} \mid \text{bit-size}(\mathfrak{g}) \leq \tau, \deg(\mathfrak{g}) \leq d\}$$

What's a 'good' random model for $\mathfrak{g}$?

Many choices of randomness 😱

Beyond pessimism I: Uniform Random Bit Polynomials & a Simple Main Theorem

Beyond pessimism I: Uniform Random Bit Polynomials & a Simple Main Theorem

$$F = \sum_{k=0}^{d} f_k X^k$$

Beyond pessimism I: Uniform Random Bit Polynomials & a Simple Main Theorem

$$F = \sum_{k=0}^{d} f_k X^k$$

s.t. $f_k \sim \mathcal{U}([-2^\tau, 2^\tau] \cap \mathbb{Z})$ independent

Beyond pessimism I: Uniform Random Bit Polynomials & a Simple Main Theorem

$$F = \sum_{k=0}^{d} f_k X^k$$

s.t. $f_k \sim \mathcal{U}([-2^{\tau}, 2^{\tau}] \cap \mathbb{Z})$ independent

SIMPLE MAIN THM

Beyond pessimism I:
Uniform Random Bit Polynomials
& a Simple Main Theorem

$$F = \sum_{k=0}^{d} f_k X^k$$

s.t. $f_k \sim \mathcal{U}([-2^\tau, 2^\tau] \cap \mathbb{Z})$ independent

SIMPLE MAIN THM

$$\mathbb{E}\, \mathrm{cost}(\mathrm{DESCARTES}, F) = \widetilde{\mathcal{O}}_B(d^2 + d\tau)$$

Beyond pessimism I: Uniform Random Bit Polynomials & a Simple Main Theorem

$$F = \sum_{k=0}^{d} f_k X^k$$

s.t. $f_k \sim \mathcal{U}([-2^{\tau}, 2^{\tau}] \cap \mathbb{Z})$ independent

SIMPLE MAIN THM

$$\mathbb{E}\,\mathrm{cost}(\text{DESCARTES}, F) = \tilde{\mathcal{O}}_B(d^2 + d\tau)$$

On average, DESCARTES is almost optimal!

Beyond pessimism II: Random Bit Polynomials

Beyond pessimism II: Random Bit Polynomials

$$f = \sum_{k=0}^{d} f_k X^k \in \mathbb{Z}[X]$$

Beyond pessimism II:

Random Bit Polynomials

$$f = \sum_{k=0}^{d} f_k X^k \in \mathbb{Z}[X]$$

s.t. f_k independent

Beyond pessimism II: Random Bit Polynomials

$$f = \sum_{k=0}^{d} f_k X^k \in \mathbb{Z}[X]$$

s.t. f_k independent

bit-size of f:

$$\tau(f) := \min\{\tau \mid \forall k, \mathbb{P}(|f_k| \leq 2^\tau) = 1\}$$

Beyond pessimism II: Random Bit Polynomials

$$f = \sum_{k=0}^{d} f_k X^k \in \mathbb{Z}[X]$$

s.t. f_k independent

bit-size of f:

$$\tau(f) := \min\{\tau \mid \forall k, \mathbb{P}(|f_k| \leq 2^\tau) = 1\}$$

weight of f:

$$w(f) := \max\{\mathbb{P}(f_k = c) \mid c \in \mathbb{R}, k \in \{0, 1, d-1, d\}\}$$

Beyond pessimism II: Random Bit Polynomials

$$f = \sum_{k=0}^{d} f_k X^k \in \mathbb{Z}[X]$$

s.t. f_k independent

bit-size of f:

$$\tau(f) := \min\{\tau \mid \forall k, \mathbb{P}(|f_k| \leq 2^\tau) = 1\}$$

weight of f:

No middle indexes!

$$w(f) := \max\{\mathbb{P}(f_k = c) \mid c \in \mathbb{R}, k \in \{0, 1, d-1, d\}\}$$

Beyond pessimism II: Random Bit Polynomials

$$f = \sum_{k=0}^{d} f_k X^k \in \mathbb{Z}[X]$$

s.t. f_k independent

bit-size of f:

$$\tau(f) := \min\{\tau \mid \forall k, \mathbb{P}(|f_k| \leq 2^\tau) = 1\}$$

weight of f:

No middle indexes!

$$w(f) := \max\{\mathbb{P}(f_k = c) \mid c \in \mathbb{R}, k \in \{0, 1, d-1, d\}\}$$

uniformity of f: $u(f) := \ln\left(w(f)\left(1 + 2^{\tau(f)+1}\right)\right)$

Beyond pessimism III:

Main Theorem

Beyond pessimism III: MAIN THEOREM

MAIN THM

$$\mathbb{E}\,\mathrm{cost}(\text{DESCARTES}, F) = \widetilde{O}_{B}\left(d^{2} + d\tau\right)\left(1 + u(F)\right)^{4}$$

Beyond pessimism III: Main Theorem

Main Thm

$$\mathbb{E}\,\mathrm{cost}(\text{Descartes}, f) = \tilde{O}_B\left(d^2 + d\tau\right)\left(1 + u(f)\right)^4$$

Note: f uniform $\Rightarrow u(f) = 0$

Beyond pessimism III: Main Theorem

Main Thm

$$\mathbb{E}\,\mathrm{cost}(\mathrm{DESCARTES}, F) = \widetilde{O}_B\left(d^2 + d\tau\right)\left(1 + u(F)\right)^4$$

Note: F uniform $\Rightarrow u(F) = 0$

Claim: For many cases, $u(F) = O(1)$

Beyond pessimism III: Main Theorem

Main Thm

$$\mathbb{E}\,\mathrm{cost}(\textsc{Descartes}, f) = \widetilde{O}_B\big(d^2 + d\tau\big)\big(1 + u(f)\big)^4$$

Note: f uniform $\Rightarrow u(f) = 0$

Claim: For many cases, $u(f) = O(1)$

If $\tau = \Omega(d)$, almost like reading!

Beyond pessimism III:

Main Theorem

Main Thm

$$\mathbb{E}\,\mathrm{cost}(\textsc{Descartes}, f) = \widetilde{\mathcal{O}}_B\left(d^2 + d\tau\right)\left(1 + u(f)\right)^4$$

Note: f uniform $\Rightarrow$ $u(f) = 0$

Claim: For many cases, $u(f) = \mathcal{O}(1)$

If $\tau = \Omega(d)$, almost like reading!

On average, Descartes is almost optimal!

Beyond pessimism IV:

Examples of Random Bit Polynomials I

Beyond pessimism IV: Examples of Random Bit Polynomials I

- Support control $\{0, 1, d-1, d\} \subseteq A$

$$f = \sum_{k \in A} f_k X^k \quad \text{with } f_k \sim \mathcal{U}\left([-2^\tau, 2^\tau] \cap \mathbb{Z}\right)$$

... then $u(f) = 0$

Beyond pessimism IV: Examples of Random Bit Polynomials I

- Support control $\{0,1,d-1,d\} \subseteq A$

$$f = \sum_{k \in A} f_k X^k \quad \text{with } f_k \sim \mathcal{U}\left([-2^\tau, 2^\tau] \cap \mathbb{Z}\right)$$

... then $u(f) = 0$

- Sign control $\sigma \in \{-1,+1\}^{\{0,\dots,d\}}$

$$f = \sum_{k=1}^{d} f_k X^k \quad \text{with } f_k \sim \mathcal{U}\left(\sigma_k\left([1,2^\tau] \cap \mathbb{N}\right)\right)$$

... then $u(f) \leq \ln 3$

Beyond pessimism V:
Examples of Random Bit Polynomials II

- Exact bitsize

$$F = \sum_{k=1}^{d} f_k X^k \quad \text{with} \quad f_k \sim \mathcal{U}(\{n \in \mathbb{Z} \mid \lfloor \log n \rfloor = \tau\})$$

$$\ldots \text{ then } u(F) \leq \ln 3$$

Beyond pessimism V:
Examples of Random Bit Polynomials II

- Exact bitsize

$$F = \sum_{k=1}^{d} f_k X^k \quad \text{with} \quad f_k \sim \mathcal{U}(\{n \in \mathbb{Z} \mid \lfloor \log n \rfloor = \tau\})$$

... then $u(F) \leq \ln 3$

+ their combinations

Beyond pessimism V:
Examples of Random Bit Polynomials II

- Exact bitsize

$$F = \sum_{k=1}^{d} f_k X^k \quad \text{with} \quad f_k \sim \mathcal{U}(\{n \in \mathbb{Z} \mid \lfloor \log n \rfloor = \tau\})$$

... then $u(F) \leq \ln 3$

+ their combinations

Our random model is flexible!

Beyond pessimism $\boxed{\mathrm{V}}$:

Smoothed case included!

Beyond pessimism $\boxed{V}$:

Smoothed case included!

$$F = \sum_{k=1}^{d} f_k X^k \qquad \text{random bit polynomial}$$

Beyond pessimism $\boxed{\text{V}}$: Smoothed case included!

$$F = \sum_{k=1}^{d} f_k X^k$$ random bit polynomial

$$g = \sum_{k=1}^{d} g_k X^k$$ fix polynomial of entries of size τ

Beyond pessimism VI: Smoothed case included!

$$F = \sum_{k=1}^{d} f_k X^k$$ random bit polynomial

$$g = \sum_{k=1}^{d} g_k X^k$$ fix polynomial of entries of size γ

$$\sigma \in \mathbb{Z} \setminus \{0\}$$

Beyond pessimism V: Smoothed case included!

$F = \sum_{k=1}^{d} F_k X^k$ random bit polynomial

$g = \sum_{k=1}^{d} g_k X^k$ fix polynomial of entries of size τ

$\sigma \in \mathbb{Z} \setminus \{0\}$

Then:

$F_\sigma = g + \sigma F$ random bit polynomial

Beyond pessimism VI: Smoothed case included!

$f = \sum_{k=1}^{d} f_k X^k$ random bit polynomial

$g = \sum_{k=1}^{d} g_k X^k$ fix polynomial of entries of size τ

$\sigma \in \mathbb{Z} \setminus \{0\}$

Then:

$f_\sigma = g + \sigma f$ random bit polynomial

& $u(f_\sigma) \leq 1 + u(f) + \max\{\tau - \tau(f), \tau(\sigma)\}$

The Ingredients of the Analysis I: Condition Numbers

The Ingredients of the Analysis I: Condition Numbers

$$C(g) := \max_{x \in [-1,1]} \frac{\sum_{k=0}^{d} |g_k|}{\max\{|g(x)|, |g'(x)|/d\}}$$

The Ingredients of the Analysis I: Condition Numbers

$$C(g) := \max_{x \in [-1,1]} \frac{\sum_{k=0}^{d} |g_k|}{\max\{|g(x)|, |g'(x)|/d\}}$$

$C(g) = \infty \Leftrightarrow g$ has a singular root in $[-1,1]$

The Ingredients of the Analysis I: Condition Numbers

$$C(g) := \max_{x \in [-1,1]} \frac{\sum_{k=0}^{d} |g_k|}{\max\{|g(x)|, |g'(x)|/d\}}$$

$$C(g) = \infty \Leftrightarrow g \text{ has a singular root in } [-1,1]$$

The Ingredients of the Analysis I: Condition Numbers

$$C(g) := \max_{x \in [-1,1]} \frac{\sum_{k=0}^{d} |g_k|}{\max\{|g(x)|, |g'(x)|/d\}}$$

$C(g) = \infty \iff g$ has a singular root in $[-1,1]$

Upper bounds on $C(g)$

The Ingredients of the Analysis I: Condition Numbers

$$C(g) := \max_{x \in [-1,1]} \frac{\sum_{k=0}^{d} |g_k|}{\max\{|g(x)|, |g'(x)|/d\}}$$

$C(g) = \infty \Leftrightarrow g$ has a singular root in $[-1,1]$

Upper bounds on $C(g)$

$\rightarrow$ Lower bounds for root separation of g

The Ingredients of the Analysis I: Condition Numbers

$$C(g) := \max_{x \in [-1,1]} \frac{\sum_{k=0}^{d} |g_k|}{\max\{|g(x)|, |g'(x)|/d\}}$$

$C(g) = \infty \Leftrightarrow g$ has a singular root in $[-1,1]$

Upper bounds on $C(g)$

$\longrightarrow$ Lower bounds for root separation of g

$\longrightarrow$ Upper bounds for depth of DESCARTES' tree

The Ingredients of the Analysis II: Bounds for Number of Complex Roots

The Ingredients of the Analysis II: Bounds for Number of Complex Roots

We only care about nearby roots!

Upper bounds for

\# complex roots of g around $[-1,1]$

The Ingredients of the Analysis II: Bounds for Number of Complex Roots

Upper bounds for

We only care about nearby roots!

$\#$ complex roots of g around $[-1,1]$

$\longrightarrow$ Upper bounds for width of DESCARTES' tree

The Ingredients of the Analysis II: Bounds for Number of Complex Roots

We only care about nearby roots!

Upper bounds for

\# complex roots of g around $[-1,1]$

$\longrightarrow$ Upper bounds for width of DESCARTES' tree

Complex analysis!

Tichmarsh thm

The Ingredients of the Analysis III: Probabilistic Toolbox

The Ingredients of the Analysis III: Probabilistic Toolbox

Ball's smoothing:

The Ingredients of the Analysis III: Probabilistic Toolbox

Ball's smoothing:

$x \in \mathbb{Z}^N$ discrete random variable

The Ingredients of the Analysis III: Probabilistic Toolbox

Ball's smoothing:

$x \in \mathbb{Z}^N$ discrete random variable

$y \in \mathbb{R}^N$ s.t. $y_i \sim \mathcal{U}((-1/2, 1/2))$ i.i.d.

The Ingredients of the Analysis III: Probabilistic Toolbox

Ball's smoothing:

$x \in \mathbb{Z}^N$ discrete random variable

$y \in \mathbb{R}^N$ s.t. $y_i \sim \mathcal{U}((-1/2, 1/2))$ i.i.d.

Then: $x + y$ continuous random var.

The Ingredients of the Analysis III: Probabilistic Toolbox

Ball's smoothing:

$x \in \mathbb{Z}^N$ discrete random variable

$y \in \mathbb{R}^N$ s.t. $y_i \sim \mathcal{U}((-\frac{1}{2}, \frac{1}{2}))$ i.i.d.

Then: $x + y$ continuous random var.

We can use our old cont. toolbox!

The Ingredients of the Analysis III: Probabilistic Toolbox

Ball's smoothing:

$x \in \mathbb{Z}^N$ discrete random variable

$y \in \mathbb{R}^N$ s.t. $y_i \sim \mathcal{U}((-1/2, 1/2))$ i.i.d.

Then: $x + y$ continuous random var.

We can use our old cont. toolbox!

⚠ I am omitting a lot of technical details.

SUMMING UP:

DESCARTES

is almost optimal on average!

Eskerrik Asko
zure arretagatik!

I.e. Thank You for your attention!