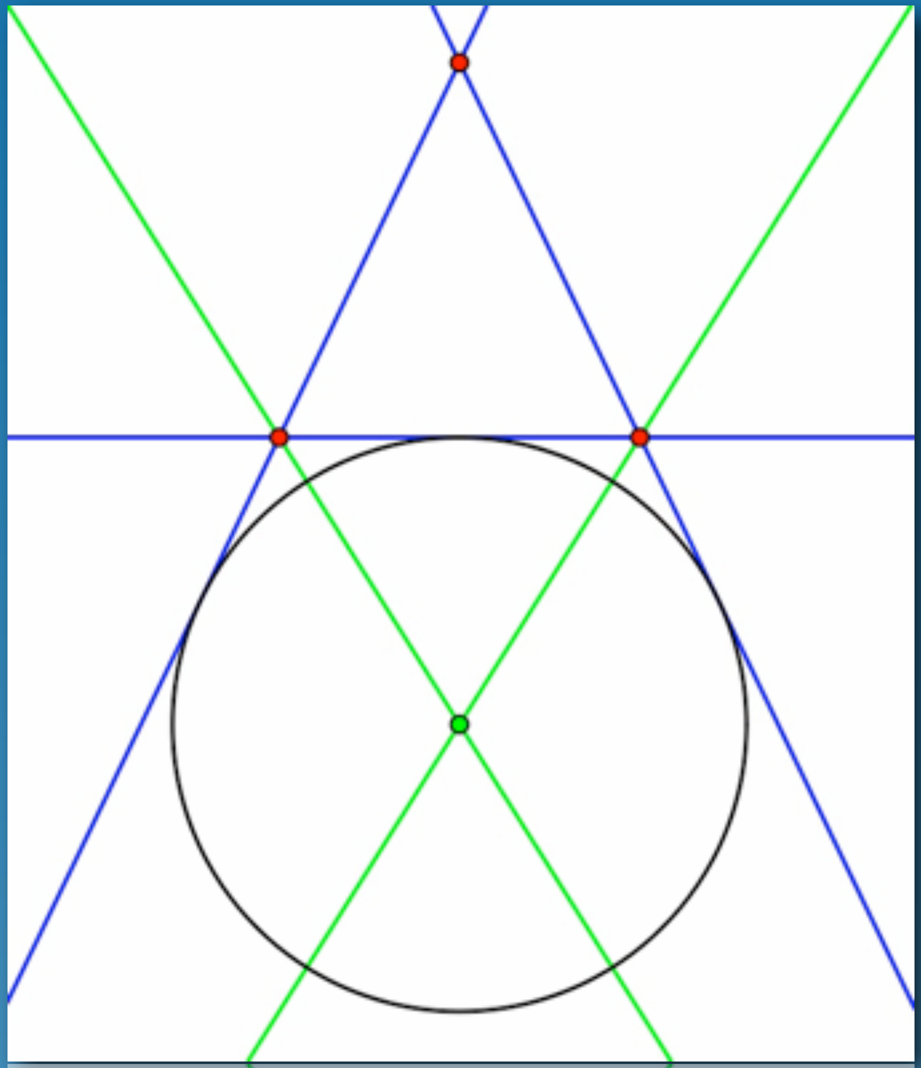
On the Complexity of the Reachability Problem in Dynamic Geometry

Thorsten Orendt

The Reachability Problem (*Reach*)

Can you walk continuously from one instance to another specific one?

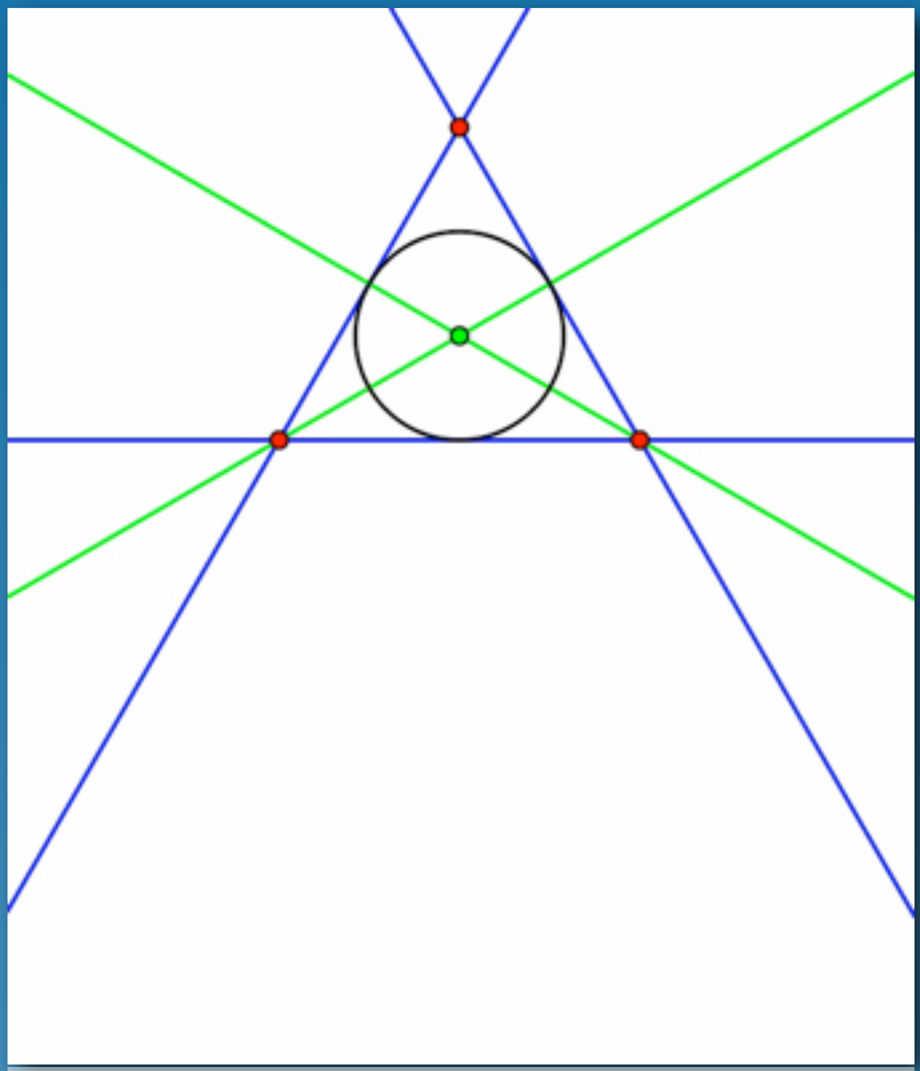


starting instance

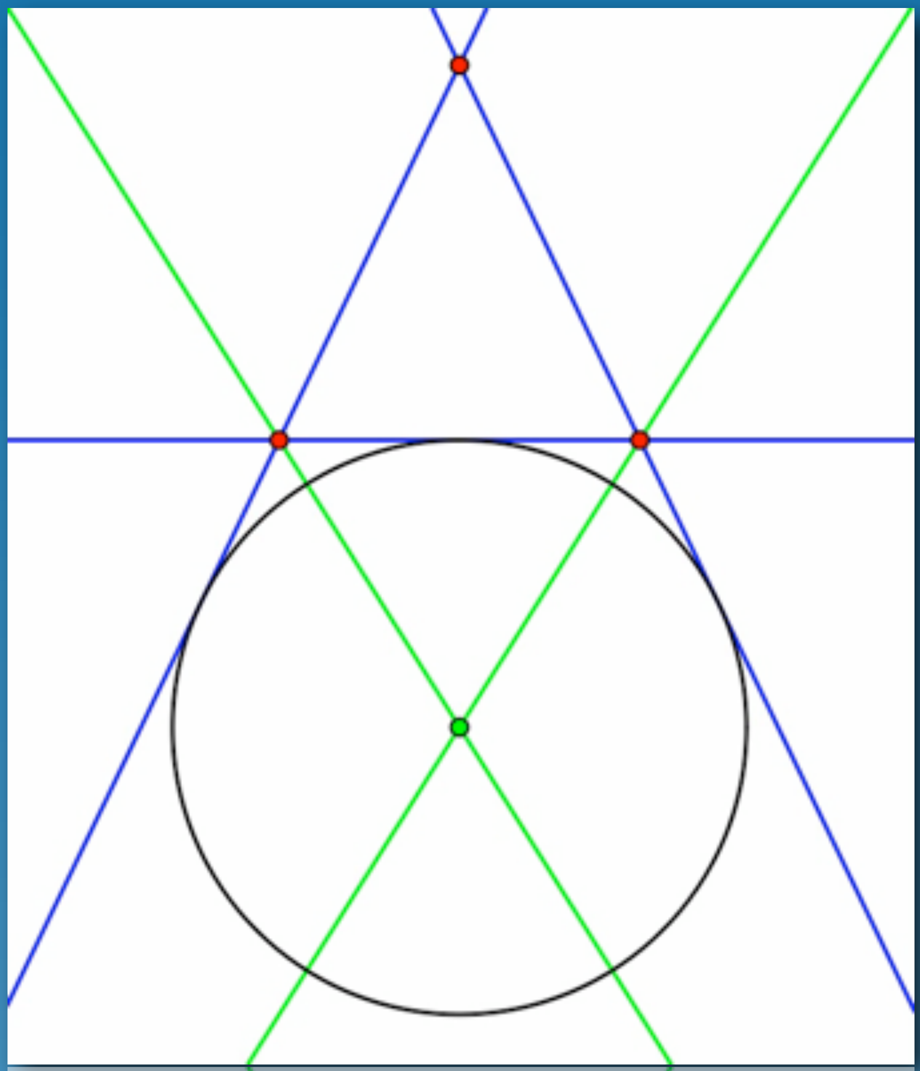
terminal instance

The Reachability Problem (*Reach*)

Can you walk continuously from one instance to another specific one?



starting instance



terminal instance

What do we know so far?

What do we know so far?

- complexity depends on ...
  - ... power of allowed geometric operations
  - ... restriction on movements of free elements

What do we know so far?

- complexity depends on ...

  ... power of allowed geometric operations

  ... restriction on movements of free elements

- if operations algebraically equivalent to solving quadratic polynomials, then *Reach* is ...

  ... decidable over $\mathbb{C}$ (Denner-Broser; ADG'04)

  ... NP-hard over $\mathbb{R}$ (Richter-Gebert, Kortenkamp; 2000)

What do we know so far?

- complexity depends on ...

  ... power of allowed geometric operations

  ... restriction on movements of free elements

- if operations algebraically equivalent to solving quadratic polynomials, then *Reach* is ...

  ... decidable over $\mathbb{C}$ (Denner-Broser; ADG'04)

  ... NP-hard over $\mathbb{R}$ (Richter-Gebert, Kortenkamp; 2000)

- no lower bounds over $\mathbb{C}$

We focus on a restriction of *Reach*

We focus on a restriction of *Reach*

- considered operations
    - constant point
    - join two points, meet two lines
    - intersection line and conic
    - intersection line and cubic curve

We focus on a restriction of *Reach*

considered operations	algebraically equivalent to
- constant point	constant
- join two points, meet two lines	arithmetics
- intersection line and conic	square root
- intersection line and cubic curve	cubic root

We focus on a restriction of *Reach*

- considered operations — algebraically equivalent to
  - constant point — constant
  - join two points, meet two lines — arithmetics
  - intersection line and conic — square root
  - intersection line and cubic curve — cubic root
- only movements of free elements with bounded length

We focus on a restriction of *Reach*

- considered operations — algebraically equivalent to
  - constant point — constant
  - join two points, meet two lines — arithmetics
  - intersection line and conic — square root
  - intersection line and cubic curve — cubic root

- only movements of free elements with bounded length

goal: show this restriction is NP-hard

From geometry to algebra

From geometry to algebra

geometric objects → complex numbers $\mathbb{C}$

From geometry to algebra

geometric objects	→	complex numbers $\mathbb{C}$
geometric operations	→	operations on $\mathbb{C}$

From geometry to algebra

geometric objects	→	complex numbers $\mathbb{C}$
geometric operations	→	operations on $\mathbb{C}$
construction sequence	→	geometric SLPs (GSPs)

From geometry to algebra

geometric objects	→	complex numbers $\mathbb{C}$
geometric operations	→	operations on $\mathbb{C}$
construction sequence	→	geometric SLPs (GSPs)
geometric instances of a construction	→	instances of a GSP

From geometry to algebra

geometric objects	→	complex numbers $\mathbb{C}$
geometric operations	→	operations on $\mathbb{C}$
construction sequence	→	geometric SLPs (GSPs)
geometric instances of a construction	→	instances of a GSP
Reach in dynamic geometry	→	*Reach* for GSPs

Considered operations

(performed at unit costs)

free	$\mathbb{C}$
constants	$\mathrm{const}_z = \{z\}$
arithmetic operations	$+ \;=\; \{(z_1, z_2, w) \in \mathbb{C}^3 \mid z_1 + z_2 = w\}$ $- \;=\; \{(z_1, z_2, w) \in \mathbb{C}^3 \mid z_1 - z_2 = w\}$ $* \;=\; \{(z_1, z_2, w) \in \mathbb{C}^3 \mid z_1 \cdot z_2 = w\}$ $/ \;=\; \{(z_1, z_2, w) \in \mathbb{C}^3 \mid z_1 = z_2 \cdot w \wedge z_2 \neq 0\}$
roots	$\sqrt{} \;=\; \{(z, w) \in \mathbb{C}^2 \mid z = w^2 \wedge z \neq 0\}$ $\sqrt[3]{} \;=\; \{(z, w) \in \mathbb{C}^2 \mid z = w^3 \wedge z \neq 0\}$

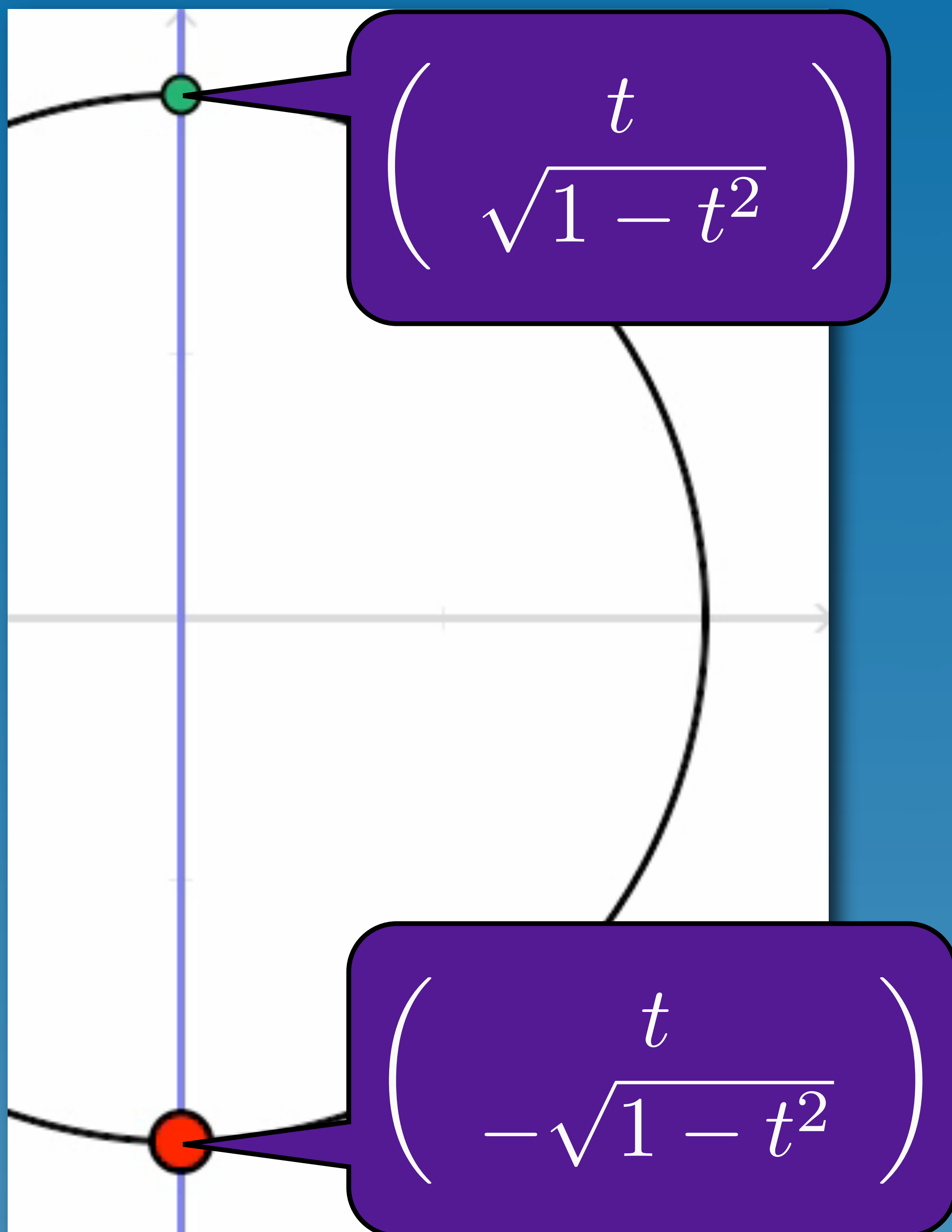
Considered operations

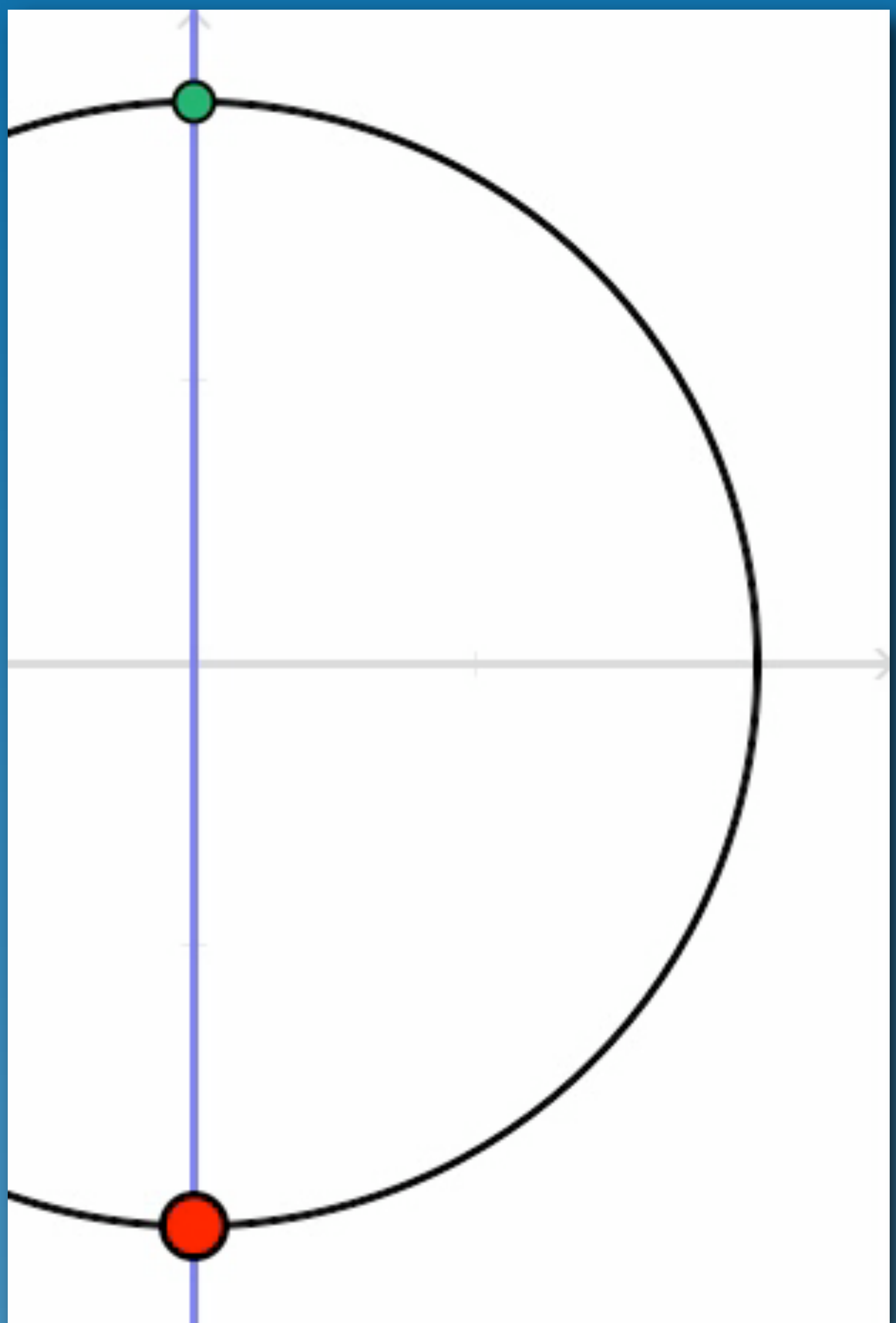
(performed at unit costs)

free	$\mathbb{C}$	
constants	$\mathrm{const}_z = \{z\}$	dependent
arithmetic operations	$+ \;=\; \{(z_1, z_2, w) \in \mathbb{C}^3 \mid z_1 + z_2 = w\}$	
	$- \;=\; \{(z_1, z_2, w) \in \mathbb{C}^3 \mid z_1 - z_2 = w\}$	
	$* \;=\; \{(z_1, z_2, w) \in \mathbb{C}^3 \mid z_1 \cdot z_2 = w\}$	
	$/ \;=\; \{(z_1, z_2, w) \in \mathbb{C}^3 \mid z_1 = z_2 \cdot w \wedge z_2 \neq 0\}$	
roots	$\sqrt{\ } \;=\; \{(z, w) \in \mathbb{C}^2 \mid z = w^2 \wedge z \neq 0\}$	
	$\sqrt[3]{\ } \;=\; \{(z, w) \in \mathbb{C}^2 \mid z = w^3 \wedge z \neq 0\}$	

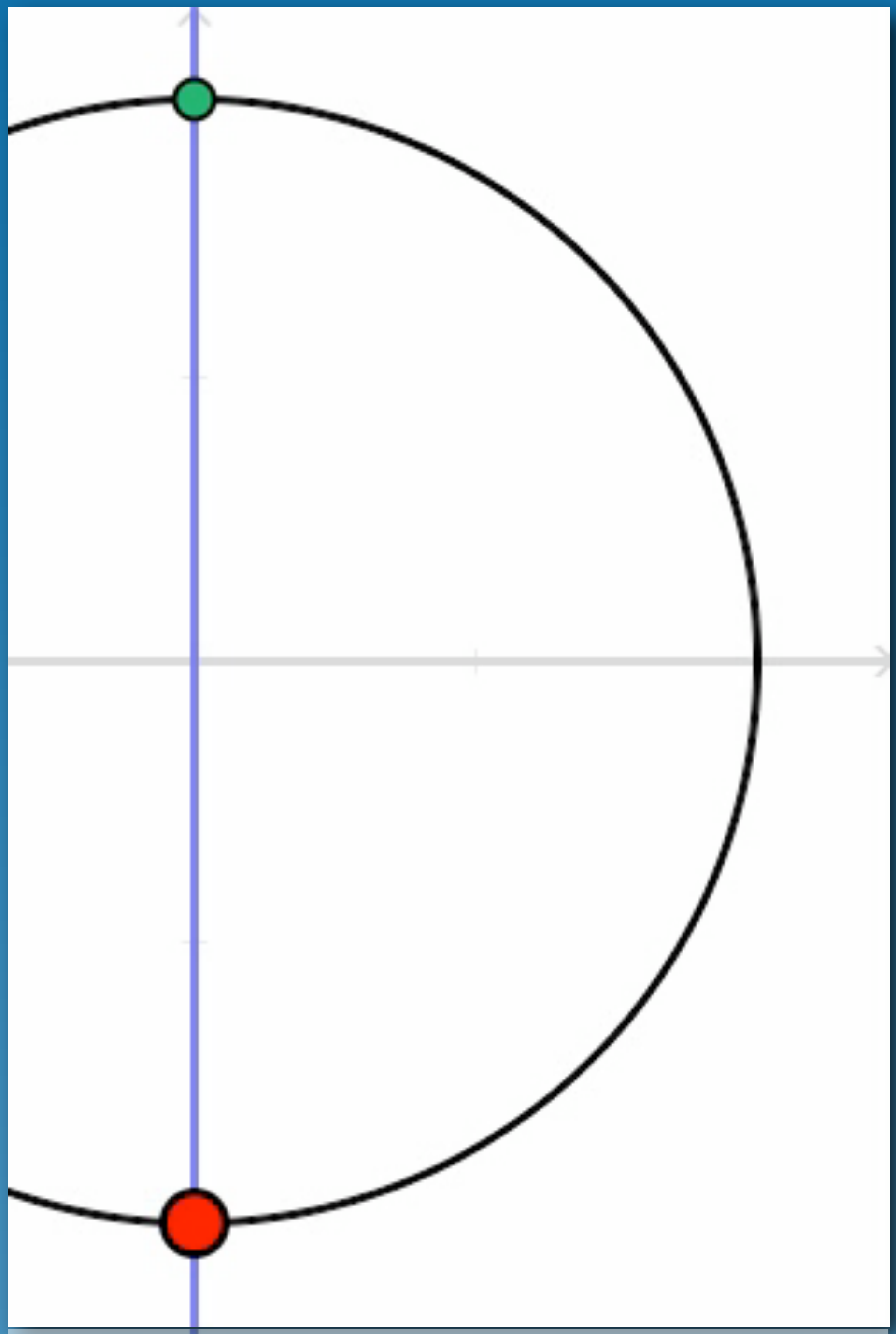
Why no roots of zero?

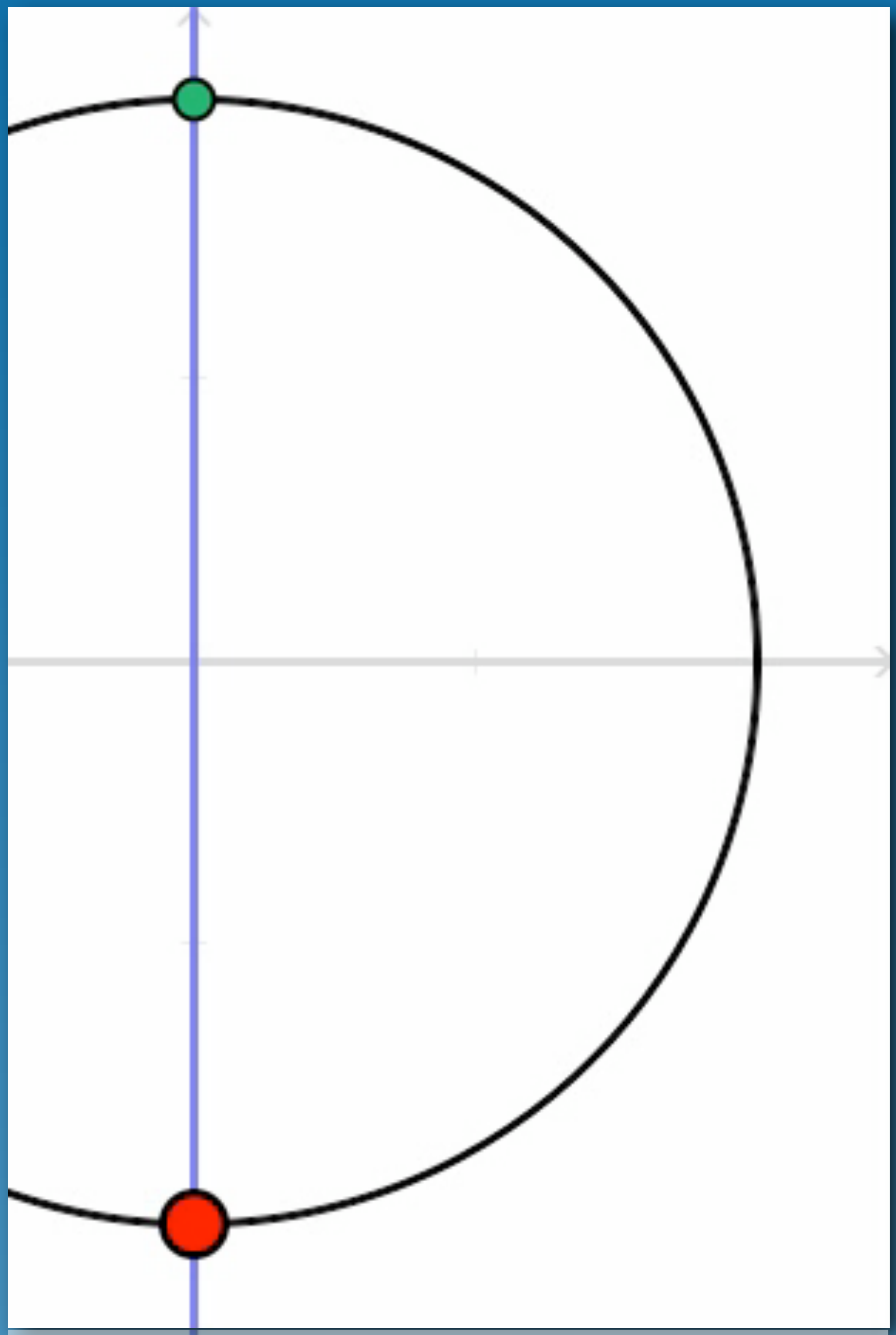
Why no roots of zero?



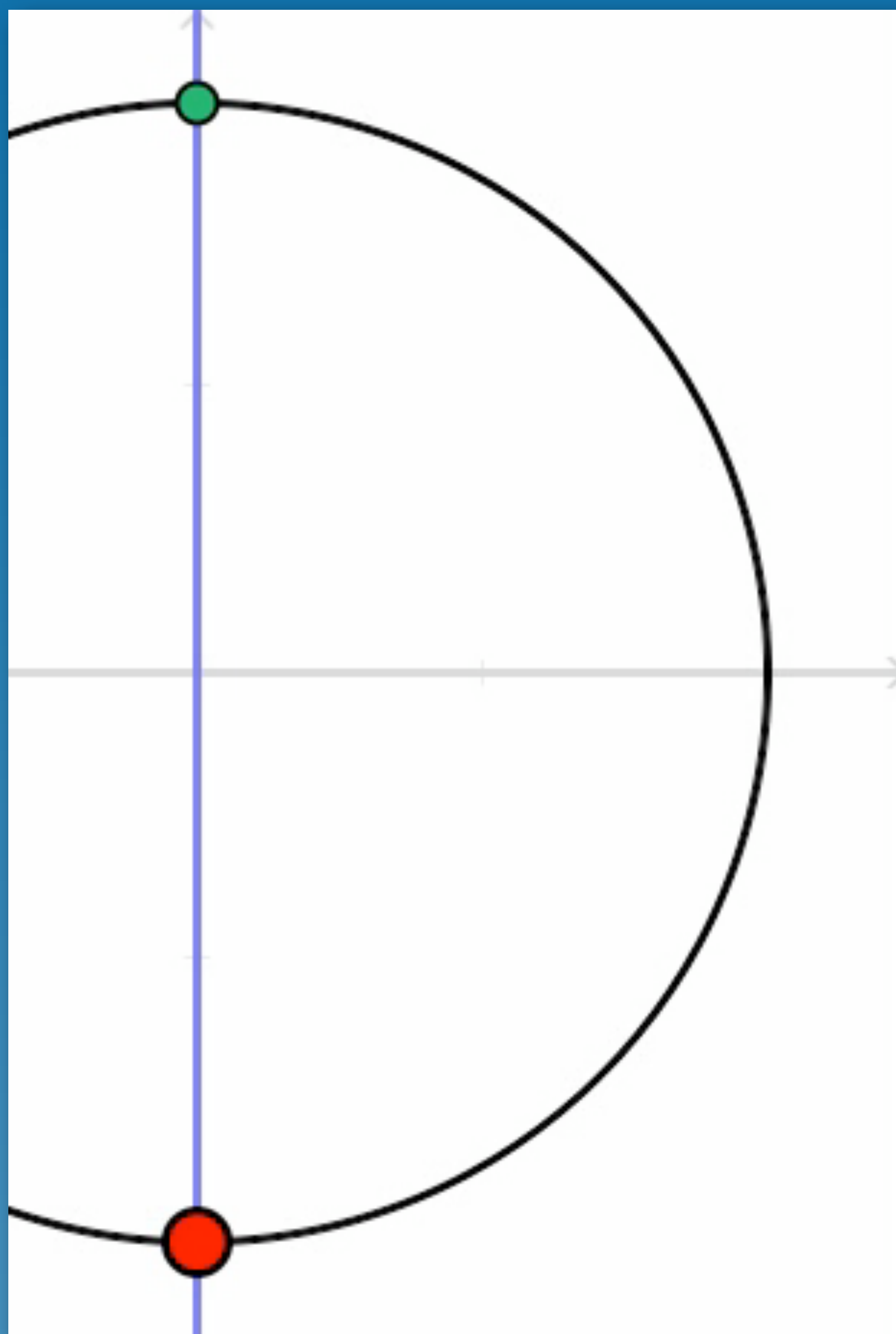


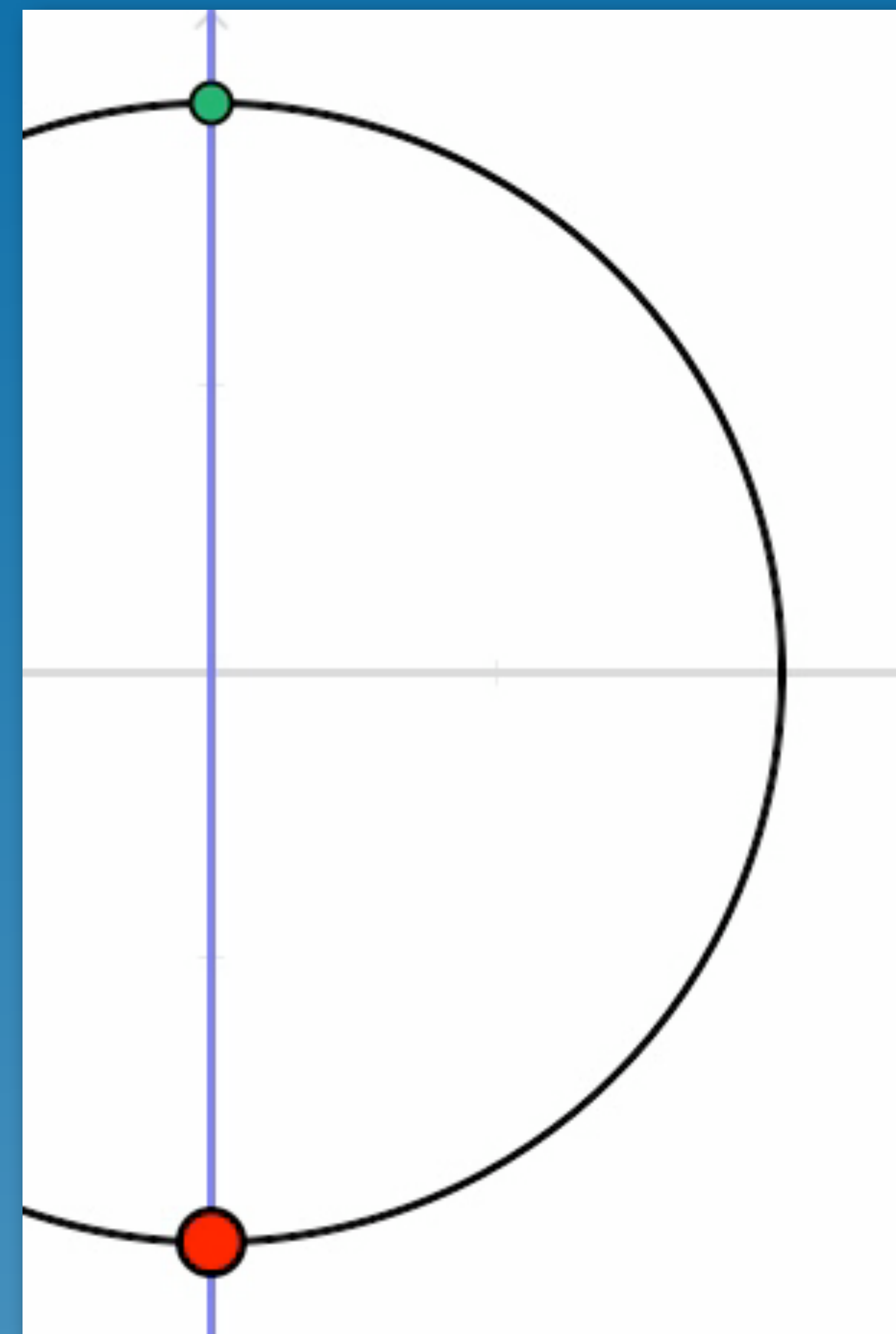
Why no roots of zero?



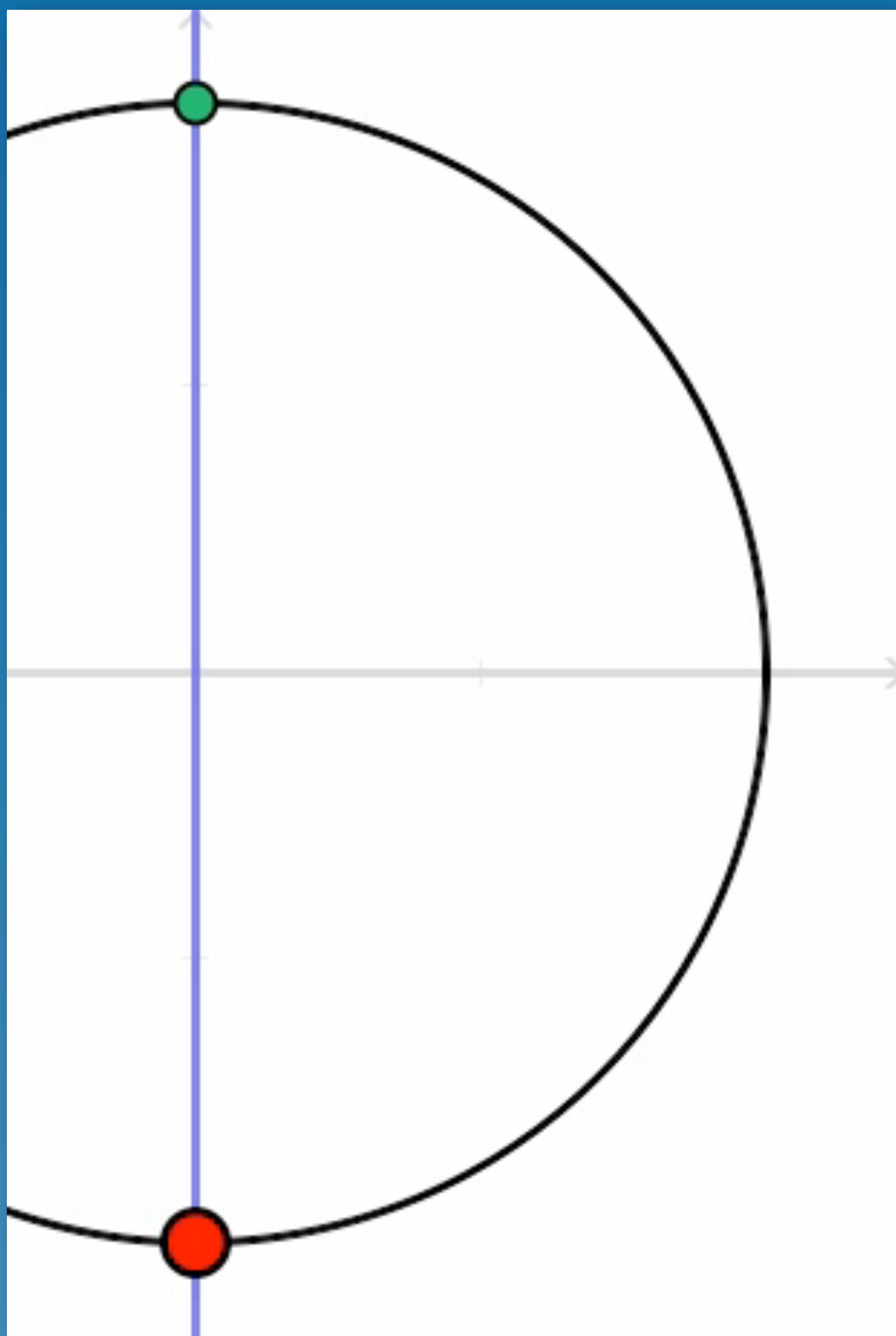


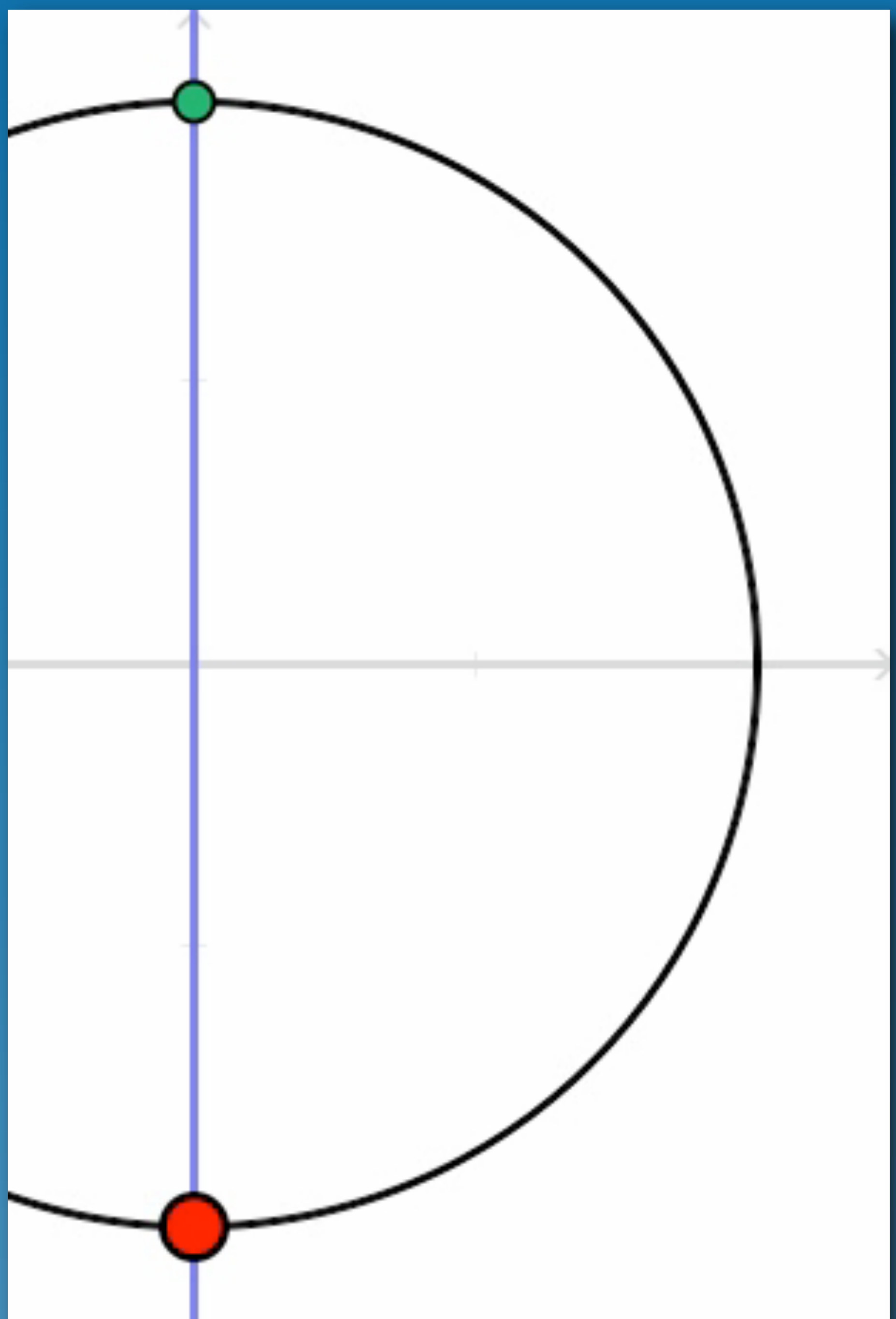
Why no roots of zero?





Why no roots of zero?





GSPs and their instances

GSPs and their instances

A **GSP** is a sequence $\underbrace{\omega_{1-p}, \ldots, \omega_0}_{\text{free}}, \underbrace{\omega_1, \ldots, \omega_q}_{\text{dependent}}$

of operations with specified input assignments

GSPs and their instances

A **GSP** is a sequence $\underbrace{\omega_{1-p}, \ldots, \omega_0}_{\text{free}}, \underbrace{\omega_1, \ldots, \omega_q}_{\text{dependent}}$

of operations with specified input assignments

An **instance** of a GSP is an assignment of complex numbers

$$Z = (z_{1-p}, \ldots, z_p) \in \mathbb{C}^{p+q}$$

so that the relations of dependent operations $\omega_1, \ldots, \omega_q$ are satisfied

Reach - algebraic version

Reach - algebraic version

given: GSP and two instance Z, W

Reach - algebraic version

given: GSP and two instance Z, W

problem: Are there continous mappings

$$\mu_{1-p}, \ldots, \mu_0 : [0,1] \to \mathbb{C}$$
$$c_1, \ldots, c_q : [0,1] \to \mathbb{C}$$

so that

$$(\mu_{1-p}(t), \ldots, c_q(t))$$

forms an instance for all $t \in [0,1]$ and

$$(\mu_{1-p}(0), \ldots, c_q(0)) = Z \wedge (\mu_{1-p}(1), \ldots, c_q(1)) = W$$

A variant of 3SAT

bool. variables	$b_1, \ldots, b_n$
literals	$\{b_1, \ldots, b_n, \neg b_1, \ldots, \neg b_n\}$
clauses	$C_j = l_{j,r} \vee l_{j,s} \vee l_{j,t} \quad (l_{j,k} \in \{b_k, \neg b_k\})$
formula	$C = C_1 \wedge \ldots \wedge C_m$
truth assign.	$\chi = (b_1, \ldots, b_n) \in \{true, false\}^n$

A variant of 3SAT

bool. variables	$b_1, \ldots, b_n$
literals	$\{b_1, \ldots, b_n, \neg b_1, \ldots, \neg b_n\}$
clauses	$C_j = l_{j,r} \vee l_{j,s} \vee l_{j,t} \quad (l_{j,k} \in \{b_k, \neg b_k\})$
formula	$C = C_1 \wedge \ldots \wedge C_m$
truth assign.	$\chi = (b_1, \ldots, b_n) \in \{true, false\}^n$

Exact 3SAT: Is there a truth assigment that makes exactly one literal true in each clause?

A variant of 3SAT

bool. variables	$b_1, \ldots, b_n$
literals	$\{b_1, \ldots, b_n, \neg b_1, \ldots, \neg b_n\}$
clauses	$C_j = l_{j,r} \vee l_{j,s} \vee l_{j,t} \quad (l_{j,k} \in \{b_k, \neg b_k\})$
formula	$C = C_1 \wedge \ldots \wedge C_m$
truth assign.	$\chi = (b_1, \ldots, b_n) \in \{true, false\}^n$

Exact 3SAT: Is there a truth assigment that makes exactly one literal true in each clause?

NP-complete

Outline reduction

1. From 3SAT formulas to functions
2. From truth assignments to analytic continuations
3. Counting multiplicities
4. Using the bounded length
5. Assembling the parts

Basic idea

Basic idea

Transfer

$l_{j,k}$ literal	→	$x_{j,k}$ function
χ truth assignment	→	γ closed path

Basic idea

Transfer

$l_{j,k}$ literal	⇨	$x_{j,k}$ function
χ truth assignment	⇨	γ closed path

so that

χ makes $l_{j,k}$

$true$	$\Longleftrightarrow$	$x^{\gamma}_{j,k}(0) = 0$
$false$	$\Longleftrightarrow$	$x^{\gamma}_{j,k}(0) = 1$

1. Formulas → Functions

$$\vee \quad \rightarrow \quad *$$

$$l_{j,k} = b_k \quad \rightarrow \quad x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

$$l_{j,k} = \neg b_k \quad \rightarrow \quad x_{j,k}(z) = \frac{\sqrt{k} + \sqrt{k+z}}{2\sqrt{k}}$$

1. Formulas → Functions

$$\vee \quad \longrightarrow \quad *$$

$$l_{j,k} = b_k \quad \longrightarrow \quad x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

$$l_{j,k} = \neg b_k \quad \longrightarrow \quad x_{j,k}(z) = \frac{\sqrt{k} + \sqrt{k+z}}{2\sqrt{k}}$$

principal branch

1. Formulas → Functions

$$\vee \quad \rightarrow \quad *$$

$$l_{j,k} = b_k \quad \rightarrow \quad x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

principal branch

$$l_{j,k} = \neg b_k \quad \rightarrow \quad x_{j,k}(z) = \frac{\sqrt{k} + \sqrt{k+z}}{2\sqrt{k}}$$

$$C_j = l_{j,r} \vee l_{j,s} \vee l_{j,t} \quad \rightarrow \quad X_j(z) = \prod_{k \in \{r,s,t\}} x_{j,k}(z)$$

1. Formulas → Functions

$$\vee \quad \to \quad *$$

$$l_{j,k} = b_k \quad \to \quad x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

$$l_{j,k} = \neg b_k \quad \to \quad x_{j,k}(z) = \frac{\sqrt{k} + \sqrt{k+z}}{2\sqrt{k}}$$

principal branch

$$C_j = l_{j,r} \vee l_{j,s} \vee l_{j,t} \quad \to \quad X_j(z) = \prod_{k \in \{r,s,t\}} x_{j,k}(z)$$

$$C = C_1 \wedge \ldots \wedge C_m \quad \to \quad X(z) = \big(X_1(z), \ldots, X_m(z)\big)$$

2. Truth assign. → analy. continuations

$$\chi = (b_1, \ldots, b_n) \in \{true, false\}^n$$

↓ encode χ by the winding numbers of a closed path γ starting at $z = 0$

$$\eta(\gamma, -k) \in \begin{cases} 2\mathbb{Z}, & b_k = true \\ 2\mathbb{Z}+1, & b_k = false \end{cases}$$

Interplay $x_{j,k}$ and γ

$$l_{j,k} = b_k \qquad\qquad b_k = true$$

Interplay $x_{j,k}$ and γ

$l_{j,k} = b_k$ → **true** ← $b_k = true$

Interplay $x_{j,k}$ and γ

$l_{j,k} = b_k$ → **true** ← $b_k = true$

↓

$$x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

Interplay $x_{j,k}$ and γ

$l_{j,k} = b_k$ → **true** ← $b_k = true$

$$x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

$$\eta(\gamma, -k) \in 2\mathbb{Z}$$

Interplay $x_{j,k}$ and γ

$l_{j,k} = b_k$ → **true** ← $b_k = true$

↓ ↓

$$x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

does not change ← $\eta(\gamma, -k) \in 2\mathbb{Z}$

Interplay $x_{j,k}$ and γ

$l_{j,k} = b_k$ → **true** ← $b_k = true$

$$x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

does not change

$$\eta(\gamma, -k) \in 2\mathbb{Z}$$

$$x^{\gamma}_{j,k}(0) = 0$$

Interplay $x_{j,k}$ and γ

$$l_{j,k} = b_k$$

false

$$b_k = false$$

$$x_{j,k}(z) = \frac{\sqrt{k} - \sqrt{k+z}}{2\sqrt{k}}$$

does change

$$\eta(\gamma, -k) \in 2\mathbb{Z} + 1$$

$$x^{\gamma}_{j,k}(0) = 1$$

What have we achieved so far?

There is γ so that

$$X_C^\gamma(0) = (0, \ldots, 0) \in \mathbb{C}^m$$

$$\Updownarrow$$

C satisfiable

What have we achieved so far?

There is γ so that

$$X_C^\gamma(0) = \big(0, \ldots, 0) \in \mathbb{C}^m$$

$$\Updownarrow$$

C satisfiable

Why we are not done?

What have we achieved so far?

There is γ so that

$$X_C^\gamma(0) = (0, \ldots, 0) \in \mathbb{C}^m$$

$$\Updownarrow$$

C satisfiable

Why we are not done?

specifying a terminal instance $\Longleftrightarrow$ knowing a satisfying truth assignment

3. Counting multiplicities

How many literals are true in a clause?

↓

How many factors vanish in $X_j^{\gamma}(0)$?

↓

idea: use the three branches of $\sqrt[3]{X_j(z)}$ as counters

3. Counting multiplicities

- set $Y_j(z) = \sqrt[3]{X_j(z)}$ (prinicipal branch)
- denote branches of cubic root by 0,1,2

3. Counting multiplicities

- set $Y_j(z) = \sqrt[3]{X_j(z)}$ (prinicipal branch)
- denote branches of cubic root by 0,1,2

$$\Longrightarrow \; Y_j^{\gamma} \quad \text{lies on branch} \quad \frac{\eta(\gamma,0)}{2} \cdot M \pmod 3$$

nested roots

number of vanishing factors in $X_j^{\gamma}(0)$

3. Counting multiplicities

How to use it?

claim in starting instance

Y_j lies on branch 0

claim in terminal instance

Y_j^γ lies on branch 1

3. Counting multiplicities

How to use it?

claim in starting instance

Y_j lies on branch 0

claim in terminal instance

$$\frac{\eta(\gamma, 0)}{2} \cdot M \pmod 3 = 1$$

3. Counting multiplicities

How to use it?

claim in starting instance

Y_j lies on branch 0

claim in terminal instance

must control it

$$\frac{\eta(\gamma,0)}{2}\cdot M \pmod 3 = 1$$

4. Using bounded length

additional claims:

- $|\gamma| < 2(n+3) + \varepsilon\,,$ where $0 < \varepsilon \leq 1$
- γ circle around $-n-2$ and 1

5. Assembling the parts

exact 3SAT has answer **YES** $\iff$ $Y_1^\gamma, \ldots, Y_m^\gamma$ lie on branch 1

5. Assembling the parts

Idea proof:

1. adjusting multiplicities in X_j
2. switching $Y_j(z) = \sqrt[3]{X_j(z)}$ to branch I
3. reverse point I

Thank you for the attention!