## Admissible Digit Sets

Jesse Hughes<sup>1,2</sup> Milad Niqui<sup>2</sup>

<sup>1</sup>Technical University of Eindhoven

<sup>2</sup>Radboud University of Nijmegen

September 28, 2004

イロト イヨト イヨト イヨト

#### 1 Digit sets

- Binary representation
- Möbius maps and digit sets
- The Stern-Brocot representation

#### Admissibility

- Admissible digit sets
- The homographic algorithm

<ロ> (四) (四) (三) (三) (三)

#### 1 Digit sets

#### Binary representation

- Möbius maps and digit sets
- The Stern-Brocot representation

#### Admissibility

- Admissible digit sets
- The homographic algorithm

- ▲圖▶ | ▲圖▶ | ▲圖

#### 1 Digit sets

- Binary representation
- Möbius maps and digit sets
- The Stern-Brocot representation

#### 2 Admissibility

- Admissible digit sets
- The homographic algorithm

▲圖▶ ▲ 国▶ ▲ 国

#### Digit sets

- Binary representation
- Möbius maps and digit sets
- The Stern-Brocot representation

#### 2 Admissibility

- Admissible digit sets
- The homographic algorithm

▲□▶ ▲ □▶ ▲ □

#### 1 Digit sets

- Binary representation
- Möbius maps and digit sets
- The Stern-Brocot representation

## 2 Admissibility

- Admissible digit sets
- The homographic algorithm

#### 1 Digit sets

- Binary representation
- Möbius maps and digit sets
- The Stern-Brocot representation

#### 2 Admissibility

- Admissible digit sets
- The homographic algorithm

#### 1 Digit sets

- Binary representation
- Möbius maps and digit sets
- The Stern-Brocot representation

## 2 Admissibility

- Admissible digit sets
- The homographic algorithm

#### Digit sets

- Binary representation
- Möbius maps and digit sets
- The Stern-Brocot representation

## 2 Admissibility

- Admissible digit sets
- The homographic algorithm

<ロ> (日) (日) (日) (日) (日)

Binary representation Möbius maps and digit sets The Stern-Brocot representation

The standard binary representation of [0, 1].

Think of binary representations in [0, 1], like

0.010100010...



Binary representation Möbius maps and digit sets The Stern-Brocot representation

The standard binary representation of [0, 1].

Think of binary representations in [0, 1], like

0.010100010...



Binary representation Möbius maps and digit sets The Stern-Brocot representation

The standard binary representation of [0, 1].

Think of binary representations in [0, 1], like

0.010100010...

$$\{0,1\}^{\omega} \longrightarrow [0,1]$$
  
 $x_1 x_2 x_3 \dots \longmapsto \sum_{i=0}^{\infty} x_i \cdot 2^{-i}$ 

(日) (日) (日) (日) (日)

문 권

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# The standard binary representation of [0, 1].



# Think: receiving one bit at a time.

Each bit restricts the set of possibilities.

- With 0 bits, x could be anything in [0, 1].
- When we see "0.0", the options are reduced.
- "0.01" reduces them further.

(日) (日) (日) (日) (日)

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# The standard binary representation of [0, 1].



Think: receiving one bit at a time.

Each bit restricts the set of possibilities.

- With 0 bits, x could be anything in [0, 1].
- When we see "0.0", the options are reduced.
- "0.01" reduces them further.

(日) (日) (日) (日) (日)

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# The standard binary representation of [0, 1].



Think: receiving one bit at a time.

Each bit restricts the set of possibilities.

- With 0 bits, x could be anything in [0, 1].
- When we see "0.0", the options are reduced.

(日) (日) (日) (日) (日)

• "0.01" reduces them further.

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# The standard binary representation of [0, 1].



Think: receiving one bit at a time.

Each bit restricts the set of possibilities.

- With 0 bits, x could be anything in [0, 1].
- When we see "0.0", the options are reduced.

(日) (日) (日) (日) (日)

• "0.01" reduces them further.

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# The standard binary representation of [0, 1].



Think: receiving one bit at a time.

Each bit restricts the set of possibilities.

- With 0 bits, x could be anything in [0, 1].
- When we see "0.0", the options are reduced.

(D) (A) (B) (B)

• "0.01" reduces them further.

Binary representation Möbius maps and digit sets The Stern-Brocot representation

## The standard binary representation of [0, 1].



 $\vec{x} = x_1 x_2 x_3 \dots$  $S_0^{\vec{x}} \supseteq S_1^{\vec{x}} \supseteq S_2^{\vec{x}} \supseteq S_3^{\vec{x}} \supseteq \dots$ 

#### Some features:

- $\bigcap S_i^{\vec{x}}$  is a singleton.
- For each x, there is a sequence x such that
   ∩ S<sub>i</sub><sup>x</sup> = {x}.

Each sequence represents some x and each x is represented.

(日) (部) (注) (注)

Binary representation Möbius maps and digit sets The Stern-Brocot representation

The standard binary representation of [0, 1].



 $\vec{x} = x_1 x_2 x_3 \dots$  $S_0^{\vec{x}} \supseteq S_1^{\vec{x}} \supseteq S_2^{\vec{x}} \supseteq S_3^{\vec{x}} \supseteq \dots$ 

# Some features:

- $\bigcap S_i^{\vec{x}}$  is a singleton.
- For each x, there is a sequence x such that
   ∩ S<sub>i</sub><sup>x</sup> = {x}.

Each sequence represents some x and each x is represented.

(日) (日) (日) (日) (日)

Binary representation Möbius maps and digit sets The Stern-Brocot representation

The standard binary representation of [0, 1].



 $\vec{x} = x_1 x_2 x_3 \dots$  $S_0^{\vec{x}} \supseteq S_1^{\vec{x}} \supseteq S_2^{\vec{x}} \supseteq S_3^{\vec{x}} \supseteq \dots$ 

#### Some features:

- $\bigcap S_i^{\vec{x}}$  is a singleton.
- For each x, there is a sequence  $\vec{x}$  such that  $\bigcap S_i^{\vec{x}} = \{x\}.$

Each sequence represents some x and each x is represented.

(日) (日) (日) (日) (日)

Binary representation Möbius maps and digit sets The Stern-Brocot representation

The standard binary representation of [0, 1].



$$\vec{x} = x_1 \, x_2 \, x_3 \, \dots$$
$$S_0^{\vec{x}} \supseteq S_1^{\vec{x}} \supseteq S_2^{\vec{x}} \supseteq S_3^{\vec{x}} \supseteq \dots$$

Some features:

- $\bigcap S_i^{\vec{x}}$  is a singleton.
- For each x, there is a sequence x such that ∩ S<sub>i</sub><sup>x</sup> = {x}.

Each sequence represents some x and each x is represented.

< □ > < □ > < □ > < □ > < □

Binary representation Möbius maps and digit sets The Stern-Brocot representation

## How to construct the sets $S_i^{\vec{X}}$



 $\phi_0(x) = \frac{x}{2}$  $\phi_1(x) = \frac{x+1}{2}$ 

イロト イヨト イヨト イヨト

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# How to construct the sets $S_i^{\vec{x}}$



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

イロト イヨト イヨト イヨト

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# How to construct the sets $S_i^{\vec{x}}$



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

$$S_0^{01...} = [0, 1]$$

$$S_1^{01...} = \phi_0([0, 1])$$

$$S_2^{01...} = \phi_0 \circ \phi_1([0, 1])$$

$$S_i^{\vec{x}} = \phi_{x_0} \circ \ldots \circ \phi_{x_i}([0, 1])$$

$$\bigcap_{i \in \mathbb{N}} S_i^{\vec{x}} = \{0.x_0 x_1 \dots\}$$

イロト イヨト イヨト イヨト

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# How to construct the sets $S_i^{\vec{x}}$



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

$$S_0^{01...} = [0, 1]$$

$$S_1^{01...} = \phi_0([0, 1])$$

$$S_2^{01...} = \phi_0 \circ \phi_1([0, 1])$$

$$S_i^{\vec{x}} = \phi_{x_0} \circ \ldots \circ \phi_{x_i}([0, 1])$$

$$\bigcap_{i \in \mathbb{N}} S_i^{\vec{x}} = \{0.x_0 x_1 \dots\}$$

イロト イヨト イヨト イヨト

Binary representation Möbius maps and digit sets The Stern-Brocot representation

# How to construct the sets $S_i^{\vec{x}}$



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

$$S_0^{01...} = [0, 1]$$

$$S_1^{01...} = \phi_0([0, 1])$$

$$S_2^{01...} = \phi_0 \circ \phi_1([0, 1])$$

$$S_i^{\vec{x}} = \phi_{x_0} \circ \ldots \circ \phi_{x_i}([0, 1])$$

$$\bigcap_{i \in \mathbb{N}} S_i^{\vec{x}} = \{0.x_0 x_1 \dots\}$$

イロト イヨト イヨト イヨト

# 1, $+\infty,$ what's the difference?



## Work with $[0,+\infty]$ or [0,1]?

The choice is arbitrary.

Squint and you can't tell the difference.

<ロ> (日) (日) (日) (日) (日)

## 1, $+\infty$ , what's the difference?



Work with  $[0, +\infty]$  or [0, 1]? The choice is arbitrary.

Squint and you can't tell the difference.

A (1) < A (1) </p>

## 1, $+\infty$ , what's the difference?



Work with  $[0, +\infty]$  or [0, 1]? The choice is arbitrary.

Squint and you can't tell the difference.

< 17 ▶

Recall 
$$\phi_0(x) = \frac{x}{2}$$
,  $\phi_1(x) = \frac{x+1}{2}$ .

Möbius map: a function

$$A(x) = \frac{ax+b}{cx+d}$$

where  $a, b, c, d \in \mathbb{R}$ .

We are interested in Möbius maps that are

- strictly increasing,
- refining  $(A: [0, +\infty] \rightarrow [0, +\infty])$ .

Recall 
$$\phi_0(x) = \frac{x}{2}$$
,  $\phi_1(x) = \frac{x+1}{2}$ .  
Möbius map: a function

$$A(x) = \frac{ax+b}{cx+d}$$

#### where $a, b, c, d \in \mathbb{R}$ .

We are interested in Möbius maps that are

- strictly increasing,
- refining  $(A : [0, +\infty] \rightarrow [0, +\infty])$ .

<ロ> (四) (四) (注) (注) ()

문 문

Recall 
$$\phi_0(x) = \frac{x}{2}$$
,  $\phi_1(x) = \frac{x+1}{2}$ .  
Möbius map: a function

$$A(x) = \frac{ax+b}{cx+d}$$

where  $a, b, c, d \in \mathbb{R}$ .

We are interested in Möbius maps that are

- strictly increasing,
- refining  $(A: [0, +\infty] \rightarrow [0, +\infty])$ .

<ロ> (四) (四) (注) (注) ()

Recall 
$$\phi_0(x) = \frac{x}{2}$$
,  $\phi_1(x) = \frac{x+1}{2}$ .  
Möbius map: a function

$$A(x) = \frac{ax+b}{cx+d}$$

where  $a, b, c, d \in \mathbb{R}$ .

We are interested in Möbius maps that are

- strictly increasing,
- refining  $(A: [0, +\infty] \rightarrow [0, +\infty])$ .

イロト イヨト イヨト イヨト

## Digit sets

#### Möbius maps are our digits.

Let  $\Phi = \{\phi_0, \dots, \phi_k\}$  be a set of Möbius maps. A sequence  $\vec{x} = \phi_{i_0} \phi_{i_1} \phi_{i_2} \dots$  represents x if

$$\bigcap_{n=0}^{\infty} \underbrace{\phi_{i_0} \circ \phi_{i_1} \circ \ldots \circ \phi_{i_n}([0, +\infty])}_{S_n^{\mathcal{X}}} = \{x\}.$$

비로 (로) (로) (로) (미)

Φ is a *digit set* if each x is represented.

## Digit sets

Möbius maps are our *digits*. Let  $\Phi = \{\phi_0, \dots, \phi_k\}$  be a set of Möbius maps. A sequence  $\vec{x} = \phi_{i_0} \phi_{i_1} \phi_{i_2} \dots$  represents x if

$$\bigcap_{n=0}^{\infty} \underbrace{\phi_{i_0} \circ \phi_{i_1} \circ \ldots \circ \phi_{i_n}([0,+\infty])}_{S_n^{\varphi}} = \{x\}.$$

비로 (로) (로) (로) (미)

Φ is a *digit set* if each x is represented.

## Digit sets



Möbius maps are our *digits*. Let  $\Phi = \{\phi_0, \dots, \phi_k\}$  be a set of Möbius maps. A sequence  $\vec{x} = \phi_{i_0} \phi_{i_1} \phi_{i_2} \dots$  represents x if

$$\bigcap_{n=0}^{\infty} \underbrace{\phi_{i_0} \circ \phi_{i_1} \circ \ldots \circ \phi_{i_n}([0,+\infty])}_{S_{\vec{x}}^{\vec{x}}} = \{x\}.$$

(日) (日) (日) (日) (日)

 $\Phi$  is a *digit set* if each x is represented.
### Digit sets



Möbius maps are our *digits*. Let  $\Phi = \{\phi_0, \dots, \phi_k\}$  be a set of Möbius maps. A sequence  $\vec{x} = \phi_{i_0} \phi_{i_1} \phi_{i_2} \dots$  represents x if

$$\bigcap_{n=0}^{\infty} \underbrace{\phi_{i_0} \circ \phi_{i_1} \circ \ldots \circ \phi_{i_n}([0,+\infty])}_{S_{\vec{n}}^{\vec{x}}} = \{x\}.$$

(日) (日) (日) (日) (日)

 $\Phi$  is a *digit set* if each x is represented.

### Digit sets



Möbius maps are our *digits*. Let  $\Phi = \{\phi_0, \dots, \phi_k\}$  be a set of Möbius maps. A sequence  $\vec{x} = \phi_{i_0} \phi_{i_1} \phi_{i_2} \dots$  represents x if

$$\bigcap_{n=0}^{\infty} \underbrace{\phi_{i_0} \circ \phi_{i_1} \circ \ldots \circ \phi_{i_n}([0,+\infty])}_{S_{\vec{n}}^{\vec{x}}} = \{x\}.$$

(日) (日) (日) (日) (日)

 $\Phi$  is a *digit set* if each x is represented.

### Digit sets



Möbius maps are our *digits*. Let  $\Phi = \{\phi_0, \dots, \phi_k\}$  be a set of Möbius maps. A sequence  $\vec{x} = \phi_{i_0} \phi_{i_1} \phi_{i_2} \dots$  represents x if

$$\bigcap_{n=0}^{\infty} \underbrace{\phi_{i_0} \circ \phi_{i_1} \circ \ldots \circ \phi_{i_n}([0,+\infty])}_{S_n^{\overrightarrow{x}}} = \{x\}.$$

(日) (日) (日) (日) (日)

 $\Phi$  is a *digit set* if each x is represented.

### Φ is a *good digit set* if

- Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
- 2 The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ .

#### Theorem

Good digit sets are digit sets.

Good digit sets yield a *total representation*, i.e.  $\Phi^{\omega} \rightarrow [0, +\infty]$  is

イロト イポト イヨト イヨト

total

•



#### $\Phi$ is a good digit set if

- Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
- 2 The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ .

#### Theorem

Good digit sets are digit sets.

Good digit sets yield a *total representation*, i.e.  $\Phi^{\omega} \rightarrow [0, +\infty]$  is

<ロ> (日) (日) (日) (日) (日)

- total,
- continuous;
- surjective.



#### $\Phi$ is a good digit set if

- Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
- 2 The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ .

### Theorem

### Good digit sets are digit sets.

Good digit sets yield a total representation, i.e.  $\Phi^\omega \to [0,+\infty]$  is

<ロ> (日) (日) (日) (日) (日)

- total,
- continuous,
- surjective.



 $\Phi$  is a good digit set if

- Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
- 2 The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ .

#### Theorem

Good digit sets are digit sets.

Good digit sets yield a total representation, i.e.  $\Phi^\omega \to [0,+\infty]$  is

- total,
- continuous,
- surjective.



 $\Phi$  is a good digit set if

- Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
- 2 The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ .

#### Theorem

Good digit sets are digit sets.

Good digit sets yield a total representation, i.e.  $\Phi^\omega \to [0,+\infty]$  is

- total,
- continuous,
- surjective.



 $\Phi$  is a good digit set if

- Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
- 2 The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ .

#### Theorem

Good digit sets are digit sets.

Good digit sets yield a total representation, i.e.  $\Phi^\omega \to [0,+\infty]$  is

・ロト ・同ト ・ヨト ・ヨ

- total,
- continuous,
- surjective.



How to make the tree: If my parents are  $\frac{a}{b}$  and  $\frac{c}{d}$ , then I am  $\frac{a+c}{b+d}$ .



### The Stern-Brocot representation is a digit set



to rationals.



The Stern-Brocot representation maps finite sequences of  $\{L, R\}$  to rationals.

Easy to show: infinite sequences yield Cauchy sequences of rationals.



The Stern-Brocot representation maps finite sequences of  $\{L, R\}$  to rationals.

Easy to show: infinite sequences yield Cauchy sequences of rationals.

Careful with that metric!



Binary representation Möbius maps and digit sets The Stern-Brocot representation

### The Stern-Brocot representation is a digit set



$$L \dots \in [0, 1]$$
  
 $LR \dots \in [\frac{1}{2}, 1]$   
 $LRR \dots \in [\frac{2}{3}, 1]$ 

A nested sequence of sets  $S_i^{\overline{X}}$ .

▲ 御 ▶ → ● 目

Each  $S_i^{\vec{x}}$  is bounded by the parents of  $x_1 x_2 \dots x_n$ 



Binary representation Möbius maps and digit sets The Stern-Brocot representation

### The Stern-Brocot representation is a digit set



$$L \dots \in [0, 1]$$
  
 $LR \dots \in [\frac{1}{2}, 1]$   
 $LRR \dots \in [\frac{2}{3}, 1]$ 

A nested sequence of sets  $S_i^{\vec{x}}$ .

Each  $S_i^{\vec{x}}$  is bounded by the parents of  $x_1 x_2 \ldots x_i$ 

< 🗇 ▶



### The Stern-Brocot representation is a digit set



$$L \dots \in [0, 1]$$
  
 $LR \dots \in [\frac{1}{2}, 1]$   
 $LRR \dots \in [\frac{2}{3}, 1]$ 

A nested sequence of sets  $S_i^{\vec{x}}$ .

Each  $S_i^{\vec{x}}$  is bounded by the parents of  $x_1 x_2 \dots x_i$ .



### The Stern-Brocot representation is a digit set



 $\{\phi_L, \phi_R\}$  is a good digit set.

#### Admissible digit sets The homographic algorithm

# Outline

### Digit sets

- Binary representation
- Möbius maps and digit sets
- The Stern-Brocot representation

### 2 Admissibility

- Admissible digit sets
- The homographic algorithm

▲□▶ ▲ 国 ▶ ▲ 国

# Φ-Computability

#### Let $\Phi$ be a good digit set.

 $f:[0,+\infty] \to [0,+\infty]$  is  $\Phi$ -computable iff f has a continuous  $\Phi^{\omega}$  lifting.

$$\begin{array}{c} \Phi^{\omega} - - \frac{f^{\sharp}}{-} \to \Phi^{\omega} \\ \downarrow & \downarrow \\ \downarrow & \downarrow \\ [0, +\infty] \xrightarrow{f} \to [0, +\infty] \end{array}$$

Good digit sets aren't very good.  $x \mapsto 2x$  isn't Stern-Brocot-computable.

(日) (日) (日) (日) (日)

### Φ-Computability

Let  $\Phi$  be a good digit set.  $f : [0, +\infty] \to [0, +\infty]$  is  $\Phi$ -computable iff f has a continuous  $\Phi^{\omega}$  lifting.



Good digit sets aren't *very* good.  $x \mapsto 2x$  isn't Stern-Brocot-computable

문 권

### Φ-Computability

Let  $\Phi$  be a good digit set.  $f: [0, +\infty] \to [0, +\infty]$  is  $\Phi$ -computable iff f has a continuous  $\Phi^{\omega}$  lifting.



Good digit sets aren't *very* good.  $x \mapsto 2x$  isn't Stern-Brocot-computable.

문 논

### $p: \Phi^\omega \to [0,+\infty]$ is an admissible representation if it is

- continuous,
- surjective,
- maximal, i.e. for every continuous *r*:



If  $\Phi^{\omega} \rightarrow [0, +\infty]$  is admissible, any continuous f is  $\Phi$ -computable.

イロト イポト イヨト イヨト

- $p: \Phi^\omega \to [0,+\infty]$  is an admissible representation if it is
  - continuous,
  - surjective,
  - maximal, i.e. for every continuous *r*:



If  $\Phi^{\omega} \rightarrow [0, +\infty]$  is admissible, any continuous f is  $\Phi$ -computable.

(日) (部) (注) (注)

- $p: \Phi^\omega \to [0,+\infty]$  is an admissible representation if it is
  - continuous,
  - surjective,
  - maximal, i.e. for every continuous r:



If  $\Phi^{\omega} \rightarrow [0, +\infty]$  is admissible, any continuous f is  $\Phi$ -computable.

- $p: \Phi^\omega \to [0,+\infty]$  is an admissible representation if it is
  - continuous,
  - surjective,
  - maximal, i.e. for every continuous r:



If  $\Phi^\omega o [0, +\infty]$  is admissible, any continuous f is  $\Phi$ -computable.

(日) (部) (종) (종) (종) (종)

Φ is an admissible digit set (ADS) if
Loosely: ∩ S<sub>i</sub><sup>x</sup> is always a singleton.
The sets φ<sub>i</sub>((0, +∞)) cover (0, +∞).
(2) replaces

The sets φ<sub>i</sub>([0, +∞]) cover [0, +∞].
 for good digit sets.

Theorem Admissible digit sets yield admissible representations.

### Vot ADS!

Φ is an admissible digit set (ADS) if
Loosely: ∩ S<sub>i</sub><sup>x</sup> is always a singleton.
The sets φ<sub>i</sub>((0, +∞)) cover (0, +∞).
(2) replaces
The sets φ<sub>i</sub>([0, +∞]) cover [0, +∞] for good digit sets.

Theorem Admissible digit sets yield admissible representations.

Not ADS!

<ロ> (日) (日) (日) (日) (日)



- $\Phi$  is an *admissible digit set* (ADS) if
  - **1** Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
  - 2 The sets  $\phi_i((0, +\infty))$  cover  $(0, +\infty)$ .

(2) replaces

• The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ . for good digit sets.

<ロ> (日) (日) (日) (日) (日)

#### Theorem

Admissible digit sets yield admissible representations.



 $\Phi$  is an *admissible digit set* (ADS) if

- Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
- 2 The sets  $\phi_i((0, +\infty))$  cover  $(0, +\infty)$ .

(2) replaces

• The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ . for good digit sets.

<ロ> (日) (日) (日) (日) (日)

#### Theorem

Admissible digit sets yield admissible representations.



- $\Phi$  is an *admissible digit set* (ADS) if
  - Loosely:  $\bigcap S_i^{\vec{x}}$  is always a singleton.
  - 2 The sets  $\phi_i((0, +\infty))$  cover  $(0, +\infty)$ .

(2) replaces

• The sets  $\phi_i([0, +\infty])$  cover  $[0, +\infty]$ . for good digit sets.

(D) (A) (B) (B)

Theorem Admissible digit sets yield admissible representations.

### The Stern-Brocot representation is not ADS



S-B is a good digit set... but not an admissible digit set

> $\phi_L([0, +\infty]) = [0, 1]$  $\phi_R([0, +\infty]) = [1, +\infty]$

> > (日) (日) (日) (日) (日)

Solution: Add  $\phi_M(x) = \frac{2x+1}{x+2}$ .

The Stern-Brocot representation is not ADS



S-B is a good digit set... but not an admissible digit set.

> $\phi_L((0, +\infty)) = (0, 1)$  $\phi_R((0, +\infty)) = (1, +\infty)$

> > (日) (日) (日) (日) (日)

Solution: Add  $\phi_M(x) = \frac{2x+1}{x+2}$ .

The Stern-Brocot representation is not ADS



S-B is a good digit set... but not an admissible digit set.

> $\phi_L((0, +\infty)) = (0, 1)$  $\phi_R((0, +\infty)) = (1, +\infty)$

> > ・日・ ・ヨ・ ・ヨ・

Solution: Add  $\phi_M(x) = \frac{2x+1}{x+2}$ .

### Why this subsection doesn't matter.

# Let $\Phi$ be an ADS. Aim: Construct an algorithm H(A, -) computing Möbius maps A. But $\Phi^{\omega} \to [0, +\infty]$ is an admissible representation. Any continuous $f : [0, +\infty] \to [0, +\infty]$ lifts to $\Phi^{\omega}$ .



But formal verifications require explicit algorithms.

・ロト ・ 日 ・ ・ ヨ ・ ・ ヨ ・

### Why this subsection doesn't matter.

Let  $\Phi$  be an ADS. Aim: Construct an algorithm H(A, -) computing Möbius maps A. But  $\Phi^{\omega} \rightarrow [0, +\infty]$  is an admissible representation.

Any continuous  $f : [0, +\infty] \to [0, +\infty]$  lifts to  $\Phi^{\omega}$ .



But formal verifications require explicit algorithms.

비로 (로) (로) (로) (토)

### Why this subsection doesn't matter.

Let  $\Phi$  be an ADS.

Aim: Construct an algorithm H(A, -) computing Möbius maps A.

But  $\Phi^{\omega} \rightarrow [0, +\infty]$  is an admissible representation.

Any continuous  $f: [0, +\infty] \rightarrow [0, +\infty]$  lifts to  $\Phi^{\omega}$ .

But formal verifications require explicit algorithms.

イロト イポト イヨト イヨト
#### Why this subsection does matter.

Let  $\Phi$  be an ADS.

Aim: Construct an algorithm H(A, -) computing Möbius maps A.

But  $\Phi^{\omega} \rightarrow [0, +\infty]$  is an admissible representation.

Any continuous  $f:[0,+\infty] \to [0,+\infty]$  lifts to  $\Phi^{\omega}$ .

$$\begin{array}{ccc}
\Phi^{\omega} - - - \rightarrow \Phi^{\omega} \\
\downarrow^{p} & \downarrow^{p} \\
\left[0, +\infty\right] \xrightarrow{f} \left[0, +\infty\right]
\end{array}$$

#### But formal verifications require explicit algorithms.

비교 (종) (종) (종) (종)

### An algorithm for computing Möbius maps

#### Let $\mathbb{M}$ be the set of refining Möbius maps.

We explicitly defined  $H: \mathbb{M} \times \Phi^{\omega} \to \Phi^{\omega}$  so that



H is the homographic algorithm.

(日) (部) (注) (注)

### An algorithm for computing Möbius maps

Let  $\mathbb{M}$  be the set of refining Möbius maps. We explicitly defined  $H : \mathbb{M} \times \Phi^{\omega} \to \Phi^{\omega}$  so that



H is the homographic algorithm.

・ロト ・回ト ・ヨト ・ヨト

### An algorithm for computing Möbius maps

Let  $\mathbb{M}$  be the set of refining Möbius maps. We explicitly defined  $H : \mathbb{M} \times \Phi^{\omega} \to \Phi^{\omega}$  so that

$$\begin{array}{c|c} \Phi^{\omega} - -\overset{H(A,-)}{-} > \Phi^{\omega} \\ \downarrow^{p} & \downarrow^{p} \\ 0, +\infty \end{array} \xrightarrow{[0, +\infty]} A > [0, +\infty] \end{array}$$

H is the homographic algorithm.

< ロト (周) (日) (日)

Digit sets Admissibile digit sets The homographic algorithm

# The very (very) rough idea behind the algorithm (but with pictures)

H is the homographic algorithm.



Least fixed point construction that

- outputs a digit when possible or
- absorbs more input when needed.

# The very (very) rough idea behind the algorithm (but with pictures)

H is the homographic algorithm.



Least fixed point construction that

- outputs a digit when possible or
- absorbs more input when needed.

# The very (very) rough idea behind the algorithm (but with pictures)

H is the homographic algorithm.



Least fixed point construction that

- outputs a digit when possible or
- absorbs more input when needed.

- ・ 同 ・ ・ き ・ ・ ・ き

# The very (very) rough idea behind the algorithm (but with pictures)

H is the homographic algorithm.



Least fixed point construction that

 outputs a digit when possible or

・ 同 ト・ ・ ヨート・ ・ ヨ

• absorbs more input when needed.

# The very (very) rough idea behind the algorithm (but with pictures)

H is the homographic algorithm.



Least fixed point construction that

 outputs a digit when possible or

・ 同 ト・ ・ ヨート・ ・ ヨ

• absorbs more input when needed.

#### Er, so what did we do?

#### • Aim: investigate representations via Möbius maps

- Found sufficient conditions for
  - total representations
  - total, admissible representations
- modified Stern-Brocot to do formal arithmetic
- explicitly computed homographic algorithm for ADS

イロト イヨト イヨト イヨト

#### Er, so what did we do?

- Aim: investigate representations via Möbius maps
- Found sufficient conditions for
  - total representations
  - total, admissible representations
- modified Stern-Brocot to do formal arithmetic
- explicitly computed homographic algorithm for ADS

(日) (部) (注) (注)

#### Admissible digit sets The homographic algorithm

#### Er, so what did we do?

- Aim: investigate representations via Möbius maps
- Found sufficient conditions for
  - total representations
  - total, admissible representations
- modified Stern-Brocot to do formal arithmetic
- explicitly computed homographic algorithm for ADS

(日) (日) (日) (日) (日)

#### The homographic algorithm

#### Er, so what did we do?

- Aim: investigate representations via Möbius maps
- Found sufficient conditions for
  - total representations
  - total, admissible representations
- modified Stern-Brocot to do formal arithmetic.

< ロト (周) (日) (日)

#### The homographic algorithm

#### Er, so what did we do?

- Aim: investigate representations via Möbius maps
- Found sufficient conditions for
  - total representations
  - total, admissible representations
- modified Stern-Brocot to do formal arithmetic.
- explicitly computed homographic algorithm for ADS

< ロト (周) (日) (日)

#### Er, so what did we do?

- Aim: investigate representations via Möbius maps
- Found sufficient conditions for
  - total representations
  - total, admissible representations
- modified Stern-Brocot to do formal arithmetic
- explicitly computed homographic algorithm for ADS

#### Outline



• Additional material

<ロ> (四) (四) (三) (三) (三)

三日 のへの

$$A(x) = \frac{ax + b}{cx + d}$$
  
Same thing: A matrix  $M_A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$   
Let  $x, y \in [0, +\infty)$ .

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{cases} \frac{x}{y} & \text{if } y \neq 0, \\ +\infty & \text{else.} \end{cases}$$

$$A(\frac{x}{y})^{"} = "M_{\mathcal{A}} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Composition of Möbius maps is the same as multiplication of matrices.

E 990

$$A(x) = \frac{ax+b}{cx+d}$$
  
Same thing: A matrix  $M_A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$   
Let  $x, y \in [0, +\infty)$ .

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{cases} \frac{x}{y} & \text{if } y \neq 0, \\ +\infty & \text{else.} \end{cases}$$

$$A(\frac{x}{y})^{"} = "M_{\mathcal{A}} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Composition of Möbius maps is the same as multiplication of matrices.

E 990

$$A(x)=rac{ax+b}{cx+d}$$
  
Same thing: A matrix  $M_A=egin{pmatrix}a&b\\c&d\end{pmatrix}$   
Let  $x,y\in [0,+\infty).$ 

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{cases} \frac{x}{y} & \text{if } y \neq 0, \\ +\infty & \text{else.} \end{cases}$$

$$A(\frac{x}{y})^{"} = "M_A \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Composition of Möbius maps is the same as multiplication of matrices. বিচাৰ আজি বিজ্ঞান হয় আজি বিজ্ঞান বিজ্ঞান বিজ্ঞান বিজ্ঞান বিজ্ঞান বিজ্ঞান বিজ্ঞান বিজ্ঞান বিজ্ঞান বিজ

$$A(x)=rac{ax+b}{cx+d}$$
  
Same thing: A matrix  $M_A=egin{pmatrix}a&b\\c&d\end{pmatrix}$ Let  $x,y\in [0,+\infty).$ 

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{cases} \frac{x}{y} & \text{if } y \neq 0, \\ +\infty & \text{else.} \end{cases}$$

$$A(\frac{x}{y}) = "M_A \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Composition of Möbius maps is the same as multiplication of matrices.

E 990

### Translating $\phi_0$ , $\phi_1$ to $[0, +\infty]$



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

イロト イヨト イヨト イヨト

포네크

### Translating $\phi_0$ , $\phi_1$ to $[0, +\infty]$



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

<ロ> (日) (日) (日) (日) (日)

포네크

### Translating $\phi_0$ , $\phi_1$ to $[0, +\infty]$



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

$$[0,1] = \phi_0([0,+\infty])$$
$$[\frac{1}{3},1] = \phi_0 \circ \phi_1([0,+\infty])$$
$$\frac{1}{\pi-1} = ".010100010...$$

- (日) (三) (三)

포네크



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

# $[0, +\infty]$ inherits a metric from [0, 1].

We use this metric to measure the "shrinking" of  $\phi_{i_1}\phi_{i_2} \ldots \phi_{i_n}([0, +\infty]).$ 

< □ > < □ > < □ >



$$\phi_0(x) = \frac{x}{2}$$
$$\phi_1(x) = \frac{x+1}{2}$$

 $[0, +\infty]$  inherits a metric from [0, 1]. We use this metric to measure the "shrinking" of  $\phi_{i_1}\phi_{i_2} \dots \phi_{i_n}([0, +\infty])$ .

- 17 ▶

4 B K 4 B



 $\mathcal{B}(\Phi, n)$  measures the maximum diameter for *n*-length sequences.

 $\mathcal{B}(\Phi, 0) = 1$  $\mathcal{B}(\Phi, 1) = rac{1}{2}$  $\mathcal{B}(\Phi, 2) = rac{1}{4}$ od:  $\lim_{i \to \infty} \mathcal{B}(\Phi, j) = 0$ 

< 177 ▶

A 3 b



 $\mathcal{B}(\Phi, n)$  measures the maximum diameter for *n*-length sequences.

 $\mathcal{B}(\Phi, 0) = 1$  $\mathcal{B}(\Phi, 1) = \frac{1}{2}$  $\mathcal{B}(\Phi, 2) = \frac{1}{4}$ Good:  $\lim_{i \to \infty} \mathcal{B}(\Phi, j) = 0$ 

# When $A(x) \in \phi_j((0, +\infty))$ no matter what x is, output the digit $\phi_j$ .

Otherwise, absorb a digit from x to refine our calculation. Define  $A \sqsubseteq \phi_j \Leftrightarrow A([0, +\infty]) \subseteq \phi_j([0, +\infty])$ .

 $\begin{cases} \phi_0 \ H(\phi_0^{-1} \circ A, \phi_{i_1} \phi_{i_2} \dots) & \text{if } A \sqsubseteq \phi_0 \\ \phi_1 \ H(\phi_1^{-1} \circ A, \phi_{i_1} \phi_{i_2} \dots) & \text{else if } A \sqsubseteq \phi_1 \\ \vdots \\ \phi_k \ H(\phi_k^{-1} \circ A, \phi_{i_1} \phi_{i_2} \dots) & \text{else if } A \sqsubseteq \phi_k \\ H(A \circ \phi_i, \phi_{i_2} \phi_{i_3} \dots) & \text{otherwise.} \end{cases}$ 

・ 同・ ・ ヨ・ ・ ヨ・

When  $A(x) \in \phi_j((0, +\infty))$  no matter what x is, output the digit  $\phi_j$ . Otherwise, absorb a digit from x to refine our calculation. Define  $A \sqsubseteq \phi_j \Leftrightarrow A([0, +\infty]) \subseteq \phi_j([0, +\infty])$ .

$$A, \phi_{i_1}\phi_{i_2}\dots) := \begin{cases} \phi_0 \ H(\phi_0^{-1} \circ A, \phi_{i_1}\phi_{i_2}\dots) & \text{if } A \sqsubseteq \phi_0 \\ \phi_1 \ H(\phi_1^{-1} \circ A, \phi_{i_1}\phi_{i_2}\dots) & \text{else if } A \sqsubseteq \phi_1 \\ \vdots \\ \phi_k \ H(\phi_k^{-1} \circ A, \phi_{i_1}\phi_{i_2}\dots) & \text{else if } A \sqsubseteq \phi_k \\ H(A \circ \phi_i, \phi_{i_2}\phi_{i_3}\dots) & \text{otherwise.} \end{cases}$$

▲圖▶ ▲理▶ ▲理▶

문 돈

When  $A(x) \in \phi_j((0, +\infty))$  no matter what x is, output the digit  $\phi_j$ . Otherwise, absorb a digit from x to refine our calculation. Define  $A \sqsubseteq \phi_j \Leftrightarrow A([0, +\infty]) \subseteq \phi_j([0, +\infty])$ .

$$H(A,\phi_{i_1}\phi_{i_2}\ldots) := \begin{cases} \phi_0 \ H(\phi_0^{-1} \circ A, \phi_{i_1}\phi_{i_2}\ldots) & \text{if } A \sqsubseteq \phi_0 \\ \phi_1 \ H(\phi_1^{-1} \circ A, \phi_{i_1}\phi_{i_2}\ldots) & \text{else if } A \sqsubseteq \phi_1 \\ \vdots \\ \phi_k \ H(\phi_k^{-1} \circ A, \phi_{i_1}\phi_{i_2}\ldots) & \text{else if } A \sqsubseteq \phi_k \\ H(A \circ \phi_i, \phi_{i_2}\phi_{i_3}\ldots) & \text{otherwise.} \end{cases}$$

▲圖▶ ▲ 문▶ ▲ 문▶ 문 문 ● ○ ○ ○

When  $A(x) \in \phi_j((0, +\infty))$  no matter what x is, output the digit  $\phi_j$ . Otherwise, absorb a digit from x to refine our calculation. Define  $A \sqsubseteq \phi_j \Leftrightarrow A([0, +\infty]) \subseteq \phi_j([0, +\infty])$ .

$$H(A,\phi_{i_1}\phi_{i_2}\ldots) := \begin{cases} \phi_0 \ H(\phi_0^{-1} \circ A,\phi_{i_1}\phi_{i_2}\ldots) & \text{if } A \sqsubseteq \phi_0 \\ \phi_1 \ H(\phi_1^{-1} \circ A,\phi_{i_1}\phi_{i_2}\ldots) & \text{else if } A \sqsubseteq \phi_1 \\ \vdots \\ \phi_k \ H(\phi_k^{-1} \circ A,\phi_{i_1}\phi_{i_2}\ldots) & \text{else if } A \sqsubseteq \phi_k \\ H(A \circ \phi_i,\phi_{i_2}\phi_{i_3}\ldots) & \text{otherwise.} \end{cases}$$

▲圖▶ ▲ 문▶ ▲ 문▶ 문 문 ● ○ ○ ○