An Uncountable Chain in $\{0,1\}^{\mathbf{N}}$

An Uncountable Chain in $\{0,1\}^{\mathbf{N}}$

Part I: the proof I showed in class, with a bit more details.

An Uncountable Chain in $\{0,1\}^{\mathbf{N}}$

Part I: the proof I showed in class, with a bit more details.

Definition. Let X and Y be two partially ordered sets.
A function $f : X \to Y$ is an *isomorphism* if

- f is bijective,
- $x_1 \leq x_2$ if and only if $f(x_1) \leq f(x_2)$.

If such an f exists, we say X and Y are *isomorphic* and write $X \cong Y$.

An Uncountable Chain in $\{0,1\}^{\mathbf{N}}$

Part I: the proof I showed in class, with a bit more details.

Definition. Let X and Y be two partially ordered sets. A function $f : X \to Y$ is an *isomorphism* if

- f is bijective,
- $x_1 \leq x_2$ if and only if $f(x_1) \leq f(x_2)$.

If such an f exists, we say X and Y are *isomorphic* and write $X \cong Y$.

Intuitive meaning: X and Y being isomorphic means that they look identical, differing only by the names of their elements.

Observation 1. $(\{0,1\}^{\mathbf{N}}, \leq)$ and $(2^{\mathbf{N}}, \subseteq)$ are isomorphic.

Observation 2. $(2^{\mathbf{N}}, \subseteq)$ and $(2^{\mathbf{Q}}, \subseteq)$ are isomorphic.

Observation 3. If X and Y are isomorphic, then X has an uncountable chain if and only if Y has an uncountable chain.

Theorem. $(2^{\mathbf{Q}}, \subseteq)$ has an uncountable chain.

Proof. For a real number x, definfe
$B_x := \{q \in \mathbf{Q} \mid q < x\}$.
Define $C := \{B_x \mid x \in \mathbf{R}\}$.

- C is a chain. Any B_x, B_y are comparable. Indeed, if $x \leq y$ then $B_x \subseteq B_y$.
- C is uncountable. Indeed, the function $f : \mathbf{R} \to C$ defined by $f(x) = B_x$ is injective.

Theorem. $(2^{\mathbf{Q}}, \subseteq)$ has an uncountable chain.

Proof. For a real number x, definfe
$B_x := \{q \in \mathbf{Q} \mid q < x\}$.
Define $C := \{B_x \mid x \in \mathbf{R}\}$.

- C is a chain. Any B_x, B_y are comparable. Indeed, if $x \leq y$ then $B_x \subseteq B_y$.
- C is uncountable. Indeed, the function $f : \mathbf{R} \to C$ defined by $f(x) = B_x$ is injective.

Corollay. $(\{0,1\}^{\mathbf{N}}, \leq)$ has an uncountable chain.

Okay, maybe this was a bit mysterious...

Let's give a (longer) proof that actually shows how the elements of the chain are constructed.

Okay, maybe this was a bit mysterious...

Let's give a (longer) proof that actually shows how the elements of the chain are constructed.

We'll define a function f that takes as input an infinite bit sequence $\mathbf{a} \in \{0,1\}^{\mathbf{N}}$ and outputs an infinite bit sequence $f(\mathbf{a}) \in \{0,1\}^{\mathbf{N}}$ such that

1. f is an injection.
2. All output elements $f(\mathbf{a})$ are comparable.

Point 1 will ensure the set of outputs is uncountable,
Point 2 will ensure it is a chain.

Example of our procedure

input sequence

$01101001\ldots$

Example of our procedure

input sequence

$01101001\ldots$

output sequence

$**\ldots$

Example of our procedure

input sequence

$01101001...$

output sequence

just infinitely many $*$ in the beginning

$**...$

01101001...

**...

read first bit of input

01101001...

**...

read first bit of input

01101001...

**...

put it here

read first bit of input

01101001...

$0***\ldots$

put it here

read first bit of input

01101001...

0***...

read first bit of input

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

01101001...

$0***\ldots$

read first bit of input

$01101001...$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$0***\ldots$$

read first bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$0**\ldots$

read first bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$0\color{red}{0}*\color{red}{0}\ldots$$

read first bit of input

$01101001...$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$00*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$00*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0\ldots$$

read next bit of input

$01101001...$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$00*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$0010*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0*0\ldots$$

read next bit of input

$0 1 1 0 1 0 0 1 \ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\ldots$

$$0010{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0{*}0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\ldots$

$$0010*010*010*010*010*010*010*010*010*010*0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$0010*010*010*010*010*010*010*010*010*010*010*0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$0010{*}010{*}010{*}010{*}010{*}010{*}010{*}010{*}010{*}010{*}010{*}0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$0010101 0 * 010 * 010 * 010 * 010 * 010 * 010 * 010 * 010 * 010 * 0 \ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\ldots$

$$00101010*010*010*010*010*010*010*010*010*0\ldots$$

read next bit of input

$01101001...$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\ldots$

$$0010101 0*0101010*0101010*0101010*0101010*0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$$\ldots$$
$$00101010*0101010*0101010*0101010*0101010*0\ldots$$

read next bit of input

$01101001...$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\ldots$

$$00101010*0101010*0101010*0101010*0101010*0\ldots$$

read next bit of input

$01101001...$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\dots$

$$0010101000101010*0101010*0101010*0101010*0\dots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\ldots$

$$0010101000101010{*}0101010{*}0101010{*}0101010{*}0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\ldots$

$$001010100010101000101010*010101000101010*0\ldots$$

read next bit of input

$01101001\ldots$

Rule 1: Read bit of input. In output, replace first $*$ by that bit.

Rule 2:

- If that bit is 0, replace every other $*$ by 0, starting with the first $*$.
- If that bit is 1, replace every other $*$ by 1, starting with the second $*$.

$\cdots$

$$001010100010101000101010*010101000101010*0\ldots$$

AND SO ON FOREVER

input $\mathbf{a}$

$01101001...$

output $f(\mathbf{a})$

$$001010100010101000101010*010101000101010*0\ldots$$

Claim. This procedure is injective and produces a chain.

Proof. Let $\mathbf{a}$ and $\mathbf{b}$ be two different input sequences.

Let i be the first coordinate where $a_i \neq b_i$.

Let's assume $a_i = 0$, $b_i = 1$.

Let's run the previous procedure on $\mathbf{a}$ and $\mathbf{b}$ and stop just before it reads the i^{th} bit.

Input:

$$\mathbf{a} = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} a_{i+2} \ldots$$

$$\mathbf{b} = a_1 a_2 \ldots a_{i-1} 1 b_{i+1} b_{i+2} \ldots$$

Input:

$\mathbf{a} = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} a_{i+2} \ldots$

$\mathbf{b} = a_1 a_2 \ldots a_{i-1} 1 b_{i+1} b_{i+2} \ldots$

Input, just before reading bit i:

$f(\mathbf{a}) = \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots *$

$f(\mathbf{b}) = \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots *$

Input:

$$\mathbf{a} = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} a_{i+2} \ldots$$

$$\mathbf{b} = a_1 a_2 \ldots a_{i-1} 1 b_{i+1} b_{i+2} \ldots$$

Input, just before reading bit i:

$$f(\mathbf{a}) = \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdots$$

$$f(\mathbf{b}) = \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdot\cdot\cdot\cdot\cdot\cdot * \ \cdots$$

These parts of $f(\mathbf{a})$ and $f(\mathbf{b})$ consist of 0's and 1's. They are equal, because the parts of $\mathbf{a}$ and $\mathbf{b}$ read so far as identical.

Input:

$\mathbf{a} = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} a_{i+2} \ldots$

Now we read the next bit of $\mathbf{a}$ and $\mathbf{b}$

$\mathbf{b} = a_1 a_2 \ldots a_{i-1} 1 b_{i+1} b_{i+2} \ldots$

Input, just before reading bit i:

$f(\mathbf{a}) = \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots$

$f(\mathbf{b}) = \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots$

Input:

$\mathbf{a} = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} a_{i+2} \ldots$

$\mathbf{b} = a_1 a_2 \ldots a_{i-1} 1 b_{i+1} b_{i+2} \ldots$

Now we read the next bit of $\mathbf{a}$ and $\mathbf{b}$

Input, just before reading bit i:

$f(\mathbf{a}) = \cdots\cdots 0 \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots$

$f(\mathbf{b}) = \cdots\cdots 1 \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots\cdots * \cdots$

Input:

$\mathbf{a} = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} a_{i+2} \ldots$

$\mathbf{b} = a_1 a_2 \ldots a_{i-1} 1 b_{i+1} b_{i+2} \ldots$

Now we read the next bit of $\mathbf{a}$ and $\mathbf{b}$

Input, just before reading bit i:

$f(\mathbf{a}) = \cdots\cdots 0 \cdots\cdots 0 \cdots\cdots * \cdots\cdots 0 \cdots\cdots * \cdots\cdots 0 \cdots$

$f(\mathbf{b}) = \cdots\cdots 1 \cdots\cdots * \cdots\cdots 1 \cdots\cdots * \cdots\cdots 1 \cdots\cdots * \cdots$

Input:

$\mathbf{a} = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} a_{i+2} \ldots$

Now we read the next bit of $\mathbf{a}$ and $\mathbf{b}$

$\mathbf{b} = a_1 a_2 \ldots a_{i-1} 1 b_{i+1} b_{i+2} \ldots$

Input, just before reading bit i:

$$f(\mathbf{a}) = \cdots\cdots 0 \cdots\cdots 0 \cdots\cdots * \cdots\cdots 0 \cdots\cdots * \cdots\cdots 0 \cdots$$

$$f(\mathbf{b}) = \cdots\cdots 1 \cdots\cdots * \cdots\cdots 1 \cdots\cdots * \cdots\cdots 1 \cdots\cdots * \cdots$$

Whatever happens from now on, it is clear that $f(\mathbf{a}) < f(\mathbf{b})$.

Input:

$$\mathbf{a} = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} a_{i+2} \ldots$$

$$\mathbf{b} = a_1 a_2 \ldots a_{i-1} 1 b_{i+1} b_{i+2} \ldots$$

Now we read the next bit of $\mathbf{a}$ and $\mathbf{b}$

Input, just before reading bit i:

$$f(\mathbf{a}) = \cdots\cdots 0 \cdots\cdots 0 \cdots\cdots * \cdots\cdots 0 \cdots\cdots * \cdots\cdots 0 \cdots$$

$$f(\mathbf{b}) = \cdots\cdots 1 \cdots\cdots * \cdots\cdots 1 \cdots\cdots * \cdots\cdots 1 \cdots\cdots * \cdots$$

Whatever happens from now on, it is clear that $f(\mathbf{a}) < f(\mathbf{b})$.

So f is injective and $\mathrm{Im}(f)$ is a chain.